

手册页第 1 部分：用户命令

版权所有 © 2012, Oracle 和/或其附属公司。保留所有权利。

本软件和相关文档是根据许可证协议提供的，该许可证协议中规定了关于使用和公开本软件和相关文档的各种限制，并受知识产权法的保护。除非在许可证协议中明确许可或适用法律明确授权，否则不得以任何形式、任何方式使用、拷贝、复制、翻译、广播、修改、授权、传播、分发、展示、执行、发布或显示本软件和相关文档的任何部分。除非法律要求实现互操作，否则严禁对本软件进行逆向工程设计、反汇编或反编译。

此文档所含信息可能随时被修改，恕不另行通知，我们不保证该信息没有错误。如果贵方发现任何问题，请书面通知我们。

如果将本软件或相关文档交付给美国政府，或者交付给以美国政府名义获得许可证的任何机构，必须符合以下规定：

U.S. GOVERNMENT END USERS:

Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

本软件或硬件是为了在各种信息管理应用领域内的一般使用而开发的。它不应被应用于任何存在危险或潜在危险的应用领域，也不是为此而开发的，其中包括可能会产生人身伤害的应用领域。如果在危险应用领域内使用本软件或硬件，贵方应负责采取所有适当的防范措施，包括备份、冗余和其它确保安全使用本软件或硬件的措施。对于因在危险应用领域内使用本软件或硬件所造成的一切损失或损害，Oracle Corporation 及其附属公司概不负责。

Oracle 和 Java 是 Oracle 和/或其附属公司的注册商标。其他名称可能是各自所有者的商标。

Intel 和 Intel Xeon 是 Intel Corporation 的商标或注册商标。所有 SPARC 商标均是 SPARC International, Inc 的商标或注册商标，并应按照许可证的规定使用。AMD、Opteron、AMD 徽标以及 AMD Opteron 徽标是 Advanced Micro Devices 的商标或注册商标。UNIX 是 The Open Group 的注册商标。

本软件或硬件以及文档可能提供了访问第三方内容、产品和服务的方式或有关这些内容、产品和服务的信息。对于第三方内容、产品和服务，Oracle Corporation 及其附属公司明确表示不承担任何种类的担保，亦不对其承担任何责任。对于因访问或使用第三方内容、产品或服务所造成的任何损失、成本或损害，Oracle Corporation 及其附属公司概不负责。

目录

前言	17
简介	21
Intro(1)	22
User Commands	27
acctcom(1)	28
adb(1)	31
addbib(1)	32
alias(1)	34
allocate(1)	38
amt(1)	40
appcert(1)	41
apptrace(1)	48
apropos(1)	52
ar(1)	54
arch(1)	59
as(1)	60
asa(1)	65
at(1)	67
atq(1)	74
atrm(1)	75
audioconvert(1)	76
audiocctl(1)	80
audioplay(1)	82
audiorecord(1)	84
audiotest(1)	86

auths(1)	87
auto_ef(1)	91
awk(1)	94
banner(1)	100
basename(1)	101
basename(1B)	103
bc(1)	104
bdiff(1)	108
bfs(1)	109
biff(1B)	113
break(1)	114
builtin(1)	116
cal(1)	118
calendar(1)	119
cat(1)	121
cd(1)	124
cdrw(1)	128
checknr(1)	134
chgrp(1)	136
chkey(1)	139
chmod(1)	141
chown(1)	165
chown(1B)	168
ckdate(1)	169
ckgid(1)	172
ckint(1)	174
ckitem(1)	176
ckkeywd(1)	179
ckpath(1)	181
ckrange(1)	184
ckstr(1)	187
cksum(1)	190
cktime(1)	192
ckuid(1)	194
ckyorn(1)	196
clear(1)	198

cmp(1)	199
col(1)	201
comm(1)	203
command(1)	205
compress(1)	209
cp(1)	213
cpio(1)	218
cpp(1)	226
cputrack(1)	232
crle(1)	237
crontab(1)	248
csh(1)	254
csplit(1)	281
ct(1C)	284
ctags(1)	286
ctrun(1)	289
ctstat(1)	292
ctwatch(1)	296
cu(1C)	298
cut(1)	305
date(1)	307
dc(1)	311
deallocate(1)	314
deroff(1)	316
df(1B)	317
dhcpinfo(1)	319
diff(1)	322
diff3(1)	326
diffmk(1)	328
digest(1)	329
dircmp(1)	331
dis(1)	332
disown(1)	334
dispgid(1)	335
dispuid(1)	336
dos2unix(1)	337

dpost(1)	339
du(1)	342
du(1B)	345
dump(1)	347
dumpps(1)	350
echo(1)	351
echo(1B)	354
ed(1)	355
edit(1)	369
egrep(1)	374
eject(1)	377
elfdump(1)	379
elfedit(1)	385
elffile(1)	394
elfsign(1)	397
elfwrap(1)	401
encrypt(1)	404
enhance(1)	408
env(1)	409
eqn(1)	411
error(1)	415
ex(1)	418
exec(1)	429
exit(1)	432
expand(1)	435
exportfs(1B)	437
expr(1)	438
expr(1B)	442
exstr(1)	445
factor(1)	449
fastboot(1B)	450
fgrep(1)	451
file(1)	454
file(1B)	457
filebench(1)	459
filesync(1)	461

find(1)	467
finger(1)	475
fmt(1)	478
fmtmsg(1)	479
fold(1)	483
from(1B)	485
ftp(1)	486
gcore(1)	501
gencat(1)	503
geniconvtbl(1)	506
genmsg(1)	509
getconf(1)	515
getfacl(1)	522
getlabel(1)	526
getopt(1)	527
getoptcvt(1)	529
getopts(1)	532
gettext(1)	541
gettxt(1)	543
getzonepath(1)	545
glob(1)	546
gprof(1)	547
grep(1)	552
groups(1)	557
groups(1B)	558
grpck(1B)	559
hash(1)	560
head(1)	562
history(1)	564
hostid(1)	574
hostname(1)	575
iconv(1)	576
indxbib(1)	579
install(1B)	580
ipcrm(1)	582
ipcs(1)	583

isainfo(1)	587
isalist(1)	589
jobs(1)	590
join(1)	598
kbd(1)	601
kdestroy(1)	607
keylogin(1)	608
keylogout(1)	609
kill(1)	610
kinit(1)	615
klist(1)	622
kmdb(1)	624
kmfcfg(1)	629
kpasswd(1)	638
krb5-config(1)	639
ksh(1)	641
ksh88(1)	691
ktutil(1)	742
kvno(1)	744
lari(1)	746
last(1)	753
lastcomm(1)	755
ld(1)	757
ldapdelete(1)	778
ldaplist(1)	783
ldapmodify(1)	788
ldapmodrdn(1)	795
ldapsearch(1)	800
ldd(1)	808
ld.so.1(1)	814
let(1)	821
lex(1)	822
lgrpinfo(1)	834
limit(1)	840
line(1)	846
list_devices(1)	847

listusers(1)	851
llc2_autoconfig(1)	852
llc2_config(1)	853
llc2_stats(1)	855
ln(1)	863
ln(1B)	866
loadkeys(1)	869
locale(1)	870
localedef(1)	873
logger(1)	878
logger(1B)	880
login(1)	882
logname(1)	890
logout(1)	891
look(1)	892
lookbib(1)	893
lorder(1)	894
ls(1)	895
ls(1B)	918
m4(1)	921
mac(1)	927
mach(1)	930
machid(1)	931
madv.so.1(1)	932
mail(1)	936
mail(1B)	942
mailcompat(1)	943
mailp(1)	945
mailq(1)	947
mailstats(1)	949
mailx(1)	951
make(1S)	976
makekey(1)	1012
man(1)	1013
mconnect(1)	1021
mcs(1)	1022

mdb(1)	1024
mesg(1)	1065
mkdir(1)	1066
mkmsgs(1)	1068
mkstr(1B)	1070
mktemp(1)	1072
moe(1)	1077
more(1)	1079
mp(1)	1086
mpss.so.1(1)	1093
msgcc(1)	1096
msgcpp(1)	1099
msgcvt(1)	1106
msgfmt(1)	1107
msggen(1)	1113
msgget(1)	1116
mt(1)	1117
mv(1)	1120
nawk(1)	1123
nc(1)	1143
ncab2clf(1)	1154
ncakmod(1)	1156
newform(1)	1157
newgrp(1)	1160
newtask(1)	1163
nice(1)	1165
nl(1)	1167
nm(1)	1170
nohup(1)	1175
nroff(1)	1179
od(1)	1182
on(1)	1188
optisa(1)	1190
pack(1)	1191
packagemanager(1)	1194
pagesize(1)	1197

pargs(1)	1198
passwd(1)	1200
paste(1)	1210
patch(1)	1213
pathchk(1)	1218
pax(1)	1221
perl(1)	1253
pfexec(1)	1260
pg(1)	1261
pgrep(1)	1266
pkcs11_inspect(1)	1269
pkg(1)	1271
pkgdepend(1)	1304
pkgdiff(1)	1309
pkgfmt(1)	1311
pkginfo(1)	1312
pkglint(1)	1314
pkgmerge(1)	1318
pkgmk(1)	1322
pkmoglify(1)	1325
pkgparam(1)	1331
pkgproto(1)	1333
pkgrecv(1)	1335
pkgrepo(1)	1339
pkgsend(1)	1347
pkgsign(1)	1351
pkgtrans(1)	1353
pklogin_finder(1)	1356
pktool(1)	1358
plabel(1)	1375
plgrp(1)	1376
plimit(1)	1380
pmadvise(1)	1382
pmap(1)	1385
pm-updatemanager(1)	1396
ppgsz(1)	1398

ppriv(1)	1401
pr(1)	1405
praliases(1)	1409
prctl(1)	1410
preap(1)	1417
print(1)	1419
printenv(1B)	1422
printf(1)	1423
priocntl(1)	1432
proc(1)	1442
prof(1)	1446
profiles(1)	1450
projects(1)	1459
ps(1)	1461
ps(1B)	1473
ptree(1)	1476
pvs(1)	1478
pwd(1)	1483
radadrgen(1)	1484
ranlib(1)	1486
rcapstat(1)	1487
rcp(1)	1491
read(1)	1495
readonly(1)	1499
refer(1)	1501
regcmp(1)	1503
renice(1)	1505
rlogin(1)	1508
rm(1)	1512
rmformat(1)	1516
rmmount(1)	1520
roffbib(1)	1522
roles(1)	1524
rpcgen(1)	1526
rpm2cpio(1)	1532
rsh(1)	1533

runat(1)	1538
rup(1)	1541
rup(1C)	1542
ruptime(1)	1543
rusage(1B)	1544
rusers(1)	1546
rwho(1)	1547
sar(1)	1548
sccs(1)	1553
sccs-admin(1)	1563
sccs-cdc(1)	1568
sccs-comb(1)	1570
sccs-delta(1)	1572
sccs-get(1)	1575
sccs-help(1)	1581
sccs-prs(1)	1582
sccs-prt(1)	1586
sccs-rmdel(1)	1589
sccs-sact(1)	1590
sccs-sccsdiff(1)	1591
sccs-unget(1)	1592
sccs-val(1)	1593
scp(1)	1595
script(1)	1597
sdiff(1)	1598
sed(1)	1600
sed(1B)	1608
set(1)	1614
setfacl(1)	1624
setlabel(1)	1628
setpgrp(1)	1630
sftp(1)	1631
sh(1)	1635
shcomp(1)	1656
shell_builtins(1)	1658
shift(1)	1663

shutdown(1B)	1665
size(1)	1666
sleep(1)	1668
soelim(1)	1670
sort(1)	1671
sortbib(1)	1678
sotruss(1)	1680
spell(1)	1682
split(1)	1684
srchtxt(1)	1686
ssh(1)	1689
ssh-add(1)	1702
ssh-agent(1)	1704
ssh-http-proxy-connect(1)	1706
ssh-keygen(1)	1708
ssh-keyscan(1)	1712
ssh-socks5-proxy-connect(1)	1714
strchg(1)	1716
strings(1)	1719
strip(1)	1721
stty(1)	1723
stty(1B)	1732
sum(1)	1739
sum(1B)	1740
suspend(1)	1741
svcprop(1)	1742
svcs(1)	1747
symorder(1)	1755
sys-suspend(1)	1756
sysV-make(1)	1757
tabs(1)	1764
tail(1)	1768
talk(1)	1771
tar(1)	1774
tbl(1)	1787
tcopy(1)	1789

tee(1)	1790
telnet(1)	1791
test(1)	1804
test(1B)	1813
tftp(1)	1815
time(1)	1818
times(1)	1821
timex(1)	1822
tip(1)	1824
touch(1)	1832
touch(1B)	1836
tplot(1)	1837
tput(1)	1838
tr(1)	1843
tr(1B)	1848
trap(1)	1849
troff(1)	1852
true(1)	1854
truss(1)	1855
tset(1B)	1862
tsort(1)	1867
tty(1)	1869
type(1)	1870
typeset(1)	1871
ul(1)	1873
umask(1)	1874
uname(1)	1878
unifdef(1)	1881
uniq(1)	1883
units(1)	1886
unix2dos(1)	1888
updatehome(1)	1890
uptime(1)	1892
userattr(1)	1893
users(1B)	1894
uucp(1C)	1895

uuencode(1C)	1899
uuglist(1C)	1903
uustat(1C)	1904
uuto(1C)	1908
uux(1C)	1911
vacation(1)	1915
vc(1)	1919
vgrind(1)	1922
vi(1)	1925
vipw(1B)	1936
volcheck(1)	1937
volrmmount(1)	1938
w(1)	1939
wait(1)	1941
wc(1)	1944
what(1)	1946
whatis(1)	1948
whereis(1B)	1949
which(1)	1951
who(1)	1952
whoami(1B)	1956
whocalls(1)	1957
whois(1)	1958
write(1)	1959
xargs(1)	1962
xgettext(1)	1967
xstr(1)	1969
yacc(1)	1971
yes(1)	1974
ypcat(1)	1975
yptest(1)	1976
yppasswd(1)	1977
ypwhich(1)	1978
zlogin(1)	1979
zonename(1)	1982
zonestat(1)	1983

前言

无论是初次使用 SunOS 操作系统的用户还是熟悉该操作系统的用户，均可通过联机手册页获取与系统及其功能有关的信息。手册页用于简要回答关于用途/用法的问题。手册页通常是参考手册的组成部分，并不用作教程。

概述

以下内容包含对手册页各部分及其所引用信息的简要说明：

- 第 1 部分按照字母顺序介绍了操作系统中提供的各种命令。
- 第 1M 部分按照字母顺序介绍了主要用于系统维护和管理的各种命令。
- 第 2 部分介绍了所有的系统调用。其中的大多数调用可能返回一个或多个错误。如果返回一个异常值，则表明有错误情况。
- 第 3 部分介绍了各种库中包含的函数，这些函数不属于第 2 部分介绍的那些直接调用 UNIX 系统原语 (primitive) 的函数。
- 第 4 部分简要介绍了各种文件的格式，并在适当之处给出了文件格式的 C 结构声明。
- 第 5 部分包含其他文档，如字符集表。
- 第 7 部分介绍了涉及特定硬件外围设备和设备驱动程序的各种特殊文件。还介绍了 STREAMS 软件驱动程序、模块和 STREAMS 通用的一组系统调用。
- 第 9E 部分介绍了 DDI (Device Driver Interface, 设备驱动程序接口) /DKI (Driver/Kernel Interface, 驱动程序/内核接口)、仅 DDI 和仅 DKI 入口点例程，开发者可以将这些例程包含在设备驱动程序中。
- 第 9F 部分介绍了可供设备驱动程序使用的内核函数。
- 第 9S 部分介绍了驱动程序用来在驱动程序和内核之间共享信息的数据结构。

下面是手册页的通用格式。每个手册的手册页部分通常遵循该顺序，但只包括需要的标题。例如，如果未报告任何已知问题，则不包括“已知问题”部分。有关每一部分的更多详细信息，请参见 `intro` 页；有关手册页的更多一般信息，请参见 `man(1)`。

名称 本部分提供了记录的命令或函数的名称，后跟其用途的简要说明。

用法概要	<p>本部分说明了命令或函数的语法。如果命令或文件不存在于标准路径中，则显示其全路径名。除非要求使用不同的参数顺序，否则选项和参数均按字母顺序排列，首先是单个字母的参数，接下来是带有参数的选项。</p> <p>本部分使用以下特殊字符：</p> <ul style="list-style-type: none">[] 方括号。括在方括号中的选项或参数是可选的。如果没有方括号，则必须指定参数。... 省略号。可以为该符号前面的参数提供多个值，或者可以多次指定该参数，例如“filename...”。 分隔符。一次只能指定一个由该字符分隔的参数。{ } 大括号。括在大括号内的选项和/或参数是相互依赖的，因此必须将大括号中的所有内容视为一个单元。
协议	本部分仅出现在第 3R 子部分，用于指示协议说明文件。
描述	本部分定义了服务的功能和行为。因此，它简明地介绍了命令执行哪些操作。它不讨论“选项”或引用“示例”。在“用法”下介绍了交互式命令、子命令、请求、宏和函数。
IOctl	本部分仅出现在第 7 部分的手册页中。只有为 <code>ioctl(2)</code> 系统调用提供了适当参数的设备类才被称为 <code>ioctl</code> ，并生成自己的标题。特定设备的 <code>ioctl</code> 调用按字母顺序显示在该特定设备的手册页中。 <code>ioctl</code> 调用用于特殊类别的设备。所有这些调用都以 <code>io</code> 结尾，例如 <code>mtio(7I)</code> 。
选项	本部分列出了各命令选项及每个选项用途的简明摘要。逐个列出各个选项，并以它们在“用法概要”部分显示的顺序排列。在选项下讨论各个选项可能的参数，还提供缺省值（如果适用）。
操作数	本部分列出了命令操作数，并介绍它们对命令操作的影响。
输出	本部分介绍了命令所生成的输出（标准输出、标准错误或输出文件）。
返回值	如果手册页记录返回值的函数，则本部分列出这些值并介绍返回这些值应满足的条件。如果函数只能返回常量值（例如 0 或 -1），则将在标记的段落中列出这些值。否则，会有单个段落介绍每个函数的返回值。声明为 <code>void</code> 的函数不返回值，因此不会在“返回值”中讨论这些函数。
错误	对于故障，大多数函数将指出它们出现故障的原因的错误代码置于全局变量 <code>errno</code> 中。本部分按字母顺序列出了函数可以生成的所有错误代码，并介绍了导致每个错误的条件。如果多个条件可以导致同一错误，则在错误代码下以单独的段落介绍每个条件。
用法	本部分列出了需要详细解释的特殊规则、功能和命令。此处列出的子部分用于说明内置功能：
	命令

	修饰符 变量 表达式 输入语法
示例	本部分提供了用法的示例，或者如何使用命令或函数的示例。会尽可能显示包括命令行输入和计算机响应的完整示例。只要给出了示例，就会显示 <code>example%</code> 提示，如果用户必须为超级用户，则提示显示为 <code>example#</code> 。示例后面跟有说明、变量替换规则或返回值。大部分示例说明了“用法概要”、“说明”、“选项”和“用法”部分的概念。
环境变量	本部分列出了命令或函数影响的所有环境变量，其后附加了关于影响的简要说明。
退出状态	本部分列出了命令返回到调用程序或 <code>shell</code> 中的值以及导致返回这些值的条件。通常，返回零表示成功完成，返回非零值表示各种错误条件。
文件	本部分列出了手册页引用的所有文件名称、相关文件以及命令创建或所需的文件。每个文件名称后面都具有描述性摘要或说明。
属性	本部分通过定义属性类型及其相应的值列出了命令、实用程序和设备驱动程序的特征。有关详细信息，请参见 <code>attributes(5)</code> 。
另请参见	本部分列出了对其他手册页、内部文档和外部出版物的引用。
诊断	本部分列出了诊断消息以及导致错误的条件的简要说明。
警告	本部分列出了有关特殊条件的警告，这些条件可能会严重影响您的工作状况。此部分不是诊断列表。
附注	本部分列出了不属于页面任何部分的其他信息。它采用对用户旁白提示的形式，包含用户特别关注的要点。此处不包含关键信息。
已知问题	本部分介绍了已知问题，并尽可能给出解决方法。

参 考 文 档

简介

引用名	Intro, intro – 命令和应用程序的介绍
描述	<p>本节介绍了此操作系统可用的命令（按字母顺序排列）。</p> <p>需特别关注的页面归类如下：</p> <p>1B 只能在 <i>SunOS/BSD 兼容性软件包</i> 中找到的命令。</p> <p>1C 用于与其他系统进行通信的命令。</p> <p>1S 特定于 SunOS 的命令。</p>
其他节	<p>有关更多信息，请参见 SunOS 参考手册的以下各节。</p> <ul style="list-style-type: none"> ■ 第 1M 节介绍了系统维护命令。 ■ 第 4 节介绍了有关文件格式的信息。 ■ 第 5 节是关于公共可用文件和杂项信息页的描述。 <p>有关这些命令和程序的教程信息，请参见 《System Administration Guide: Advanced Administration》。</p>
手册页命令语法	<p>除非另有说明，手册页的“用法概要”部分介绍的命令将按以下语法接受选项和其他参数，并按下述方式进行解释。</p> <p><i>name</i> [-<i>option</i>...] [<i>cmdarg</i>...] 其中：</p> <p>[] 将非必需的选项或 <i>cmdarg</i> 括在其中。</p> <p>... 表示有多个选项或 <i>cmdarg</i>。</p> <p><i>name</i> 可执行文件的名称。</p> <p>{ } 括在大括号内的选项和/或参数是相互依赖的，因此必须将大括号中的所有内容视为一个单元。</p> <p><i>option</i> （始终以“-”开头。） <i>noargletter</i>... 或 <i>argletter optarg</i>[,...]</p> <p><i>noargletter</i> 表示不带选项参数的选项的单个字母。注意，可以将多个 <i>noargletter</i> 选项组合在一个“-”之后（请参见下文的准则 5）。</p> <p><i>argletter</i> 表示需要选项参数的选项的单个字母。</p> <p><i>optarg</i> 符合前面的某个 <i>argletter</i> 的选项参数（字符串）。请注意，在 <i>argletter</i> 后面的 <i>optargs</i> 组必须以逗号分隔，或者以制表符或空格字符分隔并用引号括起来（请参见下文的准则 8）。</p> <p><i>cmdarg</i> 未以“-”开头的路径名（或其他命令参数），或者本身表示标准输入的“-”。</p> <p>除非另有说明，否则，只要操作数或选项参数是数字值，或者包含数字值：</p> <ul style="list-style-type: none"> ■ 该数字就被解释为十进制整数。

- 0 到 2147483647 范围内的数字从语法上可识别为数字值。
- 如果实用程序描述中指出它可接受负数作为操作数或选项参数，则 -2147483647 到 2147483647 范围内的数字从语法上可识别为数字值。
- 允许比此处列出的范围更大的范围。

命令语法标准：准则

并不是所有当前命令都遵循这些命令语法准则，但新命令很可能遵循它们。所有 shell 程序都应当使用 `getopts(1)` 来解析位置参数和检查合法选项。它支持下面的准则 3 至 10。其他准则的实施必须由命令本身来执行。

1. 命令名称（请参见上面的 *name*）的长度应该为 2 到 9 个字符。
2. 命令名称应该只包括小写字母和数字。
3. 选项名称（请参见上面的 *option*）的长度必须为一个字符。
4. 所有选项都必须以 "-" 开头。
5. 可以将没有参数的多个选项组合在单个 "-" 之后。
6. 一个选项之后的第一个选项参数（请参见上面的 *optarg*）必须以制表符或空格字符开头。
7. 选项参数不是可选的。
8. 一个选项之后的选项参数组必须以逗号分隔，或者以制表符或空格字符分隔并用引号括起来（`-o xxx,z,yy` 或 `-o"xxx z yy"`）。
9. 在命令行上所有选项都必须位于操作数（请参见上面的 *cmdarg*）之前。
10. "-" 可以用来表示选项的结尾。
11. 选项相对于彼此的顺序应当无关紧要。
12. 操作数（请参见上面的 *cmdarg*）的相对顺序能够以操作数随之出现的命令所决定的方式影响操作数的意义。
13. 只应使用前后都有一个空格字符的 "-" 来表示标准输入。

已经为 Solaris 和其他 Sun 产品开发了一组扩展的准则，称为 CLIP，即命令行界接口范例 (Command Line Interface Paradigm)。其目的是提供与 Linux 系统上广泛应用的 GNU 命令行语法更为密切一致的命令行语法，而不是为了改进现有的实用程序或将此应用于所有新的实用程序。按照计划，该组扩展准则只在合适的时候应用于正在开发中的实用程序集。

CLIP 是上面讨论的准则的一个完整超集，与 IEEE Std. 1003.1-2001 (SUSv3) 密切一致。它并未包括所有的 GNU 语法。GNU 语法允许使用与 IEEE 规则冲突的结构或具有二义性的结构。这些结构是不允许的。

扩展的 CLIP 命令行语法为：

```
utility_name -a --longopt1 -c option_argument \  
-f option_argument --longopt2=option_argument \  
--longopt3 option_argument operand
```

示例中的实用程序被命名为 `utility_name`。它后面跟有选项、选项参数和操作数，这些统称为参数。由一个连字符及紧跟其后的单个字母或数字组成的参数（如 `-a`）称为短选项。由两个连字符及紧跟其后的一系列字母、数字和连字符组成的参数（如 `--longopt1`）称为长选项。短选项和长选项统称为选项（以前称为标志）。某些选项后面跟有选项参数，如 `-c option_argument` 所示。最后一个选项和选项参数后面的参数称为操作数。一旦遇到第一个操作数，所有后续的参数都被解释为操作数。

选项参数有时显示为以空格与其短选项隔开，有时显示为直接相邻。这反映了以下情形，在某些情况下，选项参数与选项包含在同一参数字符串内；大多数情况下，它作为下一个参数。此规范要求选项是与其选项参数隔开的一个参数，但为确保以前的应用程序能够继续运行，也有一些例外。

- 如果实用程序的“用法概要”部分在短选项和选项参数之间显示了一个空格（如此示例中的 `-c option_argument` 所示），则应用程序应当将该选项及其选项参数用作隔开的参数。
- 如果未显示空格（如此示例中的 `-f option_argument` 所示），则应用程序期望选项与其选项参数在同一参数字符串中直接相邻，而不插入空格。
- 尽管存在上述要求，但无论概要行中是否存在空格，应用程序都应接受短选项和选项参数作为单个参数或隔开的参数。
- 具有选项参数的长选项始终规定如下：使用等号作为选项名称和选项参数之间的分隔符。如果实用程序的“选项”部分在长选项与其选项参数之间显示了一个等号（`=`）（如此示例中的 `--longopt2= option_argument` 所示），则应用程序应当还允许将该选项与其选项参数作为隔开的参数使用（如此示例中的 `--longopt1 option_argument` 所示）。

CLIP 扩展了前面讨论的准则，附加了下列准则：

14. 可以使用 `command subcommand [options] [operands]` 格式对类似的操作进行分组。子命令名称应与命令名称遵循相同的约定，如准则 1 和 2 中所述。
15. 长选项应该以 `--` 开头，并且只应当包含可移植字符集中的字母数字字符和连字符。选项名称通常为 1 到 3 个单词长度，并以连字符分隔每个单词。
16. 应当使用 `--name=argument` 来指定长选项的选项参数。还允许 `--name argument` 格式。
17. 所有实用程序都应该支持两个标准长选项：`--version`（具有短选项同义词 `-v`）和 `--help`（具有短选项同义词 `-?`）。如果 `--version` 的该首选短选项同义词已被使用，可以更换其短选项同义词（但必须提供一个短选项同义词）。在遇到这两个选项时，都会停止进一步处理参数，并且在显示相应的输出后，实用程序会成功退出。
18. 每个短选项只应当有一个对应的长选项，每个长选项只应当有一个对应的短选项。为了与以前的做法或等效实用程序的社区版本兼容，允许使用同义选项。
19. 短选项名称应该根据以下规则从长选项名称中获取：

1. 使用长选项名称的第一个字母作为短选项名称。
 2. 如果第一个字母与其他短选项名称冲突，则选择一个显眼的辅音字母。
 3. 如果第一个字母和显眼的辅音字母均与其他短选项名称冲突，则选择一个显眼的元音字母。
 4. 如果长选项名称中的字母都不可用，则选择一个任意字符。
20. 如果长选项名称由单个字符组成，它必须使用同一字符作为短选项名称。应避免单字符的长选项。只有单个字符是最具描述性的名称时，才允许单字符的长选项，这种情况极少。
21. 附加的 CLIP 准则的准则 1 中描述的这种格式的子命令一般是必需的。如果省略了该子命令，则命令将不接受任何操作数，并且只允许所定义的在遇到时会停止进一步处理参数的选项。在不提供子命令和参数的情况下使用此格式调用命令是错误的。提供此准则是为了允许以命令-子命令结构接受常用格式的命令 `--help`、命令 `-?`、命令 `--version` 以及命令 `-V`。

其中的一些准则只有实用程序的创作者才会感兴趣。在此处提供它们是为了方便想要编写遵循此语法的实用程序的人使用。

致谢

Oracle America, Inc. 非常感谢 The Open Group 允许复制版权归其所有的部分文档。可以通过 <http://www.opengroup.org/bookstore/> 在线访问 The Open Group 的原始文档。

美国电气及电子工程师学会 (Institute of Electrical and Electronics Engineers, IEEE) 与 The Open Group 已授予我们部分翻印其文档的权限。

在以下陈述中，短语“此文本”指部分的系统文档。

Sun 操作系统参考手册中以电子形式从以下来源翻印和复制了此文本的一部分：IEEE Std 1003.1, 2004 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, 版权所有 (C) 2001-2004, 美国电气及电子工程师学会与 The Open Group。如果这些版本与原始 IEEE 和 The Open Group 标准之间存在任何差异，请以原始 IEEE 和 The Open Group 标准为准。原始标准可通过 <http://www.opengroup.org/unix/online.html> 在线访问。

在所有包含该资料的产品中都应提供此声明。

另请参见

`getopts(1)`、`wait(1)`、`exit(2)`、`getopt(3C)`

诊断

在终止时，每个命令返回两个状态字节，一个由系统提供并给出终止原因，在“正常”终止的情况下，另一个由程序提供 [请参见 `exit(2)`]。前一个字节为 0，表示正常终止。后一个字节通常为 0，表示成功执行，而非零则表示出现故障，例如参数错误、数据错误或无法访问数据。它的名称不一，可称为“退出代码”、“退出状态”或“返回代码”，仅在涉及特殊约定时才会进行描述。

警告

当处理包含空字符的文件时，某些命令会产生异常结果。这些命令通常将文本输入行视为字符串，因此在行中遇到空字符（字符串结束符）时会出现混乱。

参 考 文 档

User Commands

引用名	acctcom – 搜索并输出进程记帐文件
用法概要	<pre>acctcom [-abfhikmqrtv] [-C sec] [-e time] [-E time] [-g group] [-H factor] [-I chars] [-l line] [-n pattern] [-o output-file] [-O sec] [-s time] [-S time] [-u user] [filename]...</pre>
描述	<p>acctcom 实用程序读取 <code>acct.h(3HEAD)</code> 描述的格式的 <i>filename</i>、标准输入或者 <code>/var/adm/pacct</code>，并将选定的记录写入到标准输出。每个记录代表一个进程的执行。输出中显示 <code>COMMAND NAME</code>、<code>USER</code>、<code>TTYNAME</code>、<code>START TIME</code>、<code>END TIME</code>、<code>REAL (SEC)</code>、<code>CPU (SEC)</code> 以及 <code>MEAN SIZE (K)</code>，还可能会显示 <code>F (fork()/exec() 标志: 1 表示不带 exec() 的 fork())</code>、<code>STAT (系统退出状态)</code>、<code>HOG FACTOR</code>、<code>KCORE MIN</code>、<code>CPU FACTOR</code>、<code>CHARS TRNSFD</code> 以及 <code>BLOCKS READ (读取和写入的总块数)</code>。</p> <p>如果命令是以超级用户特权执行的，则命令名称前会附加一个 '#'。如果某个进程没有与一个已知终端关联，则在 <code>TTYNAME</code> 字段中会输出 '?'。</p> <p>如果未指定 <i>filename</i> 并且标准输入与某个终端或者 <code>/dev/null</code> 关联（就像在 shell 中使用 '&' 时一样），则将读取 <code>/var/adm/pacct</code>；否则，读取标准输入。</p> <p>如果指定了任何 <i>filename</i> 参数，则会按照文件的各自顺序读取这些文件。每个文件通常是正向读取的，也就是说，按照进程完成时间的先后顺序读取。文件 <code>/var/adm/pacct</code> 通常是要检查的当前文件；一个较忙的系统可能需要若干个这样的文件，除当前文件外，这些文件均位于 <code>/var/adm/pacctincr</code> 中</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -a 显示有关选定进程的某些平均统计信息。这些统计信息将显示在输出记录之后。 -b 反向读取，先显示最后的命令。在读取标准输入时，此选项不起作用。 -f 在输出中列出 <code>fork()/exec()</code> 标志列和系统退出状态列。此选项的数字输出是八进制的。 -h 显示进程在执行过程中使用的总可用 CPU 时间的比率，而不是平均内存大小。该“扰乱因子”=(总 CPU 时间)/(已用时间)。 -i 在输出中列出包含 I/O 计数的列。 -k 显示总 <code>kcore-minutes</code> 而不是内存大小。 -m 显示平均核心大小（缺省值）。 -q 不列出任何输出记录，只是像使用 -a 选项时一样列出平均统计信息。 -r 显示 CPU 因子 (用户时间/(系统时间+用户时间))。 -t 为系统和用户显示单独的 CPU 时间。

- v 从输出中排除列标题。
- C *sec* 只显示总 CPU 时间 (系统时间 + 用户时间) 超过 *sec* 秒的进程。
- e *time* 选择在 *time* 时间或其之前存在的进程。
- E *time* 选择在 *time* 时间或其之前结束的进程。为 -S 和 -E 使用相同的 *time* 将显示在 *time* 时间存在的进程。
- g *group* 只显示属于 *group* 的进程。*group* 可由组 ID 或组名指定。
- H *factor* 只显示超过 *factor* 的进程，其中，*factor* 是上文中在介绍 -h 选项时解释的“扰乱因子”。
- I *chars* 只显示传送的字符数大于 *chars* 指定的限定数的进程。
- l *line* 只显示属于终端 /dev/term/line 的进程。
- n *pattern* 只显示与 *pattern* 匹配的命令，其中，*pattern* 可以是一个正则表达式，除了 + 表示出现一次或多次，这与在 [regcmp\(3C\)](#) 中时一样。
- o *output-file* 将选定的进程记录以输入数据格式复制到 *output-file*；禁止输出到标准输出。
- O *sec* 只显示 CPU 系统时间超出 *sec* 秒的进程。
- s *time* 显示在 *time* 时间或其之后存在的进程，该时间以 *hr* [:*min* [:*sec*]] 格式指定。
- S *time* 选择在 *time* 时间或其之后开始的进程。
- u *user* 只显示属于 *user* 的进程。该用户可以通过用户 ID、稍后会被转换为用户 ID 的登录名、'#'（这指定仅显示以超级用户特权执行的那些进程）或者 '?'（这指定仅显示与未知用户 ID 关联的那些进程）指定。

文件

- /etc/group 系统组文件
- /etc/passwd 系统口令文件
- /var/adm/pacctincr 活动进程记帐文件

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/accounting/legacy-accounting
CSI	Enabled (已启用)

另请参见

[ps\(1\)](#)、[acct\(1M\)](#)、[acctcms\(1M\)](#)、[acctcon\(1M\)](#)、[acctmerge\(1M\)](#)、[acctprc\(1M\)](#)、[acctsh\(1M\)](#)、
《Oracle Solaris 管理：常见任务》

附注

acctcom 只报告已终止的进程；对于活动进程，请使用 [ps\(1\)](#)。

引用名 adb – 通用调试器

用法概要 adb [-kw] [-I *dir*] [-P *prompt*] [-V *mode*] [*object* [*core*]]

描述 adb 实用程序是一个交互式的通用调试器。它可用于检查文件并为程序的执行提供一个可控的环境。

目前，adb 实用程序是作为指向 [mdb\(1\)](#) 实用程序的链接实现的。mdb(1) 是一个低层次调试器，可用于检查用户进程以及当前操作系统或操作系统故障转储。新的 mdb(1) 实用程序完全向后兼容 adb 命令的现有语法和功能，包括支持处理 adb 宏文件。《[Oracle Solaris Modular Debugger Guide](#)》与 [mdb\(1\)](#) 手册页描述了 mdb 功能，包括其 adb 兼容性模式。在执行 adb 链接时，该兼容模式将缺省激活。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	developer/debug/mdb

另请参见 [mdb\(1\)](#)、[attributes\(5\)](#)

《[Oracle Solaris Modular Debugger Guide](#)》

引用名 addbib – create or extend a bibliographic database

用法概要 addbib [-a] [-p *promptfile*] *database*

描述 When addbib starts up, answering y to the initial Instructions? prompt yields directions. Typing n (or RETURN) skips the directions. addbib then prompts for various bibliographic fields, reads responses from the terminal, and sends output records to *database*. A null response (just RETURN) means to leave out that field. A '-' (minus sign) means to go back to the previous field. A trailing backslash allows a field to be continued on the next line. The repeating Continue? prompt allows the user either to resume by typing y (or RETURN), to quit the current session by typing n or q, or to edit *database* with any system editor (see vi(1), ex(1), ed(1)).

选项 The following options are supported:

- a Suppresses prompting for an abstract. Asking for an abstract is the default. Abstracts are ended with a Control-D.
- p *promptfile* Uses a new prompting skeleton, defined in *promptfile*. This file should contain prompt strings, a TAB, and the key-letters to be written to the *database*.

用法

Bibliography Key Letters

The most common key-letters and their meanings are given below. addbib insulates you from these key-letters, since it gives you prompts in English, but if you edit the bibliography file later on, you will need to know this information.

- %A Author's name
- %B Book containing article referenced
- %C City (place of publication)
- %D Date of publication
- %E Editor of book containing article referenced
- %F Footnote number or label (supplied by refer)
- %G Government order number
- %H Header commentary, printed before reference
- %I Issuer (publisher)
- %J Journal containing article
- %K Keywords to use in locating reference
- %L Label field used by -k option of refer
- %M Bell Labs Memorandum (undefined)

%N	Number within volume
%O	Other commentary, printed at end of reference
%P	Page number(s)
%Q	Corporate or Foreign Author (unreversed)
%R	Report, paper, or thesis (unpublished)
%S	Series title
%T	Title of article or book
%V	Volume number
%X	Abstract — used by roffbib, not by refer
%Y,Z	Ignored by refer

示例

示例 1 Editing the bibliography file

Except for A, each field should be given just once. Only relevant fields should be supplied.

```
%A  Mark Twain
%T  Life on the Mississippi
%I  Penguin Books
%C  New York
%D  1978
```

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	text/doctools

另请参见

[ed\(1\)](#), [ex\(1\)](#), [indxbib\(1\)](#), [lookbib\(1\)](#), [refer\(1\)](#), [roffbib\(1\)](#), [sortbib\(1\)](#), [vi\(1\)](#), [attributes\(5\)](#)

引用名	alias, unalias – create or remove a pseudonym or shorthand for a command or series of commands
用法概要	<pre>/usr/bin/alias [alias-name[= string...]] /usr/bin/unalias alias-name... /usr/bin/unalias -a</pre>
csh	<pre>alias [name [def]] unalias pattern</pre>
ksh88	<pre>alias [-tx] [name[= value]...] unalias name... unalias [-a]</pre>
ksh	<pre>alias [-ptx] [name[= value]...] unalias [-a] [name...]</pre>
描述	<p>The <code>alias</code> and <code>unalias</code> utilities create or remove a pseudonym or shorthand term for a command or series of commands, with different functionality in the C-shell and Korn shell environments.</p>
<code>/usr/bin/alias</code>	<p>The <code>alias</code> utility creates or redefines alias definitions or writes the values of existing alias definitions to standard output. An alias definition provides a string value that replaces a command name when it is encountered.</p> <p>An alias definition affects the current shell execution environment and the execution environments of the subshells of the current shell. When used as specified by this document, the alias definition does not affect the parent process of the current shell nor any utility environment invoked by the shell.</p>
<code>/usr/bin/unalias</code>	<p>The <code>unalias</code> utility removes the definition for each alias name specified. The aliases are removed from the current shell execution environment. The <code>-a</code> option removes all alias definitions from the current execution environment.</p>
csh	<p><code>alias</code> assigns <i>def</i> to the alias <i>name</i>. The assigned <i>def</i> is a list of words that can contain escaped history-substitution metasyntax. <i>name</i> is not allowed to be <code>alias</code> or <code>unalias</code>. If <i>def</i> is omitted, the alias <i>name</i> is displayed along with its current definition. If both <i>name</i> and <i>def</i> are omitted, all aliases are displayed.</p> <p>Because of implementation restrictions, an alias definition must have been entered on a previous command line before it can be used.</p> <p><code>unalias</code> discards aliases that match (filename substitution) <i>pattern</i>. All aliases can be removed by <code>'unalias *'</code>.</p>

`ksh88` `alias` with no arguments prints the list of aliases in the form *name=value* on standard output. An `alias` is defined for each name whose *value* is specified. A trailing space in *value* causes the next word to be checked for alias substitution. The `-t` flag is used to set and list tracked aliases. The value of a tracked alias is the full pathname corresponding to the specified *name*. The value becomes undefined when the value of `PATH` is reset but the aliases remained tracked. Without the `-t` flag, for each *name* in the argument list for which no *value* is specified, the name and value of the alias is printed. The `-x` flag is used to set or print *exported aliases*. An exported alias is defined for scripts invoked by *name*. The exit status is non-zero if a *name* is specified, but no value, and no alias has been defined for the *name*.

The `alias`s specified by the list of *names* can be removed from the `alias` list with `unalias`.

`ksh` `alias` creates or redefines alias definitions or writes the existing alias definitions to standard output.

An alias definition provides a string value that replaces a command name when the command is read. Alias names can contain any printable character that is not special to the shell. If an alias value ends in a `SPACE` or `TAB`, the word following the command name the alias replaces is also checked to see whether it is an alias.

If no names are specified, the names and values of all aliases are written to standard output. Otherwise, for each name that is specified, and `=value` is not specified, the current value of the alias corresponding to name is written to standard output. If `=value` is specified, the alias name is created or redefined.

`alias` is built-in to the shell as a declaration command so that field splitting and pathname expansion are not performed on the arguments. Tilde expansion occurs on *value*. An alias definition only affects scripts read by the current shell environment. It does not affect scripts run by this shell.

`unalias` removes the definition of each named alias from the current shell execution environment, or all aliases if `-a` is specified. It does not affect any commands that have already been read and subsequently executed.

选项

The following option is supported by `unalias`:

`-a` Removes all alias definitions from the current shell execution environment.

`ksh88` The following option is supported by `alias`:

`-t` Sets and lists tracked aliases.

`ksh` The following options are supported by `alias`:

`-p` Causes the output to be in the form of `alias` commands that can be used as input to the shell to recreate the current aliases.

`-t` Specifies tracked aliases.

Tracked aliases connect a command name to the command's pathname, and are reset when the PATH variable is unset. The tracked aliases feature is now obsolete.

-x Ignored, this option is obsolete.

The following option is supported by `unalias`:

-a Causes all alias definitions to be removed. *name* operands are optional and ignored if specified.

操作数

The following operands are supported:

`alias` *alias-name* Write the alias definition to standard output.

`unalias` *alias-name* The name of an alias to be removed.

alias-name=string Assign the value of *string* to the alias *alias-name*.

If no operands are specified, all alias definitions are written to standard output.

Output

The format for displaying aliases (when no operands or only *name* operands are specified) is:

```
"%s=%s\n" name, value
```

The *value* string is written with appropriate quoting so that it is suitable for reinput to the shell.

示例

示例 1 Modifying a Command's Output

This example specifies that the output of the `ls` utility is columnated and more annotated:

```
example% alias ls="ls -CF"
```

示例 2 Repeating Previous Entries in the Command History File

This example creates a simple “redo” command to repeat previous entries in the command history file:

```
example% alias r='fc -s'
```

示例 3 Specifying a Command's Output Options

This example provides that the `du` utility summarize disk output in units of 1024 bytes:

```
example% alias du="du -k"
```

示例 4 Dealing with an Argument That is an Alias Name

This example sets up the `nohup` utility so that it can deal with an argument that is an alias name:

```
example% alias nohup="nohup "
```

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `alias` and `unalias`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态 The following exit values are returned:

0 Successful completion.

`alias` >0 One of the *alias-name* operands specified did not have an alias definition, or an error occurred.

`unalias` >0 One of the *alias-name* operands specified did not represent a valid alias definition, or an error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

`cs`, `cs`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

`ks`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

另请参见 [cs\(1\)](#), [ks\(1\)](#), [ks88\(1\)](#), [shell_builtins\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名 allocate – 设备分配

用法概要 allocate [-s] [-w] [-F] [-U *uname*] [-z *zonename*] *device*
allocate [-s] [-w] [-F] [-U *uname*] [-z *zonename*] -g *dev-type*

描述 allocate 实用程序通过其分配机制管理设备的所有权。它确保每个设备一次只被一个合格用户使用。

device 参数指定要操控的设备。为维护设备所有者的完整性，分配操作将针对与该设备关联的所有特殊文件执行。

缺省的分配操作将与 *device* 关联的设备特殊文件分配给当前进程的 `uid`。

只有经授权的用户才能分配设备。 `device_allocate(4)` 中指定了所需的授权。

如果系统配置有 Trusted Extensions，则 allocate 在向设备的调用者授予访问权限之前会为该设备运行清理程序。对于具有可移除介质并且该介质具有可挂载的文件系统的设备，如果调用者选择了该设备，则 allocate 将挂载该介质。

选项 支持以下选项：

- F *device* 强制分配空闲的或预分配的设备。此选项通常与 -U 选项一起使用来为特定用户分配或重新分配设备。仅允许具有 `solaris.device.revoke` 授权的用户使用此选项。
- g *dev-type* 分配其设备类型与 *dev-type* 匹配的设备。 *dev-type* 参数指定要操作的设备的类型。
- s 无提示。抑制任何诊断信息的输出。
- U *uname* 执行分配操作时，使用用户 ID *uname* 而不是使用当前进程的用户 ID。仅允许具有 `solaris.device.revoke` 授权的用户使用此选项。

如果配置了 Trusted Extensions，则还支持以下选项：

- w 在窗口环境中运行设备清理程序。如果该程序的窗口版本存在，则会使用窗口版本。否则，将在终端窗口中运行标准版本。
- z *zonename* 将设备分配到由 *zonename* 指定的区域中。

操作数 支持下列操作数：

device 指定要分配的设备的名称。

退出状态 将返回以下退出值：

- 0 成功完成。
- 20 对于指定的设备没有任何项。
- 其他值** 出现错误。

文件

`/etc/security/device_allocate`

`/etc/security/device_maps`

`/etc/security/dev/*`

`/etc/security/lib/*`

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

调用为 "Uncommitted"（未确定）。选项为 "Uncommitted"（未确定）。输出为 "Not-an-Interface"（不是接口）。

另请参见 [deallocate\(1\)](#)、[list_devices\(1\)](#)、[device_allocate\(1M\)](#)、[dminfo\(1M\)](#)、[mkdevalloc\(1M\)](#)、[mkdevmaps\(1M\)](#) 和 [mkdevmaps\(1M\)](#)。

“控制对设备的访问”

附注 只有 Solaris 审计功能启用后，本手册页中描述的功能才可用。

只有 [device_allocate\(1M\)](#) 服务启用后，本手册页中描述的功能才可用。

在配置有 Trusted Extensions 的系统上，该功能是缺省启用的。

Solaris 操作环境的将来发行版可能不再支持 `/etc/security/dev`、[mkdevalloc\(1M\)](#) 和 [mkdevmaps\(1M\)](#)。

引用名 amt – 运行抽象机测试

用法概要 amt [-s]

描述 amt 命令在经通用标准 (Common Criteria) 安全认证的系统中使用。该命令用来验证用于实现受控安全访问保护框架 (Controlled Access Protection Profile, CAPP) 的对象重用需求的低层次功能是否正常工作。/usr/bin/amt 是一个 shell 脚本，执行特定于您的系统的测试。对于 32 位系统，这些测试作为 32 位应用程序运行。对于 64 位系统，这些测试执行两次：一次作为 32 位应用程序，一次作为 64 位应用程序。

amt 会列出它执行的每个测试的结果：**通过**或**失败**，除非使用 -s 选项抑制了输出。

选项 支持以下选项：

-s 抑制输出。

退出状态 将返回以下错误值：

0 所有测试都通过。

>0 失败测试的数目。

<0 错误的命令行参数。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed (已确定)

另请参见 [attributes\(5\)](#)

引用名	appcert – 用于检查应用程序级产品对 Solaris 接口的不稳定使用。
用法概要	appcert [-h] [-n] [-f <i>infile</i>] [-w <i>working_dir</i>] [-B] [-L] [-S] { <i>obj</i> <i>dir</i> }...
描述	<p>appcert 实用程序用于检查应用程序是否与 Solaris 应用程序二进制接口 (Application Binary Interface, ABI) 兼容。Solaris ABI 定义了 Solaris 中可以供应用程序安全稳定地使用的运行时库接口。更准确地说，appcert 实用程序能够识别依赖于不稳定运行时接口的依赖项以及可能导致产品在 Solaris 的未来发行版中运行失败的其他特定风险。</p> <p>appcert 可检查：</p> <ul style="list-style-type: none"> ■ 使用 Solaris 库中的专用符号。 这些是专用符号，即不打算供开发者使用的函数或数据。它们是 Solaris 库之间用来互相调用的接口。这些符号可能会改变其语义行为或者甚至完全消失（也称为降级的符号）。因此，确保应用程序不依赖于任何专用接口是一个很好的做法。 ■ 静态链接。 具体来说，这是指静态链接 <code>libc.a</code>、<code>libsocket.a</code> 和 <code>libnsl.a</code> 归档文件，而不是动态链接对应的共享目标文件 <code>.so</code>。因为 Solaris 库之间的专用符号调用的语义在不同的发行版之间可能会发生更改，所以最好不要以静态方式将库代码固定到二进制目标文件中。 ■ 未绑定的符号。 这些是在运行 appcert 时动态链接程序无法解析的库符号（即，函数或数据）。未绑定的符号表明可能存在环境问题（如 <code>LD_LIBRARY_PATH</code>）或构建问题（如在编译时未指定 <code>-llib</code> 和/或 <code>-z defs</code>）。将未绑定的符号标记出来是为了指出这些问题，防止出现更严重的问题。 <p>如果产品是很多程序和支持共享目标文件的集合，通过让 appcert 引用应用程序的安装目录，可以很容易地利用 appcert 检查整个应用程序。</p> <p>为执行该任务，appcert 将为产品内的每个目标文件（不管是可执行目标文件还是共享目标文件）构建一个接口依赖项配置文件，以确定该产品依赖的所有 Solaris 系统接口。（注：appcert 使用 Solaris 运行时链接程序来执行该确定操作。）然后，这些依赖项配置文件将与 Solaris ABI 的定义进行比较以确定是否存在专用接口（这些专用接口在应用程序级使用时不安全且不稳定）。</p> <p>appcert 会生成一个简单的汇总报告，其中指出产品的哪些组件有缺陷（如果有）以及它们的缺陷是什么。该报告对正在检查其产品的发行版间稳定性的开发人员很有帮助。</p> <p>注意，appcert 生成的是完整的接口依赖项信息，包括公用的 Solaris 接口（安全稳定）和专用（非 ABI）接口。如有需要，还可以为每个产品组件检查该信息。</p> <p>重要说明： appcert 必须与被检查的应用程序在同一环境中运行。请参见“附注”部分。</p>

- 选项** 支持以下选项：
- B 如果 `appcert` 在批处理模式下运行，则输出报告中将为每个二进制目标文件包含一行，以 `PASS` 开头表示没有检测到该二进制目标文件的问题，以 `FAIL` 开头表示发现问题，以 `INC` 开头表示无法完整检查该二进制目标文件。解析这些标签时不要太拘泥于字面意义。例如，`PASS` 只是意味着未触发任何 `appcert` 警告。这些字符串是左对齐的，因此可以使用 `grep^FAIL ...` 等进行选定。
 - f *infile* 指定包含要检查的文件（每个文件一行）的列表的 *infile*。该文件列表将附加到由命令行操作数确定的列表中（请参见下文的“操作数”部分）。
 - h 输出 `appcert` 的用法信息。
 - L `appcert` 检查产品中是否存在共享目标文件。如果找到一些共享目标文件，`appcert` 会将共享目标文件所驻留的目录附加到 `LD_LIBRARY_PATH`。使用 `-L` 标志可以阻止 `appcert` 的这一行为。
 - n 当在目录中搜索要检查的二进制目标文件时，该选项不跟踪符号链接。请参见 `find(1)`。
 - S 将 Solaris 库目录（即 `/usr/openwin/lib:/usr/dt/lib`）附加到 `LD_LIBRARY_PATH`。
 - w *working_dir* 指定要在其中运行库组件并创建临时文件的目录（缺省目录为 `/tmp`）。
- 操作数** 支持下列操作数：
- `{obj | dir} ...` 要检查的目标文件的完整列表和/或包含构成要检查的产品的目标文件的目录完整列表。`appcert` 以递归方式搜索目录来查找目标文件；非目标文件将被忽略。
- 退出状态** 将返回以下退出值：
- 0 `appcert` 运行成功，未发现潜在的二进制稳定性问题。
 - 1 `appcert` 未能成功运行。
 - 2 某些已检查的目标文件存在潜在的二进制稳定性问题。
 - 3 未找到任何可检查的二进制目标文件。
- 限制** 如果要检查的目标文件依赖于库，必须在目标文件中记录这些依赖项（使用编译器的 `-l` 开关）。
- 如果要检查的目标文件依赖于其他共享库，则在运行 `appcert` 时必须能够通过 `LD_LIBRARY_PATH` 或 `RUNPATH` 访问这些库。

要检查 64 位应用程序，计算机必须运行 64 位 Solaris 内核。请参见 `isalist(1)`。另外，当前不能对 64 位应用程序执行静态链接检查。?

appcert 不能检查以下内容：

- 完全或部分静态链接的目标文件。

完全静态链接的目标文件将被报告为不稳定的。?

- 没有设置执行权限的可执行文件。

将跳过这些文件。没有执行权限的共享目标文件不会跳过。?

- 充当 `setuid root` 的目标文件。??

由于 `ldd(1)` 中的限制，将跳过这些文件。要检查这些文件，请复制并/或更改权限。

- 非 ELF 可执行文件，如 shell 脚本。
- 针对 Solaris，并非 C 语言接口，如 C++ 接口和 Java 接口。

代码本身无需为 C 语言，但是对 Solaris 库的调用必须使用 C 语言。

输出文件

appcert 会将结果记录到工作目录（缺省为 `/tmp/appcert.?????`）中的下列文件。

Index	所检查的二进制目标文件与工作目录中此二进制目标文件的特定输出所在子目录之间的映射。
Report	运行 appcert 时在 <code>stdout</code> 上显示的汇总报告的副本。
Skipped	包含要求 appcert 检查但强制跳过的二进制目标文件的列表以及跳过每个二进制目标文件的简单原因。

此外，在 `appcert.????/objects/` 目录下的子目录中还包括针对每个目标文件的信息，这些信息在以下文件中：

<code>check.demoted_symbols</code>	包含怀疑为降级的 Solaris 符号的符号列表。
<code>check.dynamic.private</code>	包含与目标文件直接绑定的专用 Solaris 符号的列表。
<code>check.dynamic.public</code>	包含与目标文件直接绑定的公用 Solaris 符号的列表。
<code>check.dynamic.unbound</code>	包含运行 <code>ldd -r</code> 时动态链接程序没有绑定的符号的列表。为方便起见，还包括 <code>ldd</code> 输出的包含 <code>file not found</code> 的行。
<code>summary.dynamic</code>	所检查目标文件的动态绑定的格式整齐的摘要，其中包括从每个 Solaris 库使用的公用符号和专用符号的表。

其他文件是 `appcert` 在内部使用的临时文件。

输出消息

专用符号的使用

专用符号是 Solaris 库中的一些不打算供开发者或外部使用的函数或数据变量。这些符号是 Solaris 库之间进行互相调度和通信的接口。在 `pvs(1)` 输出中，这些符号的符号版本名称标记为 `SUNWprivate`。

专用符号可能会改变其语义行为或者甚至完全消失（**降级的**或者**废弃的**符号）。因此，您的应用程序不应依赖于任何专用符号。

降级的符号

降级符号是某个 Solaris 库中曾经专用于该库，但在后续 Solaris 发行版中已经删除（或其作用域被限制为该库本地）的函数或数据变量。如果应用程序直接调用了某个降级的符号，则该应用程序在删除了该符号的发行版中及后续发行版中将无法运行（重定位错误）。

在极少数情况下，在某个后续发行版中会恢复某个降级的符号。尽管如此，应用程序仍不能在某些发行版中运行。

在从 Solaris 2.5.1 到 2.6 的转换中，Oracle Corporation 完成了大部分库作用域限定工作。这一措施是为了提高二进制文件稳定性。通过使这些完全内部的接口不可见（即它们无法被动态链接），开发者无法有意或无意地调用这些接口。有关更多信息，请参见《[链接程序和库指南](#)》，特别是介绍版本控制的章节。

未绑定的符号

未绑定的符号是指在应用程序引用的库符号中，在运行 `appcert` 时动态链接程序无法解析的那些库符号。**注：**`appcert` 并不真正运行应用程序，因此，影响动态链接的某个环境方面可能没有正确设置。

未绑定的符号不是一定表明存在潜在的二进制稳定性问题。它们唯一表明的是在运行 `appcert` 时，运行时动态链接程序无法解析这些符号。

存在未绑定的符号的原因可能是 `LD_LIBRARY_PATH` 设置得不正确。请确保正确设置该变量，这样您所有的二进制目标文件才能找到它们依赖的所有库（您的产品自己的库，Solaris 库，或者第三方的库）。然后，重新运行 `appcert`。

您可能会发现编写一个 shell 脚本，使用该脚本来正确设置环境并针对您要检查的二进制目标文件运行 `appcert` 比较方便。

存在未绑定的符号的另一个常见原因是测试中的共享目标文件未记录其动态依赖项，也就是说，在构建时**没有**将 `-l` 开关提供给编译器和 `ld(1)`。因此，共享目标文件要求基于它链接的**可执行文件**记录正确的依赖项。

注意，这样的共享目标文件可以通过标准形式进行链接（即在构建可执行文件时指定）或者动态打开（例如，可执行文件在运行时可能会针对共享目标文件调用 `dlopen(3C)`）。在以上两种情况下，运行 `appcert` 时都会存在未绑定的符号。在第一种情况下，可通过在运行 `appcert` 之前正确设置 `LD_LIBRARY_PATH` 来解决未绑定的符号

问题。在第二种情况 (dlopen) 下，很难解决未绑定的符号问题。在某些情况下，可以正确设置 `LD_PRELOAD` 来预装入所需的库，但该过程并不是总能奏效。

如何了解环境是否已经正确设置并可以避免出现未绑定的符号？如果运行 `ldd -r` 时未出现 "file not found" 或 "symbol not found" 错误，表明环境已经正确设置。有关动态链接的更多信息，请参见 `ld.so.1(1)` 和 `ldd(1)`。

在任何情况下，`appcert` 都会将未绑定的符号标记为警告，以防还暗藏着更严重的问题。未绑定的符号可能表明存在依赖于降级的符号（已从库中删除的符号或作用域调整为该库本地的符号）的依赖项。依赖于降级的符号的依赖项会导致严重的二进制稳定性问题。

不过，正确地设置环境应该能够消除大多数未绑定的符号。通常，比较好的做法是尽可能地在构建时记录库的依赖项，这样有助于更好地定义二进制目标文件并使其成为自包含的。另外，建议在构建共享目标文件时使用 `-z defs` 标志来强制在编译时解析所有符号。有关更多信息，请参见 `ld(1)`。

找不到绑定

`appcert` 针对每个要测试的二进制目标文件运行 `/bin/ldd -r`。该命令对环境变量进行如下设置：`LD_DEBUG="files,bindings"`。（有关更多信息，请参见 `ldd(1)` 和 `ld.so.1(1)`）。如果该命令由于某种原因而失败，`appcert` 将不能获得任何动态符号绑定信息并将找不到绑定。

存在以下任一情况时，`appcert` 可能会失败：

- 二进制目标文件没有读权限。
- 二进制目标文件是 SUID 或 SGID，并且用户没有足够的权限。
- 二进制目标文件是一个没有设置执行权限位的可执行文件。
- 二进制目标文件是完全静态链接的。
- 二进制目标文件没有记录库依赖项信息。

同样，还存在其他一些情况，如内存不足。通常，该标志意味着由于权限或环境问题，`appcert` 无法完整地检查目标文件。请尝试修改权限或环境以便能够记录动态绑定。

过时的库

过时的库是指已不赞成使用并且可能会在将来的某个发行版中从 Solaris 中完全删除的库。`appcert` 将这些库标出是因为依赖于此类库的应用程序可能无法在将来的 Solaris 发行版中运行。过时的库中的所有接口（包括专用接口）都被冻结且不会更改。

sys_errlist 或 sys_nerr 的使用

直接使用 `sys_errlist` 或 `sys_nerr` 符号会存在风险，导致引用可能超出 `sys_errlist` 数组的结尾。这些符号在 32 位版本的 Solaris 中不赞成使用，而在 64 位版本中完全不存在。请改用 `strerror(3C)`。

强符号与弱符号的使用

与弱符号（如 `socket`）关联的强符号（如 `_socket`）保留为专用符号（其行为在将来可能会改变）。您的应用程序只应直接引用弱符号（强符号通常以 "_" 开头）。

注：在某些构建环境下，尽管源代码看起来没有引用专用符号，也会在您的二进制代码中记录强符号/专用符号依赖性而不是记录弱符号/公用符号依赖性。不过，应采取措施来弄清发生此情况的原因并纠正此依赖性。

附注

appcert 应该与要检查的应用程序在相同的环境中运行。否则，它将无法将引用正确解析到 Solaris 库中的接口。请采取下列步骤：

1. 确保将 LD_LIBRARY_PATH 及环境的任何其他方面设置为应用程序在运行时使用的设置。另外，请确保该设置包含产品中的任何非 Solaris 共享目标文件所在的目录，以确保在引用这些目标文件时可以找到它们。
2. 确保要检查的所有二进制目标文件满足以下条件：
 - 是动态链接的 ELF 目标文件
 - 设置了对可执行文件的执行权限（共享目标文件不必满足此条件）。
 - 不是 SUID root 用户（否则，必须是 root 用户才能检查；必要时，请创建非 SUID 副本并检查这些副本）。

您可能会发现编写一个 shell 脚本，使用该脚本来正确设置环境并运行 appcert 比较方便。

可能会遇到下面一些潜在的问题：

- appcert 将看起来是 Solaris 库的一部分的符号报告为未绑定的符号。

当应用程序使用 `dlopen(3C)` 访问没有记录其 Solaris 依赖项的共享目标文件时，可能会发生这种情况。appcert 在这种情况下无法解析符号的使用，因为从不会针对共享目标文件调用动态链接程序，并且没有任何其他依赖项信息可用来分析 Solaris 符号绑定。对于非 Solaris 符号也会发生这种情况。

为避免该问题，请确保在构建共享目标文件时，在编译行上使用 `-llib` 选项来显式记录共享目标文件对 Solaris 库的依赖项信息（请参见 `cc(1)` 和 `ld(1)`）。
- appcert 报告指出应用程序使用了应用程序的源代码中没有引用的 Solaris 专用符号。

该问题很有可能是由引用该符号的 Solaris 库的静态链接造成的。由于 appcert 使用动态链接程序来解析符号，因此，在 appcert 看来，静态链接的库是应用程序代码的一部分（从某种意义上来说，它们确实是）。有时，Solaris 头文件中的宏替换也会导致该问题。

为避免该问题，请尽量不要将 Solaris 库归档文件静态链接到您的应用程序中。
- appcert 不能识别作为 Solaris 的一部分的库。

有些过时的 Solaris 库太旧了，以致于还未能能够对其符号进行版本化，这些库就已被废弃了。因此，appcert 无法将其识别为 Solaris 的一部分。

已知问题

遗憾的是，将术语“公用”和“专用”分别等同于“稳定”和“不稳定”让人有点困惑。需要特别说明的是，实验性的或者发展中的接口从某种意义上来说是公用的，因为它们记录在案的并且鼓励使用它们。但这些接口是不稳定的，因为构建有这些接口的应用程

序可能无法在后续发行版中运行。因此，为了方便 `appcert` 检查，这些接口被归类为专用接口，直到它们不再改进。与此相反，过时的接口最终将会消失，因此是不稳定的，即使它们在过去是公用且稳定的并且 `appcert` 当前仍将其视为公用接口。幸运的是，这两种情况极少见。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	developer/appcert
接口稳定性	Committed（已确定）

另请参见

`cc(1)`、`find(1)`、`isalist(1)`、`ld(1)`、`ldd(1)`、`ld.so.1(1)`、`pvs(1)`、`dlopen(3C)`、`strerror(3C)`

引用名	aptrace – 跟踪对 Solaris 共享库的应用函数调用
用法概要	<pre>aptrace [-f] [-F [!] <i>tracefromlist</i>] [-T [!] <i>tracetolist</i>] [-o <i>outputfile</i>] [[-tv] [!] <i>call</i> ,...] <i>command</i> [<i>command arguments</i>]</pre>
描述	<p>aptrace 实用程序运行 <i>command</i> 所指定的可执行程序，并跟踪程序 <i>command</i> 对 Solaris 共享库所进行的所有函数调用。对于可跟踪的每个函数调用，aptrace 会报告所调用的库接口的名称、所传递的参数值以及返回值。</p> <p>缺省情况下，aptrace 跟踪从可执行目标文件到所依赖的任何共享目标文件的直接调用。缺省情况下不报告间接调用（即可执行文件所依赖的共享目标文件之间的调用）。</p> <p>可以使用 -F 或 -T 选项（请参见下文）跟踪更多共享目标文件之间的调用。</p> <p>缺省报告格式为每个调用一行，且不提供通过引用传递的参数以及数据结构的格式化输出。</p> <p>可使用 -v 选项（请参见下文）获得提供更多参数详细信息的格式化输出。</p> <p>缺省情况下，会跟踪共享目标文件所提供的每个接口（如果已调用）。但是，可以使用 -t 和/或 -v 选项限制要跟踪的接口组。</p> <p>由于一般情况下可以跟踪在运行时链接的任何动态目标文件（可执行目标文件以及所依赖的任何共享目标文件）之间的调用，因此所跟踪的每个调用的报告将提供发出调用的目标文件名称。</p> <p>aptrace 会跟踪动态目标文件之间通过过程链接表发生的所有过程调用，因此仅跟踪通过该表绑定的那些过程调用。请参见《链接程序和库指南》。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -f 追踪 fork(2) 所创建的所有子项。此选项还将在每行的开头输出进程 ID。 -F [!] <i>tracefromlist</i> 跟踪从一个逗号分隔的共享目标文件列表发出的调用。仅跟踪从这些共享目标文件发出的调用。缺省值为仅跟踪从主可执行对象发出的调用。仅要求提供共享目标文件的基名。例如，<i>libc</i> 将匹配 <i>/usr/lib/libc.so.1</i>。此外，还支持 shell 样式通配符（如 fnmatch(5) 中所述）。列表前面带有 "!" 表示不应跟踪从列表中目标文件发出的调用。如果要求跟踪从 <i>command</i> 发出的调用，则 <i>command</i> 必须是 <i>tracefromlist</i> 的成员。 -o <i>outputfile</i> aptrace 输出将定向到 <i>outputfile</i>。缺省情况下，aptrace 输出放置在所跟踪的进程的标准错误流上。 -t [!] <i>call</i>, ... 跟踪或排除函数调用。将跟踪逗号分隔列表 <i>call</i> 中指定的调用。如果列表以 ! 开头，将从跟踪输出中排除指定的函数调用。缺省值为 -t *。允许使用 shell 样式通配符。

- T [!]tracetolist** 跟踪对逗号分隔的共享目标文件列表的调用。缺省值为跟踪对所有共享目标文件的调用。如上所述，只需提供基名，且允许使用通配符。列表前面带有"!"表示不应跟踪对列表中目标文件的调用。
- v [!]call, ...** 提供详细的格式化的参数输出，以及所指定的函数调用的返回值（如上面的 **-t** 选项所述）。与 **truss(1)** 不同，**-v** 选项指定的调用无需再由 **-t** 选项指定。例如，**appttrace -v open** 等同于 **truss -t open -v open**。

示例

示例1 跟踪日期命令

```
% appttrace date
-> date      -> libc.so.1:atexit(0xff3bf9ac, 0x22000, 0x0) ** NR
-> date      -> libc.so.1:atexit(0x11550, 0xfefef80, 0xab268) ** NR
-> date      -> libc.so.1:setlocale(0x6, 0x11560, 0x0) ** NR
-> date      -> libc.so.1:textdomain(0x11564, 0xfefce156, 0xff160200) ** NR
-> date      -> libc.so.1:int getopt(int = 0x1,
                    const char * * = 0xffbffa5c,
                    const char * = 0x11574 "a:u")
<- date      -> libc.so.1:getopt() = 0xffffffff
-> date      -> libc.so.1:time_t time(time_t * = 0x225c0)
<- date      -> libc.so.1:time() = 0x41ab6e82
-> date      -> libc.so.1:char * nl_langinfo(nl_item = 0x3a)
<- date      -> libc.so.1:nl_langinfo() = 0xfefd3e10
-> date      -> libc.so.1:struct tm * localtime(const time_t * = 0x225c0)
<- date      -> libc.so.1:localtime() = 0xff160240
-> date      -> libc.so.1:memcpy(0xffbfff9cc, 0xff160240, 0x24) ** NR
-> date      -> libc.so.1:size_t strftime(char * = 0x225c4 "",
                    size_t = 0x400,
                    const char * = 0xfefd3e10 "%a %b %e %T %Z %Y",
                    const struct tm * = 0xffbfff9cc)
<- date      -> libc.so.1:strftime() = 0x1c
-> date      -> libc.so.1:int puts(const char * = 0x225c4
                    "Mon Nov 29 10:46:26 PST 2004")
                    Mon Nov 29 10:46:26 PST 2004
<- date      -> libc.so.1:puts() = 0x1d
-> date      -> libc.so.1:exit(0x0, 0x22400, 0x0) ** NR
```

示例2 跟踪设置了详细级别的特定接口组

```
% appttrace -v localtime, strftime, puts date
-> date      -> libc.so.1:struct tm * localtime(const time_t * = 0x225c0)
                    arg0 = (const time_t *) 0x225c0
                    return = (struct tm *) 0xff160280 (struct tm) {
                    tm_sec: (int) 0x4
                    tm_min: (int) 0x34
                    tm_hour: (int) 0xa
```

示例2 跟踪设置了详细级别的特定接口组 (续)

```

    tm_mday: (int) 0x1d
    tm_mon: (int) 0xa
    tm_year: (int) 0x68
    tm_wday: (int) 0x1
    tm_yday: (int) 0x14d
    tm_isdst: (int) 0
}
<- date      -> libc.so.1:localtime() = 0xff160280
-> date      -> libc.so.1:size_t strftime(char * = 0x225c4 "",
    size_t = 0x400,
    const char * = 0xfefd3e10 "%a %b %e %T %Z %Y",
    const struct tm * = 0xffb99c)
arg0 = (char *) 0x225c4 ""
arg1 = (size_t) 0x400
arg2 = (const char *) 0xfefd3e10 "%a %b %e %T %Z %Y"
arg3 = (const struct tm *) 0xffb99c (struct tm) {
tm_sec: (int) 0x4
tm_min: (int) 0x34
tm_hour: (int) 0xa
tm_mday: (int) 0x1d
tm_mon: (int) 0xa
tm_year: (int) 0x68
tm_wday: (int) 0x1
tm_yday: (int) 0x14d
tm_isdst: (int) 0
}
return = (size_t) 0x1c
<- date      -> libc.so.1:strftime() = 0x1c
-> date      -> libc.so.1:int puts(const char * = 0x225c4
    "Mon Nov 29 10:52:04 PST 2004")
arg0 = (const char *) 0x225c4 "Mon Nov 29 10:52:04 PST 2004"
    Mon Nov 29 10:52:04 PST 2004
return = (int) 0x1d
<- date      -> libc.so.1:puts() = 0x1d

```

** NR—不跟踪函数调用的返回值。

文件

Solaris 运行时链接程序 (`ld.so.1(1)`) 的链接审计功能提供了对 `appttrace` 的基本运行时支持，并且此工具对 `appttrace` 命令的使用依赖于保存在 `/usr/lib/abi` 中的一个审计目标文件 (`appttrace.so.1`)。

限制

通常，`appttrace` 无法跟踪对接受变量参数列表的函数的调用。在某些情况下，可通过一些巧妙的编码来解决此问题，尤其是在 `printf` 和 `scanf` 系列中。

`appttrace` 实用程序无法跟踪返回类型为 `struct` 或 `union` 的函数调用的返回值。

无法跟踪试图探测栈或提取调用者相关信息的函数。例如，`[gs]etcontext()`、`[sig]longjmp()`、`[sig]setjmp()` 和 `vfork()`。

对于 `exit(2)` 等不会返回的函数，不会跟踪其返回值。

为安全起见，只有具有相应特权的进程可以使用 `apprace` 跟踪 `setuid/setgid` 程序。

在跟踪需要包含 `<varargs.h>` 才能使用的函数（例如 `wprintw(3XCURSES)` 和 `wscanw(3XCURSES)`）时，将不会提供参数的格式化输出。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	developer/appcert（32 位） SUNWcstlx（64 位）
接口稳定性	Uncommitted（未确定）

另请参见

`ld.so.1(1)`、`truss(1)`、`wprintw(3XCURSES)`、`wscanw(3XCURSES)`、`attributes(5)`、`fnmatch(3)`

《链接程序和库指南》

引用名 `apropos` – locate commands by keyword lookup

用法概要 `apropos keyword...`

描述 The `apropos` utility displays the manual page name, section number, subsection name, the *keyword*, and a short description for each manual page that contains *keyword*.

This information is contained in the index files that are either automatically created by an SMF service as described in [man\(1\)](#) and [man\(5\)](#), or manually created using [catman\(1M\)](#) with `-w` option.

Each word is considered separately and the case of letters is ignored. Stemming on English words and section matching are also supported. Words which are part of other words are considered. For example, when looking for `compile`, `apropos` finds all instances of `compiler` as well. .

As `apropos` is simply the `-k` option to the [man\(1\)](#) command, see [man\(1\)](#) for more details.

示例 **示例 1** Finding a Manual Page with a Name Line Containing *keyword*

Try

```
example% apropos password
```

and

```
example% apropos editor
```

If the line starts *filename(section) . . .*, you can run

```
man -s section filename
```

to display the manual page for *filename*.

示例 2 Finding a Manual Page for the `printf()` Subroutine

Try

```
example% apropos format
```

and then

```
example% man -s 3s printf
```

to get the manual page on the subroutine `printf()`.

文件 `/usr/share/man/man_index/*` Table of Contents and keyword database.

Generated files include:

- `/usr/share/man/man_index/man.idx`
- `/usr/share/man/man_index/man.dic`
- `/usr/share/man/man_index/man.frq`

- /usr/share/man/man_index/man.pos

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools
CSI	Enabled
Interface Stability	Committed

另请参见

[man\(1\)](#), [whatis\(1\)](#), [catman\(1M\)](#), [attributes\(5\)](#), [man\(5\)](#)

引用名

ar – maintain portable archive or library

用法概要

```
/usr/bin/ar -d [-SVv] archive file...
/usr/bin/ar -m [-abiSVv] [posname] archive file...
/usr/bin/ar -p [-SsVv] archive [file]...
/usr/bin/ar -q [-cSVv] archive file...
/usr/bin/ar -r [-abciuSVv] [posname] archive file...
/usr/bin/ar -t [-SsVv] archive [file]...
/usr/bin/ar -x [-CSsTVv] archive [file]...
/usr/xpg4/bin/ar -d [-SVv] archive file...
/usr/xpg4/bin/ar -m [-abiSVv] [posname] archive file...
/usr/xpg4/bin/ar -p [-SsVv] archive [file]...
/usr/xpg4/bin/ar -q [-cSVv] archive file...
/usr/xpg4/bin/ar -r [-abciuSVv] [posname] archive file...
/usr/xpg4/bin/ar -t [-SsVv] archive [file]...
/usr/xpg4/bin/ar -x [-CSsTVv] archive [file]...
```

描述

The ar utility maintains groups of files combined into a single archive file. Its main use is to create and update library files. However, it can be used for any similar purpose. The magic string and the file headers used by ar consist of printable ASCII characters. If an archive is composed of printable files, the entire archive is printable.

When ar creates an archive, it creates headers in a format that is portable across all machines. The portable archive format and structure are described in detail in [ar.h\(3HEAD\)](#). The archive symbol table described there is used by the link editor [ld\(1\)](#) to effect multiple passes over libraries of object files in an efficient manner. An archive symbol table is only created and maintained by ar when there is at least one object file in the archive. The archive symbol table is in a specially named file that is always the first file in the archive. This file is never mentioned or accessible to the user. Whenever the ar command is used to create or update the contents of such an archive, the symbol table is rebuilt. The -s option described below forces the symbol table to be rebuilt.

选项

The following options are supported:

-a

Positions new files in archive after the file named by the *posname* operand.

-b

Positions new files in archive before the file named by the *posname* operand.

-
- c
Suppresses the diagnostic message that is written to standard error by default when *archive* is created.
 - C
Prevents extracted files from replacing like-named files in the file system. This option is useful when -T is also used to prevent truncated file names from replacing files with the same prefix.
 - d
Deletes one or more *files* from *archive*.
 - i
Positions new *files* in *archive* before the file named by the *posname* operand. This option is equivalent to -b.
 - m
Moves *files*. If -a, -b, or -i with the *posname* operand are specified, the -m option moves *files* to the new position. Otherwise, -m moves *files* to the end of *archive*.
 - p
Prints the contents of *files* in *archive* to standard output. If no *files* are specified, the contents of all files in *archive* are written in the order of the archive.
 - q
Quickly appends *files* to the end of *archive*. Positioning options -a, -b, and -i are invalid. The command does not check whether the added *files* are already in *archive*. This option is useful to avoid quadratic behavior when creating a large archive piece-by-piece.
 - r
Replaces or adds *files* in *archive*. If *archive* does not exist, a new archive file is created and a diagnostic message is written to standard error, unless the -c option is specified. If no *files* are specified and the *archive* exists, the results are undefined. Files that replace existing files do not change the order of the archive. If the -u option is used with the -r option, only those files with dates of modification later than the archive files are replaced. If the -a, -b, or -i option is used, the *posname* argument must be present and specifies that new files are to be placed after (-a) or before (-b or -i) *posname*. Otherwise, the new files are placed at the end.
 - s
Forces the regeneration of the archive symbol table even if ar is not invoked with an option that will modify the archive contents. This command is useful to restore the archive symbol table after the `strip(1)` command has been used on the archive.
 - S
When building the archive symbol table, force the use of the 64-bit capable symbol table format. By default, the 32-bit format is used for all archives smaller than 4GB, and the larger format is used for larger archives that exceed the 32-bit limit.

- t
Prints a table of contents of *archive*. The files specified by the *file* operands are included in the written list. If no *file* operands are specified, all files in *archive* are included in the order of the archive.
- T
Allows file name truncation of extracted files whose archive names are longer than the file system can support. By default, extracting a file with a name that is too long is an error. In that case, a diagnostic message is written and the file is not extracted.
- u
Updates older files. When used with the -r option, files within *archive* are replaced only if the corresponding *file* has a modification time that is at least as new as the modification time of the file within *archive*.
- v
Gives verbose output. When used with options -d, -r, or -x, the -v option writes a detailed file-by-file description of the archive creation and the constituent *files*, and maintenance activity. When used with -p, -v writes the name of the file to the standard output before writing the file itself to the standard output. When used with -t, -v includes a long listing of information about the files within the archive. When used with -x, -v prints the filename preceding each extraction. When writing to an archive, -v writes a message to the standard error.
- V
Prints its version number on standard error.

/usr/xpg4/bin/ar The following options are supported for /usr/xpg4/bin/ar:

- v Same as the /usr/bin/ar version, except when writing to an archive, no message is written to the standard error.
- x Extracts the files named by the *file* operands from *archive*. The contents of *archive* are not changed. If no *file* operands are given, all files in *archive* are extracted. If the file name of a file extracted from *archive* is longer than that supported in the directory to which it is being extracted, the results are undefined. The modification time of each *file* extracted is set to the time *file* is extracted from *archive*.

操作数

The following operands are supported:

- archive* A path name of the archive file.
- file* A path name. Only the last component is used when comparing against the names of files in the archive. If two or more *file* operands have the same last path name component (see [basename\(1\)](#)), the results are unspecified. The implementation's archive format will not truncate valid file names of files added to or replaced in the archive.

posname The name of a file in the archive file, used for relative positioning. See options `-m` and `-r`.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `ar`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, and `NLSPATH`.

`TMPDIR` Determine the pathname that overrides the default directory for temporary files, if any.

`TZ` Determine the timezone used to calculate date and time strings written by `ar -tv`. If `TZ` is unset or null, an unspecified default timezone is used.

退出状态

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/ar`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/linker
Interface Stability	Committed

`/usr/xpg4/bin/ar`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[basename\(1\)](#), [cpio\(1\)](#), [elffile\(1\)](#), [file\(1\)](#), [ld\(1\)](#), [lorder\(1\)](#), [strip\(1\)](#), [tar\(1\)](#), [ar.h\(3HEAD\)](#), [a.out\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注

If the same file is mentioned twice in an argument list, it may be put in the archive twice.

By convention, archives are suffixed with “`.a`”.

When inserting ELF objects into an archive file, `ar` might add `\n` characters to pad these objects to an 8-byte boundary. Such padding improves the efficiency with which [ld\(1\)](#) can access the archive. Only ELF object files are padded in this way. Other archive members are not altered. When an object with such padding is extracted from an archive, the padding is not included in the resulting output.

It is faster to create a new archive from scratch than to insert individual files into an existing archive via separate calls to `ar`. When possible, the recommended strategy is to remove the existing archive, and recreate it with a single `ar` invocation.

The overall size of an archive is allowed to exceed 4GB. However, the size of any individual file within an archive is limited to 4GB by the archive file format. See [ar.h\(3HEAD\)](#).

The maximum user ID and group ID for an individual file within an archive are limited to 6 decimal digits by the archive file format. Any file with a user or group ID greater than 999999 is quietly set to user ID “nobody” (60001) or group ID “nobody” (6001). See [ar.h\(3HEAD\)](#).

引用名 arch – display the architecture of the current host

用法概要 arch [-k | *archname*]

描述 The `arch` utility displays the application architecture of the current host system. Due to extensive historical use of this command without any options, all SunOS 5.x SPARC based systems will return "sun4" as their application architecture. Use of this command is discouraged. See NOTES section below.

Systems can be broadly classified by their *architectures*, which define what executables will run on which machines. A distinction can be made between *kernel* architecture and *application* architecture (or, commonly, just “architecture”). Machines that run different kernels due to underlying hardware differences may be able to run the same application programs.

选项 -k Displays the kernel architecture, such as sun4u. This defines which specific SunOS kernel will run on the machine, and has implications only for programs that depend on the kernel explicitly (for example, [ps\(1\)](#)).

操作数 The following operand is supported:

archname Use *archname* to determine whether the application binaries for this application architecture can run on the current host system. The *archname* must be a valid application architecture, such as sun4, i86pc, and so forth.

If *application* binaries for *archname* can run on the current host system, TRUE (0) is returned. Otherwise, FALSE (1) is returned.

退出状态 The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [mach\(1\)](#), [ps\(1\)](#), [uname\(1\)](#), [attributes\(5\)](#)

附注 This command is provided for compatibility with previous releases and its use is discouraged. Instead, the `uname` command is recommended. See [uname\(1\)](#) for usage information.

引用名

as – assembler

用法概要

SPARC

```
as [ -hwcap={1|0} ] [ -L ] [ -m ] [ -m32 ] [ -m64 ]
  [ -n ] [ -o outfile ] [ -ul ] [ -P ]
  [ -Dname ] [ -Dname=def ] [ -Ipath ]
  [ -Uname... ] [ -Q[y|n] ] [ -s ]
  [ -S[a|b|c|l|A|B|C|L] ] [ -V ] [ -xarch=v ]
  [ -xF ] [ -Y[m|c],path ] [ -YI,path ] filename...
```

x86

```
as [ -a32 ] [ -m ] [ -m32 ] [ -m64 ] [ -n ]
  [ -H ] [ -nH ] [ -o outfile ]
  [ -P ] [ -Dname ] [ -Dname=def ] [ -Ipath ]
  [ -Uname... ]
  [ -KPIC ] [ -Q[y|n] ] [ -s ] [ -S[a|b|c|l|A|B|C|L] ] [ -V ]
  [ -xchip=v ] [ -xmodel=a ]
  [ -Y[m|d],path ] [ -YI,path ] filename...
```

描述

The `as` command creates object files from assembly language source files.

选项

This section is divided into three:

- Common Options (options common to both SPARC and x86)
- SPARC Options
- x86 Options

Common Options

-Dname**-Dname=*def***

When the `-P` option is in effect, these options are passed to the `cpp(1)` preprocessor without interpretation by the `as` command; otherwise, they are ignored.

-I *path*

When the `-P` option is in effect, this option is passed to the `cpp(1)` preprocessor without interpretation by the `as` command; otherwise, it is ignored.

-i

Instructs `as` to ignore line-number information from the preprocessor.

-m

Run the `m4(1)` macro processor on the input to the assembler.

-m32|-m64

Generate 32-bit or 64-bit ELF format object code.

-n

Suppress all the warnings while assembling.

- o *outfile***
Put the output of the assembly in *outfile*. By default, the output file name is formed by removing the `.s` suffix, if there is one, from the input file name and appending an `.o` suffix.
- P**
Run `cpp(1)`, the C preprocessor, on the files being assembled. The preprocessor is run separately on each input file, not on their concatenation. The preprocessor output is passed to the assembler.
- Q[*y*|*n*]**
If the *y* option is specified, it produces the “assembler version” information in the comment section of the output object file. If the *n* option is specified, the information is suppressed.
- S[*a*|*b*|*c*|*l*|*A*|*B*|*C*|*L*]**
Produces a disassembly of the emitted code to the standard output. Adding each of the following characters to the `-S` option produces:
- a** Disassembling with address
 - b** Disassembling with `.bof`
 - c** Disassembling with comments
 - l** Disassembling with line numbers.
- Capital letters switch the corresponding option off. The default is `-Sc`.
- s**
Place all stabs in the `.stabs` section. By default, stabs are placed in `stabs.excl` sections, which are stripped out by the static linker, `ld(1)`, during final execution. When the `-s` option is used, stabs remain in the final executable because `.stab` sections are not stripped by the static linker.
- U*name***
When the `-P` option is in effect, this option is passed to the `cpp(1)` preprocessor without interpretation by the `as` command; otherwise, it is ignored.
- Y*m,path***
Specify path to the version of `m4` to use.
- YI,*path***
Indicate path to search for `#include` header files.
- SPARC Options**
- hwcap={*1*|*0*}**
Enable (*1*) or suppress (*0*) the generation of the Hardware Capabilities section. Default is to generate the section.
- L**
Save all symbols, including temporary labels that are normally discarded to save space, in the ELF symbol table.

- ul
Treat all undefined symbols as local.
- Yc,*path*
Specify path to the version of cpp to use.
- xarch=sparc
Enables the assembler to accept instructions defined in the SPARC-V9 architecture. The resulting object code is in ELF32 format when compiled with -m32, ELF64 format with -m64. It will not execute on a Oracle Solaris V8 system (a machine with a V8 processor). It will execute on a Oracle Solaris V8+ system.
- xarch=sparcvis
Enables the assembler to accept instructions defined in the SPARC-V9 architecture plus the instructions in the Visual Instruction Set (VIS) version 1.0. The resulting object code is in V8+ ELF32 format when compiled with -m32, ELF64 format with -m64. It will not execute on a Oracle Solaris system with a V8 processor. It will execute on a Oracle Solaris system with a V8+ processor.
- xarch=sparcvis2
Enables the assembler to accept instructions defined in the SPARC-V9 architecture, plus the instructions in the Visual Instruction Set (VIS) version 2.0, with UltraSPARC-III extensions. The resulting object code is in V8+ ELF32 format when compiled with -m32, ELF64 format with -m64.
- xarch=sparcvis3
Accept instructions defined for the SPARC VIS version 3 of the SPARC-V9 ISA which are instructions from the SPARC-V9 instruction set, plus the UltraSPARC extensions, including the Visual Instruction Set (VIS) version 1.0, the UltraSPARC-III extensions, including the Visual Instruction Set (VIS) version 2.0, the fused multiply-add instructions, and the Visual Instruction Set (VIS) version 3.0.
- xarch=sparcfmaf
Accept instructions defined for the `sparcfmaf` version of the SPARC-V9 ISA, plus the UltraSPARC extensions, including the Visual Instruction Set (VIS) version 1.0, the UltraSPARC-III extensions, including the Visual Instruction Set (VIS) version 2.0, and the SPARC64 VI extensions for floating-point multiply-add.
- xarch=sparcima
Accept instructions defined for the `sparcima` version of the SPARC-V9 ISA which are instructions from the SPARC-V9 instruction set, plus the UltraSPARC extensions, including the Visual Instruction Set (VIS) version 1.0, the UltraSPARC-III extensions, including the Visual Instruction Set (VIS) version 2.0, the SPARC64 VI extensions for floating-point multiply-add, and the SPARC64 VII extensions for integer multiply-add.
- xarch=sparc4
Accept instructions defined for the `sparc4` version of the SPARC-V9 ISA, which are instructions from the SPARC-V9 instruction set, plus the extensions, which includes VIS

1.0, the UltraSPARC-III extensions, which includes VIS 2.0, the fused floating-point multiply-add instructions, VIS 3.0, and SPARC4 instructions.

-xarch=v9

Equivalent to: -m64 -xarch=sparc

-xarch=v9a

Equivalent to: -m64 -xarch=sparcv9a

-xarch=v9b

Equivalent to: -m64 -xarch=sparcv9b

-xF

Generates additional information for use by the Oracle Solaris Studio performance Analyzer. If the input file does not contain any stabs (debugging directives), then the assembler will generate the default stabs needed by the Oracle Solaris Studio analyzer. Also see the dbx(1) Oracle Sun Studio manual page.

x86 Options

-a32

Allow 32-bit addresses in 64-bit mode.

-H

Generate the Hardware Capabilities section. (This is the default.)

-nH

Suppress the generation of the Hardware Capabilities section.

-KPIC

Check for address referencing with absolute relocation and issue warning.

-xchip=v

When there is a choice between several possible encodings, choose the one that is appropriate for the stated chip. In particular, use the appropriate no-op byte sequence to fill code alignment padding, and warn when instructions not defined for the stated chip are used.

The assembler accepts the instruction sets for the following recognized -xchip values:

generic	generic x86 instruction set.
native	this host processor.
core2	Intel Core2 processor.
nehalem	Intel Nehalem processor.
opteron	AMD Opteron processor.
penryn	Intel Penryn processor.
pentium	Intel Pentium architecture.
pentium_pro	Intel Pentium Pro architecture.

pentium3 Intel Pentium 3 style processor.
 pentium4 Intel Pentium 4 style processor.
 sandybridge Intel Sandy Bridge processor.
 westmere Intel Westmere processor.
 amdfam10 AMD FAM10 processor.
 ivybridge Intel Ivy Bridge processor.
 haswell Intel Haswell processor.

-xmodel=[small | medium | kernel]

For -m64 only, generate R_X86_64_32S relocatable type for data access under kernel.

Otherwise, generate R_X86_64_32 under small. SHN_AMD64_LCOMMON and .lbcomm support added under medium. small is the default.

-Yd,path

Specify path to the version of cm4defs to use.

环境变量

TMPDIR

as normally creates temporary files in the directory /tmp. You may specify another directory by setting the environment variable TMPDIR to your chosen directory. (If TMPDIR is not a valid directory, then as will use /tmp).

文件

By default, as creates its temporary files in /tmp.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/
Interface Stability	Committed

另请参见

[cpp\(1\)](#), [ld\(1\)](#), [m4\(1\)](#), [nm\(1\)](#), [strip\(1\)](#), [tmpnam\(3C\)](#), [a.out\(4\)](#), [attributes\(5\)](#)

附注

On SPARC platforms, the cpp symbol `__sparc` is set when the flag -P appears, as well as `__sparcv8` with the -m32 flag, and `__sparcv9` with the -m64 flag.

On x86/x64, the symbol `__i386` is set when the flag -P appears, as well as `__amd64` with the -m64 flag.

If the -m (invoke the [m4\(1\)](#) macro processor) option is used, keywords for m4 cannot be used as symbols (variables, functions, labels) in the input file since m4 cannot determine which keywords are assembler symbols and which keywords are real m4 macros.

Whenever possible, you should access the assembler through a compilation system interface program such as the Oracle Solaris Studio C compiler, cc, to ensure proper library linking. See the [cc\(1\)](#) Oracle Solaris Studio man page.

引用名	asa – convert FORTRAN carriage-control output to printable form
用法概要	asa [-f] [<i>file</i>]...
描述	<p>The <code>asa</code> utility will write its input files to standard output, mapping carriage-control characters from the text files to line-printer control sequences.</p> <p>The first character of every line will be removed from the input, and the following actions will be performed.</p> <p>If the character removed is:</p> <p>SPACE The rest of the line will be output without change.</p> <p>0 It is replaced by a NEWLINE control sequence followed by the rest of the input line.</p> <p>1 It is replaced by a NEWPAGE control sequence followed by the rest of the input line.</p> <p>+ It is replaced by a control sequence that causes printing to return to the first column of the previous line, where the rest of the input line is printed.</p> <p>For any other character in the first column of an input line, <code>asa</code> skips the character and prints the rest of the line unchanged.</p> <p>If <code>asa</code> is called without providing a <i>filename</i>, the standard input is used.</p>
选项	<p>The following option is supported:</p> <p>-f Start each file on a new page.</p>
操作数	<p>The following operand is supported:</p> <p><i>file</i> A pathname of a text file used for input. If no <i>file</i> operands are specified, or <code>-</code> is specified, the standard input will be used.</p>
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of <code>asa</code> : LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.
退出状态	<p>The following exit values are returned:</p> <p>0 All input files were output successfully.</p> <p>>0 An error occurred.</p>
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名

at, batch – execute commands at a later time

用法概要

```

/usr/bin/at [-c | -k | -s] [-m] [-f file] [-p project]
           [-q queueName] -t time

/usr/bin/at [-c | -k | -s] [-m] [-f file] [-p project]
           [-q queueName] timespec...

/usr/bin/at -l [-p project] [-q queueName] [at_job_id. ...]

/usr/bin/at -r at_job_id. ..

/usr/bin/batch [-p project]

/usr/xpg4/bin/at [-c | -k | -s] [-m] [-f file] [-p project]
                [-q queueName] -t time

/usr/xpg4/bin/at [-c | -k | -s] [-m] [-f file] [-p project]
                [-q queueName] timespec...

/usr/xpg4/bin/at -l [-p project] [-q queueName]
                [at_job_id. ...]

/usr/xpg4/bin/at -r at_job_id. ..

/usr/xpg4/bin/batch [-p project]

```

描述

at

The at utility reads commands from standard input and groups them together as an *at-job*, to be executed at a later time.

The at-job is executed in a separate invocation of the shell, running in a separate process group with no controlling terminal, except that the environment variables, current working directory, file creation mask (see [umask\(1\)](#)), and system resource limits (for sh and ksh88 only, see [ulimit\(1\)](#)) in effect when the at utility is executed is retained and used when the at-job is executed.

When the at-job is submitted, the *at_job_id* and scheduled time are written to standard error. The *at_job_id* is an identifier that is a string consisting solely of alphanumeric characters and the period character. The *at_job_id* is assigned by the system when the job is scheduled such that it uniquely identifies a particular job.

User notification and the processing of the job's standard output and standard error are described under the -m option.

Users are permitted to use at and batch (see below) if their name appears in the file `/usr/lib/cron/at.allow`. If that file does not exist, the file `/usr/lib/cron/at.deny` is checked to determine if the user should be denied access to at. If neither file exists, only a user with the `solaris.jobs.user` authorization is allowed to submit a job. If only `at.deny` exists and is empty, global usage is permitted. The `at.allow` and `at.deny` files consist of one user name per line.

cron and at jobs are not be executed if the user's account is locked. Only accounts which are not locked as defined in [shadow\(4\)](#) will have their job or process executed.

batch

The batch utility reads commands to be executed at a later time.

Commands of the forms:

```
/usr/bin/batch [-p project]
/usr/xpg4/bin/batch [-p project]
```

are respectively equivalent to:

```
/usr/bin/at -q b [-p project] now
/usr/xpg4/bin/at -q b -m [-p project] now
```

where queue b is a special at queue, specifically for batch jobs. Batch jobs are submitted to the batch queue for immediate execution. Execution of submitted jobs can be delayed by limits on the number of jobs allowed to run concurrently. See [queuedefs\(4\)](#).

选项

If the -c, -k, or -s options are not specified, the SHELL environment variable by default determines which shell to use.

For `/usr/xpg4/bin/at` and `/usr/xpg4/bin/batch`, if SHELL is unset or NULL, `/usr/xpg4/bin/sh` is used.

For `usr/bin/at` and `usr/bin/batch`, if SHELL is unset or NULL, `/bin/sh` is used.

The following options are supported:

- c C shell. [csh\(1\)](#) is used to execute the at-job.
- k Korn shell. `/bin/ksh` is used to execute the at-job.
- s Bourne shell. [sh\(1\)](#) is used to execute the at-job.
- f *file* Specifies the path of a file to be used as the source of the at-job, instead of standard input.
- l (The letter ell.) Reports all jobs scheduled for the invoking user if no *at_job_id* operands are specified. If *at_job_ids* are specified, reports only information for these jobs.
- m Sends mail to the invoking user after the at-job has run, announcing its completion. Standard output and standard error produced by the at-job are mailed to the user as well, unless redirected elsewhere. Mail is sent even if the job produces no output.

If -m is not used, the job's standard output and standard error is provided to the user by means of mail, unless they are redirected elsewhere; if there is no such output to provide, the user is not notified of the job's completion.

- p project* Specifies under which project the at or batch job is run. When used with the *-l* option, limits the search to that particular project. Values for *project* is interpreted first as a project name, and then as a possible project ID, if entirely numeric. By default, the user's current project is used.
- q queueName* Specifies in which queue to schedule a job for submission. When used with the *-l* option, limits the search to that particular queue. Values for *queueName* are limited to the lower case letters a through z. By default, at-jobs are scheduled in queue a. In contrast, queue b is reserved for batch jobs. Since queue c is reserved for cron jobs, it can not be used with the *-q* option.
- r at_job_id* Removes the jobs with the specified *at_job_id* operands that were previously scheduled by the at utility.
- t time* Submits the job to be run at the time specified by the *time* option-argument, which must have the format as specified by the [touch\(1\)](#) utility.

操作数

The following operands are supported:

- at_job_id* The name reported by a previous invocation of the at utility at the time the job was scheduled.
- timespec* Submit the job to be run at the date and time specified. All of the *timespec* operands are interpreted as if they were separated by space characters and concatenated. The date and time are interpreted as being in the timezone of the user (as determined by the TZ variable), unless a timezone name appears as part of *time* below.

In the C locale, the following describes the three parts of the time specification string. All of the values from the LC_TIME categories in the C locale are recognized in a case-insensitive manner.

- time* The *time* can be specified as one, two or four digits. One- and two-digit numbers are taken to be hours, four-digit numbers to be hours and minutes. The time can alternatively be specified as two numbers separated by a colon, meaning *hour:minute*. An AM/PM indication (one of the values from the *am_pm* keywords in the LC_TIME locale category) can follow the time; otherwise, a 24-hour clock time is understood. A timezone name of GMT, UCT, or ZULU (case insensitive) can follow to specify that the time is in Coordinated Universal Time. Other timezones can be specified using the TZ environment variable. The *time* field can also be one of the following tokens in the C locale:

midnight Indicates the time 12:00 am (00:00).

noon Indicates the time 12:00 pm.
now Indicate the current day and time. Invoking at now submits an at-job for potentially immediate execution (that is, subject only to unspecified scheduling delays).

date An optional *date* can be specified as either a month name (one of the values from the mon or abmon keywords in the LC_TIME locale category) followed by a day number (and possibly year number preceded by a comma) or a day of the week (one of the values from the day or abday keywords in the LC_TIME locale category). Two special days are recognized in the C locale:

today Indicates the current day.

tomorrow Indicates the day following the current day.

If no *date* is given, today is assumed if the given time is greater than the current time, and tomorrow is assumed if it is less. If the given month is less than the current month (and no year is given), next year is assumed.

increment The optional *increment* is a number preceded by a plus sign (+) and suffixed by one of the following: minutes, hours, days, weeks, months, or years. (The singular forms are also accepted.) The keyword next is equivalent to an increment number of + 1. For example, the following are equivalent commands:

```
at 2pm + 1 week
at 2pm next week
```

用法

The format of the at command line shown here is guaranteed only for the C locale. Other locales are not supported for midnight, noon, now, mon, abmon, day, abday, today, tomorrow, minutes, hours, days, weeks, months, years, and next.

Since the commands run in a separate shell invocation, running in a separate process group with no controlling terminal, open file descriptors, traps and priority inherited from the invoking environment are lost.

示例

```
at 示例1 Typical Sequence at a Terminal
This sequence can be used at a terminal:
$ at -m 0730 tomorrow
sort < file >outfile
<EOT>
```

示例2 Redirecting Output

This sequence, which demonstrates redirecting standard error to a pipe, is useful in a command procedure (the sequence of output redirection specifications is significant):

```
$ at now + 1 hour <<!
diff file1 file2 2>&1 >outfile | mailx mygroup
```

示例3 Self-rescheduling a Job

To have a job reschedule itself, `at` can be invoked from within the `at`-job. For example, this daily-processing script named `my.daily` runs every day (although `crontab` is a more appropriate vehicle for such work):

```
# my.daily runs every day
at now tomorrow < my.daily
daily-processing
```

示例4 Various Time and Operand Presentations

The spacing of the three portions of the C locale *timespec* is quite flexible as long as there are no ambiguities. Examples of various times and operand presentations include:

```
at 0815am Jan 24
at 8 :15amjan24
at now "+ 1day"
at 5 pm FRIDay
at '17
    utc+
    30minutes'
```

batch

示例5 Typical Sequence at a Terminal

This sequence can be used at a terminal:

```
$ batch
sort <file >outfile
<EOT>
```

示例6 Redirecting Output

This sequence, which demonstrates redirecting standard error to a pipe, is useful in a command procedure (the sequence of output redirection specifications is significant):

```
$ batch <<!
diff file1 file2 2>&1 >outfile | mailx mygroup
!
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `at` and `batch`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, `NLSPATH`, and `LC_TIME`.

- DATEMSK** If the environment variable DATEMSK is set, `at` uses its value as the full path name of a template file containing format strings. The strings consist of format specifiers and text characters that are used to provide a richer set of allowable date formats in different languages by appropriate settings of the environment variable LANG or LC_TIME. The list of allowable format specifiers is located in the [getdate\(3C\)](#) manual page. The formats described in the OPERANDS section for the *time* and *date* arguments, the special names noon, midnight, now, next, today, tomorrow, and the *increment* argument are not recognized when DATEMSK is set.
- SHELL** Determine a name of a command interpreter to be used to invoke the at-job. If the variable is unset or NULL, sh is used. If it is set to a value other than sh, the implementation uses that shell; a warning diagnostic is printed telling which shell will be used.
- TZ** Determine the timezone. The job is submitted for execution at the time specified by *timespec* or `-t time` relative to the timezone specified by the TZ variable. If *timespec* specifies a timezone, it overrides TZ. If *timespec* does not specify a timezone and TZ is unset or NULL, an unspecified default timezone is used.

退出状态

The following exit values are returned:

- 0 The `at` utility successfully submitted, removed or listed a job or jobs.
- >0 An error occurred, and the job will not be scheduled.

文件

`/usr/lib/cron/at.allow` names of users, one per line, who are authorized access to the `at` and `batch` utilities

`/usr/lib/cron/at.deny` names of users, one per line, who are denied access to the `at` and `batch` utilities

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/at`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Not enabled
Interface Stability	Committed
Standard	See standards(5) .

`/usr/xpg4/bin/at`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Not enabled
Interface Stability	Standard

/usr/bin/batch

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Standard

/usr/xpg4/bin/batch

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Standard

另请参见

[auths\(1\)](#), [crontab\(1\)](#), [csh\(1\)](#), [date\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [touch\(1\)](#), [ulimit\(1\)](#), [umask\(1\)](#), [cron\(1M\)](#), [getdate\(3C\)](#), [auth_attr\(4\)](#), [shadow\(4\)](#), [queuedefs\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注

Regardless of queue used, [cron\(1M\)](#) has a limit of 100 jobs in execution at any time.

There can be delays in `cron` at job execution. In some cases, these delays can compound to the point that `cron` job processing appears to be hung. All jobs are executed eventually. When the delays are excessive, the only workaround is to kill and restart `cron`.

引用名 atq – display the jobs queued to run at specified times

用法概要 atq [-c] [-n] [username]...

描述 The atq utility displays the at jobs queued up for the current user. [at\(1\)](#) is a utility that allows users to execute commands at a later date. If invoked by a user with the `solaris.jobs.admin` authorization, atq will display all jobs in the queue.

If no options are given, the jobs are displayed in chronological order of execution.

When an authorized user invokes atq without specifying *username*, the entire queue is displayed; when a *username* is specified, only those jobs belonging to the named user are displayed.

选项 The following options are supported:

-c Displays the queued jobs in the order they were created (that is, the time that the at command was given).

-n Displays only the total number of jobs currently in the queue.

文件 /var/spool/cron/atjobs spool area for at jobs.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [at\(1\)](#), [atrm\(1\)](#), [auths\(1\)](#), [cron\(1M\)](#), [auth_attr\(4\)](#), [attributes\(5\)](#)

引用名	atrm – remove jobs spooled by at or batch				
用法概要	atrm [-afi] [[job #] [user]...]				
描述	<p>The <code>atrm</code> utility removes delayed-execution jobs that were created with the at(1) command, but have not yet executed. The list of these jobs and associated job numbers can be displayed by using atq(1).</p> <p><code>atrm</code> removes each job-number you specify, and/or all jobs belonging to the user you specify, provided that you own the indicated jobs.</p> <p>You can only remove jobs belonging to other users if you have <code>solaris.jobs.admin</code> privileges.</p>				
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -a All. Removes all unexecuted jobs that were created by the current user. If invoked by the privileged user, the entire queue is flushed. -f Force. All information regarding the removal of the specified jobs is suppressed. -i Interactive. <code>atrm</code> asks if a job should be removed. If the response is affirmative, the job is removed. 				
文件	<code>/var/spool/cron/atjobs</code> Spool area for at jobs				
环境变量	<p>See environ(5) for descriptions of the following environment variables that affect the execution of <code>atrm</code>: <code>LANG</code>, <code>LC_ALL</code>, <code>LC_COLLATE</code>, <code>LC_CTYPE</code>, <code>LC_MESSAGES</code>, and <code>NLSPATH</code>.</p> <p>Affirmative responses are processed using the extended regular expression defined for the <code>yesexpr</code> keyword in the <code>LC_MESSAGES</code> category of the user's locale. The locale specified in the <code>LC_COLLATE</code> category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for <code>yesexpr</code>. The locale specified in <code>LC_CTYPE</code> determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the <code>yesexpr</code>. See locale(5)</p>				
属性	See attributes(5) for descriptions of the following attributes:				
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>system/core-os</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	system/core-os
ATTRIBUTE TYPE	ATTRIBUTE VALUE				
Availability	system/core-os				
另请参见	at(1) , atq(1) , auths(1) , cron(1M) , auth_attr(4) , attributes(5) , environ(5) , locale(5)				

引用名 audioconvert – 转换音频文件格式

用法概要 audioconvert [-pF] [-f *outfmt*] [-o *outfile*]
[[-i *infmt*] [*file*]...] ...

描述 `audioconvert` 用于在一组受支持的音频编码和文件格式之间转换音频数据。该命令可用于压缩和解压缩音频数据，向原始音频数据文件添加音频文件头，以及在标准数据编码（例如 `-law` 和线性 PCM）之间进行转换。

如果未提供文件名，`audioconvert` 将从标准输入流中读取数据，然后将音频文件写入标准输出。否则，将按顺序处理输入文件，进行串联，然后写入输出文件。

输入文件应包含用于识别音频数据格式的音频文件头。如果音频数据不包含可识别的文件头，则必须通过 `-i` 选项指定格式，使用 `rate`、`encoding` 和 `channels` 关键字来识别输入数据格式。

输出文件格式是通过在 `-f` 规范中使用格式选项更新第一个输入文件的格式派生而来的。如果不指定 `-p`，所有后续的输入文件都将转换为所得到的这一格式，然后串联在一起。输出文件将包含音频文件头，除非在输出格式选项中指定了 `format=raw`。

可以使用 `-p` 选项就地转换输入文件。当 `-p` 起作用时，可根据 `-f` 选项修改每个输入文件的格式以确定输出格式。然后，将使用转换后的数据覆盖现有文件。

`file(1)` 命令可解码和输出 Sun 音频文件的音频数据格式。

选项 支持以下选项：

- `-p` *In Place*: 输入文件分别转换为 `-f` 选项所指定的格式，并进行重写。如果目标文件是符号链接，将重写底层文件。`-o` 选项不能与 `-p` 一起指定。
- `-F` *Force*: 此选项可强制 `audioconvert` 忽略由 `-i` 选项指定了格式的输入文件的任何文件头。如果未指定 `-F`，`audioconvert` 将对包含有效音频文件头的输入文件忽略 `-i` 选项。
- `-f outfmt` *Output Format*: 此选项用于指定输出文件的文件格式和数据编码。未指定的字段的缺省值派生自输入文件格式。下一部分列出了有效的关键字和值。
- `-o outfile` *Output File*: 所有输入文件将进行串联，转换为输出格式，然后写入所指定的输出文件。如果未指定 `-o` 和 `-p`，串联的输出将写入标准输出。`-p` 选项不能与 `-o` 一起指定。
- `-i infmt` *Input Format*: 此选项用于指定原始输入文件的数据编码。通常，输入数据格式派生自音频文件头。在转换开头没有有效音频文件头的音频数据时，需要使用此选项。如果为包含音频文件头的输入文件指定了 `-i`，将忽略输入格式字符串，除非指定了 `-F` 选项。格式规范语法与 `-f` 输出文件格式相同。

可以指定多个输入格式。输入格式规定所有输入文件都要遵守该规范，直到指定了新的输入格式。

file *File Specification*: 所指定的音频文件将进行串联, 转换为输出格式, 然后写出。如果未提供文件名, 或者指定了特殊的文件名 '-', 将从标准输入读取音频数据。

-? *Help*: 输出命令行用法消息。

格式规范

输入和输出格式规范的语法为:

keyword=value[,keyword=value ...]

中间没有空格。可以在开头没有 *keyword=* 的情况下使用明确的值。

rate 音频抽样率以每秒的样例数进行指定。如果一个数字后面跟有字母 *k*, 表示乘以 1000 (例如 44.1k = 44100)。常用抽样率的标准有: 8k、16k、32k、44.1k 和 48k。

channels 使用一个整数指定交错声道的数量。还可以使用 *mono* 和 *stereo* 这两个词来分别指定单声道和双声道数据。

encoding 此选项用于指定数字音频数据的表现形式。编码可隐式指定精度 (*ulaw* 表示 8 位精度), 或者在名称中显式指定精度 (例如, *linear16*)。有效的编码值为:

ulaw CCITT G.711 -law 编码。这是主要用于电话音质语音的 8 位格式。

alaw CCITT G.711 A-law 编码。这是欧洲主要用于电话音质语音的 8 位格式。

linear8,
linear16,
linear32

线性脉冲编码调制 (Pulse Code Modulation, PCM) 编码。该名称可标识精度位数。*linear16* 通常用于高质量音频数据。

pcm 同 *linear16*。

g721 CCITT G.721 压缩格式。此编码使用具有 4 位精度的自适应差分脉冲编码调制 (Adaptive Delta Pulse Code Modulation, ADPCM)。该编码主要用于压缩 -law 语音数据 (可达到 2:1 的压缩率)。

g723 CCITT G.723 压缩格式。此编码使用具有 3 位精度的自适应差分脉冲编码调制 (Adaptive Delta Pulse Code Modulation, ADPCM)。该编码主要用于压缩 -law 语音数据 (可达到 8:3 的压缩率)。音频质量类似于 G.721, 但在用于非语音数据时质量较低。

还可以使用以下编码值作为速记方式来设置抽样率、声道和编码:

voice 等同于 `encoding=ulaw,rate=8k,channels=mono`。
 cd 等同于 `encoding=linear16,rate=44.1k,channels=stereo`。
 dat 等同于 `encoding=linear16,rate=48k,channels=stereo`。

格式化 此选项用于指定音频文件格式。有效格式有：

sun Sun 兼容文件格式（缺省）。
 raw 在读取或写入原始音频数据（没有音频头）时可使用此格式；该格式也可以与 `offset` 结合使用以导入外来音频文件格式。

`offset` (*-i only*) 指定字节偏移以定位音频数据的开始位置。此选项可用于导入包含无法识别的文件头的音频数据。

用法 当遇到的文件大于或等于 2 GB (2^{31} 字节) 时，有关 `audioconvert` 行为的说明，请参见 [largefile\(5\)](#)。

示例

示例 1 在存储语音数据之前进行录制和压缩

在将语音数据存储到文件中之前对其进行录制和压缩：

```
example% audiorecord | audioconvert -f g721 > mydata.au
```

示例 2 串联两个音频文件

串联两个 Sun 格式的音频文件（不管是何种数据格式），然后输出一个 8 位 ulaw、16 kHz、单声道文件：

```
example% audioconvert -f ulaw,rate=16k,mono -o outfile.au infile1 infile2
```

示例 3 将目录转换为 Sun 格式

将包含原始语音数据文件的目录就地转换为 Sun 格式（向每个文件添加文件头）：

```
example% audioconvert -p -i voice -f sun *.au
```

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
体系结构	SPARC、x86
可用性	audio/audio-utilities
接口稳定性	Committed（已确定）

另请参见

[audioplay\(1\)](#)、[audiorecord\(1\)](#)、[file\(1\)](#)、[attributes\(5\)](#)、[largefile\(5\)](#)

附注

只需将声道相加，即可实施用于将多声道数据转换为单声道数据的算法。如果输入数据完全同相（如同单声道文件转换为立体声又转换回单声道的情况），得到的数据可能会有些失真。

引用名 audioctl – 音频混音器控制命令行应用程序

用法概要

```
audioctl list-devices
audioctl show-device [-v] [-d device]
audioctl show-control [-v] [-d device] [control ...]
audioctl set-control [-v] [-d device] control value
audioctl save-controls [-d device] [-f file]
audioctl load-controls [-d device] file
```

描述 audioctl 命令用于控制音频混音器的各种功能以及获取有关音频混音器和音频设备的信息。可以对以下数据类型运行 audioctl 命令：

device 音频设备，例如 `audiohd#0`。接受此设备的子命令会将其作为选项 `-d` 的参数。如果未提供音频设备，则假定使用缺省音频设备。与音频设备关联的任何设备节点也正常运行，例如 `/dev/sound/0`、`/dev/dsp1` 或 `/dev/audio`。

control 混音器控制名称，例如 `volume`。

value 控制的值。具体格式取决于控制的类型。单声道值通常使用 `0` 到 `100`（含 `0` 和 `100`）之间的一个整数。立体声值使用一对此类数字，表示左右两个声道。布尔值指示 `on` 或 `off`。枚举采用一个或多个名称的单个值。

file 控制设置的 ASCII 文本文件。

选项 每个子命令都具有自己的专用选项集合。但是，某些子命令支持特殊标志 `-v`，该标志表示要求更详细的输出。

子命令 支持以下子命令：

```
audioctl list-devices
    列出系统上的所有音频设备。
```

```
audioctl show-device [-v] [-d devices]
    显示有关设备的常规信息。
```

```
audioctl show-control [-v] [-d device] [control ..]
    显示设备的控制设置值。显示指定的控制。如果未提供控制名称，则显示所有控制值。
```

```
audioctl set-control [-v] [-d device] control value
    将控制值更改为所提供的值。
```

```
audioctl save-controls [-f] [-d device] file
    将所有混音器控制值的当前状态保存到指定文件。如果该文件已存在，此命令会安全中止，除非指定 -f。
```

```
audioctl load-controls [-d device] file
    针对所有混音器控制恢复先前保存在指定文件中的状态。
```


环境变量 **AUDIODEV** 如果未指定 `-d` 和 `-a` 选项，则参考 **AUDIODEV** 环境变量。如果已设置这两个选项，**AUDIODEV** 将包含用户的缺省音频设备的全路径名。

文件 `/dev/audiocctl /dev/sound/{0..n}ctl`

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
体系结构	SPARC、x86
可用性	system/io/audio
接口稳定性	请参见下文。

`audiocctl` 命令及其子命令是 "Committed"（已确定）。用户可读的输出 "Not An Interface"（不是接口）。设备名称、控制名称和值是 "Uncommitted"（未确定）。`save-controls` 和 `load-controls` 子命令使用的状态文件的格式“已提交为专用”。

另请参见 [audioconvert\(1\)](#)、[audioplay\(1\)](#)、[audiorecord\(1\)](#)、[open\(2\)](#)、[attributes\(5\)](#)

引用名 `audioplay` – 播放音频文件

用法概要 `audioplay [-iV] [-v vol] [-d dev] [file]...`

描述 `audioplay` 实用程序将指定的音频文件复制到音频设备（如果未提供文件名，则将标准输入复制到音频设备）。如果未指定输入文件且标准输入为 `tty`，该程序将退出并显示一条错误消息。

输入文件必须包含有效的音频文件头。此文件头中的编码信息与音频设备的功能相匹配，如果数据格式不兼容，则会显示一条错误消息且跳过此文件。压缩的 ADPCM (G.721) 单声道音频数据在播放之前会自动进行解压缩。

通常会忽略抽样频率中的微小偏差（即，小于 1%）。例如，允许抽样频率为 8012 Hz 的数据在仅支持 8000 Hz 的音频设备上播放。如果存在 `-v` 选项，则会以警告消息标记此类偏差。

选项 支持以下选项：

`-d dev` *Device*: `dev` 参数指定输出应定向到的备用音频设备。如果未指定 `-d` 选项，则参考 `AUDIODEV` 环境变量（请参见下文）。否则，`/dev/audio` 会用作缺省音频设备。

`-i` *Immediate*: 如果音频设备不可用（即，另一个进程当前已进行写访问），`audioplay` 通常会等待，直到其可以访问设备为止。当存在 `-i` 选项时，若设备处于忙碌状态，`audioplay` 会显示一条错误消息并立即退出。

`-v vol` *Volume*: 在开始播放之前将输出音量设置为指定的值，在 `audioplay` 退出后将输出音量重置为先前的级别。`vol` 参数是 0 到 100（包含 0 和 100）之间的一个整数值。如果未指定此参数，输出音量会保持最近由任何进程设置的级别。

`-V` *Verbose*: 显示在等待访问音频设备或检测到抽样率偏差时出现的标准错误的相关消息。

`-\?` *Help*: 输出命令行用法消息。

操作数 `file` *File Specification*: 按顺序播放命令行上指定的音频文件。如果未提供文件名，则播放标准输入流（如果不是 `tty`），输入流也必须包含音频文件头。特殊文件名 `-` 可用于读取标准输入流而不是文件。如果提供的是相对路径名，则参考 `AUDIOPATH` 环境变量（请参见下文）。

用法 当遇到的文件大于或等于 2 GB (2^{31} 字节) 时，有关 `audioplay` 行为的说明，请参见 [largefile\(5\)](#)。

环境变量 `AUDIODEV` 要写入的音频设备的全路径名（如果未提供 `-d` 参数）。如果未设置 `AUDIODEV` 变量，将使用 `/dev/audio`。

`AUDIOPATH` 以冒号分隔的目录列表，要在这些目录中搜索使用相对路径名提供名称的音频文件。可在搜索路径中显式指定当前目录 (`.`)。如果未设置 `AUDIOPATH` 变量，将仅在当前目录中进行搜索。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
体系结构	SPARC、x86
可用性	audio/audio-utilities
接口稳定性	Committed（已确定）

另请参见 [audioconvert\(1\)](#)、[audiocctl\(1\)](#)、[audiorecord\(1\)](#)、[attributes\(5\)](#)、[largefile\(5\)](#)、[audio\(7I\)](#)

已知问题 `audioplay` 当前支持一组限定的音频格式转换。如果音频文件的格式不受音频设备支持，则必须先对其进行转换。例如，要转换成动态语音格式，请使用以下命令：

```
example% audioconvert -f voice myfile | audioplay
```

格式转换无法始终与音频输出保持同步。在这种情况下，您应该在播放数据之前转换成临时文件。

引用名 `audiorecord` – 录制音频文件

用法概要 `audiorecord [-af] [-v vol] [-c channels] [-s rate]`
 `[-e encoding] [-t time] [-i info] [-d dev]`
 `[-T au | aif[f] | wav] [file[.au|.aif[f]]|.wav]`

描述 `audiorecord` 实用程序将音频数据从音频设备复制到指定的音频文件，或者在未提供文件名的情况下复制到标准输出。如果未指定输出文件且标准输出为 `tty`，该程序将退出并显示一条错误消息。

缺省情况下，单声道音频数据以 8 kHz 的频率进行录制，且以 `law` 格式进行编码。如果音频设备支持其他配置，可以使用 `-c`、`-s` 和 `-e` 选项指定数据格式。输出文件前附带音频文件头，该文件头用于识别文件中的编码数据格式。

立即开始录制，并一直录制直到收到 `SIGINT` 信号（例如 `Ctrl-C`）为止。如果指定了 `-t` 选项，`audiorecord` 会在录制了指定数量的数据后停止。

如果音频设备不可用，即，如果另一个进程当前已进行读访问，`audiorecord` 会显示一条错误消息并立即退出。

选项 支持以下选项：

- `- \?` *Help*: 输出命令行用法消息。
- `-a` *Append*: 在指定音频文件的末尾附加数据。音频设备必须支持现有文件的音频数据格式。
- `-c channels` *Channels*: 指定音频通道的数量（1 或 2）。可将该值指定为一个整数或字符串 `mono` 或 `stereo`。缺省值为 `mono`。
- `-d dev` *Device*: `dev` 参数指定应从其获取输入的备用音频设备。如果未指定 `-d` 选项，则参考 `AUDIODEV` 环境变量（请参见下文）。否则，`/dev/audio` 会用作缺省音频设备。
- `-e encoding` *Encoding*: 指定音频数据编码。该值可以是 `ulaw`、`alaw` 或 `linear` 之一。缺省编码是 `ulaw`。
- `-f` *Force*: 若指定了 `-a` 标志，则音频设备的抽样率必须与录制原始文件的抽样率匹配。如果还指定了 `-f` 标志，则忽略抽样率差异，但会显示一条有关标准错误的警告消息。
- `-i info` *Information*: 输出文件头的 'information'（信息）字段会设置为 `info` 参数所指定的字符串。不能将此选项指定为与 `-a` 参数一起使用。
- `-s rate` *Sample Rate*: 指定抽样率，以每秒的样例数为单位。如果一个数字后面跟有字母 `k`，表示乘以 1000（例如 `44.1k = 44100`）。缺省抽样率为 8 kHz。

	<code>-t time</code>	<i>Time</i> : <code>time</code> 参数指定录制的最长时间。可将时间指定为表示秒数的浮点值, 或者指定为 <code>hh:mm:ss.dd</code> 形式 (其中, 小时和分钟规范为可选)。
	<code>-T au aif[f] wav</code>	指定要创建的音频文件类型。如果使用 <code>-a</code> 选项, 文件类型必须与其将附加到的文件匹配。无论文件后缀为何, 都要设置为该选项中指定的类型。如果未指定此选项, 则由文件后缀确定文件类型。
	<code>-v vol</code>	<i>Volume</i> : 在开始录制之前将录制增益设置为指定的值, 在 <code>audiorecord</code> 退出后将录制增益重置为先前的级别。 <code>vol</code> 参数是 0 到 100 (包含 0 和 100) 之间的一个整数值。如果未指定此参数, 输入音量会保持最近由任何进程设置的级别。
操作数	<code>file[.au .aif[f]].wav</code>	<i>File Specification</i> : 重写或附加指定的音频文件。如果未提供文件名且标准输出不是 <code>tty</code> , 或者如果指定了特殊文件名“-”, 则输出会定向到标准输出。 如果未指定 <code>-T</code> 选项, 则由文件后缀确定文件类型。如果无法识别后缀, 则缺省值为 <code>.au</code> 。如果指定 <code>-T</code> 选项, 则无论文件后缀为何, 都将使用该文件类型。
用法		当遇到的文件大于或等于 2 GB (2^{31} 字节) 时, 有关 <code>audiorecord</code> 行为的说明, 请参见 largefile(5) 。
环境变量	<code>AUDIODEV</code>	要从其录制的音频设备的全路径名 (如果未提供 <code>-d</code> 参数)。如果未设置 <code>AUDIODEV</code> 变量, 将使用 <code>/dev/audio</code> 。
属性		有关下列属性的说明, 请参见 attributes(5) :

属性类型	属性值
体系结构	SPARC、x86
可用性	audio/audio-utilities
接口稳定性	Committed (已确定)

另请参见 [audioconvert\(1\)](#)、[audiocctl\(1\)](#)、[audioplay\(1\)](#)、[attributes\(5\)](#)、[largefile\(5\)](#)、[audio\(7I\)](#)

引用名 audiotest – 测试音频设备

用法概要 audiotest [-24571] [dev] ...

描述 `audiotest` 实用程序对指定的音频设备运行测试（如果未指定音频设备，则对系统上找到的所有音频设备运行测试）。测试包括通过每个声道播放音频样例并测量回放的时钟偏移率。

选项 支持以下选项：

- 1 循环模式。测试运行进入死循环。
- 2 立体声（双声道）模式。这是缺省模式。回放假定存在 2 个声道。
- 4 四声道模式（4 声道环绕声）。测试假定存在四个环绕声声道。
- 5 环绕声模式 (5.1)。该测试将检查左、右、环绕声左、环绕声右和中央声道。不测试低频音效声道。
- 7 环绕声模式 (7.1)。该测试将检查左、右、环绕声左、环绕声右、后环绕声左、后环绕声右和中央声道。不测试低频音效声道。

操作数 `dev` 要测试的设备的路径，例如 `/dev/dsp0`。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
体系结构	SPARC、x86
可用性	audio/audio-utilities
接口稳定性	Committed（已确定）

另请参见 [audioconvert\(1\)](#)、[audiocctl\(1\)](#)、[audiorecord\(1\)](#)、[attributes\(5\)](#)、[audio\(7I\)](#)

已知问题 `audiotest` 无法检测物理设备支持的实际音频通道的数量。

`audiotest` 不测试低频音效 (low-frequency effects, LFE) 声道。

不测试音频捕获、音量控制或其他高级设备功能。

引用名	auths – 管理和列出授权
用法概要	<pre>auths [user]... auths list [-S repository] [-v] [-u user] auths info [-S repository] [-v] [authorization] auths check [-u user] authorization auths add [-S repository] -t description [-h help_file_path] authorization auths modify [-S repository] [-t description] [-h help_file_path] authorization auths remove [-S repository] authorization</pre>
描述	<p>auths 命令在标准输出上显示已授予您或者随意指定的用户或角色的授权。授权是某些特权程序在确定用户是否可以执行受限功能时要检查的权限。</p> <p>此命令还可以在 auth_attr(4) 数据库中创建和修改授权及其属性（在本地文件名称服务或 LDAP 名称服务中）。auths 命令还在标准输出上显示已授予您或者随意指定的用户或角色的授权。</p> <p>管理员必须具有 "Rights Management"（权限管理）配置文件才能使用添加、修改或删除子命令管理 auth_attr(4) 数据库中的授权。</p> <p>每个用户可以具有零个或多个授权。授权由全限定名表示，该名称可标识创建授权的组织及其控制的功能。按照 Java 约定，授权的分层组件由句点(.)分隔，以创建组织的逆序 Internet 域名开头，以授权类中的特定函数结尾。授权不能以句点(.)结尾。</p> <p>星号(*)表示类中的所有授权。</p> <p>在 user_attr(4) 和 <code>/etc/security/policy.conf</code> 文件中查找用户的授权（请参见 policy.conf(4)）。可在 user_attr(4) 中直接指定授权，或通过 prof_attr(4) 间接指定授权。也可以将授权作为缺省授权直接分配给系统中的每个用户，或者作为 <code>/etc/security/policy.conf</code> 文件中的缺省配置文件间接分配给每个用户。</p>
子命令	<pre>add [-S repository] -t description [-h help_file_path] authorization</pre> <p>在指定的名称服务系统信息库 (<i>repository</i>) 中创建指定的授权 (<i>authorization</i>)。</p> <p>如果未指定系统信息库选项，则在文件名称服务中创建授权。</p> <pre>check [-u user] authorization</pre> <p>检查指定的授权 (<i>authorization</i>) 是否已授予指定的用户名 (<i>user</i>) 或当前用户。</p> <p>如果用户具有相应的授权，则 auths 会以退出代码 0 退出。否则，该命令会返回大于 1 的退出代码。</p>

info [-S *repository*] [-v] [*authorization*]

检查指定的授权 (*authorization*) 是否存在于指定的名称服务系统信息库 (*repository*)，或基于 `nsswitch.conf(4)` 进行查找。如果指定的授权存在，则会列出该授权，且 `auths` 以返回代码 0 退出。

如果未指定授权，`auths` 会输出指定的名称服务系统信息库中存在的所有授权，或基于 `nsswitch.conf(4)` 输出所有授权。

list [-S *repository*] [-v] [-u *user*]

列出分配给指定用户 (*user*) 或当前用户的所有授权，如果未指定用户名，则基于名称服务系统信息库 (*repository*) 列出所有授权。

如果未指定系统信息库，则基于 `nsswitch.conf(4)` 查找信息。

modify [-S *repository*] [-t *description*] [-h *help_file_path*]

修改指定名称服务系统信息库中的现有授权。如果未指定系统信息库，则修改基于 `nsswitch.conf(4)` 找到的第一个名称服务中的授权。

remove [-S *repository*] *authorization*

在指定的名称服务系统信息库 (*repository*) 中删除现有授权 (*authorization*)。

如果未指定系统信息库，则从基于 `nsswitch.conf(4)` 找到的第一个名称服务中删除授权。

选项

`auths` 子命令支持以下选项：

- h *help_file_path* 设置包含授权相关信息的帮助文件的位置。
- S *repository* 指定要修改或搜索的名称服务系统信息库 (*repository*)。支持的系统信息库选项为 `files` 和 `ldap`。
如果省略此选项，将基于 `nsswitch.conf(4)` 进行查找。
- t *description* 指定授权的文本描述。
- u *user* 指定要列出或检查其授权的用户名 (*user*)。
如果省略此选项，将使用当前用户。
- v 输出授权描述。

示例

示例 1 使用 `auths` 命令

`auths` 的输出内容如下所示：

```
example% auths tester01 tester02
tester01 : solaris.system.date,solaris.jobs.admin
tester02 : solaris.system.*
example%
```

`tester01` 中的授权名称由逗号分隔，逗号后不加空格。

示例1 使用 auths 命令 (续)

以下命令列出分配给用户 `tester01` 的授权。

```
example% auths list -u tester01
```

```
tester01:
solaris.jobs.admin
solaris.system.date
```

示例2 列出授权

以下命令列出分配给用户 `tester01` 的授权及说明。

```
example% auths list -v -u tester01
```

```
tester01:
solaris.jobs.admin
Manage All Jobs
solaris.system.date
Set Date & Time
```

示例3 列出授权

以下命令列出名称服务中的授权和描述。

```
example% auths info -v solaris.user.manage
solaris.user.manage:
Manage user accounts
example%
```

示例4 添加授权

以下命令将授权 `solaris.foo.manage` 以及描述 `manage foo` 和帮助文件 `AuthFoo.html` 添加到文件名称服务系统信息库中。

```
example% auths add -t "manage foo"\
-h /home/abc/AuthFoo.html solaris.foo.manage
```

示例5 修改授权

以下示例在 LDAP 中修改授权 `solaris.foo.manage`，将描述设置为 `manage foo and bar`，将帮助文件设置为 `AuthFooBar.html`。

```
example% auths -S ldap modify -t "manage foo and bars"\
-h /home/abc/AuthFooBar.html solaris.foo.manage
```

退出状态

将返回以下退出值：

- 0 成功完成。
- 1 出现错误。

2 用户未经授权。

文件

/etc/user_attr

/etc/security/auth_attr

/etc/security/policy.conf

/etc/security/prof_attr

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[profiles\(1\)](#)、[roles\(1\)](#)、[getauthattr\(3C\)](#)、[auth_attr\(4\)](#)、[policy.conf\(4\)](#)、[prof_attr\(4\)](#)、[user_at](#)

引用名	auto_ef - 自动编码查找器																																												
用法概要	<pre>/usr/bin/auto_ef [-e <i>encoding_list</i>] [-a] [-l <i>level</i>] [<i>file</i> ...] /usr/bin/auto_ef -h</pre>																																												
描述	<p>auto_ef 实用程序可识别给定文件的编码。该实用程序通过以下方法判断编码：使用 iconv 代码转换、确定对文件的特定代码转换是否成功，以及对文件中出现的字符序列执行频率分析。</p> <p>如果字符串是二进制、字符表、本地化数字列表或计时图，或者如果字符串或文件很小（例如，小于 100 字节），auto_ef 实用程序可能会生成意外的输出。</p> <table> <tr><td>ASCII</td><td></td></tr> <tr><td>ISO-2022-JP</td><td>JIS</td></tr> <tr><td>eucJP</td><td>日文 EUC</td></tr> <tr><td>PCK</td><td>日文 PC 汉字、CP932、Shift JIS</td></tr> <tr><td>UTF-8</td><td></td></tr> <tr><td>ko_KR.euc</td><td>韩文 EUC</td></tr> <tr><td>ko_KR.cp949</td><td>统一朝鲜文</td></tr> <tr><td>ISO-2022-KR</td><td>ISO-2022 韩文</td></tr> <tr><td>zh_CN.iso2022-CN</td><td>ISO-2022 CN/CN-EXT</td></tr> <tr><td>zh_CN.euc</td><td>简体中文 EUC、GB2312</td></tr> <tr><td>GB18030</td><td>简体中文 GB18030/GBK</td></tr> <tr><td>zh_TW-big5</td><td>BIG5</td></tr> <tr><td>zh_TW-euc</td><td>繁体中文 EUC</td></tr> <tr><td>zh_TW.hkscs</td><td>香港 BIG5</td></tr> <tr><td>iso-8859-1</td><td>西欧文及类似语言</td></tr> <tr><td>iso-8859-2</td><td>东欧文及类似语言</td></tr> <tr><td>iso-8859-5</td><td>西里尔文及类似语言</td></tr> <tr><td>iso-8859-6</td><td>阿拉伯文</td></tr> <tr><td>iso-8859-7</td><td>希腊文</td></tr> <tr><td>iso-8859-8</td><td>希伯来文</td></tr> <tr><td>CP1250</td><td>windows-1250，对应于 ISO-8859-2</td></tr> <tr><td>CP1251</td><td>windows-1251，对应于 ISO-8859-5</td></tr> </table>	ASCII		ISO-2022-JP	JIS	eucJP	日文 EUC	PCK	日文 PC 汉字、CP932、Shift JIS	UTF-8		ko_KR.euc	韩文 EUC	ko_KR.cp949	统一朝鲜文	ISO-2022-KR	ISO-2022 韩文	zh_CN.iso2022-CN	ISO-2022 CN/CN-EXT	zh_CN.euc	简体中文 EUC、GB2312	GB18030	简体中文 GB18030/GBK	zh_TW-big5	BIG5	zh_TW-euc	繁体中文 EUC	zh_TW.hkscs	香港 BIG5	iso-8859-1	西欧文及类似语言	iso-8859-2	东欧文及类似语言	iso-8859-5	西里尔文及类似语言	iso-8859-6	阿拉伯文	iso-8859-7	希腊文	iso-8859-8	希伯来文	CP1250	windows-1250，对应于 ISO-8859-2	CP1251	windows-1251，对应于 ISO-8859-5
ASCII																																													
ISO-2022-JP	JIS																																												
eucJP	日文 EUC																																												
PCK	日文 PC 汉字、CP932、Shift JIS																																												
UTF-8																																													
ko_KR.euc	韩文 EUC																																												
ko_KR.cp949	统一朝鲜文																																												
ISO-2022-KR	ISO-2022 韩文																																												
zh_CN.iso2022-CN	ISO-2022 CN/CN-EXT																																												
zh_CN.euc	简体中文 EUC、GB2312																																												
GB18030	简体中文 GB18030/GBK																																												
zh_TW-big5	BIG5																																												
zh_TW-euc	繁体中文 EUC																																												
zh_TW.hkscs	香港 BIG5																																												
iso-8859-1	西欧文及类似语言																																												
iso-8859-2	东欧文及类似语言																																												
iso-8859-5	西里尔文及类似语言																																												
iso-8859-6	阿拉伯文																																												
iso-8859-7	希腊文																																												
iso-8859-8	希伯来文																																												
CP1250	windows-1250，对应于 ISO-8859-2																																												
CP1251	windows-1251，对应于 ISO-8859-5																																												

CP1252	windows-1252, 对应于 ISO-8859-1
CP1253	windows-1253, 对应于 ISO-8859-7
CP1255	windows-1255, 对应于 ISO-8859-8
koi8-r	对应于 iso-8859-5

缺省情况下, `auto_ef` 会为指定文件中的文本返回一个最有可能的编码。要获取该文件的所有可能的编码, 请使用 `-a` 选项。

此外, 缺省情况下, `auto_ef` 会使用最快的过程来检查文件。要获取更为准确的结果, 请使用 `-l` 选项。

要使用一组限定的编码检查数据, 请使用 `-e` 选项。

选项

支持以下选项:

`-a` 按可能性顺序显示所有可能的编码 (评分在 0.0 到 1.0 范围内)。评分较高表示可能性较大。例如,

```
example% auto_ef -a test_file
eucJP          0.89
zh_CN.euc      0.04
ko_KR.euc      0.01
```

若不使用此选项, 则仅会显示具有最高评分的那个编码。

`-e encoding_list` 仅使用指定编码检查数据。例如, 当 `encoding_list` 指定为 "ko_KR.euc:ko_KR.cp949" 时, `auto_ef` 会仅使用 CP949 和 ko_KR.euc 检查文本。若不使用此选项, `auto_ef` 会使用所有编码检查文本。可以通过使用冒号 (:) 分隔编码指定多个编码。

`-h` 显示用法消息。

`-l level` 指定判断级别。level 的值可以是 0、1、2 或 3。级别 3 可生成最佳结果, 但速度可能很慢。级别 0 速度最快, 但结果的准确性可能要比高级别的结果差。缺省值为级别 0。

操作数

支持下列操作数:

`file` 要检查的文件名。

示例

示例 1 检查文件的编码

```
example% auto_ef file_name
```

示例 2 以级别 2 检查文件的编码。

```
example% auto_ef -l 2 file_name
```

示例 3 仅使用 eucJP 或 ko_KR.euc 检查文件的编码

```
example% auto_ef -e "eucJP:ko_KR.euc" file_name
```

退出状态

将返回以下退出值：

- 0 成功完成
- 1 出现错误。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	text/auto_ef
接口稳定性	Committed（已确定）

另请参见

[auto_ef\(3EXT\)](#)、[libauto_ef\(3LIB\)](#)、[attributes\(5\)](#)

《国际语言环境指南》

引用名 awk – pattern scanning and processing language

用法概要

```
/usr/bin/awk [-f progfile] [-Fc] [' prog ' ] [parameters]
[filename]...
```

```
/usr/xpg4/bin/awk [-FcERE] [-v assignment]... 'program' -f progfile...
[argument]...
```

描述 The `/usr/xpg4/bin/awk` utility is described on the [nawk\(1\)](#) manual page.

The `/usr/bin/awk` utility scans each input *filename* for lines that match any of a set of patterns specified in *prog*. The *prog* string must be enclosed in single quotes (`'`) to protect it from the shell. For each pattern in *prog* there can be an associated action performed when a line of a *filename* matches the pattern. The set of pattern-action statements can appear literally as *prog* or in a file specified with the `-f progfile` option. Input files are read in order; if there are no files, the standard input is read. The file name `'-'` means the standard input.

选项 The following options are supported:

`-f progfile` awk uses the set of patterns it reads from *progfile*.

`-Fc` Uses the character *c* as the field separator (FS) character. See the discussion of FS below.

用法

Input Lines Each input line is matched against the pattern portion of every pattern-action statement; the associated action is performed for each matched pattern. Any *filename* of the form *var=value* is treated as an assignment, not a filename, and is executed at the time it would have been opened if it were a filename. *Variables* assigned in this manner are not available inside a BEGIN rule, and are assigned after previously specified files have been read.

An input line is normally made up of fields separated by white spaces. (This default can be changed by using the FS built-in variable or the `-Fc` option.) The default is to ignore leading blanks and to separate fields by blanks and/or tab characters. However, if FS is assigned a value that does not include any of the white spaces, then leading blanks are not ignored. The fields are denoted `$1`, `$2`, . . . ; `$0` refers to the entire line.

Pattern-action Statements A pattern-action statement has the form:

```
pattern { action }
```

Either pattern or action can be omitted. If there is no action, the matching line is printed. If there is no pattern, the action is performed on every input line. Pattern-action statements are separated by newlines or semicolons.

Patterns are arbitrary Boolean combinations (`!`, `||`, `&&`, and parentheses) of relational expressions and regular expressions. A relational expression is one of the following:

expression relop expression
expression matchop regular_expression

where a *relop* is any of the six relational operators in C, and a *matchop* is either ~ (contains) or !~ (does not contain). An *expression* is an arithmetic expression, a relational expression, the special expression

var in array

or a Boolean combination of these.

Regular expressions are as in [egrep\(1\)](#). In patterns they must be surrounded by slashes. Isolated regular expressions in a pattern apply to the entire line. Regular expressions can also occur in relational expressions. A pattern can consist of two patterns separated by a comma; in this case, the action is performed for all lines between the occurrence of the first pattern to the occurrence of the second pattern.

The special patterns BEGIN and END can be used to capture control before the first input line has been read and after the last input line has been read respectively. These keywords do not combine with any other patterns.

Built-in Variables

Built-in variables include:

FILENAME	name of the current input file
FS	input field separator regular expression (default blank and tab)
NF	number of fields in the current record
NR	ordinal number of the current record
OFMT	output format for numbers (default %.6g)
OFS	output field separator (default blank)
ORS	output record separator (default new-line)
RS	input record separator (default new-line)

An action is a sequence of statements. A statement can be one of the following:

```
if ( expression ) statement [ else statement ]
while ( expression ) statement
do statement while ( expression )
for ( expression ; expression ; expression ) statement
for ( var in array ) statement
break
continue
{ [ statement ] . . . }
expression      # commonly variable = expression
print [ expression-list ] [ >expression ]
```

```
printf format [ ,expression-list ] [ >expression ]
next          # skip remaining patterns on this input line
exit [expr]   # skip the rest of the input; exit status is expr
```

Statements are terminated by semicolons, newlines, or right braces. An empty expression-list stands for the whole input line. Expressions take on string or numeric values as appropriate, and are built using the operators +, -, *, /, %, ^ and concatenation (indicated by a blank). The operators ++, --, +=, -=, *=, /=, %=, ^=, >, >=, <, <=, ==, !=, and ?: are also available in expressions. Variables can be scalars, array elements (denoted $x[i]$), or fields. Variables are initialized to the null string or zero. Array subscripts can be any string, not necessarily numeric; this allows for a form of associative memory. String constants are quoted (""), with the usual C escapes recognized within.

The `print` statement prints its arguments on the standard output, or on a file if *>expression* is present, or on a pipe if *|cmd* is present. The output resulted from the `print` statement is terminated by the output record separator with each argument separated by the current output field separator. The `printf` statement formats its expression list according to the format (see [printf\(3C\)](#)).

Built-in Functions

The arithmetic functions are as follows:

<code>cos(x)</code>	Return cosine of x , where x is in radians. (In <code>/usr/xpg4/bin/awk</code> only. See awk(1) .)
<code>sin(x)</code>	Return sine of x , where x is in radians. (In <code>/usr/xpg4/bin/awk</code> only. See awk(1) .)
<code>exp(x)</code>	Return the exponential function of x .
<code>log(x)</code>	Return the natural logarithm of x .
<code>sqrt(x)</code>	Return the square root of x .
<code>int(x)</code>	Truncate its argument to an integer. It is truncated toward 0 when $x > 0$.

The string functions are as follows:

<code>index(s, t)</code>	Return the position in string s where string t first occurs, or 0 if it does not occur at all.
<code>int(s)</code>	truncates s to an integer value. If s is not specified, $\$0$ is used.
<code>length(s)</code>	Return the length of its argument taken as a string, or of the whole line if there is no argument.
<code>split(s, a, fs)</code>	Split the string s into array elements $a[1]$, $a[2]$, \dots , $a[n]$, and returns n . The separation is done with the regular expression fs or with the field separator FS if fs is not given.

`sprintf(fmt, expr, expr, ...)` Format the expressions according to the [printf\(3C\)](#) format given by *fmt* and returns the resulting string.

`substr(s, m, n)` returns the *n*-character substring of *s* that begins at position *m*.

The input/output function is as follows:

`getline` Set `$0` to the next input record from the current input file. `getline` returns 1 for successful input, 0 for end of file, and -1 for an error.

Large File Behavior

See [largefile\(5\)](#) for the description of the behavior of `awk` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Printing Lines Longer Than 72 Characters

The following example is an `awk` script that can be executed by an `awk -f examplescript` style command. It prints lines longer than seventy two characters:

```
length > 72
```

示例 2 Printing Fields in Opposite Order

The following example is an `awk` script that can be executed by an `awk -f examplescript` style command. It prints the first two fields in opposite order:

```
{ print $2, $1 }
```

示例 3 Printing Fields in Opposite Order with the Input Fields Separated

The following example is an `awk` script that can be executed by an `awk -f examplescript` style command. It prints the first two input fields in opposite order, separated by a comma, blanks or tabs:

```
BEGIN { FS = ",[ \t]*|[ \t]+" }  
      { print $2, $1 }
```

The example only works with `/usr/xpg4/bin/awk`.

示例 4 Adding Up the First Column, Printing the Sum and Average

The following example is an `awk` script that can be executed by an `awk -f examplescript` style command. It adds up the first column, and prints the sum and average:

```
{ s += $1 }  
END { print "sum is", s, " average is", s/NR }
```

示例 5 Printing Fields in Reverse Order

The following example is an `awk` script that can be executed by an `awk -f examplescript` style command. It prints fields in reverse order:

示例 5 Printing Fields in Reverse Order (续)

```
{ for (i = NF; i > 0; --i) print $i }
```

示例 6 Printing All lines Between start/stop Pairs

The following example is an awk script that can be executed by an `awk -f examplescript` style command. It prints all lines between start/stop pairs.

```
/start/, /stop/
```

示例 7 Printing All Lines Whose First Field is Different from the Previous One

The following example is an awk script that can be executed by an `awk -f examplescript` style command. It prints all lines whose first field is different from the previous one.

```
$1 != prev { print; prev = $1 }
```

示例 8 Printing a File and Filling in Page numbers

The following example is an awk script that can be executed by an `awk -f examplescript` style command. It prints a file and fills in page numbers starting at 5:

```
/Page/ { $2 = n++; }
        { print }
```

示例 9 Printing a File and Numbering Its Pages

Assuming this program is in a file named `prog`, the following example prints the file input numbering its pages starting at 5:

```
example% awk -f prog n=5 input
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `awk`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, `NLSPATH`, and `PATH`.

LC_NUMERIC Determine the radix character used when interpreting numeric input, performing conversions between numeric and string values and formatting numeric output. Regardless of locale, the period character (the decimal-point character of the POSIX locale) is the decimal-point character recognized in processing awk programs (including assignments in command-line arguments).

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/awk

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Not Enabled

/usr/xpg4/bin/awk

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[egrep\(1\)](#), [grep\(1\)](#), [nawk\(1\)](#), [sed\(1\)](#), [printf\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注

Input white space is not preserved on output if fields are involved.

There are no explicit conversions between numbers and strings. To force an expression to be treated as a number, add `0` to it. To force an expression to be treated as a string, concatenate the null string (`""`) to it.

引用名 banner – make posters

用法概要 banner *strings*

描述 banner prints its arguments (each up to 10 characters long) in large letters on the standard output.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/extended-system-utilities

另请参见 [echo\(1\)](#), [attributes\(5\)](#)

引用名	basename, dirname – deliver portions of path names				
用法概要	<pre>/usr/bin/basename <i>string</i> [<i>suffix</i>]</pre> <pre>/usr/xpg4/bin/basename <i>string</i> [<i>suffix</i>]</pre> <pre>dirname <i>string</i></pre>				
描述	The <code>basename</code> utility deletes any prefix ending in <code>/</code> and the <i>suffix</i> (if present in <i>string</i>) from <i>string</i> , and prints the result on the standard output. It is normally used inside substitution marks (<code>' '</code>) within shell procedures.				
/usr/bin	The <i>suffix</i> is a pattern defined on the expr(1) manual page.				
/usr/xpg4/bin	The <i>suffix</i> is a string with no special significance attached to any of the characters it contains.				
	The <code>dirname</code> utility delivers all but the last level of the path name in <i>string</i> .				
示例	<p>示例 1 Setting environment variables</p> <p>The following example, invoked with the argument <code>/home/sms/personal/mail</code> sets the environment variable <code>NAME</code> to the file named <code>mail</code> and the environment variable <code>MYMAILPATH</code> to the string <code>/home/sms/personal</code>:</p> <pre>example% NAME='basename \$HOME/personal/mail'</pre> <pre>example% MYMAILPATH='dirname \$HOME/personal/mail'</pre> <p>示例 2 Compiling a file and moving the output</p> <p>This shell procedure, invoked with the argument <code>/usr/src/bin/cat.c</code>, compiles the named file and moves the output to <code>cat</code> in the current directory:</p> <pre>example% cc \$1</pre> <pre>example% mv a.out 'basename \$1 .c'</pre>				
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of <code>basename</code> and <code>dirname</code> : <code>LANG</code> , <code>LC_ALL</code> , <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>NLSPATH</code> .				
退出状态	The following exit values are returned: <ul style="list-style-type: none"> 0 Successful completion. >0 An error occurred. 				
属性	See attributes(5) for descriptions of the following attributes:				
/usr/bin	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">ATTRIBUTE TYPE</th> <th style="text-align: center;">ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>system/core-os</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	system/core-os
ATTRIBUTE TYPE	ATTRIBUTE VALUE				
Availability	system/core-os				

basename(1)

/usr/xpg4/bin

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[expr\(1\)](#), [basename\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名 `basename` – display portions of pathnames

用法概要 `/usr/ucb/basename string [suffix]`

描述 The `basename` utility deletes any prefix ending in `'/'` and the *suffix*, if present in *string*. It directs the result to the standard output, and is normally used inside substitution marks (`' '`) within shell procedures. The *suffix* is a string with no special significance attached to any of the characters it contains.

示例 示例 1 Using the `basename` command.

This shell procedure invoked with the argument `/usr/src/bin/cat.c` compiles the named file and moves the output to `cat` in the current directory:

```
example% cc $1
example% mv a.out `basename $1 .c`
```

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [sh\(1\)](#), [attributes\(5\)](#)

引用名	bc – arbitrary precision arithmetic language
用法概要	<code>/usr/bin/bc [-c] [-l] [file]...</code> <code>/usr/xpg6/bin/bc [-c] [-l] [file]...</code>
描述	The bc utility implements an arbitrary precision calculator. It takes input from any files given, then reads from the standard input. If the standard input and standard output to bc are attached to a terminal, the invocation of bc is <i>interactive</i> , causing behavioral constraints described in the following sections. bc processes a language that resembles C and is a preprocessor for the desk calculator program dc, which it invokes automatically unless the -c option is specified. In this case the dc input is sent to the standard output instead.
用法	The syntax for bc programs is as follows: <i>L</i> Means a letter a–z, <i>E</i> Means an expression: a (mathematical or logical) value, an operand that takes a value, or a combination of operands and operators that evaluates to a value, <i>S</i> Means a statement.
Comments	Enclosed in <code>/*</code> and <code>*/</code> .
Names (Operands)	Simple variables: <i>L</i> . Array elements: <i>L</i> [<i>E</i>] (up to BC_DIM_MAX dimensions). The words <code>ibase</code> , <code>obase</code> (limited to BC_BASE_MAX), and <code>scale</code> (limited to BC_SCALE_MAX).
Other Operands	Arbitrarily long numbers with optional sign and decimal point. Strings of fewer than BC_STRING_MAX characters, between double quotes ("). (<i>E</i>) <code>sqrt (<i>E</i>)</code> Square root <code>length (<i>E</i>)</code> Number of significant decimal digits. <code>scale (<i>E</i>)</code> Number of digits right of decimal point. <code>L (<i>E</i>, ..., <i>E</i>)</code>
Operators	<code>+ - * / % ^</code> (% is remainder; ^ is power) <code>++ ---</code> (prefix and postfix; apply to names) <code>== <= >= != < ></code> <code>= += -= *= /= %= ^=</code>
Statements	<i>E</i> <code>{ <i>S</i>; ... ; <i>S</i> }</code> <code>if (<i>E</i>) <i>S</i></code>


```

while ( E ) S
for ( E ; E ; E ) S
null statement
break
quit

.string

```

```

Function Definitions  define L ( L , . . . , L ) {
                    auto L , . . . , L
                    S ; . . . S
                    return ( E )
                    }

```

```

Functions in -l Math  s ( x )      sine
Library              c ( x )      cosine
                    e ( x )      exponential
                    l ( x )      log
                    a ( x )      arctangent
                    j ( n , x )  Bessel function

```

All function arguments are passed by value.

The value of a statement that is an expression is printed unless the main operator is an assignment. Either semicolons or new-lines may separate statements. Assignment to `scale` influences the number of digits to be retained on arithmetic operations in the manner of `dc`. Assignments to `ibase` or `obase` set the input and output number radix respectively.

The same letter may be used as an array, a function, and a simple variable simultaneously. All variables are global to the program. `auto` variables are stacked during function calls. When using arrays as function arguments or defining them as automatic variables, empty square brackets must follow the array name.

选项

The following operands are supported:

```

- c      Compiles only. The output is dc commands that are sent to the standard output.
/usr/bin/bc  - l      Defines the math functions and initializes scale to 20, instead of the default zero.
/usr/xpg6/bin/bc - l      Defines the math functions and initializes scale to 20, instead of the default zero. All
                    math results have the scale of 20.

```

操作数

The following operands are supported:

file A pathname of a text file containing bc program statements. After all cases of *file* have been read, bc reads the standard input.

示例

示例 1 Setting the precision of a variable

In the shell, the following assigns an approximation of the first ten digits of n to the variable x :

```
x=$(printf "%s\n" 'scale = 10; 104348/33215' | bc)
```

示例 2 Defining a computing function

Defines a function to compute an approximate value of the exponential function:

```
scale = 20
define e(x){
    auto a, b, c, i, s
    a = 1
    b = 1
    s = 1
    for(i=1; 1==1; i++){
        a = a*x
        b = b*i
        c = a/b
        if(c == 0) return(s)
        s = s+c
    }
}
```

示例 3 Printing the approximate values of the function

Prints approximate values of the exponential function of the first ten integers:

```
for(i=1; i<=10; i++) e(i)
```

or

```
for (i = 1; i <= 10; ++i) {          e(i) }
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of bc: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

0 All input files were processed successfully.

unspecified An error occurred.

文件

/usr/lib/lib.b mathematical library

/usr/include/limits.h to define BC_ parameters

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[dc\(1\)](#), [awk\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注

The bc command does not recognize the logical operators && and | |.

The for statement must have all three expressions (*E*'s).

引用名 bdiff – big diff

用法概要 bdiff *filename1 filename2* [*n*] [-s]

描述 bdiff is used in a manner analogous to diff to find which lines in *filename1* and *filename2* must be changed to bring the files into agreement. Its purpose is to allow processing of files too large for diff. If *filename1* (*filename2*) is –, the standard input is read.

bdiff ignores lines common to the beginning of both files, splits the remainder of each file into *n*-line segments, and invokes diff on corresponding segments. If both optional arguments are specified, they must appear in the order indicated above.

The output of bdiff is exactly that of diff, with line numbers adjusted to account for the segmenting of the files (that is, to make it look as if the files had been processed whole). Note: Because of the segmenting of the files, bdiff does not necessarily find a smallest sufficient set of file differences.

选项 *n* The number of line segments. The value of *n* is 3500 by default. If the optional third argument is given and it is numeric, it is used as the value for *n*. This is useful in those cases in which 3500-line segments are too large for diff, causing it to fail.

-s Specifies that no diagnostics are to be printed by bdiff (silent option). Note: However, this does not suppress possible diagnostic messages from diff, which bdiff calls.

用法 See [largefile\(5\)](#) for the description of the behavior of bdiff when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

文件 /tmp/bd?????

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	enabled

另请参见 [diff\(1\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

诊断 Use help for explanations.

引用名	bfs – big file scanner						
用法概要	<code>/usr/bin/bfs [-] filename</code>						
描述	<p>The <code>bfs</code> command is (almost) like <code>ed(1)</code> except that it is read-only and processes much larger files. Files can be up to 1024K bytes and 32K lines, with up to 512 characters, including new-line, per line (255 for 16-bit machines). <code>bfs</code> is usually more efficient than <code>ed(1)</code> for scanning a file, since the file is not copied to a buffer. It is most useful for identifying sections of a large file where <code>csplit(1)</code> can be used to divide it into more manageable pieces for editing.</p> <p>Normally, the size of the file being scanned is printed, as is the size of any file written with the <code>w</code> (write) command. The optional <code>-</code> suppresses printing of sizes. Input is prompted with <code>*</code> if <code>P</code> and a carriage return are typed, as in <code>ed(1)</code>. Prompting can be turned off again by inputting another <code>P</code> and carriage return. Note that messages are given in response to errors if prompting is turned on.</p> <p>All address expressions described under <code>ed(1)</code> are supported. In addition, regular expressions may be surrounded with two symbols besides <code>/</code> and <code>?</code>:</p> <ul style="list-style-type: none"> > indicates downward search without wrap-around, and < indicates upward search without wrap-around. <p>There is a slight difference in mark names; that is, only the letters <code>a</code> through <code>z</code> may be used, and all 26 marks are remembered.</p>						
bfs Commands	<p>The <code>e</code>, <code>g</code>, <code>v</code>, <code>k</code>, <code>p</code>, <code>w</code>, <code>=</code>, <code>!</code>, and null commands operate as described under <code>ed(1)</code>. Commands such as <code>---</code>, <code>+++</code>, <code>+++</code>, <code>-12</code>, and <code>+4p</code> are accepted. Note that <code>1</code>, <code>10p</code> and <code>1</code>, <code>10</code> will both print the first ten lines. The <code>f</code> command only prints the name of the file being scanned; there is no <i>remembered</i> file name. The <code>w</code> command is independent of output diversion, truncation, or crunching (see the <code>xo</code>, <code>xt</code>, and <code>xc</code> commands, below). The following additional commands are available:</p> <table border="0" style="width: 100%;"> <tr> <td style="vertical-align: top; padding-right: 20px;"><code>xf file</code></td> <td>Further commands are taken from the named <i>file</i>. When an end-of-file is reached, an interrupt signal is received or an error occurs, reading resumes with the file containing the <code>xf</code>. The <code>xf</code> commands may be nested to a depth of 10.</td> </tr> <tr> <td style="vertical-align: top; padding-right: 20px;"><code>xn</code></td> <td>List the marks currently in use (marks are set by the <code>k</code> command).</td> </tr> <tr> <td style="vertical-align: top; padding-right: 20px;"><code>xo [file]</code></td> <td>Further output from the <code>p</code> and null commands is diverted to the named <i>file</i>, which, if necessary, is created mode 666 (readable and writable by everyone), unless your <code>umask</code> setting (see <code>umask(1)</code>) dictates otherwise. If</td> </tr> </table>	<code>xf file</code>	Further commands are taken from the named <i>file</i> . When an end-of-file is reached, an interrupt signal is received or an error occurs, reading resumes with the file containing the <code>xf</code> . The <code>xf</code> commands may be nested to a depth of 10.	<code>xn</code>	List the marks currently in use (marks are set by the <code>k</code> command).	<code>xo [file]</code>	Further output from the <code>p</code> and null commands is diverted to the named <i>file</i> , which, if necessary, is created mode 666 (readable and writable by everyone), unless your <code>umask</code> setting (see <code>umask(1)</code>) dictates otherwise. If
<code>xf file</code>	Further commands are taken from the named <i>file</i> . When an end-of-file is reached, an interrupt signal is received or an error occurs, reading resumes with the file containing the <code>xf</code> . The <code>xf</code> commands may be nested to a depth of 10.						
<code>xn</code>	List the marks currently in use (marks are set by the <code>k</code> command).						
<code>xo [file]</code>	Further output from the <code>p</code> and null commands is diverted to the named <i>file</i> , which, if necessary, is created mode 666 (readable and writable by everyone), unless your <code>umask</code> setting (see <code>umask(1)</code>) dictates otherwise. If						

	<i>file</i> is missing, output is diverted to the standard output. Note that each diversion causes truncation or creation of the file.
<code>:label</code>	This positions a <i>label</i> in a command file. The <i>label</i> is terminated by newline, and blanks between the : (colon) and the start of the <i>label</i> are ignored. This command may also be used to insert comments into a command file, since labels need not be referenced.
<code>(.,.)xb/regular expression/label</code>	<p>A jump (either upward or downward) is made to <i>label</i> if the command succeeds. It fails under any of the following conditions:</p> <ol style="list-style-type: none"> 1. Either address is not between 1 and \$. 2. The second address is less than the first. 3. The regular expression does not match at least one line in the specified range, including the first and last lines. <p>On success, . (dot) is set to the line matched and a jump is made to <i>label</i>. This command is the only one that does not issue an error message on bad addresses, so it may be used to test whether addresses are bad before other commands are executed. Note that the command, <code>xb/^/label</code>, is an unconditional jump.</p> <p>The <code>xb</code> command is allowed only if it is read from someplace other than a terminal. If it is read from a pipe, only a downward jump is possible.</p>
<code>xt number</code>	Output from the <code>p</code> and null commands is truncated to, at most, <i>number</i> characters. The initial number is 255.
<code>xv [digit] [spaces] [value]</code>	<p>The variable name is the specified <i>digit</i> following the <code>xv</code>. The commands <code>xv5100</code> or <code>xv5 100</code> both assign the value 100 to the variable 5. The command <code>xv61,100p</code> assigns the value 1,100p to the variable 6. To reference a variable, put a % in front of the variable name. For example, using the above assignments for variables 5 and 6:</p> <pre>1,%5p 1,%5 %6</pre> <p>will all print the first 100 lines.</p>

```
g/%5/p
```

would globally search for the characters `100` and print each line containing a match. To escape the special meaning of `%`, a `\` must precede it.

```
g/".*\%[cds]/p
```

could be used to match and list `%c`, `%d`, or `%s` formats (for example, `printf`-like statements) of characters, decimal integers, or strings. Another feature of the `xv` command is that the first line of output from a UNIX system command can be stored into a variable. The only requirement is that the first character of *value* be an `!`. For example:

```
.w junk
xv5!cat junk
!rm junk
!echo "%5"
xv6!expr %6 + 1
```

would put the current line into variable `35`, print it, and increment the variable `36` by one. To escape the special meaning of `!` as the first character of *value*, precede it with a `\`.

```
xv7\!date
```

stores the value `!date` into variable `7`.

```
xbz label
xbn label
```

These two commands will test the last saved *return code* from the execution of a UNIX system command (`!command`) or nonzero value, respectively, to the specified label. The two examples below both search for the next five lines containing the string `size`:

Example 1:

```
xv55
: l
/size/
xv5!expr %5 - 1
!if 0%5 != 0 exit 2
xvn l
```

Example 2:

```
xv45
: l
/size/
xv4!expr %4 - 1
!if 0%4 = 0 exit 2
xbz l
```

xc [switch]

If `switch` is 1, output from the `p` and null commands is crunched; if `switch` is 0, it is not. Without an argument, `xc` reverses `switch`. Initially, `switch` is set for no crunching. Crunched output has strings of tabs and blanks reduced to one blank and blank lines suppressed.

操作数

The following operand is supported:

filename Any file up to 1024K bytes and 32K lines, with up to 512 characters, including new-line, per line (255 for 16-bit machines). *filename* can be a section of a larger file which has been divided into more manageable sections for editing by the use of [csplit\(1\)](#).

退出状态

The following exit values are returned:

- 0 Successful completion without any file or command errors.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[csplit\(1\)](#), [ed\(1\)](#), [umask\(1\)](#), [attributes\(5\)](#)

诊断

Message is ? for errors in commands, if prompting is turned off. Self-explanatory error messages are displayed when prompting is on.

引用名 `biff` – give notice of incoming mail messages

用法概要 `/usr/ucb/biff [y | n]`

描述 `biff` turns mail notification on or off for the terminal session. With no arguments, `biff` displays the current notification status for the terminal.

If notification is allowed, the terminal rings the bell and displays the header and the first few lines of each arriving mail message. `biff` operates asynchronously. For synchronized notices, use the `MAIL` variable of `sh(1)` or the `mail` variable of `csh(1)`.

A '`biff y`' command can be included in your `~/.login` or `~/.profile` file for execution when you log in.

选项 `y` Allow mail notification for the terminal.

`n` Disable notification for the terminal.

文件 `~/.login` User's login file

`~/.profile` User's profile file

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [csh\(1\)](#), [mail\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#)

引用名 break, continue – shell built-in functions to escape from or advance within a controlling while, for, foreach, or until loop

用法概要

sh	break [<i>n</i>] continue [<i>n</i>]
cs	break continue
ksh88	*break [<i>n</i>] *continue [<i>n</i>]
ksh	+break [<i>n</i>] +continue [<i>n</i>]

描述

sh The break utility exits from the enclosing for or while loop, if any. If *n* is specified, break *n* levels.

The continue utility resumes the next iteration of the enclosing for or while loop. If *n* is specified, resume at the *n*-th enclosing loop.

cs The break utility resumes execution after the end of the nearest enclosing foreach or while loop. The remaining commands on the current line are executed. This allows multilevel breaks to be written as a list of break commands, all on one line.

The continue utility continues execution of the next iteration of the nearest enclosing while or foreach loop.

ksh88 The break utility exits from the enclosed for, while, until, or select loop, if any. If *n* is specified, then break *n* levels. If *n* is greater than the number of enclosing loops, the outermost enclosing loop shall be exited.

The continue utility resumes the next iteration of the enclosed for, while, until, or select loop. If *n* is specified then resume at the *n*-th enclosed loop. If *n* is greater than the number of enclosing loops, the outermost enclosing loop shall be used.

On this manual page, ksh88(1) commands that are preceded by one or two * (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.

4. Words that follow a command preceded by ****** that are in the format of a variable assignment are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign, and also that word splitting and file name generation are not performed.

ksh

`break` is a shell special built-in that exits the smallest enclosing `for`, `select`, `while`, or `until` loop. It also exits the *n*th enclosing loop if *n* is specified. Execution continues at the command following the loop or loops.

If *n* is specified, it must be a positive integer ≥ 1 . If *n* is larger than the number of enclosing loops, the last enclosing loop is exited.

`continue` is a shell special built-in that continues execution at the top of the smallest enclosing `for`, `select`, `while`, or `until` loop, if any; or of the top of the *n*th enclosing loop if *n* is specified.

If *n* is specified, it must be a positive integer ≥ 1 . If *n* is larger than the number of enclosing loops, the last enclosing loop is used.

On this manual page, [ksh\(1\)](#) commands that are preceded by one or two + symbols are special built-in commands and are treated the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Built-in commands are not valid function names.
5. Words following a command preceded by **++** that are in the format of a variable assignment are expanded with rules as a variable assignment. This means that tilde substitution is performed after the = sign and field splitting and file name generation are not performed.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[csh\(1\)](#), [exit\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#)

引用名	builtin – 用于添加、删除或显示 shell 内置命令的 ksh 内置函数
用法概要	builtin [-ds] [-f lib] [pathname ...]
描述	<p>kshbuiltin 命令可在当前 shell 环境中添加、删除或显示内置命令。内置命令在当前 shell 进程中执行，可能会在当前 shell 中产生负面影响。在大多数系统上，内置命令的调用时间比创建单独进程的命令低一或两个数量级。</p> <p>对于指定的每个 <i>pathname</i>，由路径名的基名确定内置命令的名称。对于每个基名，该 shell 会在当前 shell 中查找 C 级函数（通过内置名称前置 <i>b_</i> 来确定其名称）。如果 <i>pathname</i> 包含正斜杠 (/)，内置命令会绑定到 <i>pathname</i>。仅当 <i>pathname</i> 是在路径搜索期间找到的第一个可执行对象时，才会执行绑定到该路径名的内置命令。否则，将在执行路径搜索之前查找内置命令。</p> <p>如果未指定 <i>pathname</i>，builtin 会在标准输出上显示当前内置命令列表，或者在指定了 -s 选项的情况下仅显示特殊内置命令。将显示已绑定到路径名的内置命令的全路径名。</p> <p>可使用 -f 选项指定包含内置命令的库。如果库中包含名为 <code>lib_init()</code> 的函数，则会在装入该库时调用此函数（带有参数 0）。<code>lib_init()</code> 函数可以通过调用相应的 C 级函数装入内置命令。在这种情况下，对 C 级函数名称没有限制。</p> <p>调用 C 级函数时带有三个参数。前两个参数与 <code>main()</code> 相同，第三个是一个指针。</p> <p>无法从限定的 shell 调用 kshbuiltin 命令。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none">-d 删除每个指定内置命令。无法删除特殊内置命令。-f lib 在具有动态链接的系统上，在共享库 <i>lib</i> 中装入和搜索内置命令。 搜索 <code>\$PATH</code> 和系统相关库目录中的库。可省略系统相关共享库的前缀或后缀。装入某库后，其符号即可用于 builtin 的当前和后续调用。在单个 builtin 调用中可以指定多个库。按指定库时的反向顺序搜索库。-s 仅显示特殊内置命令。
操作数	<p>支持下列操作数：</p> <p><i>pathname</i> 指定 <i>pathname</i>。由路径名的基名确定内置命令的名称。</p>
退出状态	<p>将返回以下退出值：</p> <ul style="list-style-type: none">0 成功完成。>0 出现错误。

示例 示例1 装入 builtin 命令
 以下示例从库 libfoo.so 装入 builtin 命令 mycmd:
 example% builtin -f foo mycmd

作者 David Korn, dgk@research.att.com

属性 有关下列属性的说明, 请参见 [attributes\(5\)](#):

属性类型	属性值
可用性	system/core-os
接口稳定性	Uncommitted (未确定)

另请参见 [ksh\(1\)](#)、[whence\(1\)](#)、[attributes\(5\)](#)

引用名 cal – display a calendar

用法概要 cal [*month*] *year*

描述 The cal utility writes a Gregorian calendar to standard output. If the *year* operand is specified, a calendar for that year is written. If no operands are specified, a calendar for the current month is written.

操作数 The following operands are supported:

month Specify the month to be displayed, represented as a decimal integer from 1 (January) to 12 (December). The default is the current month.

year Specify the year for which the calendar is displayed, represented as a decimal integer from 1 to 9999. The default is the current year.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of cal: LANG, LC_ALL, LC_CTYPE, LC_TIME, LC_MESSAGES, and NLSPATH.

TZ Determine the timezone used to calculate the value of the current month.

退出状态 The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [calendar\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注 An unusual calendar is printed for September 1752. That is the month 11 days were skipped to make up for lack of leap year adjustments. To see this calendar, type:

```
cal 9 1752
```

The command cal 83 refers to the year 83, not 1983.

The year is always considered to start in January.

引用名	calendar – reminder service
用法概要	calendar [-]
描述	<p>The calendar utility consults the file calendar in the current directory and writes lines that contain today's or tomorrow's date anywhere in the line to standard output. Most reasonable month-day dates such as Aug. 24, august 24, 8/24, and so forth, are recognized, but not 24 August or 24/8. On Fridays and weekends “tomorrow” extends through Monday. calendar can be invoked regularly by using the crontab(1) or at(1) commands.</p> <p>When the optional argument - is present, calendar does its job for every user who has a file calendar in his or her login directory and sends them any positive results by mail(1). Normally this is done daily by facilities in the UNIX operating system (see cron(1M)).</p> <p>If the environment variable DATEMSK is set, calendar will use its value as the full path name of a template file containing format strings. The strings consist of conversion specifications and text characters and are used to provide a richer set of allowable date formats in different languages by appropriate settings of the environment variable LANG or LC_TIME; see environ(5). See strftime(3C) for the list of allowable conversion specifications.</p>
示例	<p>示例 1 Possible contents of a template</p> <p>The following example shows the possible contents of a template:</p> <pre>%B %eth of the year %Y</pre> <p>%B represents the full month name, %e the day of month and %Y the year (4 digits).</p> <p>If DATEMSK is set to this template, the following calendar file would be valid:</p> <pre>March 7th of the year 1989 <Reminder></pre>
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of calendar: LC_CTYPE, LC_TIME, LC_MESSAGES, NLS_PATH, and TZ.
退出状态	<p>0 Successful completion.</p> <p>>0 An error occurred.</p>
文件	<p>/etc/passwd system password file</p> <p>/tmp/cal* temporary files used by calendar</p> <p>/usr/lib/calprog program used to determine dates for today and tomorrow</p>
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[at\(1\)](#), [crontab\(1\)](#), [mail\(1\)](#), [cron\(1M\)](#), [ypbind\(1M\)](#), [strftime\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注

Appropriate lines beginning with white space will not be printed.

Your calendar must be public information for you to get reminder service.

calendar's extended idea of "tomorrow" does not account for holidays.

The `-` argument works only on calendar files that are local to the machine; calendar is intended not to work on calendar files that are mounted remotely with NFS. Thus, 'calendar -' should be run only on diskful machines where home directories exist; running it on a diskless client has no effect.

calendar is no longer in the default root crontab. Because of the network burden 'calendar -' can induce, it is inadvisable in an environment running [ypbind\(1M\)](#) with a large `passwd.byname` map. If, however, the usefulness of calendar outweighs the network impact, the super-user may run 'crontab -e' to edit the root crontab. Otherwise, individual users may wish to use 'crontab -e' to edit their own crontabs to have cron invoke calendar without the `-` argument, piping output to mail addressed to themselves.

引用名	cat – concatenate and display files
用法概要	<code>/usr/bin/cat [-nbsvet] [file...]</code>
描述	<p>The cat utility reads each <i>file</i> in sequence and writes it on the standard output. Thus:</p> <pre>example% cat file</pre> <p>prints <i>file</i> on your terminal, and:</p> <pre>example% cat file1 file2 >file3</pre> <p>concatenates <i>file1</i> and <i>file2</i>, and writes the results in <i>file3</i>. If no input file is given, cat reads from the standard input file.</p>
选项	<p>The following options are supported by <code>/usr/bin/cat</code>:</p> <ul style="list-style-type: none"> -b Number the lines, as -n, but omit the line numbers from blank lines. -n Precede each line output with its line number. -s cat is silent about non-existent files. -u The output is not buffered. <p>Buffered output is the default.</p> <ul style="list-style-type: none"> -v Non-printing characters, with the exception of tabs, NEWLINES and form feeds, are printed visibly. ASCII control characters (octal 000 – 037) are printed as ^<i>n</i>, where <i>n</i> is the corresponding ASCII character in the range octal 100 – 137 (@, A, B, C, . . . , X, Y, Z, [, \,], ^, and _); the DEL character (octal 0177) is printed ^?. Other non-printable characters are printed as M-<i>x</i>, where <i>x</i> is the ASCII character specified by the low-order seven bits. <p>When used with the -v option, the following options can be used:</p> <ul style="list-style-type: none"> -e A \$ character is printed at the end of each line, prior to the NEWLINE. -t Tabs are printed as ^Is and form feeds to be printed as ^Ls. <p>The -e and -t options are ignored if the -v option is not specified.</p>
操作数	<p>The following operand is supported:</p> <p><i>file</i> A path name of an input file. If no <i>file</i> is specified, the standard input is used. If <i>file</i> is <code>-</code>, cat reads from the standard input at that point in the sequence. cat does not close and reopen standard input when it is referenced in this way, but accepts multiple occurrences of <code>-</code> as <i>file</i>.</p>
用法	<p>See largefile(5) for the description of the behavior of cat when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).</p>

示例**示例 1** Concatenating a File

The following command writes the contents of the file `myfile` to standard output:

```
example% cat myfile
```

示例 2 Concatenating Two files into One

The following command concatenates the files `doc1` and `doc2` and writes the result to `doc.all`.

```
example% cat doc1 doc2 > doc.all
```

示例 3 Concatenating Two Arbitrary Pieces of Input with a Single Invocation

When standard input is a terminal, the following command gets two arbitrary pieces of input from the terminal with a single invocation of `cat`:

```
example% cat start - middle - end > file
```

when standard input is a terminal, gets two arbitrary pieces of input from the terminal with a single invocation of `cat`.

If standard input is a regular file,

```
example% cat start - middle - end > file
```

would be equivalent to the following command:

```
cat start - middle /dev/null end > file
```

because the entire contents of the file would be consumed by `cat` the first time `-` was used as a *file* operand and an end-of-file condition would be detected immediately when `-` was referenced the second time.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `cat`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

- `0` All input files were output successfully.
- `>0` An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[touch\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注

Redirecting the output of `cat` onto one of the files being read causes the loss of the data originally in the file being read. For example,

```
example% cat filename1 filename2 > filename1
```

causes the original data in `filename1` to be lost.

引用名 cd, chdir, pushd, popd, dirs – change working directory

用法概要 /usr/bin/cd [*directory*]

sh cd [*argument*]

chdir [*argument*]

csh cd [*dir*]

chdir [*dir*]

pushd [+*n* | *dir*]

popd [+*n*]

dirs [-*l*]

ksh88, ksh cd [-L] [-P] [*arg*]

cd *old new*

描述

/usr/bin/cd The /usr/bin/cd utility changes the current directory in the context of the cd utility only. This is in contrast to the version built into the shell. /usr/bin/cd has no effect on the invoking process but can be used to determine whether or not a given directory can be set as the current directory.

sh The Bourne shell built-in cd changes the current directory to *argument*. The shell parameter HOME is the default *argument*. The shell parameter CDPATH defines the search path for the directory containing *argument*. Alternative directory names are separated by a colon (:). The default path is <null> (specifying the current directory). The current directory is specified by a null path name, which can appear immediately after the equal sign or between the colon delimiters anywhere else in the path list. If *argument* begins with '/', '.', or './', the search path is not used. Otherwise, each directory in the path is searched for *argument*. cd must have execute (search) permission in *argument*. Because a new process is created to execute each command, cd would be ineffective if it were written as a normal command; therefore, it is recognized by and is internal to the shell. (See [pwd\(1\)](#), [sh\(1\)](#), and [chdir\(2\)](#)).

chdir is just another way to call cd.

csh If *dir* is not specified, the C shell built-in cd uses the value of shell parameter HOME as the new working directory. If *dir* specifies a complete path starting with '/', '.', or './', *dir* becomes the new working directory. If neither case applies, cd tries to find the designated directory relative to one of the paths specified by the CDPATH shell variable. CDPATH has the same syntax as, and similar semantics to, the PATH shell variable. cd must have execute (search) permission in *dir*. Because a new process is created to execute each command, cd would be ineffective if it were written as a normal command; therefore, it is recognized by and is internal to the C-shell. (See [pwd\(1\)](#), [sh\(1\)](#), and [chdir\(2\)](#)).

`cd` *r* changes the shell's working directory to directory *dir*. If no argument is given, change to the home directory of the user. If *dir* is a relative pathname not found in the current directory, check for it in those directories listed in the `cdpath` variable. If *dir* is the name of a shell variable whose value starts with a `/`, change to the directory named by that value.

`pushd` pushes a directory onto the directory stack. With no arguments, exchange the top two elements.

`+n` Rotate the *n*'th entry to the top of the stack and `cd` to it.

dir Push the current working directory onto the stack and change to *dir*.

`popd` pops the directory stack and `cd` to the new top directory. The elements of the directory stack are numbered from 0 starting at the top.

`+n` Discard the *n*'th entry in the stack.

`dirs` prints the directory stack, most recent to the left; the first directory shown is the current directory. With the `-l` argument, produce an unabbreviated printout; use of the `~` notation is suppressed.

ksh88, ksh

The Korn shell built-in `cd` command can be in either of two forms. In the first form it changes the current directory to *arg*. If *arg* is `-` the directory is changed to the previous directory. The shell variable `HOME` is the default *arg*. The environment variable `PWD` is set to the current directory. If the `PWD` is changed, the `OLDPWD` environment variable shall also be changed to the value of the old working directory, that is, the current working directory immediately prior to the call to change directory (`cd`). The shell variable `CDPATH` defines the search path for the directory containing *arg*. Alternative directory names are separated by a colon (`:`). The default path is `null` (specifying the current directory). The current directory is specified by a null path name, which can appear immediately after the equal sign or between the colon delimiters anywhere else in the path list. If *arg* begins with a `'/'`, `'.'`, or `'..'`, then the search path is not used. Otherwise, each directory in the path is searched for *arg*. If unsuccessful, `cd` attempts to change directories to the pathname formed by the concatenation of the value of `PWD`, a slash character, and *arg*.

`-L` Handles the operation dot-dot (`..`) logically. Symbolic link components are *not* resolved before dot-dot components are processed.

`-P` Handles the operand dot-dot physically. Symbolic link components *are* resolved before dot-dot components are processed.

If both `-L` and `-P` options are specified, the last option to be invoked is used and the other is ignored. If neither `-L` nor `-P` is specified, the operand is handled dot-dot logically.

The second form of `cd` substitutes the string *new* for the string *old* in the current directory name, `PWD` and tries to change to this new directory.

The `cd` command cannot be executed by `rsh`. Because a new process is created to execute each command, `cd` would be ineffective if it were written as a normal command; therefore, it is recognized by and is internal to the Korn shell. (See [pwd\(1\)](#), [sh\(1\)](#), and [chdir\(2\)](#)).

操作数

The following operands are supported:

directory An absolute or relative pathname of the directory that becomes the new working directory. The interpretation of a relative pathname by `cd` depends on the `CDPATH` environment variable.

Output

If a non-empty directory name from `CDPATH` is used, an absolute pathname of the new working directory is written to the standard output as follows:

```
"%s\n", <new directory>
```

Otherwise, there is no output.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `cd`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

`CDPATH` A colon-separated list of pathnames that refer to directories. If the *directory* operand does not begin with a slash (/) character, and the first component is not dot or dot-dot, `cd` searches for *directory* relative to each directory named in the `CDPATH` variable, in the order listed. The new working directory sets to the first matching directory found. An empty string in place of a directory pathname represents the current directory. If `CDPATH` is not set, it is treated as if it were an empty string.

`HOME` The name of the home directory, used when no *directory* operand is specified.

`OLDPWD` A pathname of the previous working directory, used by `cd -`.

`PWD` A pathname of the current working directory, set by `cd` after it has changed to that directory.

退出状态

The following exit values are returned by `cd`:

0 The directory was successfully changed.

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

csh, ksh88, sh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Standard	See standards(5) .

ksh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

另请参见

[csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [pwd\(1\)](#), [sh\(1\)](#), [chdir\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名 cdrw – CD 读取和写入

用法概要

```

cdrw -i [-vSCO] [-d device] [-p speed] [image-file]
cdrw -a [-vSCO] [-d device] [-p speed] [-T audio-type] audio-file1
    [audio-file2]...
cdrw -x [-v] [-d device] [-T audio-type] track-number out-file
cdrw -c [-vSC] [-d device] [-p speed] [-m tmp-dir]
    [-s src-device]
cdrw -b [-v] [-d device] all | session | fast
cdrw -L [-v] [-d device]
cdrw -M [-v] [-d device]
cdrw -l [-v]
cdrw -h

```

描述

`cdrw` 命令可提供创建数据和音频 CD 的功能。此命令还提供从音频 CD 提取声道以及创建数据 DVD 的功能。CD 或 DVD 设备必须符合 MMC，才能使用 `cdrw` 命令创建 CD 或 DVD。

除非您使用 `-d` 选项指定某设备，否则 `cdrw` 将搜索已连接到系统的 CD 或 DVD 写入器。如果 `cdrw` 找到一个此类设备，它会将该设备用作此命令的缺省 CD 或 DVD 写入器。

若系统上连接有多个 CD 或 DVD 写入器，请使用 `-d` 选项指明所需的设备。可以采用下列方式之一指定设备名称：`/dev/rdisk/cNtNdNsN`、`cNtNdNsN`、`cNtNdN` 或卷管理器使用的名称（例如，`cdrom` 或 `cdrom1`）。使用 `-l` 选项可提供 CD 或 DVD 写入器的列表。

有关向系统中添加符合 USB 海量存储类的 CD-RW 或 DVD-RW 设备的说明，请参见 [scsa2usb\(7D\)](#)。

创建数据 CD

在创建数据 CD 时，`cdrw` 使用轨道一次刻录 (Track-At-Once) 模式进行写入。使用 `-i` 选项指定包含要写入到 CD 介质上的数据的文件。如果不指定此选项，`cdrw` 将从标准输入读取数据。

无论在哪种情况下，通常都会准备好数据，方法是使用 `mkisofs` 命令将文件和文件信息转换成 CD 上使用的 High Sierra 格式。请参见包括此命令用法的示例。

创建数据 DVD

`cdrw` 可以使用从 `mkisofs` 生成的映像 DVD+RW 或 DVD-RW 设备上创建单会话数据 DVD。可将这些磁盘作为 HFS 文件系统挂载。在创建数据 DVD 时，`cdrw` 使用光盘一次刻录 (Disk-At-Once, DAO) 模式进行写入，这会在写入完成时关闭介质并防止添加任何其他会话。因为 DAO 模式要求事先知道映像的大小，所以在将映像写入 DVD 介质之前应事先准备好映像。

- 创建音频 CD** 使用 `-a` 选项创建音频 CD。可以使用此选项指定单个或多个音频文件。所有音频文件都应采用支持的音频格式。当前批准的格式包括：
- `sun` 数据采用 Red Book CDDA 格式的 Sun .au 文件
 - `wav` 数据采用 Red Book CDDA 格式的 RIFF (.wav) 文件
 - `cda` 具有原始 CD 音频数据（即，抽样率为 44.1 KHz 时采用小尾数法 (little-endian) 的 16 位 PCM 立体声）的 .cda 文件
 - `aur` 具有采用大尾数法 (big-endian) 的原始 CD 数据的 .aur 文件
- 如果未指定音频格式，`cdrw` 会尝试基于文件扩展名识别音频文件格式。忽略扩展名中字符的大小写。如果使用 `-T` 选项指定了格式，则假定为适用于所有指定文件的音频文件类型。另外，写入声道后使用 `-c` 选项关闭会话。因此，应在单个命令行中指定要写入的磁道。
- 提取音频** 还可使用 `cdrw` 通过 `-x` 选项从音频 CD 提取音频数据。CD 应具有 Red Book CDDA 格式的磁道。缺省情况下，输出格式基于文件扩展名。用户可以使用 `-T` 选项指定 `sun`、`wav`、`cda` 或 `aur` 输出格式。
- 复制 CD** 可使用 `cdrw` 复制单会话数据 CD-ROM 和 Red Book 音频 CD。在复制 CD 时，`cdrw` 会查找指定的源设备。如果在使用 `-c` 选项时未指定源设备，会假定当前 CD 写入器是源。`cdrw` 将一个或多个磁道提取到临时文件并在当前 CD 写入器中查找空的可写入 CD-R/RW 介质。如果没有找到介质，请在当前 CD 写入器中插入空的可写入 CD 介质。如果缺省临时目录没有足够的空间，可以使用 `-m` 选项指定备用目录。
- 擦除 CD-RW 或 DVD-RW 介质** 用户必须先擦除 CD-RW 介质，才可以对其进行重写。通过 `-b` 选项，当前支持以下类型的擦除：
- `session` 擦除最后一个会话。
 - `fast` 最小程度地擦除介质。
 - `all` 擦除整个介质。
- 如果使用 `session` 擦除类型，`cdrw` 将擦除最后一个会话。如果 CD-RW 上仅记录了一个会话（例如，由此工具创建的数据或音频 CD-RW），则 `session` 擦除仅会擦除所记录的部分，留下一个空磁盘。这比擦除整个介质速度要快。对于 DVD 介质，使用 `-b session` 擦除整个介质。
- `fast` 擦除类型通过删除第一个会话的 PMA 和 TOC 最小程度地擦除整个介质。它不会擦除介质上的用户数据和后续磁道，但会将介质视为空磁盘。如果需要完全擦除介质，请使用 `all` 选项。
- 如果是多会话磁盘、最后一个会话未关闭或磁盘状态未知而您想要擦除该磁盘，应使用 `all` 擦除类型。使用此擦除类型，`cdrw` 会擦除整个磁盘。

DVD+RW 介质不支持擦除。要重新使用 DVD+RW 介质，仅需在介质上写入一个新映像。cdrw 会自动格式化并覆盖现有介质。

检查设备列表或介质状态

您可以使用 `-l` 选项列出系统的 CD 或 DVD 写入器。另外，对于特定介质，您也可以使用 `-M` 选项获取清空状态和目录。`-M` 选项还可显示有关最后一个会话的起始地址和下一可写入地址的信息。此信息可与 `-o` 选项一起用于创建多会话 CD。有关更多信息，请参见 SUNWfsman 软件包中的 `mkisofs(8)` 手册页 (`/usr/share/man/man8/mkisofs.8`)。

选项

支持以下选项：

- a 创建音频磁盘。必须至少指定一个 *audio-file* 名称。一个 CD 具有的声音不能超过 99 个，所以指定的音频文件也不得超出 99 个。
- b 清空 CD-RW 或 DVD-RW 介质。必须使用 `all`、`fast` 或 `session` 参数指定擦除类型。DVD+RW 介质不支持清空，但可以在无需清空的情况下进行重写。
- c 复制 CD。如果没有指定任何其他参数，则将缺省 CD 写入设备假定为源设备。在这种情况下，复制操作会将源介质读取到临时目录，并提示您将介质放入驱动器中以继续执行复制操作。
- C 此选项已过时。
此选项以前用于指示 `cdrw` 查询驱动器来确定介质容量。现在这是缺省行为。
- d 指定 CD 或 DVD 写入设备。
- h 帮助。显示用法消息。
- i 指定用于创建数据 CD 或 DVD 的映像文件。文件大小应小于介质上可写入的大小。另外，请考虑提供本地文件，而不是挂载 NFS 的文件系统上的文件。CD 写入进程期望数据持续可用，没有中断。
- l 列出系统上可用的所有 CD 或 DVD 写入器。
- L 关闭磁盘。如果在完成最后一个写入操作后介质仍处于打开状态，则将其关闭以阻止任何其他写入。只能在可重写的 CD-RW 介质上执行此操作。
- m 复制 CD 或 DVD 时，使用备用临时目录（而不是缺省临时目录）来存储磁道数据。备用临时目录可能是必需的，因为 CD 上的数据量可能非常大。例如，80 分钟的音频 CD 的数据量可能达到 800 MB，而 DVD 可能要达到 4.7 GB。缺省临时目录可能没有那么多可用空间。
- M 报告介质状态。`cdrw` 报告介质是否为空、其目录、最后一个会话的起始地址，以及下一可写入地址（如果磁盘已打开）。DVD+RW 不支持擦除且始终在介质上具有某些内容。
- O 保持磁盘打开。`cdrw` 会关闭会话而保持磁盘打开，以便稍后添加其他会话来创建多会话磁盘。

- p 设置 CD 写入速度。例如，`-p 4` 会将速度设置为 4X。如果未指定此选项，`cdrw` 会使用 CD 写入器的缺省速度。如果指定此选项，`cdrw` 会尝试将驱动器写入速度设置为此值，但不能保证驱动器使用的实际速度。
- s 指定用于复制 CD 或 DVD 的源设备。
- S 模拟模式。在此模式中，`cdrw` 在驱动器激光关闭的情况下运行，所以不会向介质中写入任何内容。使用此选项可验证系统是否能够以最利于 CD 写入的速率提供数据。

CD-R、CD-RW（非 MRW 格式）、DVD-R 和 DVD-RW 介质支持模拟模式（-S）。DVD-RAM、DVD+R、DVD+RW、任何 MRW 格式介质以及某些其他介质不支持模拟模式（-S）。
- T 提取音频文件或读取音频文件以创建音频 CD 时要使用的音频格式。*audio-type* 可以是 `sun`、`wav`、`cda` 或 `aur`。
- v 详细模式。
- x 从声道提取音频数据。

示例

示例 1 创建数据 CD 或 DVD

```
example% cdrw -i /local/iso_image
```

示例 2 从目录创建 CD 或 DVD

本示例显示了如何从目录树 `/home/foo` 创建 CD 或 DVD。

```
example% mkisofs -r /home/foo 2>/dev/null | cdrw -i -p 1
```

示例 3 提取声道编号

本示例显示了如何将声道编号 1 提取到 `/home/foo/song1.wav`。

```
example% cdrw -x -T wav 1 /home/foo/song1.wav
```

示例 4 使用 wav 文件

本示例显示了如何从磁盘上的 `wav` 文件创建音频 CD。

```
example% cdrw -a song1.wav song2.wav song3.wav song4.wav
```

示例 5 擦除 CD-RW 或 DVD-RW 介质

本示例显示了如何擦除可重写介质。

```
example% cdrw -b all
```

示例 6 使用多个驱动器创建数据 CD 或 DVD

本示例显示了如何在系统上使用多个 CD、DVD-R 或 DVD-RW 驱动器创建数据 CD 或 DVD。

示例 6 使用多个驱动器创建数据 CD 或 DVD (续)

```
example% cdrw -d c1t6d0s2 -i /home/foo/iso-image
```

示例 7 检查数据传送速率

本示例显示了如何验证系统是否能够以满足写入操作的速率向 CD-RW 或 DVD 驱动器提供数据。

```
example% cdrw -S -i /home/foo/iso-image
```

示例 8 以较高优先级运行

本示例显示了如何以较高优先级运行 cdrw (仅限 root 用户)。

```
example# priocntl -e -p 60 cdrw -i /home/foo/iso-image
```

示例 9 创建多会话磁盘

本示例显示了如何通过使用 mkisofs 创建第一个会话映像并在不关闭磁盘的情况下将其记录到磁盘上。

```
example% cdrw -O -i /home/foo/iso-image
```

通过使用 mkisofs 创建映像 (采用 cdrw 报告的会话起始地址和下一可写入地址), 可向打开的磁盘中添加其他会话。

```
example% cdrw -M
```

```
Track No. |Type      |Start address
-----+-----+-----
1          |Data      |0
Leadout    |Data      |166564
```

```
Last session start address: 162140
```

```
Next writable address: 173464
```

```
example% mkisofs -o /tmp/image2 -r -C 0,173464 -M \
/dev/rdisk/c0t2d0s2 /home/foo
```

属性

有关下列属性的说明, 请参见 [attributes\(5\)](#):

属性类型	属性值
可用性	media/cdrw

另请参见

[audioconvert\(1\)](#)、[priocntl\(1\)](#)、[policy.conf\(4\)](#)、[attributes\(5\)](#)、[rbac\(5\)](#)、[scsa2usb\(7D\)](#)、[sd\(7D\)](#)

SUNWfsman 软件包中的 [mkisofs\(8\)](#) (/usr/share/man/man8/mkisofs.8)

《Oracle Solaris 11.1 管理：设备和文件系统》

附注

CD 写入进程要求以恒定速率向驱动器提供数据。在写入 CD 时，将 I/O 活动保持在最小程度并关闭任何相关的 I/O 应用程序。

在创建副本或提取声道时，请使用符合 MMC 的源 CD-ROM 驱动器。可以使用 CD 写入器执行这些操作。

在写入 CD 之前，通过使用 `-M` 选项确保该介质为空。您可以使用 `-s` 模拟模式测试系统，以确保系统可以按所需速率提供数据。`cdrw` 会为支持它的驱动器打开缓冲区欠载保护并从大多数停顿进行恢复。如果系统无法以恒定速率提供数据或者频繁发生停顿，您可以使用 `-p` 选项减慢速度。也可以通过使用 `prionctl(1)` 命令尝试以较高优先级运行 `cdrw`。

如果您知道 CD-R/RW 驱动器可以按不同的写入速度运行，请使用 `-p` 选项。有些市售驱动器处理驱动器速度设置命令的方式有所不同，所以请明智地使用此选项。

`cdrw` 命令使用 `rbac(5)` 控制用户对设备的访问权。缺省情况下，所有用户都可以访问 `cdrw`，但可设置为仅限单个用户访问。有关更多信息，请参见《[Oracle Solaris 11.1 管理：设备和文件系统](#)》。

要以非 `root` 用户身份刻录 CD，必须启用 `hal` 且用户必须在控制台上。`hal`（即 `svc:/system/hal` SMF 服务）在缺省情况下处于启用状态，因此，通常不需要特殊操作。

用户必须登录到控制台。`/dev/console` 也是正确的。以前，用户可以远程登录（例如，通过使用 `telnet` 或 `ssh`）且能够刻录 CD。除非管理员已将缺省配置更改为拒绝 `solaris.device.cdrw` 授权，否则以上情况仍有效。请参见 `policy.conf(4)`。

引用名 checknr – check nroff and troff input files; report possible errors

用法概要 checknr [-fs] [-a .x1 .y1 .x2 .y2xn .yn]
[-c .x1 .x2 .x3xn] [filename]...

描述 checknr checks a list of [nroff\(1\)](#) or [troff\(1\)](#) input files for certain kinds of errors involving mismatched opening and closing delimiters and unknown commands. If no files are specified, checknr checks the standard input. Delimiters checked are:

- Font changes using `\fx ... \fP`.
- Size changes using `\sx ... \s0`.
- Macros that come in open ... close forms, for example, the `.TS` and `.TE` macros which must always come in pairs.

checknr knows about the [ms\(5\)](#) and [me\(5\)](#) macro packages.

checknr is intended to be used on documents that are prepared with checknr in mind. It expects a certain document writing style for `\f` and `\s` commands, in that each `\fx` must be terminated with `\fP` and each `\sx` must be terminated with `\s0`. While it will work to directly go into the next font or explicitly specify the original font or point size, and many existing documents actually do this, such a practice will produce complaints from checknr. Since it is probably better to use the `\fP` and `\s0` forms anyway, you should think of this as a contribution to your document preparation style.

选项

- f Ignore `\f` font changes.
- s Ignore `\s` size changes.
- a .x1 .y1 ... Add pairs of macros to the list. The pairs of macros are assumed to be those (such as `.DS` and `.DE`) that should be checked for balance. The -a option must be followed by groups of six characters, each group defining a pair of macros. The six characters are a period, the first macro name, another period, and the second macro name. For example, to define a pair `.BS` and `.ES`, use `'-a.BS.ES'`
- c .x1 ... Define commands which checknr would otherwise complain about as undefined.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

另请参见 [eqn\(1\)](#), [nroff\(1\)](#), [troff\(1\)](#), [attributes\(5\)](#), [me\(5\)](#), [ms\(5\)](#)

已知问题

There is no way to define a one-character macro name using the -a option.

引用名 chgrp – change file group ownership

用法概要 chgrp [-fhR] *group file...*

chgrp -s [-fhR] *groupsid file...*

chgrp -R [f] [-H | -L | -P] *group file...*

chgrp -s -R [f] [-H | -L | -P] *groupsid file...*

描述 The chgrp utility will set the group ID of the file named by each *file* operand to the group ID specified by the *group* operand.

For each *file* operand, it will perform actions equivalent to the [chown\(2\)](#) function, called with the following arguments:

- The *file* operand will be used as the *path* argument.
- The user ID of the file will be used as the *owner* argument.
- The specified group ID will be used as the *group* argument.

Unless chgrp is invoked by a process with appropriate privileges, the set-user-ID and set-group-ID bits of a regular file will be cleared upon successful completion; the set-user-ID and set-group-ID bits of other file types may be cleared.

The operating system has a configuration option `_POSIX_CHOWN_RESTRICTED`, to restrict ownership changes. When this option is in effect, the owner of the file may change the group of the file only to a group to which the owner belongs. Only the super-user can arbitrarily change owner IDs, whether or not this option is in effect. To set this configuration option, include the following line in `/etc/system`:

```
set rstchown = 1
```

To disable this option, include the following line in `/etc/system`:

```
set rstchown = 0
```

`_POSIX_CHOWN_RESTRICTED` is enabled by default. See [system\(4\)](#) and [fpathconf\(2\)](#).

选项 The following options are supported.

`/usr/bin/chgrp` and
`/usr/xpg4/bin/chgrp`

- f Force. Does not report errors.
- h If the file is a symbolic link, this option changes the group of the symbolic link. Without this option, the group of the file referenced by the symbolic link is changed.
- H If the file specified on the command line is a symbolic link referencing a file of type directory, this option changes the group of the directory referenced by the symbolic link and all the files in the file hierarchy below it. If a symbolic link is encountered when traversing a file hierarchy, the group of the target file is changed, but no recursion takes place.

- L If the file is a symbolic link, this option changes the group of the file referenced by the symbolic link. If the file specified on the command line, or encountered during the traversal of the file hierarchy, is a symbolic link referencing a file of type directory, then this option changes the group of the directory referenced by the symbolic link and all files in the file hierarchy below it.
- P If the file specified on the command line or encountered during the traversal of a file hierarchy is a symbolic link, this option changes the group of the symbolic link. This option does not follow the symbolic link to any other part of the file hierarchy.
- s The specified group is Windows SID. This option requires a file system that supports storing SIDs, such as ZFS.

Specifying more than one of the mutually-exclusive options -H, -L, or -P is not considered an error. The last option specified determines the behavior of `chgrp`.

- `/usr/bin/chgrp` -R Recursive. `chgrp` descends through the directory, and any subdirectories, setting the specified group ID as it proceeds. When a symbolic link is encountered, the group of the target file is changed, unless the -h or -P option is specified. However, no recursion takes place, unless the -H or -L option is specified.
- `/usr/xpg4/bin/chgrp` -R Recursive. `chgrp` descends through the directory, and any subdirectories, setting the specified group ID as it proceeds. When a symbolic link is encountered, the group of the target file is changed, unless the -h or -P option is specified. Unless the -H, -L, or -P option is specified, the -L option is used as the default mode.

操作数

The following operands are supported:

- group* A group name from the group database or a numeric group ID. Either specifies a group ID to be given to each file named by one of the *file* operands. If a numeric *group* operand exists in the group database as a group name, the group ID number associated with that group name is used as the group ID.
- file* A path name of a file whose group ID is to be modified.

用法

See [largefile\(5\)](#) for the description of the behavior of `chgrp` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `chgrp`: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 The utility executed successfully and all requested changes were made.
- >0 An error occurred.

文件 /etc/group group file

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/chgrp

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled. See NOTES.
Interface Stability	Committed
Standard	See standards(5) .

/usr/xpg4/bin/chgrp

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled. See NOTES.
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [chmod\(1\)](#), [chown\(1\)](#), [id\(1M\)](#), [chown\(2\)](#), [fpathconf\(2\)](#), [group\(4\)](#), [passwd\(4\)](#), [system\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注 chgrp is CSI-enabled except for the *group* name.

引用名	chkey – change user's secure RPC key pair
用法概要	chkey [-p] [-s nis files ldap] [-m <mechanism>]
描述	<p>chkey is used to change a user's secure RPC public key and secret key pair. chkey prompts for the old secure-rpc password and verifies that it is correct by decrypting the secret key. If the user has not already used keylogin(1) to decrypt and store the secret key with keyserv(1M), chkey registers the secret key with the local keyserv(1M) daemon. If the secure-rpc password does not match the login password, chkey prompts for the login password. chkey uses the login password to encrypt the user's secret Diffie-Hellman (192 bit) cryptographic key. chkey can also encrypt other Diffie-Hellman keys for authentication mechanisms configured.</p> <p>chkey ensures that the login password and the secure-rpc password(s) are kept the same, thus enabling password shadowing. See shadow(4).</p> <p>The key pair can be stored in the <code>/etc/publickey</code> file (see publickey(4)) or the NIS <code>publickey</code> map. If a new secret key is generated, it will be registered with the local keyserv(1M) daemon.</p> <p>Keys for specific mechanisms can be changed or re-encrypted using the <code>-m</code> option followed by the authentication mechanism name. Multiple <code>-m</code> options can be used to change one or more keys.</p> <p>If the source of the <code>publickey</code> is not specified with the <code>-s</code> option, chkey consults the <code>publickey</code> entry in the name service switch configuration file. See nsswitch.conf(4). If the <code>publickey</code> entry specifies one and only one source, then chkey will change the key in the specified name service. However, if multiple name services are listed, chkey can not decide which source to update and will display an error message. The user should specify the source explicitly with the <code>-s</code> option.</p> <p>Non root users are not allowed to change their key pair in the <code>files</code> database.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -p Re-encrypt the existing secret key with the user's login password. -s nis Update the NIS database. -s files Update the <code>files</code> database. -s ldap Update the LDAP database. -m <mechanism> Changes or re-encrypt the secret key for the specified mechanism.
文件	<p><code>/etc/nsswitch.conf</code></p> <p><code>/etc/publickey</code></p>

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[keylogin\(1\)](#), [keylogout\(1\)](#), [keyserv\(1M\)](#), [newkey\(1M\)](#), [nsswitch.conf\(4\)](#), [publickey\(4\)](#), [shadow\(4\)](#), [attributes\(5\)](#)

引用名	chmod – change the permissions mode of a file
用法概要	<pre>chmod [-fR] <i>absolute-mode</i> file...</pre> <pre>chmod [-fR] <i>symbolic-mode-list</i> file...</pre> <pre>chmod [-fR] <i>acl_operation</i> file...</pre> <pre>chmod [-fR] [-@ <i>named_attribute</i>]...<i>attribute_specification_list</i> file...</pre>
描述	<p>The chmod utility changes or assigns the mode of a file.</p> <p>chmod can also be used to modify Access Control Lists (ACLs) on files and directories, and to modify boolean read-write system attributes on regular files, directories, and opaque extended attribute files.</p>
Absolute Mode	<p>An absolute mode command line has the following format:</p> <pre>chmod [<i>options</i>] <i>absolute-mode</i> file...</pre> <p>where <i>absolute-mode</i> is specified using octal numbers <i>nmmn</i> defined as follows:</p> <p><i>n</i></p> <p>a number from 0 to 7. An absolute mode is constructed from the OR of any of the following modes:</p> <p>4000 Set user ID on execution.</p> <p>20 # 0 Set group ID on execution if # is 7, 5, 3, or 1. Enable mandatory locking if # is 6, 4, 2, or 0.</p> <p>For directories, files are created with BSD semantics for propagation of the group ID. With this option, files and subdirectories created in the directory inherit the group ID of the directory, rather than of the current process. For directories, the setgid bit can only be set or cleared by using symbolic mode.</p> <p>1000 Turn on sticky bit. See chmod(2).</p> <p>0400 Allow read by owner.</p> <p>0200 Allow write by owner.</p> <p>0100 Allow execute (search in directory) by owner.</p> <p>0700 Allow read, write, and execute (search) by owner.</p>

0040

Allow read by group.

0020

Allow write by group.

0010

Allow execute (search in directory) by group.

0070

Allow read, write, and execute (search) by group.

0004

Allow read by others.

0002

Allow write by others.

0001

Allow execute (search in directory) by others.

0007

Allow read, write, and execute (search) by others.

For directories, the `setgid` bit cannot be set (or cleared) in absolute mode; it must be set (or cleared) in symbolic mode using `g+s` (or `g-s`).

Symbolic Mode

A symbolic mode command line has the following format:

```
chmod [options] symbolic-mode-list file . . .
```

where *symbolic-mode-list* is a comma-separated list (with no intervening white space) of symbolic mode expressions of the form:

[*who*] *operator* [*permissions*]

Operations are performed in the order given. Multiple *permissions* letters following a single operator cause the corresponding operations to be performed simultaneously.

who

zero or more of the characters `u`, `g`, `o`, and `a` specifying whose permissions are to be changed or assigned:

`u`

user's permissions

`g`

group's permissions

`o`

others' permissions

a
all permissions (user, group, and other)

If *who* is omitted, it defaults to **a**, but the setting of the file mode creation mask (see [umask](#) in [sh\(1\)](#) or [csh\(1\)](#) for more information) is taken into account. When *who* is omitted, `chmod` does not override the restrictions of your user mask.

operator

either +, -, or =, signifying how permissions are to be changed:

+
Add permissions.

If *permissions* are omitted, nothing is added.

If *who* is omitted, add the file mode bits represented by *permissions*, *except* for the those with corresponding bits in the file mode creation mask.

If *who* is present, add the file mode bits represented by the *permissions*.

-
Take away permissions.

If *permissions* are omitted, do nothing.

If *who* is omitted, clear the file mode bits represented by *permissions*, *except* for those with corresponding bits in the file mode creation mask.

If *who* is present, clear the file mode bits represented by *permissions*.

=
Assign permissions absolutely.

If *who* is omitted, clear all file mode bits; if *who* is present, clear the file mode bits represented by *who*.

If *permissions* are omitted, do nothing else.

If *who* is omitted, add the file mode bits represented by *permissions*, *except* for the those with corresponding bits in the file mode creation mask.

If *who* is present, add the file mode bits represented by *permissions*.

Unlike other symbolic operations, = has an absolute effect in that it resets all other bits represented by *who*. Omitting *permissions* is useful only with = to take away all permissions.

permission

any compatible combination of the following letters:

l
mandatory locking

r
read permission

s
user or group set-ID

t
sticky bit

w
write permission

x
execute permission

X
execute permission if the file is a directory or if there is execute permission for one of the other user classes

u,g,o
indicate that *permission* is to be taken from the current user, group or other mode respectively.

Permissions to a file can vary depending on your user identification number (UID) or group identification number (GID). Permissions are described in three sequences each having three characters:

User	Group	Other
rxw	rxw	rxw

This example (user, group, and others all have permission to read, write, and execute a given file) demonstrates two categories for granting permissions: the access class and the permissions themselves.

The letter *s* is only meaningful with *u* or *g*, and *t* only works with *u*.

Mandatory file and record locking (*l*) refers to a file's ability to have its reading or writing permissions locked while a program is accessing that file.

In a directory which has the set-group-ID bit set (reflected as either `-----s---` or `-----l---` in the output of `'ls -ld'`), files and subdirectories are created with the group-ID of the parent directory—not that of current process.

It is not possible to permit group execution and enable a file to be locked on execution at the same time. In addition, it is not possible to turn on the set-group-ID bit and enable a file to be locked on execution at the same time. The following examples, therefore, are invalid and elicit error messages:


```
chmod g+x,+l file
chmod g+s,+l file
```

Only the owner of a file or directory (or the super-user) can change that file's or directory's mode. Only the super-user can set the sticky bit on a non-directory file. If you are not super-user, chmod masks the sticky-bit but does not return an error. In order to turn on a file's set-group-ID bit, your own group ID must correspond to the file's and group execution must be set.

ACL Operation

An Access Control List (ACL) is a list of Access Control Entries (ACEs), each of which define access permissions for a particular class of user. The list of ACEs is numbered, starting from zero. The position of an ACE within an ACL is called an *index*. This index is used as an argument in many of the chmod commands described below. See 《Oracle Solaris 11.1 管理：ZFS 文件系统》 for further description of ACLs and ACEs.

Oracle Solaris utilities, including chmod, support both the NFSv4 and the newer POSIX-draft ACL specifications. These specifications spell out the syntax and semantics of the *acl_specification* field shown below. These two ACL specifications are described in their respective subsections, below.

An ACL Operation command line has the following format:

```
chmod [options] A[index]- file ...
chmod [options] A-acl_specification file ...
chmod [options] A[index]{+|=}acl_specification file ...
```

...where *acl_specification* is a comma-separated list (with no intervening whitespace) of the form:

A[index]+acl_specification

Prepends the access control entries (ACE) specified in *acl_specification* to the beginning of the file's ACL. Depending on the file system, the ACL can be reordered when applied to the file. If the optional *index* is specified, then new ACEs are inserted before specified *index*.

A-

Removes all ACEs for current ACL on file and replaces current ACL with new ACL that represents only the current mode of the file.

Aindex-

Removes ACE specified by *index* number.

A-*acl_specification*

Removes ACEs specified by *acl_specification*, if they exist in current file's ACL.

A=*acl_specification*

Replaces a files entire ACL with *acl_specification*.

A[*index*]=*acl_specification*

Replaces ACEs starting at a specific index number in the current ACL on the file. If multiple ACEs are specified, then each subsequent ACE in *acl_specification* replaces the

corresponding ACE in the current ACL.

POSIX-draft ACL Specification (as supported by UFS)

POSIX-draft ACLs (as supported by UFS) are specified as colon (:) separated fields of the following.

`user::perms`

File owner permissions.

`user:username:perms`

Permissions for a specific user.

`group::perms`

File group member permissions.

`group:groupname:perms`

Permissions for a specific group.

`other::perms`

Permissions for user other than the file owner or members of file group.

`mask:perms`

The ACL mask. The mask entry specifies the maximum permissions allowed for user (other than that the owner) and for groups.

`default:user::perms`

Default file owner permissions.

`default:user:username:perms`

Default permissions for a specific user.

`default:group::perms`

Default file group member permissions.

`default:group:groupname:perms`

Default permissions for a specific group.

`default:other:perms`

Default permissions for user other than the file owner or members of the file group.

`default:mask:perms`

Default ACL mask.

The above specification allows for ACLs to be specified such as:

```
user:tom:rw-,mask:rwx,group:staff:r-x
```

NFSv4 ACL Specification (as supported by NFSv4 and ZFS)

NFSv4 ACLs provide richer ACL semantics. They provide both allow and deny entries, finer-grained permissions, and enhanced inheritance control.

NFSv4 ACLs are specified as colon (:) separated fields of the following.

owner@:<perms>[:inheritance flags]:<allow|deny>
Permissions for file owner.

group@:<perms>[:inheritance flags]:<allow|deny>
Permissions for file group member.

everyone@:<perms>[:inheritance flags]:<allow|deny>
Permissions for everyone, including file owner and group member.

user:<username>:<perms>[:inheritance flags]:<allow|deny>
Permissions for a specific user.

usersid:<sid string>:<perms>[:inheritance flags]:<allow|deny>
Permissions for a specific user, but user is specified by SID.

group:<groupname>:<perms>[:inheritance flags]:<allow|deny>
Permissions for a specific group.

groupsid:<sid string>:<perms>[:inheritance flags]:<allow|deny>
Permissions for a specific group, but group is specified by SID.

sid:<sid string>:<perms>[:inheritance flags]:<allow|deny>
Permissions for a specific SID, but it doesn't matter if it is a user or a group.

Permissions can be specified in three different chmod ACL formats: verbose, compact, or positional. The verbose format uses words to indicate that the permissions are separated with a forward slash (/) character. Compact format uses the permission letters and positional format uses the permission letters or the hyphen (-) to identify no permissions.

The permissions for verbose mode and their abbreviated form in parentheses for compact and positional mode are described as follows:

read_data (r)
Permission to read the data of a file.

list_directory (r)
Permission to list the contents of a directory.

write_data (w)
Permission to modify a file's data. anywhere in the file's offset range.

add_file (w)
Permission to add a new file to a directory.

append_data (p)
The ability to modify a file's data, but only starting at EOF.

Currently, this permission is not supported.

add_subdirectory (p)
Permission to create a subdirectory to a directory.

read_xattr (R)

Ability to read the extended attributes of a file.

write_xattr (W)

Ability to create extended attributes or write to the extended attribute directory.

execute (x)

Permission to execute a file.

read_attributes (a)

The ability to read basic attributes (non-ACLs) of a file.

write_attributes (A)

Permission to change the times associated with a file or directory to an arbitrary value.

delete (d)

Permission to delete a file.

For more information about delete permission behavior, see the [《Oracle Solaris 11.1 管理：ZFS 文件系统》](#).

delete_child (D)

Permission to delete a file within a directory.

For more information about delete permission behavior, see the [《Oracle Solaris 11.1 管理：ZFS 文件系统》](#)

read_acl (c)

Permission to read the ACL of a file.

write_acl (C)

Permission to write the ACL of a file.

write_owner (o)

Permission to change the owner of a file.

synchronize (s)

Permission to access file locally at server with synchronize reads and writes.

Currently, this permission is not supported.

Using the compact ACL format, permissions are specified by using 14 unique letters to indicate permissions.

Using the positional ACL format, permissions are specified as positional arguments similar to the `ls -V` format. The hyphen (-), which indicates that no permission is granted at that position, can be omitted and only the required letters have to be specified.

The letters above are listed in the order they would be specified in positional notation.

Permissions can be specified with these letters in the following way:

```
rx--D-----
```

The hyphens can be removed to compact the string as follows:

```
rxD
```

Several special permission sets or aliases are also supported. The following permission sets are used the same way that verbose permissions are specified.

```
full_set
```

All permissions.

```
modify_set
```

All permissions except `write_acl` and `write_owner`.

```
read_set
```

`read_data`, `read_acl`, `read_attributes`, and `read_xattr`.

```
write_set
```

`write_data`, `append_data`, `write_attributes`, and `write_xattr`

The optional inheritance flags can be specified in the three formats. The first format uses words to indicate the various inheritance flags separated with a forward slash (/) character.

```
file_inherit (f)
```

Inherit to all newly created files.

```
dir_inherit (d)
```

Inherit to all newly created directories.

```
inherit_only (i)
```

When placed on a directory, do not apply to the directory, only to newly created files and directories. This flag requires that either `file_inherit` and or `dir_inherit` is also specified.

```
no_propagate (n)
```

Indicates that ACL entries should be inherited to objects in a directory, but inheritance should stop after descending one level. This flag is dependent upon either `file_inherit` and or `dir_inherit` also being specified.

The inheritance flags listed can also be specified in the compact format or as positional arguments similar to the `ls -V` format. A hyphen character indicates that the inheritance flag at that position is not specified in the positional ACL format.

The inheritance flags can be specified with these letters in any of the following equivalent ways.

```
file_inherit/dir_inherit/no_propagate
```

```
fd-n--
```

```
fdn
```

With this inheritance model, an ACL entry can be specified such as:

```
user:tom:read_data/write_data/read_attributes:file_inherit:allow
user:fred:read_data:file_inherit/dir_inherit:deny
user:bob:read_data:allow
```

Attribute Operation An attribute operation command line has the following format:

```
chmod [options] attribute_specification_list file ...
```

where *attribute_specification_list* is the character *S* followed by a comma-separated list of one or more *attribute_specifications*. Each *attribute_specification* is of the form:

```
[operator]attribute_specifier
```

An *operator* is one of the following:

- +
Each attribute specified by the associated *attribute_specifier* is adjusted to match the value specified by the *attribute_specifier*.
- Each attribute specified by the associated *attribute_specifier* is adjusted to match the inverse of the value specified by the *attribute_specifier*.
- =
Each attribute specified by the associated *attribute_specifier* is adjusted to match the value specified by the *attribute_specifier*. Any boolean read-write extended system attributes associated with the current file that are not specified by *attribute_specifier* is cleared.

If an *operator* is not specified in an *attribute_specification*, chmod behaves as if + had been specified.

An *attribute_specifier* takes one of the following forms:

- a
Set all boolean read-write extended system attributes associated with the current file.
- c [*compact_attribute_list*]
c ' { '*compact_attribute_list*' } '
Set each boolean read-write extended system attribute identified by *compact_attribute_list*.
- v [*verbose_attribute_setting*]
v [' { '*verbose_attribute_setting_list*' } '
Set each boolean read-write extended system attribute identified by *verbose_attribute_setting*.

A *compact_attribute_list* is a list of zero or more adjacent attribute abbreviation characters from list of *Attribute Names and Abbreviation Characters* later in this section. An arbitrary number of hyphen (-) characters can be included in a *compact_attribute_list*. These are ignored.

A *verbose_attribute_setting* is an attribute name from the list of *Attribute Names and Abbreviation Characters* later in this section, optionally, immediately preceded by *no*. If the attribute name is used without *no*, the attribute is set; otherwise the attribute is cleared.

A *verbose_attribute_setting_list* is zero or more comma-separated *verbose_attribute_settings*.

Multiple operations specified for a file are accumulated and are all set for a file operand as a single attribute setting operation. If an attribute is specified more than once in an *attribute_specification_list*, the last specified operation is applied.

The following is a list of *Attribute Names and Abbreviation Characters*:

Attribute Name

Abbreviation Character

hidden

H

sparse

s

system

S

readonly

R

archive

A

nounlink

u

immutable

i

appendonly

a

nodump

d

av_quarantined

q

av_modified

m

选项

The following options are supported:

-f

Force. chmod does not complain if it fails to change the mode of a file.

-R

Recursively descend through directory arguments, setting the mode for each file. When symbolic links are encountered, the mode of the target file is changed, but no recursion takes place.

-@ *named_attribute*

Perform the attribute operation on the named extended attribute file of each file operand instead of the file operand itself. If multiple -@ operations are supplied, the attribute specification mode is applied to each of the named attribute files.

A named attribute of * carries meaning to chmod, and is considered to mean all extended attribute files associated with a file operand. This does not refer to the special files . and ..

A named attribute of . . carries special meaning to chmod, and is considered to mean the file operand itself. This allows chmod, in a single call, to apply the attribute specification mode to the specified named attribute file of the file operand and the file operand itself.

操作数

The following operands are supported:

absolute-mode

symbolic-mode-list

Represents the change to be made to the file mode bits of each file named by one of the *file* operands. See Absolute Mode and Symbolic Mode in the DESCRIPTION section of this manual page for more information.

acl_operation

Represents the modification to be performed on the file's ACL. See ACL Operation in the DESCRIPTION section for more information.

acl_operation is one of the following:

A[*number*] -

A-*acl_specification*

A[*index*]{+|=}*acl_specification*

attribute_specification_list

Represents the modification to be performed on the file's attributes. See Attribute Operation in the DESCRIPTION section of this manual page for more information.

file

A path name of a file whose file mode bits are to be modified.

用法

See [largefile\(5\)](#) for the description of the behavior of chmod when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Denying execute Permission

The following example denies execute permission to everyone:

```
% chmod a-x file
```


示例 2 Allowing read-only Permission

The following example allows only read permission to everyone:

```
% chmod 444 file
```

示例 3 Making a File readable and writable

The following example makes a file readable and writable by the group and others:

```
% chmod go+rw file
```

```
% chmod 066 file
```

示例 4 Locking a File From Access

The following example locks a file from access:

```
$ chmod +l file
```

示例 5 Granting read, write, execute, and set group-ID Permission on a File

The following example grants everyone read, write, and execute permissions on the file, and turns on the set group-ID:

```
$ chmod a=rwx,g+s file
```

```
$ chmod 2777 file
```

示例 6 Prepending a New ACL Entry on a ZFS File

The following example prepends a new ACL entry on a ZFS file.

First, display the current ACL:

```
$ ls -v file.3
-rw-r--r-- 1 marks  staff          0 Oct  9 15:49 file.3
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/
  write_attributes/write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/
  write_attributes/write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/
  synchronize:allow
```

Issue the following command:

```
$ chmod A+user:lp:read_data:deny file.3
```

Display the new ACL:

```
$ ls -v file.3
-rw-r--r--+ 1 marks  staff          0 Oct  9 15:49 file.3
0:user:lp:read_data:deny
```

示例 6 Prepending a New ACL Entry on a ZFS File (续)

```
1:owner@:execute:deny
2:owner@:read_data/write_data/append_data/write_xattr/
  write_attributes/write_acl/write_owner:allow
3:group@:write_data/append_data/execute:deny
4:group@:read_data:allow
5:everyone@:write_data/append_data/write_xattr/execute/
  write_attributes/write_acl/write_owner:deny
6:everyone@:read_data/read_xattr/read_attributes/read_acl/
  synchronize:allow
```

示例 7 Prepending a New POSIX-draft ACL Entry on a UFS File

The following example prepends a new POSIX-draft ACL entry on a UFS file.

First, display the current ACL:

```
$ ls -v file.2
-rw-r--r--  1 marks   staff           0 Oct  9 15:52 file.2
 0:user::rw-
 1:group::r--          #effective:r--
 2:mask:r--
 3:other:r--
```

Issue the following command:

```
$ chmod A+user:lp:-wx file.2
```

Display the new ACL:

```
$ ls -v file.2
-rw-r--r--+ 1 marks   staff           0 Oct  9 15:52 file.2
 0:user::rw-
 1:user:lp:-wx          #effective:---
 2:group::r--          #effective:r--
 3:mask:r--
 4:other:r--
```

示例 8 Inserting an ACL Entry in a Specific Position on a ZFS file

The following example inserts an ACL entry in a specific position on a ZFS file system. It also illustrates the compact ACL format.

First, display the ACL to pick a location to insert a new ACE.

```
% ls -V file.1
-rw-r--r--+ 1 root    root           0 Oct  6 12:16 file.1
  user:lp:rw-----:-----:allow
  owner@:--x-----:-----:deny
  owner@:rw-p---A-W-Co-:-----:allow
```

示例 8 Inserting an ACL Entry in a Specific Position on a ZFS file (续)

```
group@:-wpx-----:-----:deny
group@:r-----:-----:allow
everyone@:-wpx---A-W-Co-:-----:deny
everyone@:r-----a-R-c--s:-----:allow
```

Next, insert a new entry in location 3. This causes the entries that are currently in position 3 - 6 to be pushed down.

Issue the following command:

```
$ chmod A3+user:marks:r:deny file.1
```

Display the new ACL:

```
$ ls -V file.1
-rw-r--r--+ 1 root    staff          0 Feb  3 14:13 file.1
  user:lp:rw-----:-----:allow
  owner@:-x-----:-----:deny
  owner@:rw-p---A-W-Co-:-----:allow
  user:marks:r-----:-----:deny
  group@:-wpx-----:-----:deny
  group@:r-----:-----:allow
  everyone@:-wpx---A-W-Co-:-----:deny
  everyone@:r-----a-R-c--s:-----:allow
```

示例 9 Inserting a POSIX-draft ACL in a Specific Position on a UFS File

The file system reorders ACLs when they are stored in the file system. The following example illustrates this behavior.

```
$ ls -v file.1
-rw-r--r--+ 1 root    root          0 Sep 29 16:10 file.1
  0:user::rw-
  1:user:lp:rw-      #effective:r--
  2:group::r--      #effective:r--
  3:mask:r--
  4:other:r--
```

Now, insert an entry at index position 3. The command works, but the file system reorders the ACL.

```
$ chmod A3+user:marks:rw- file.1
$ ls -v file.1
-rw-r--r--+ 1 root    root          0 Sep 29 16:10 file.1
  0:user::rw-
  1:user:lp:rw-      #effective:r--
  2:user:marks:rw-   #effective:r--
  3:group::r--      #effective:r--
```

示例 9 Inserting a POSIX-draft ACL in a Specific Position on a UFS File (续)

```
4:mask:r--
5:other:r--
```

Rather than inserting the ACL entry in position 3 as requested, it actually ends up in position 2.

示例 10 Removing an ACL Entry on a ZFS File

The following example removes the lp entry from an ACL:

```
$ ls -v file.3
-rw-r--r--+ 1 marks    staff          0 Oct  9 15:49 file.3
 0:user:lp:read_data:deny
 1:owner@:execute:deny
 2:owner@:read_data/write_data/append_data/write_xattr/
  write_attributes/write_acl/write_owner:allow
 3:group@:write_data/append_data/execute:deny
 4:group@:read_data:allow
 5:everyone@:write_data/append_data/write_xattr/execute/
  write_attributes/write_acl/write_owner:deny
 6:everyone@:read_data/read_xattr/read_attributes/read_acl/
  synchronize:allow

$ chmod A-user:lp:read_data:deny file.3
$ ls -v file.3
-rw-r--r--  1 marks    staff          0 Oct  9 15:49 file.3
 0:owner@:execute:deny
 1:owner@:read_data/write_data/append_data/write_xattr/
  write_attributes/write_acl/write_owner:allow
 2:group@:write_data/append_data/execute:deny
 3:group@:read_data:allow
 4:everyone@:write_data/append_data/write_xattr/execute/
  write_attributes/write_acl/write_owner:deny
 5:everyone@:read_data/read_xattr/read_attributes/read_acl/
  synchronize:allow
```

示例 11 Removing a POSIX-draft ACL on a UFS File

The following example removes the lp entry from an ACL:

```
$ ls -v file.2
-rw-r--r--+ 1 marks    staff          0 Oct  9 15:52 file.2
 0:user::rw-
 1:user:lp:-wx      #effective:---
 2:group::r--      #effective:r--
 3:mask:r--
 4:other:r--
```

示例 11 Removing a POSIX-draft ACL on a UFS File (续)

```
$ chmod A-user:lp:-wx file.2
$ ls -v file.2
-rw-r--r--  1 marks  staff           0 Oct  9 15:52 file.2
 0:user::rw-
 1:group::r--          #effective:r--
 2:mask:r--
 3:other:r--
```

示例 12 Removing a Specific ACL Entry by Index Number on a ZFS File

Consider the following ACL:

```
$ ls -v file
 0:group:staff:read_data/write_data/execute/read_acl:allow
 1:user:bin:read_data:deny
 2:user:bin:read_data:allow
 3:owner@:write_data/append_data:deny
 4:owner@:read_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 5:group@:write_data/append_data:deny
 6:group@:read_data/execute:allow
 7:everyone@:write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:deny
 8:everyone@:read_data/read_xattr/execute/read_attributes/read_acl
  /synchronize:allow
```

Remove the second user entry for bin.

```
$ chmod A2- file
$ ls -v file
 0:group:staff:read_data/write_data/execute/read_acl:allow
 1:user:bin:read_data:deny
 2:owner@:write_data/append_data:deny
 3:owner@:read_data/write_xattr/execute/write_attributes/write_acl
  /write_owner:allow
 4:group@:write_data/append_data:deny
 5:group@:read_data/execute:allow
 6:everyone@:write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:deny
 7:everyone@:read_data/read_xattr/execute/read_attributes/read_acl
  /synchronize:allow
```

示例 13 Removing a Specific POSIX-draft ACL Entry on a UFS File

The following example removes the lp entry by index number from the following ACL:

```
$ ls -v file.1
-rw-r--r--+  1 root  root           0 Sep 29 16:10 file.1
 0:user::rw-
```

示例 13 Removing a Specific POSIX-draft ACL Entry on a UFS File (续)

```

1:user:lp:rw-          #effective:r--
2:group::r--          #effective:r--
3:mask:r--
4:other:r--

$ chmod A1- file.1
$ ls -v
-rw-r--r--+ 1 root      root          0 Sep 29 16:10 file.1
0:user::rw-
1:group::r--          #effective:r--
2:mask:r--
3:other:r--

```

示例 14 Removing All ACLs From a File

The following command works with either NFSv4/ZFS or POSIX-draft ACLs.

Consider the following ACL:

```

$ ls -v file.3
-rw-r--r--+ 1 marks    staff          0 Oct  9 15:49 file.3
0:user:lp:read_data/write_data:allow
1:user:marks:read_acl:allow
2:owner@:execute:deny
3:owner@:read_data/write_data/append_data/write_xattr/
  write_attributes/write_acl/write_owner:allow
4:group@:write_data/append_data/execute:deny
5:group@:read_data:allow
6:everyone@:write_data/append_data/write_xattr/execute/
  write_attributes/write_acl/write_owner:deny
7:everyone@:read_data/read_xattr/read_attributes/read_acl/
  synchronize:allow

```

The existing ACL is effectively removed and is replaced with an ACL that represents the permission bits of the file.

```

$ chmod A- file.3
$ ls -v file.3
-rw-r--r-- 1 marks    staff          0 Oct  9 15:49 file.3
0:owner@:execute:deny
1:owner@:read_data/write_data/append_data/write_xattr/
  write_attributes/write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/
  write_attributes/write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/
  synchronize:allow

```

示例 15 Replacing an Entire ACL Entry on a ZFS File

Use the following chmod syntax if you want to replace an ACL in its entirety:

```
$ chmod A=owner@:read_data/write_data:allow,group@:read_data/
      write_data:allow,user:lp:read_data:allow file.4
$ ls -v file.4
-rw-rw----+ 1 marks  staff          0 Oct  9 16:12 file.4
      0:owner@:read_data/write_data:allow
      1:group@:read_data/write_data:allow
      2:user:lp:read_data:allow
```

示例 16 Replacing an Entire POSIX-draft ACL on a UFS File

This operation is a little more complicated. The replacement ACL needs the necessary entries to represent the file owner, file group owner, other, mask and any additional entries you wish to set.

```
$ chmod A=user::rw-,group::rw-,other::---,mask:r--,
      user:lp:r-- file.3
$ ls -v file.3
-rw-r-----+ 1 root  root          0 Oct  9 16:14 file.3
      0:user::rw-
      1:user:lp:r--          #effective:r--
      2:group::rw-          #effective:r--
      3:mask:r--
      4:other:---
```

示例 17 Replacing a Specific Entry on a ZFS File

Consider the following ACL.

```
$ ls -v file.5
-rw-r--r--+ 1 marks  staff          0 Oct  9 16:18 file.5
      0:user:marks:read_data:allow
      1:owner@:execute:deny
      2:owner@:read_data/write_data/append_data/write_xattr/
      write_attributes/write_acl/write_owner:allow
      3:group@:write_data/append_data/execute:deny
      4:group@:read_data:allow
      5:everyone@:write_data/append_data/write_xattr/execute/
      write_attributes/write_acl/write_owner:deny
      6:everyone@:read_data/read_xattr/read_attributes/read_acl/
      synchronize:allow
```

Now, change the allow access to a deny for user marks:

```
$ chmod A0=user:marks:read_data:deny file.5
$ ls -v file.5
-rw-r--r--+ 1 marks  staff          0 Aug 23 09:11 file.5
      0:user:marks:read_data:deny
```

示例 17 Replacing a Specific Entry on a ZFS File (续)

```
1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
  /write_acl/write_owner:allow
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

示例 18 Replacing a Specific POSIX-draft ACL on a UFS File

Consider the following ACL.

```
$ ls -v file.4
-rw-r--r--+ 1 marks    staff          0 Oct  9 16:21 file.4
  0:user::rw-
  1:user:lp:rwx      #effective:r--
  2:group::r--      #effective:r--
  3:mask:r--
  4:other:r--
```

Now, change the permission on lp from rwx to r--:

```
$ chmod A1=user:lp:r-- file.4

$ ls -v file
-rw-r--r--+ 1 marks    staff          0 Oct  9 16:21 file.4
  0:user::rw-
  1:user:lp:r--      #effective:r--
  2:group::r--      #effective:r--
  3:mask:r--
  4:other:r--
```

示例 19 Setting ACL Inheritance Flags on a ZFS File

You can only set inheritance flags on ZFS files. When setting ACLs on directories, several inheritance flags can be optionally set.

Suppose you have an ACL entry for user lp that you want to be inherited to newly created files in a directory. First, you need to create an inheritable ACL entry on the directory:

```
$ chmod A+user:lp:read_data:file_inherit:allow test.dir
$ ls -dv test.dir
drwxr-xr-x+ 2 marks    staff          2 Aug 23 09:08 test.dir/
0:user:lp:read_data:file_inherit:allow
1:owner@::deny
2:owner@:list_directory/read_data/add_file/write_data/add_subdirectory
  /append_data/write_xattr/execute/write_attributes/write_acl
```


示例 19 Setting ACL Inheritance Flags on a ZFS File (续)

```

/write_owner:allow
3:group@:add_file/write_data/add_subdirectory/append_data:deny
4:group@:list_directory/read_data/execute:allow
5:everyone@:add_file/write_data/add_subdirectory/append_data/write_xattr
/write_attributes/write_acl/write_owner:deny
6:everyone@:list_directory/read_data/read_xattr/execute/read_attributes
/read_acl/synchronize:allow

```

The `lp` entry is inherited to newly created files in the directory `test.dir`.

```

$ touch test.dir/file.test
$ ls -v test.dir/file.test
-rw-r--r--+ 1 marks    staff      0 Oct  9 16:29 test.dir/file.test
  0:user:lp::deny
  1:user:lp:read_data:allow
  2:owner@:execute:deny
  3:owner@:read_data/write_data/append_data/write_xattr/
    write_attributes/write_acl/write_owner:allow
  4:group@:write_data/append_data/execute:deny
  5:group@:read_data:allow
  6:everyone@:write_data/append_data/write_xattr/execute/
    write_attributes/write_acl/write_owner:deny
  7:everyone@:read_data/read_xattr/read_attributes/read_acl/
    synchronize:allow

```

The user `lp` entry is inherited to the newly created file. Multiple combinations of the inheritance flags can be specified. For example, if you wanted the `lp` entry to also be inherited to directories, then the following command can be used:

```

$ chmod A+user:lp:read_data:file_inherit/\
    dir_inherit:allow test.dir

```

示例 20 Replacing System Attributes of a ZFS File

The following examples replace system attributes of a ZFS file:

```

$ chmod S=v{archive,hidden,readonly,system,appendonly,\
    nonodump,immutable,noav_modified,noav_quarantined,\
    nounlink} file1

```

or

```

$ chmod S=c{AHRsaiu} file1

```

or

```

$ chmod S=c{AHRsa-i--u} file1

```

or

示例 20 Replacing System Attributes of a ZFS File (续)

```
$ chmod S=cAHRsaiu file1
```

or

```
$ chmod -@ '..' S=cAHRsaiu file1
```

Assuming appropriate privileges, this results in the following system attributes of `file1` being set: `archive`, `hidden`, `readonly`, `system`, `appendonly`, `immutable`, and `nounlink`. Assuming appropriate privileges, the following system attributes of `file1` are cleared: `nodump`, `av_modified`, and `av_quarantined`.

示例 21 Clearing All System Attributes of a ZFS File

The following examples clears all system attributes of a ZFS file:

```
$ chmod S-a file1
```

or

```
$ chmod -@ '..' S-a file1
```

Assuming appropriate privileges, all boolean read-write system attributes are cleared on `file1`.

示例 22 Setting a System Attribute of a Named Attribute File of a ZFS File

The following example sets a system attribute of a named attribute file of a ZFS file, but not of the file itself:

```
$ chmod -@ myattr S+vhidden file1
```

This results in the hidden system attribute being set for the named attribute file `myattr` of `file1`, but not the file itself.

示例 23 Setting a System Attribute of All Named Attribute File of a ZFS File

The following example sets a system attribute of all named attribute files of a ZFS file, but not of the file itself:

```
$ chmod -@ '*' S+a file1
```

示例 24 Setting a System Attribute of All Named Attribute Files of a ZFS File

The following example sets a system attribute of all named attribute files of a ZFS file, as well as of the file itself:

```
$ chmod -@ '..' -@ '*' S+vhidden file1
```

This results in the hidden system attribute being set for all named attribute files of `file1`, as well as the file itself.

示例 25 Recursively Descending Through a Directory Hierarchy

The following example recursively descends through a directory hierarchy, and sets all system attributes of all named attribute files, the ZFS file operands, as well as of the directory itself:

```
$ chmod -R -@ '..' -@ '*' S+a directory1
```

This results in the hidden system attribute being set for all named attribute files of all regular files and directories within the directory hierarchy of `directory1`, as well as of `directory1` itself.

示例 26 Setting the hidden and system System Attributes of a ZFS File

The following examples set the hidden and system system attributes of a ZFS file:

```
$ chmod S+cHS file1
```

or

```
$ chmod S+vhiddden,+vsystem file1
```

or

```
$ chmod S+v{hiddden,system} file1
```

or

```
$ chmod S+c{-HS-----} file1
```

or

```
$ chmod S-v{nohiddden,nosystem} file1
```

or

```
$ chmod S-v{hiddden,system},+v{hiddden,system} file1
```

示例 27 Clearing All System Attributes of a ZFS File

The following example clears all system attributes of a ZFS file:

```
$ chmod S-a file1
```

or

```
$ chmod S=v{} file1
```

In the following two examples, the last attribute operation specified takes precedence.

In this example, the replacement attribute name list (`{}`) clears all system attributes for `file1`:

```
$ chmod S+cHS,=v{} file1
```

In this example, the clear attributes operation (`-a`) clears all system attributes of `file1`:

```
$ chmod S+vhiddden,+vsystem,-a file1
```

示例 28 Setting the Values of All Boolean read-write System Attributes of a File

The following example sets the values of all boolean read-write system attributes of a file to the same as the boolean read-write system attributes of another file:

```
$ chmod S=v'ls -/v file1|sed -n '2s/.*/{p}' file2
```

Assuming appropriate privileges and that `file1` and `file2` have the same supported system attributes, all system attributes of `file1` that are set are also set on `file2`. All system attributes of `file1` that are cleared are also cleared on `file2`.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `chmod`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

- 0
Successful completion.
- >0
An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed

另请参见

[getfacl\(1\)](#), [ls\(1\)](#), [setfacl\(1\)](#), [chmod\(2\)](#), [fgetattr\(3C\)](#), [acl\(5\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

《Oracle Solaris 11.1 管理：ZFS 文件系统》

附注

Absolute changes do not work for the set-group-ID bit of a directory. You must use `g+s` or `g-s`.

`chmod` permits you to produce useless modes so long as they are not illegal (for instance, making a text file executable). `chmod` does not check the file type to see if mandatory locking is meaningful.

If the filesystem is mounted with the `nosuid` option, `setuid` execution is not allowed.

If you use `chmod` to change the file group owner permissions on a file with ACL entries, both the file group owner permissions and the ACL mask are changed to the new permissions. Be aware that the new ACL mask permissions can change the effective permissions for additional users and groups who have ACL entries on the file. Use the [getfacl\(1\)](#) or [ls\(1\)](#) command to make sure the appropriate permissions are set for all ACL entries.

引用名 chown – change file ownership

用法概要

```
/usr/bin/chown        /usr/bin/chown [-fhR] owner[:group] file...
                     /usr/bin/chown -s [-fhR] ownersid[:groupsid] file...
                     /usr/bin/chown -R [-f] [-H | -L | -P] owner[:group] file...
                     /usr/bin/chown -s -R [-f] [-H | -L | -P] ownersid[:groupsid] file...

/usr/xpg4/bin/chown /usr/xpg4/bin/chown [-fhR] owner[:group] file...
                     /usr/xpg4/bin/chown -s [-fhR] ownersid[:groupsid] file...
                     /usr/xpg4/bin/chown -R [-f] [-H | -L | -P] owner[:group] file...
                     /usr/xpg4/bin/chown -s -R [-f] [-H | -L | -P] ownersid[:groupsid] file...
```

描述

`/usr/bin/chown` and `/usr/xpg4/bin/chown` The `chown` utility sets the user ID of the file named by each `file` to the user ID specified by `owner`, and, optionally, sets the group ID to that specified by `group`.

If `chown` is invoked by other than the super-user, the set-user-ID bit is cleared.

Only the owner of a file (or the super-user) can change the owner of that file.

The file system has a mountpoint option, `rstchown`, to restrict ownership changes. When this option is in effect the owner of the file is prevented from changing the owner ID of the file. Only the super-user can arbitrarily change owner IDs, whether or not this option is in effect.

`chown` changes the ownership of each file to `owner`. `owner` can be specified as either a user name or a numeric user id. The group ownership of each file can also be changed to `group` by appending `:group` to the user name.

选项

`/usr/bin/chown` and `/usr/xpg4/bin/chown` The following options are supported:

- f Force. Does not report errors.
- h If the file is a symbolic link, this option changes the owner of the symbolic link. Without this option, the owner of the file referenced by the symbolic link is changed.
- H If the file specified on the command line is a symbolic link referencing a file of type directory, this option changes the owner of the directory referenced by the symbolic link and all the files in the file hierarchy below it. If a symbolic link is encountered when traversing a file hierarchy, the owner of the target file is changed, but no recursion takes place.

- L If the file is a symbolic link, this option changes the owner of the file referenced by the symbolic link. If the file specified on the command line, or encountered during the traversal of the file hierarchy, is a symbolic link referencing a file of type directory, then this option changes the owner of the directory referenced by the symbolic link and all files in the file hierarchy below it.
- P If the file specified on the command line or encountered during the traversal of a file hierarchy is a symbolic link, this option changes the owner of the symbolic link. This option does not follow the symbolic link to any other part of the file hierarchy.
- s The owner and/or group arguments are Windows SID strings. This option requires a file system that supports storing SIDs, such as ZFS.

Specifying more than one of the mutually-exclusive options `-H`, `-L`, or `-P` is not considered an error. The last option specified determines the behavior of `chown`.

`/usr/bin/chown`

The following options are supported:

- R Recursive. `chown` descends through the directory, and any subdirectories, setting the specified ownership ID as it proceeds. When a symbolic link is encountered, the owner of the target file is changed, unless the `-h` or `-P` option is specified. However, no recursion takes place, unless the `-H` or `-L` option is specified.

`/usr/xpg4/bin/chown`

The following options are supported:

- R Recursive. `chown` descends through the directory, and any subdirectories, setting the specified ownership ID as it proceeds. When a symbolic link is encountered, the owner of the target file is changed, unless the `-h` or `-P` option is specified. Unless the `-H`, `-L`, or `-P` option is specified, the `-L` option is used as the default mode.

操作数

The following operands are supported:

owner[:group] A user ID and optional group ID to be assigned to *file*. The *owner* portion of this operand must be a user name from the user database or a numeric user ID. Either specifies a user ID to be given to each file named by *file*. If a numeric *owner* exists in the user database as a user name, the user ID number associated with that user name is used as the user ID. Similarly, if the *group* portion of this operand is present, it must be a group name from the group database or a numeric group ID. Either specifies a group ID to be given to each file. If a numeric group operand exists in the group database as a group name, the group ID number associated with that group name is used as the group ID.

file A path name of a file whose user ID is to be modified.

用法

See [largefile\(5\)](#) for the description of the behavior of `chown` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Changing Ownership of All Files in the Hierarchy

The following command changes ownership of all files in the hierarchy, including symbolic links, but not the targets of the links:

```
example% chown -R -h owner[:group] file...
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of chown: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

0 The utility executed successfully and all requested changes were made.

>0 An error occurred.

文件

/etc/passwd System password file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/chown

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled. See NOTES.
Interface Stability	Committed
Standard	See standards(5) .

/usr/xpg4/bin/chown

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled. See NOTES.
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[chgrp\(1\)](#), [chmod\(1\)](#), [chown\(2\)](#), [fpathconf\(2\)](#), [passwd\(4\)](#), [system\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注

chown is CSI-enabled except for the *owner* and *group* names.

引用名 chown – change owner

用法概要 /usr/ucb/chown [-fR] owner[.group] filename...

描述 chown changes the owner of the *filenames* to *owner*. The owner can be either a decimal user ID (UID) or a login name found in the password file. An optional *group* can also be specified. The group can be either a decimal group ID (GID) or a group name found in the GID file.

In the default case, only the super-user of the machine where the file is physically located can change the owner. The system configuration option {_POSIX_CHOWN_RESTRICTED} and the privileges PRIV_FILE_CHOWN and PRIV_FILE_CHOWN_SELF also affect who can change the ownership of a file. See [chown\(2\)](#) and [privileges\(5\)](#).

选项 The following options are supported:

- f Do not report errors.
- R Recursively descend into directories setting the ownership of all files in each directory encountered. When symbolic links are encountered, their ownership is changed, but they are not traversed.

用法 See [largefile\(5\)](#) for the description of the behavior of chown when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

文件 /etc/passwd Password file

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [chgrp\(1\)](#), [chown\(2\)](#), [group\(4\)](#), [passwd\(4\)](#), [attributes\(5\)](#), [largefile\(5\)](#), [privileges\(5\)](#)

引用名 ckdate, errdate, helpdate, valdate – prompts for and validates a date

用法概要

```
ckdate [-Q] [-W width] [-f format] [-d default] [-h help]
      [-e error] [-p prompt] [-k pid [-s signal]]

/usr/sadm/bin/errdate [-W width] [-e error] [-f format]
/usr/sadm/bin/helpdate [-W width] [-h help] [-f format]
/usr/sadm/bin/valdate [-f format] input
```

描述

The ckdate utility prompts a user and validates the response. It defines, among other things, a prompt message whose response should be a date, text for help and error messages, and a default value (which will be returned if the user responds with a RETURN). The user response must match the defined format for a date.

All messages are limited in length to 70 characters and are formatted automatically. Any white space used in the definition (including newline) is stripped. The -W option cancels the automatic formatting. When a tilde is placed at the beginning or end of a message definition, the default text will be inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under NOTES) will be displayed.

Three visual tool modules are linked to the ckdate command. They are errdate (which formats and displays an error message), helpdate (which formats and displays a help message), and valdate (which validates a response). These modules should be used in conjunction with FML objects. In this instance, the FML object defines the prompt. When format is defined in the errdate and helpdate modules, the messages will describe the expected format.

选项

The following options are supported:

- d *default* Defines the default value as *default*. The default does not have to meet the format criteria.
- e *error* Defines the error message as *error*.
- f *format* Specifies the format against which the input will be verified. Possible formats and their definitions are:
 - %b = abbreviated month name (jan, feb, mar)
 - %B = full month name %d = day of month (01 - 31)
 - %D = date as %m/%d/%y (the default format)
 - %e = day of month (1 - 31; single digits are preceded by a blank)
 - %h = abbreviated month name, identical to %b%
 - %m = month number (01 - 12)

- `%y =` year within century (for instance, 89)
- `%Y =` year as CCYY (for instance, 1989)
- `-h help` Defines the help messages as *he lp*.
- `-k pid` Specifies that process ID *pid* is to be sent a signal if the user chooses to abort.
- `-p prompt` Defines the prompt message as *prompt*.
- `-Q` Specifies that quit will not be allowed as a valid response.
- `-s signal` Specifies that the process ID *pid* defined with the `-k` option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- `-W width` Specifies that prompt, help and error messages will be formatted to a line length of *width*.

操作数

The following operand is supported:

input Input to be verified against format criteria.

退出状态

The following exit values are returned:

- `0` Successful execution.
- `1` EOF on input, or negative width on `-W` option, or usage error.
- `3` User termination (quit).
- `4` Garbled format argument.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	system/core-os

另请参见

[attributes\(5\)](#)

附注

The default prompt for `ckdate` is:

```
Enter the date [?,q]:
```

The default error message is:

```
ERROR - Please enter a date. Format is <format>.
```

The default help message is:

```
Please enter a date. Format is <format>.
```

When the quit option is chosen (and allowed), `q` is returned along with the return code 3. The `valdate` module will not produce any output. It returns zero for success and non-zero for failure.

引用名 ckgid, errgid, helpgid, valgid – prompts for and validates a group id

用法概要 ckgid [-Q] [-W *width*] [-m] [-d *default*] [-h *help*]
 [-e *error*] [-p *prompt*] [-k *pid* [-s *signal*]]

 /usr/sadm/bin/errgid [-W *width*] [-e *error*]

 /usr/sadm/bin/helpgid [-W *width*] [-m] [-h *help*]

 /usr/sadm/bin/valgid *input*

描述 ckgid prompts a user and validates the response. It defines, among other things, a prompt message whose response should be an existing group ID, text for help and error messages, and a default value (which will be returned if the user responds with a carriage return).

All messages are limited in length to 70 characters and are formatted automatically. Any white space used in the definition (including newline) is stripped. The -W option cancels the automatic formatting. When a tilde is placed at the beginning or end of a message definition, the default text will be inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under NOTES) will be displayed.

Three visual tool modules are linked to the ckgid command. They are `errgid` (which formats and displays an error message), `helpgid` (which formats and displays a help message), and `valgid` (which validates a response). These modules should be used in conjunction with FML objects. In this instance, the FML object defines the prompt.

选项 The following options are supported:

- d *default* Defines the default value as *default*. The default is not validated and so does not have to meet any criteria.
- e *error* Defines the error message as *error*.
- h *help* Defines the help messages as *help*.
- k *pid* Specifies that process ID *pid* is to be sent a signal if the user chooses to abort.
- m Displays a list of all groups when help is requested or when the user makes an error.
- p *prompt* Defines the prompt message as *prompt*.
- Q Specifies that quit will not be allowed as a valid response.
- s *signal* Specifies that the process ID *pid* defined with the -k option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- W *width* Specifies that prompt, help and error messages will be formatted to a line length of *width*.

操作数 The following operand is supported:
input Input to be verified against /etc/group.

退出状态 The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on -W option, or usage error.
- 3 User termination (quit).

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [attributes\(5\)](#)

附注 The default prompt for `ckgid` is:

Enter the name of an existing group [?,q]:

The default error message is:

ERROR: Please enter one of the following group names: [*List*]

If the `-m` option of `ckgid` is used, a list of valid groups is displayed here.

The default help message is:

ERROR: Please enter one of the following group names: [*List*]

If the `-m` option of `ckgid` is used, a list of valid groups is displayed here.

When the `quit` option is chosen (and allowed), `q` is returned along with the return code 3. The `valgid` module will not produce any output. It returns 0 for success and non-zero for failure.

引用名 ckint, errint, helpint, valint – display a prompt; verify and return an integer value

用法概要

```
ckint [-Q] [-W width] [-b base] [-d default] [-h help]
      [-e error] [-p prompt] [-k pid [-s signal]]
/usr/sadm/bin/errint [-W width] [-b base] [-e error]
/usr/sadm/bin/helpint [-W width] [-b base] [-h help]
/usr/sadm/bin/valint [-b base] input
```

描述

The `ckint` utility prompts a user, then validates the response. It defines, among other things, a prompt message whose response should be an integer, text for help and error messages, and a default value (which will be returned if the user responds with a carriage return).

All messages are limited in length to 70 characters and are formatted automatically. Any white space used in the definition (including newline) is stripped. The `-W` option cancels the automatic formatting. When a tilde is placed at the beginning or end of a message definition, the default text will be inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under NOTES) will be displayed.

Three visual tool modules are linked to the `ckint` command. They are `errint` (which formats and displays an error message), `helpint` (which formats and displays a help message), and `valint` (which validates a response). These modules should be used in conjunction with FML objects. In this instance, the FML object defines the prompt. When *base* is defined in the `errint` and `helpint` modules, the messages will include the expected base of the input.

选项

The following options are supported:

- b *base* Defines the base for input. Must be 2 to 36, default is 10.
- d *default* Defines the default value as *default*. The default is not validated and so does not have to meet any criteria.
- e *error* Defines the error message as *error*.
- h *help* Defines the help messages as *help*.
- k *pid* Specifies that process ID *pid* is to be sent a signal if the user chooses to abort.
- p *prompt* Defines the prompt message as *prompt*.
- Q Specifies that quit will not be allowed as a valid response.
- s *signal* Specifies that the process ID *pid* defined with the `-k` option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- W *width* Specifies that prompt, help and error messages will be formatted to a line length of *width*.

操作数 The following operand is supported:
input Input to be verified against *base* criterion.

退出状态 The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on `-W` option, or usage error.
- 3 User termination (quit).

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [attributes\(5\)](#)

附注 The default base 10 prompt for `ckint` is:

Enter an integer [?,q]:

The default base 10 error message is:

ERROR - Please enter an integer.

The default base 10 help message is:

Please enter an integer.

The messages are changed from "integer" to "base *base* integer" if the base is set to a number other than 10.

When the quit option is chosen (and allowed), `q` is returned along with the return code 3. The `valint` module will not produce any output. It returns 0 for success and non-zero for failure.

引用名 ckitem, erritem, helpitem – build a menu; prompt for and return a menu item

用法概要

```
ckitem [-Q] [-W width] [-uno] [-f filename] [-l label]
      [ [-i invis] [,]... ] [-m max] [-d default] [-h help]
      [-e error] [-p prompt] [-k pid] [-s signal]
      [choice [...]]

/usr/sadm/bin/erritem [-W width] [-e error] [choice [...]]

/usr/sadm/bin/helpitem [-W width] [-h help] [choice [...]]
```

描述

The `ckitem` utility builds a menu and prompts the user to choose one item from a menu of items. It then verifies the response. Options for this command define, among other things, a prompt message whose response will be a menu item, text for help and error messages, and a default value (which will be returned if the user responds with a carriage return).

By default, the menu is formatted so that each item is prepended by a number and is printed in columns across the terminal. Column length is determined by the longest choice. Items are alphabetized.

All messages are limited in length to 70 characters and are formatted automatically. Any white space used in the definition (including newline) is stripped. The `-W` option cancels the automatic formatting. When a tilde is placed at the beginning or end of a message definition, the default text will be inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under NOTES) will be displayed.

Two visual tool modules are linked to the `ckitem` command. They are `erritem` (which formats and displays an error message) and `helpitem` (which formats and displays a help message). These modules should be used in conjunction with FML objects. In this instance, the FML object defines the prompt. When `choice` is defined in these modules, the messages will describe the available menu choice (or choices).

选项

The following options are supported:

- `-d default` Define the default value as *default*. The default is not validated and so does not have to meet any criteria.
- `-e error` Define the error message as *error*.
- `-f filename` Define a file, *filename*, which contains a list of menu items to be displayed. (The format of this file is: `token<tab>description`. Lines beginning with a pound sign (#) are designated as comments and ignored.)
- `-h help` Define the help messages as *help*.
- `-i invis` Define invisible menu choices (those which will not be printed in the menu). (For example, "all" used as an invisible choice would mean it is a legal option)

- but does not appear in the menu. Any number of invisible choices may be defined.) Invisible choices should be made known to a user either in the prompt or in a help message.
- k *pid* Specify that the process ID *pid* is to be sent a signal if the user chooses to abort.
 - l *label* Define a label, *label*, to print above the menu.
 - m *max* Define the maximum number of menu choices that the user can choose. The default is 1.
 - n Specify that menu items should not be displayed in alphabetical order.
 - o Specify that only one menu token will be returned.
 - p *prompt* Define the prompt message as *prompt*.
 - Q Specify that quit will not be allowed as a valid response.
 - s *signal* Specify that process ID *pid* defined with the -k option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
 - u Specify that menu items should be displayed as an unnumbered list.
 - W *width* Specify that prompt, help and error messages will be formatted to a line length of *width*.

操作数

The following operand is supported:

choice Define menu items. Items should be separated by white space or newline.

退出状态

The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on -W option, or inability to open file on -f option, or usage error.
- 3 User termination (quit).
- 4 No choices from which to choose.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[attributes\(5\)](#)

附注

The user may input the number of the menu item if choices are numbered or as much of the string required for a unique identification of the item. Long menus are paged with 10 items per page.

When menu entries are defined both in a file (by using the `-f` option) and also on the command line, they are usually combined alphabetically. However, if the `-n` option is used to suppress alphabetical ordering, then the entries defined in the file are shown first, followed by the options defined on the command line.

The default prompt for `ckitem` is:

```
Enter selection [?,??,q]:
```

One question mark will give a help message and then redisplay the prompt. Two question marks will give a help message and then redisplay the menu label, the menu and the prompt.

The default error message if you typed a number is:

```
ERROR: Bad numeric choice specification
```

The default error message if you typed a string is:

```
ERROR: Entry does not match available menu selection. Enter the number of the menu item you wish to select, the token which is associated with the menu item, or a partial string which uniquely identifies the token for the menu item. Enter ?? to reprint the menu.
```

The default help message is:

```
Enter the number of the menu item you wish to select, the token which is associated with the menu item, or a partial string which uniquely identifies the token for the menu item. Enter ? to reprint the menu.
```

When the quit option is chosen (and allowed), `q` is returned along with the return code 3.

引用名	ckkeywd – prompts for and validates a keyword
用法概要	ckkeywd [-Q] [-W <i>width</i>] [-d <i>default</i>] [-h <i>help</i>] [-e <i>error</i>] [-p <i>prompt</i>] [-k <i>pid</i> [-s <i>signal</i>]] <i>keyword</i> [...]
描述	<p>ckkeywd prompts a user and validates the response. It defines, among other things, a prompt message whose response should be one of a list of keywords, text for help and error messages, and a default value (which will be returned if the user responds with a carriage return). The answer returned from this command must match one of the defined list of keywords.</p> <p>All messages are limited in length to 70 characters and are formatted automatically. Any white space used in the definition (including newline) is stripped. The -W option cancels the automatic formatting. When a tilde is placed at the beginning or end of a message definition, the default text will be inserted at that point, allowing both custom text and the default text to be displayed.</p> <p>If the prompt, help or error message is not defined, the default message (as defined under NOTES) will be displayed.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -d <i>default</i> Defines the default value as <i>default</i>. The default is not validated and so does not have to meet any criteria. -e <i>error</i> Defines the error message as <i>error</i>. -h <i>help</i> Defines the help messages as <i>help</i>. -k <i>pid</i> Specifies that process ID <i>pid</i> is to be sent a signal if the user chooses to abort. -p <i>prompt</i> Defines the prompt message as <i>prompt</i>. -Q Specifies that quit will not be allowed as a valid response. -s <i>signal</i> Specifies that the process ID <i>pid</i> defined with the -k option is to be sent signal <i>signal</i> when quit is chosen. If no signal is specified, SIGTERM is used. -W <i>width</i> Specifies that prompt, help and error messages will be formatted to a line length of <i>width</i>.
操作数	<p>The following operand is supported:</p> <p><i>keyword</i> Defines the keyword, or list of keywords, against which the answer will be verified.</p>
退出状态	<p>The following exit values are returned:</p> <ul style="list-style-type: none"> 0 Successful execution. 1 EOF on input, or negative width on -W option, or no keywords from which to choose, or usage error.

3 User termination (quit).

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[attributes\(5\)](#)

附注

The default prompt for ckkeywd is:

Enter appropriate value [*keyword*, [. . .], ?, q]:

The default error message is:

ERROR: Please enter one of the following keywords: *keyword*, [. . .], q

The default help message is:

keyword, [. . .], q

When the quit option is chosen (and allowed), q is returned along with the return code 3.

引用名 ckpath, errpath, helppath, valpath – display a prompt; verify and return a pathname

用法概要

```
ckpath [-Q] [-W width] [-a | l] [-b | c | f | y]
      [-n [o | z]] [-rtwx] [-d default] [-h help]
      [-e error] [-p prompt] [-k pid [-s signal]]

/usr/sadm/bin/errpath [-W width] [-a | l] [-b | c | f | y]
      [-n [o | z]] [-rtwx] [-e error]

/usr/sadm/bin/helppath [-W width] [-a | l] [-b | c | f | y]
      [-n [o | z]] [-rtwx] [-h help]

/usr/sadm/bin/valpath [-a | l] [-b | c | f | y]
      [-n [o | z]] [-rtwx] input
```

描述

The ckpath utility prompts a user and validates the response. It defines, among other things, a prompt message whose response should be a pathname, text for help and error messages, and a default value (which is returned if the user responds with a RETURN).

The pathname must obey the criteria specified by the first group of options. If no criteria is defined, the pathname must be for a normal file that does not yet exist. If neither -a (absolute) or -l (relative) is given, then either is assumed to be valid.

All messages are limited in length to 79 characters and are formatted automatically. Tabs and newlines are removed after a single white space character in a message definition, but spaces are not removed. When a tilde is placed at the beginning or end of a message definition, the default text is inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under EXAMPLES) is displayed.

Three visual tool modules are linked to the ckpath command. They are `errpath` (which formats and displays an error message on the standard output), `helppath` (which formats and displays a help message on the standard output), and `valpath` (which validates a response).

选项

The following options are supported:

- a Pathname must be an absolute path.
- b Pathname must be a block special file.
- c Pathname must be a character special file.
- d *default* Defines the default value as *default*. The default is not validated and so does not have to meet any criteria.
- e *error* Defines the error message as *error*.
- f Pathname must be a regular file.
- h *help* Defines the help message as *help*.

- k *pid* Specifies that process ID *pid* is to be sent a signal if the user chooses to quit.
- l Pathname must be a relative path.
- n Pathname must not exist (must be new).
- o Pathname must exist (must be old).
- p *prompt* Defines the prompt message as *prompt*.
- Q Specifies that quit is not allowed as a valid response.
- r Pathname must be readable.
- s *signal* Specifies that the process ID *pid* defined with the -k option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- t Pathname must be creatable (touchable). Pathname will be created if it does not already exist.
- w Pathname must be writable.
- W *width* Specify that prompt, help and error messages be formatted to a line length of *width*.
- x Pathname must be executable.
- y Pathname must be a directory.
- z Pathname must have a file having a size greater than zero bytes.

操作数

The following operand is supported:

input Input to be verified against validation options.

示例

The text of the default messages for ckpath depends upon the criteria options that have been used.

示例1 Default prompt

An example default prompt for ckpath (using the -a option) is:

```
example% ckpath -a
Enter an absolute pathname [?,q]
```

示例2 Default error message

An example default error message (using the -a option) is:

```
example% /usr/sadm/bin/errpath -a
ERROR: A pathname is a filename, optionally preceded by parent
directories.
The pathname you enter: - must begin with a slash (/)
```

示例 3 Default help message

An example default help message (using the `-a` option) is:

```
example% /usr/sadm/bin/helppath -a
A pathname is a filename, optionally preceded by parent directories.
The pathname you enter: - must begin with a slash (/)
```

示例 4 The quit option

When the quit option is chosen (and allowed), `q` is returned along with the return code 3. Quit input gets a trailing newline.

示例 5 Using the `valpath` module

The `valpath` module will produce a usage message on `stderr`. It returns 0 for success and non-zero for failure.

```
example% /usr/sadm/bin/valpath
usage: valpath [-[a|l][b|c|f|y][n|[o|z]]rtwx] input
.
.
.
```

退出状态

The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on `-W` option, or usage error.
- 2 Mutually exclusive options.
- 3 User termination (quit).
- 4 Mutually exclusive options.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[signal.h\(3HEAD\)](#), [attributes\(5\)](#)

引用名 ckrange, errrange, helprange, valrange – prompts for and validates an integer

用法概要

```
ckrange [-Q] [-W width] [-l lower] [-u upper] [-b base]
        [-d default] [-h help] [-e error] [-p prompt]
        [-k pid [-s signal]]

/usr/sadm/bin/errrange [-W width] [-e error] [-l lower]
        [-u upper] [-b base]

/usr/sadm/bin/helprange [-W width] [-h help] [-l lower]
        [-u upper] [-b base]

/usr/sadm/bin/valrange [-l lower] [-u upper] [-b base] input
```

描述

The `ckrange` utility prompts a user for an integer between a specified range and determines whether this response is valid. It defines, among other things, a prompt message whose response should be an integer in the range specified, text for help and error messages, and a default value (which is returned if the user responds with a RETURN).

This command also defines a range for valid input. If either the lower or upper limit is left undefined, then the range is bounded on only one end.

All messages are limited in length to 79 characters and are formatted automatically. Tabs and newlines are removed after a single whitespace character in a message definition, but spaces are not removed. When a tilde is placed at the beginning or end of a message definition, the default text will be inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under EXAMPLES) is displayed.

Three visual tool modules are linked to the `ckrange` command. They are `errrange` (which formats and displays an error message on the standard output), `helprange` (which formats and displays a help message on the standard output), and `valrange` (which validates a response).

Note: Negative "input" arguments confuse `getopt` in `valrange`. By inserting a "-" before the argument, `getopt` processing will stop. See [getopt\(1\)](#) and [Intro\(1\)](#) about `getopt` parameter handling. `getopt` is used to parse positional parameters and to check for legal options.

选项

The following options are supported:

- b *base* Defines the base for input. Must be 2 to 36, default is 10. Base conversion uses [strtol\(3C\)](#). Output is always base 10.
- d *default* Defines the default value as *default*. *default* is converted using [strtol\(3C\)](#) in the desired base. Any characters invalid in the specified base will terminate the `strtol` conversion without error.
- e *error* Defines the error message as *error*.

- h *help* Defines the help message as *help*.
- k *pid* Specifies that process ID *pid* is to be sent a signal if the user chooses to quit.
- l *lower* Defines the lower limit of the range as *lower*. Default is the machine's largest negative long.
- p *prompt* Defines the prompt message as *prompt*.
- Q Specifies that quit will not be allowed as a valid response.
- s *signal* Specifies that the process ID *pid* defined with the -k option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- u *upper* Defines the upper limit of the range as *upper*. Default is the machine's largest positive long.
- W *width* Specifies that prompt, help and error messages will be formatted to a line length of *width*.

操作数

The following operand is supported:

input Input to be verified against upper and lower limits and base.

示例

示例 1 Default base 10 prompt

The default base 10 prompt for ckrange is:

```
example% ckrange
Enter an integer between lower_bound and
upper_bound [lower_bound-upper_bound,?,q]:
```

示例 2 Default base 10 error message

The default base 10 error message is:

```
example% /usr/sadm/bin/errrange
ERROR: Please enter an integer between lower_bound \
and upper_bound.
```

示例 3 Default base 10 help message

The default base 10 help message is:

```
example% /usr/sadm/bin/helpprange
Please enter an integer between lower_bound and upper_bound.
```

示例 4 Changing messages for a base other than 10

The messages are changed from "integer" to "base *base* integer" if the base is set to a number other than 10. For example,

```
example% /usr/sadm/bin/helpprange -b 36
```

示例 5 Using the quit option

When the quit option is chosen (and allowed), q is returned along with the return code 3. Quit input gets a trailing newline.

示例 6 Using the valrange module

The valrange module will produce a usage message on stderr. It returns 0 for success and non-zero for failure.

```
example% /usr/sadm/bin/valrange
usage: valrange [-l lower] [-u upper] [-b base] input
```

退出状态

The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on -W option, or usage error.
- 2 Usage error.
- 3 User termination (quit).

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[Intro\(1\)](#), [getopt\(1\)](#), [strtol\(3C\)](#), [attributes\(5\)](#), [signal.h\(3HEAD\)](#)

引用名 ckstr, errstr, helpstr, valstr – display a prompt; verify and return a string answer

用法概要

```
ckstr [-Q] [-W width] [ [-r regexp] [...] ] [-l length]
      [-d default] [-h help] [-e error] [-p prompt]
      [-k pid [-s signal]]

/usr/sadm/bin/errstr [-W width] [-e error] [-l length]
      [ [-r regexp] [...] ]

/usr/sadm/bin/helpstr [-W width] [-h help] [-l length]
      [ [-r regexp] [...] ]

/usr/sadm/bin/valstr [-l length] [ [-r regexp] [...] ] input
```

描述

The `ckstr` utility prompts a user and validates the response. It defines, among other things, a prompt message whose response should be a string, text for help and error messages, and a default value (which are returned if the user responds with a RETURN).

The answer returned from this command must match the defined regular expression and be no longer than the length specified. If no regular expression is given, valid input must be a string with a length less than or equal to the length defined with no internal, leading or trailing white space. If no length is defined, the length is not checked.

All messages are limited in length to 79 characters and are formatted automatically. Tabs and newlines are removed after a single white space character in a message definition, but spaces are not removed. When a tilde is placed at the beginning or end of a message definition, the default text will be inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under EXAMPLES) is displayed.

Three visual tool modules are linked to the `ckstr` command. They are `errstr` (which formats and displays an error message on the standard output), `helpstr` (which formats and displays a help message on the standard output), and `valstr` (which validates a response).

选项

The following options are supported:

- d *default* Defines the default value as *default*. The default is not validated and so does not have to meet any criteria.
- e *error* Defines the error message as *error*.
- h *help* Defines the help message as *help*.
- k *pid* Specifies that process ID *pid* is to be sent a signal if the user chooses to quit.
- l *length* Specifies the maximum length of the input.
- p *prompt* Defines the prompt message as *prompt*.
- Q Specifies that quit will not be allowed as a valid response.

- *r regexp* Specifies a regular expression, *regexp*, against which the input should be validated. May include white space. If multiple expressions are defined, the answer need match only one of them.
- *s signal* Specifies that the process ID *pid* defined with the -k option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- *W width* Specifies that prompt, help and error messages will be formatted to a line length of *width*.

操作数

The following operand is supported:

input Input to be verified against format length and/or regular expression criteria.

示例

示例 1 Default prompt

The default prompt for `ckstr` is:

```
example% ckstr  
Enter an appropriate value [?,q]:
```

示例 2 Default error message

The default error message is dependent upon the type of validation involved. The user will be told either that the length or the pattern matching failed. The default error message is:

```
example% /usr/sadm/bin/errstr  
ERROR: Please enter a string which contains no embedded,  
leading or trailing spaces or tabs.
```

示例 3 Default help message

The default help message is also dependent upon the type of validation involved. If a regular expression has been defined, the message is:

```
example% /usr/sadm/bin/helpstr -r regexp  
Please enter a string which matches the following pattern:  
regexp
```

Other messages define the length requirement and the definition of a string.

示例 4 Using the quit option

When the quit option is chosen (and allowed), `q` is returned along with the return code 3. Quit input gets a trailing newline.

示例 5 Using the valstr module

The `valstr` module will produce a usage message on `stderr`. It returns 0 for success and non-zero for failure.

示例 5 Using the valstr module (续)

```
example% /usr/sadm/bin/valstr
usage: valstr [-l length] [[-r regexp] [ . . . ]] input
```

退出状态

The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on -W option, or usage error.
- 2 Invalid regular expression.
- 3 User termination (quit).

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[signal.h\(3HEAD\)](#), [attributes\(5\)](#)

引用名	cksum – write file checksums and sizes
用法概要	<code>/usr/bin/cksum [file...]</code>
描述	<p>The <code>cksum</code> command calculates and writes to standard output a cyclic redundancy check (CRC) for each input file, and also writes to standard output the number of octets in each file.</p> <p>For each file processed successfully the <code>cksum</code> method writes in the following format:</p> <pre>"%u %d %s\n" <checksum>, <# of octets>, <path name></pre> <p>If no file operand was specified, the path name and its leading space is omitted.</p> <p>The CRC used is based on the polynomial used for CRC error checking in the referenced Ethernet standard.</p> <p>The encoding for the CRC checksum is defined by the generating polynomial:</p> $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ <p>Mathematically, the CRC value corresponding to a given file is defined by the following procedure:</p> <ol style="list-style-type: none">1. The n bits to be evaluated are considered to be the coefficients of a mod 2 polynomial $M(x)$ of degree $n-1$. These n bits are the bits from the file, with the most significant bit being the most significant bit of the first octet of the file and the last bit being the least significant bit of the last octet, padded with zero bits (if necessary) to achieve an integral number of octets, followed by one or more octets representing the length of the file as a binary value, least significant octet first. The smallest number of octets capable of representing this integer is used.2. $M(x)$ is multiplied by x^{32} (that is, shifted left 32 bits) and divided by $G(x)$ using mod 2 division, producing a remainder $R(x)$ of degree ≤ 31.3. The coefficients of $R(x)$ are considered to be a 32-bit sequence.4. The bit sequence is complemented and the result is the CRC.
操作数	<p>The following operand is supported:</p> <p><i>file</i> A path name of a file to be checked. If no <i>file</i> operands are specified, the standard input is used.</p>
用法	<p>The <code>cksum</code> command is typically used to quickly compare a suspect file against a trusted version of the same, such as to ensure that files transmitted over noisy media arrive intact. However, this comparison cannot be considered cryptographically secure. The chances of a damaged file producing the same CRC as the original are astronomically small; deliberate deception is difficult, but probably not impossible.</p>

Although input files to `cksum` can be any type, the results need not be what would be expected on character special device files. Since this document does not specify the block size used when doing input, checksums of character special files need not process all of the data in those files.

The algorithm is expressed in terms of a bitstream divided into octets. If a file is transmitted between two systems and undergoes any data transformation (such as moving 8-bit characters into 9-bit bytes or changing Little Endian byte ordering to Big Endian), identical CRC values cannot be expected. Implementations performing such transformations can extend `cksum` to handle such situations.

See [largefile\(5\)](#) for the description of the behavior of `cksum` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `cksum`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

- 0 All files were processed successfully.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[digest\(1\)](#), [sum\(1\)](#), [bart\(1M\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

引用名 cktime, errtime, helptime, valtime – display a prompt; verify and return a time of day

用法概要

```
cktime [-Q] [-W width] [-f format] [-d default] [-h help]
      [-e error] [-p prompt] [-k pid [-s signal]]

/usr/sadm/bin/errtime [-W width] [-e error] [-f format]
/usr/sadm/bin/helptime [-W width] [-h help] [-f format]
/usr/sadm/bin/valtime [-f format] input
```

描述

The `cktime` utility prompts a user and validates the response. It defines, among other things, a prompt message whose response should be a time, text for help and error messages, and a default value (which is returned if the user responds with a RETURN). The user response must match the defined format for the time of day.

All messages are limited in length to 70 characters and are formatted automatically. Any white space used in the definition (including NEWLINE) is stripped. The `-W` option cancels the automatic formatting. When a tilde is placed at the beginning or end of a message definition, the default text is inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under NOTES) is displayed.

Three visual tool modules are linked to the `cktime` command. They are `errtime` (which formats and displays an error message), `helptime` (which formats and displays a help message), and `valtime` (which validates a response). These modules should be used in conjunction with FML objects. In this instance, the FML object defines the prompt. When format is defined in the `errtime` and `helptime` modules, the messages will describe the expected format.

选项

The following options are supported:

- `-d default` Defines the default value as *default*. The default is not validated and so does not have to meet any criteria.
- `-e error` Defines the error message as *error*.
- `-f format` Specifies the format against which the input will be verified. Possible formats and their definitions are:
 - `%H` = hour (00 - 23)
 - `%I` = hour (00 - 12)
 - `%M` = minute (00 - 59)
 - `%p` = ante meridian or post meridian
 - `%r` = time as %I:%M:%S %p
 - `%R` = time as %H:%M (the default format)
 - `%S` = seconds (00 - 59)
 - `%T` = time as %H:%M:%S
- `-h help` Defines the help messages as *help*.

- k *pid* Specifies that process ID *pid* is to be sent a signal if the user chooses to abort.
- p *prompt* Defines the prompt message as *prompt*.
- Q Specifies that quit will not be allowed as a valid response.
- s *signal* Specifies that the process ID *pid* defined with the -k option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- W *width* Specifies that prompt, help and error messages will be formatted to a line length of *width*.

操作数

The following operand is supported:

input Input to be verified against format criteria.

退出状态

The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on -W option, or usage error .
- 3 User termination (quit) .
- 4 Garbled format argument.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	system/core-os

另请参见

[attributes\(5\)](#)

附注

The default prompt for `cktime` is:

Enter a time of day [?,q]:

The default error message is:

ERROR: Please enter the time of day. Format is <format>.

The default help message is:

Please enter the time of day. Format is <format>.

When the quit option is chosen (and allowed), q is returned along with the return code 3. The `valtime` module will not produce any output. It returns 0 for success and non-zero for failure.

引用名 ckuid, erruid, helpuid, valuid – prompts for and validates a user ID

用法概要 ckuid [-Q] [-W *width*] [-m] [-d *default*] [-h *help*]
[-e *error*] [-p *prompt*] [-k *pid* [-s *signal*]]

/usr/sadm/bin/erruid [-W *width*] [-e *error*]

/usr/sadm/bin/helpuid [-W *width*] [-m] [-h *help*]

/usr/sadm/bin/valuid *input*

描述 The ckuid utility prompts a user and validates the response. It defines, among other things, a prompt message whose response should be an existing user ID, text for help and error messages, and a default value (which are returned if the user responds with a RETURN).

All messages are limited in length to 70 characters and are formatted automatically. Any white space used in the definition (including NEWLINE) is stripped. The -W option cancels the automatic formatting. When a tilde is placed at the beginning or end of a message definition, the default text is inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under NOTES) is displayed.

Three visual tool modules are linked to the ckuid command. They are *erruid* (which formats and displays an error message), *helpuid* (which formats and displays a help message), and *valuid* (which validates a response). These modules should be used in conjunction with FML objects. In this instance, the FML object defines the prompt.

选项 The following options are supported:

- d *default* Defines the default value as *default*. The default is not validated and so does not have to meet any criteria.
- e *error* Defines the error message as *error*.
- h *help* Defines the help messages as *help*.
- k *pid* Specifies that process ID *pid* is to be sent a signal if the user chooses to abort.
- m Displays a list of all logins when help is requested or when the user makes an error.
- p *prompt* Defines the prompt message as *prompt*.
- Q Specifies that quit will not be allowed as a valid response.
- s *signal* Specifies that the process ID *pid* defined with the -k option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- W *width* Specifies that prompt, help and error messages will be formatted to a line length of *width*.

操作数 The following operand is supported:
input Input to be verified against /etc/passwd.

退出状态 The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on -W option, or usage error.
- 2 Usage error.
- 3 User termination (quit).

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [attributes\(5\)](#)

附注 The default prompt for ckuid is:
 Enter the login name of an existing user [?,q]:

The default error message is:
 ERROR - Please enter the login name of an existing user.

If the -m option is used, the default error message is:
 ERROR: Please enter one of the following login names: <List>

The default help message is:
 Please enter the login name of an existing user.

If the -m option is used, the default help message is:
 Please enter one of the following login names: <List>

When the quit option is chosen (and allowed), q is returned along with the return code 3. The valuid module will not produce any output. It returns 0 for success and non-zero for failure.

引用名 ckyornd, erryornd, helpyornd, valyornd – prompts for and validates yes/no

用法概要 ckyornd [-Q] [-W *width*] [-d *default*] [-h *help*] [-e *error*]
 [-p *prompt*] [-k *pid*] [-s *signal*]
 /usr/sadm/bin/erryornd [-W *width*] [-e *error*]
 /usr/sadm/bin/helpyornd [-W *width*] [-h *help*]
 /usr/sadm/bin/valyornd *input*

描述 ckyornd prompts a user and validates the response. It defines, among other things, a prompt message for a yes or no answer, text for help and error messages, and a default value (which is returned if the user responds with a RETURN).

All messages are limited in length to 70 characters and are formatted automatically. Any white space used in the definition (including newline) is stripped. The -W option cancels the automatic formatting. When a tilde is placed at the beginning or end of a message definition, the default text is inserted at that point, allowing both custom text and the default text to be displayed.

If the prompt, help or error message is not defined, the default message (as defined under NOTES) is displayed.

Three visual tool modules are linked to the ckyornd command. They are erryornd (which formats and displays an error message), helpyornd (which formats and displays a help message), and valyornd (which validates a response).

选项 The following options are supported:

- d *default* Defines the default value as *default*. The default is not validated and so does not have to meet any criteria.
- e *error* Defines the error message as *error*.
- h *help* Defines the help messages as *help*.
- k *pid* Specifies that process ID *pid* is to be sent a signal if the user chooses to abort.
- p *prompt* Defines the prompt message as *prompt*.
- Q Specifies that quit will not be allowed as a valid response.
- s *signal* Specifies that the process ID *pid* defined with the -k option is to be sent signal *signal* when quit is chosen. If no signal is specified, SIGTERM is used.
- W *width* Specifies that prompt, help and error messages will be formatted to a line length of *width*.

操作数 The following operand is supported:

- input* Input to be verified as y, yes, or n, no (in any combination of upper- and lower-case letters).

退出状态 The following exit values are returned:

- 0 Successful execution.
- 1 EOF on input, or negative width on -W option, or usage error.
- 2 Usage error.
- 3 User termination (quit).

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [attributes\(5\)](#)

附注 The default prompt for ckyornd is:

Yes or No [y,n,?,q]:

The default error message is:

ERROR - Please enter yes or no.

The default help message is:

To respond in the affirmative, enter y, yes, Y, or YES.

To respond in the negative, enter n, no, N, or NO.

When the quit option is chosen (and allowed), q is returned along with the return code 3. The vakyornd module will not produce any output. It returns 0 for success and non-zero for failure.

引用名 clear – clear the terminal screen

用法概要 clear [*term*]

描述 The `clear` utility clears the terminal screen if this is possible. It looks in the environment for the terminal type, if this is not already specified by the *term* operand, and then looks up the terminfo database to figure out how to clear the screen.

操作数 *term* Indicates the type of terminal. Normally, this operand is unnecessary because the default is taken from the environment variable TERM.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [tput\(1\)](#), [attributes\(5\)](#)

引用名	cmp – compare two files
用法概要	<code>/usr/bin/cmp [-l -s] file1 file2 [skip1] [skip2]</code>
描述	<p>cmp compares two files <i>file1</i> and <i>file2</i>. cmp writes no output if the files are the same. By default, if the files differ, the byte and line number at which the first difference occurred are written to standard output. Bytes and lines are numbered beginning with 1.</p> <p><i>skip1</i> and <i>skip2</i> are initial byte offsets into <i>file1</i> and <i>file2</i> respectively, and can be either octal or decimal. A leading 0 denotes octal.</p> <p>If either <i>file1</i> or <i>file2</i> is -, cmp uses standard input for that operand.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -l Write the decimal byte number and the differing bytes (in octal) for each difference. -s Write nothing for differing files. Return non-zero exit status only.
操作数	<p>The following operands are supported:</p> <p><i>file1</i> A path name of the first file to be compared. If <i>file1</i> is -, the standard input is used.</p> <p><i>file2</i> A path name of the second file to be compared. If <i>file2</i> is -, the standard input is used.</p> <p>If both <i>file1</i> and <i>file2</i> refer to standard input or refer to the same FIFO special, block special or character special file, an error results.</p>
用法	See largefile(5) for the description of the behavior of cmp when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).
示例	<p>示例 1 Comparing Files Byte for Byte</p> <p>The following example does a byte for byte comparison of <i>file1</i> and <i>file2</i>:</p> <pre>example% cmp file1 file2 0 1024</pre> <p>It skips the first 1024 bytes in <i>file2</i> before starting the comparison.</p>
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of cmp: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.
退出状态	<p>The following error values are returned:</p> <ul style="list-style-type: none"> 0 The files are identical. 1 The files are different. This includes the case where one file is identical to the first part of the other. >1 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[comm\(1\)](#), [diff\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

引用名 col – reverse line-feeds filter

用法概要 col [-bfp x]

描述

The col utility reads from the standard input and writes to the standard output. It performs the line overlays implied by reverse line-feeds, and by forward and reverse half-line-feeds. Unless -x is used, all blank characters in the input will be converted to tab characters wherever possible. col is particularly useful for filtering multi-column output made with the .rt command of `nroff(1)` and output resulting from use of the `tbl(1)` preprocessor.

The ASCII control characters SO and SI are assumed by col to start and end text in an alternative character set. The character set to which each input character belongs is remembered, and on output SI and SO characters are generated as appropriate to ensure that each character is written in the correct character set.

On input, the only control characters accepted are space, backspace, tab, carriage-return and newline characters, SI, SO, VT, reverse line-feed, forward half-line-feed and reverse half-line-feed. The VT character is an alternative form of full reverse line-feed, included for compatibility with some earlier programs of this type. The only other characters to be copied to the output are those that are printable.

The ASCII codes for the control functions and line-motion sequences mentioned above are as given in the table below. ESC stands for the ASCII escape character, with the octal code 033; ESC- means a sequence of two characters, ESC followed by the character x.

reverse line-feed	ESC-7
reverse half-line-feed	ESC-8
forward half-line-feed	ESC-9
vertical-tab (VT)	013
start-of-text (SO)	016
end-of-text (SI)	017

选项

- b Assume that the output device in use is not capable of backspacing. In this case, if two or more characters are to appear in the same place, only the last one read will be output.
- f Although col accepts half-line motions in its input, it normally does not emit them on output. Instead, text that would appear between lines is moved to the next lower full-line boundary. This treatment can be suppressed by the -f (fine) option; in this case, the output from col may contain forward half-line-feeds (ESC-9), but will still never contain either kind of reverse line motion.
- p Normally, col will ignore any escape sequences unknown to it that are found in its input; the -p option may be used to cause col to output these sequences as regular

characters, subject to overprinting from reverse line motions. The use of this option is highly discouraged unless the user is fully aware of the textual position of the escape sequences.

- x Prevent col from converting blank characters to tab characters on output wherever possible. Tab stops are considered to be at each column position n such that n modulo 8 equals 1.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of col: LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following error values are returned:

- 0 Successful completion.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	system/core-os
CSI	enabled

另请参见

[nroff\(1\)](#), [tbl\(1\)](#), [ascii\(5\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注

The input format accepted by col matches the output produced by nroff with either the -T37 or -Tlp options. Use -T37 (and the -f option of col) if the ultimate disposition of the output of col will be a device that can interpret half-line motions, and -Tlp otherwise.

col cannot back up more than 128 lines or handle more than 800 characters per line.

Local vertical motions that would result in backing up over the first line of the document are ignored. As a result, the first line must not have any superscripts.

引用名	comm – select or reject lines common to two files
用法概要	<code>/usr/bin/comm [-options] file1 file2</code>
描述	<p>comm reads two files <i>file1</i> and <i>file2</i> which should be ordered in the collating sequence of the current locale, and produces three text columns as output:</p> <ol style="list-style-type: none"> 1 Lines only in <i>file1</i>. 2 Lines only in <i>file2</i>. 3 Lines in both files. <p>If lines in either file are not ordered according to the collating sequence of the current locale, the results are not specified.</p> <p>If either <i>file1</i> or <i>file2</i> is -, comm uses standard input starting at the current location.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -1 Suppresses the output column of lines unique to <i>file1</i>. -2 Suppresses the output column of lines unique to <i>file2</i>. -3 Suppresses the output column of lines duplicated in <i>file1</i> and <i>file2</i>.
操作数	<p>The following operands are supported:</p> <p><i>file1</i> A path name of the first file to be compared. If <i>file1</i> is -, the standard input is used.</p> <p><i>file2</i> A path name of the second file to be compared. If <i>file2</i> is -, the standard input is used.</p>
用法	See largefile(5) for the description of the behavior of comm when encountering files greater than or equal to 2 Gbyte (2 ³¹ bytes).
示例	<p>示例 1 Printing a list of utilities specified by files</p> <p>If <i>file1</i>, <i>file2</i>, and <i>file3</i> each contain a sorted list of utilities, the command</p> <pre>example% comm -23 file1 file2 comm -23 - file3</pre> <p>prints a list of utilities in <i>file1</i> not specified by either of the other files. The entry:</p> <pre>example% comm -12 file1 file2 comm -12 - file3</pre> <p>prints a list of utilities specified by all three files. And the entry:</p> <pre>example% comm -12 file2 file3 comm -23 -file1</pre> <p>prints a list of utilities specified by both <i>file2</i> and <i>file3</i>, but not specified in <i>file1</i>.</p>

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of comm: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 All input files were successfully output as specified.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[cmp\(1\)](#), [diff\(1\)](#), [sort\(1\)](#), [uniq\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

引用名 `command` – execute a simple command

用法概要

`/usr/bin/command` `command [-p] command_name [argument] ...`

`command [-v | -V] command_name`

`ksh` `command [-pvxV] [command_name [argument...]]`

描述

The `command` utility causes the shell to treat the arguments as a simple command, suppressing the shell function lookup.

If the *command_name* is the same as the name of one of the special built-in utilities, the special properties do not occur. In every other respect, if *command_name* is not the name of a function, the effect of `command` (with no options) are the same as omitting `command`.

The `command` utility also provides information concerning how a command name is interpreted by the shell. See `-v` and `-V`.

`ksh` Without the `-v` or `-V` option, `command` executes *command_name* with arguments specified by *argument*, suppressing the shell function lookup that normally occurs. In addition, if *command* is a special built-in command, the special properties are removed so that failures do not cause the script that executes it to terminate.

If the `-v` or `-V` options are specified, `command` is equivalent to [whence\(1\)](#).

选项

The following options are supported by `/usr/bin/command`:

- `-p` Performs the command search using a default value for `PATH` that is guaranteed to find all of the standard utilities.
- `-v` Writes a string to standard output that indicates the path or command that is be used by the shell, in the current shell execution environment to invoke *command_name*, but does not invoke *command_name*.
 - Utilities, regular built-in utilities, *command_names* including a slash character, and any implementation-provided functions that are found using the `PATH` variable is written as absolute path names.
 - Shell functions, special built-in utilities, regular built-in utilities not associated with a `PATH` search, and shell reserved words are written as just their names.
 - An alias is written as a command line that represents its alias definition.
 - Otherwise, no output is written and the exit status reflects that the name was not found.
- `-V` Writes a string to standard output that indicates how the name specified in the *command_name* operand is interpreted by the shell, in the current shell execution environment, but does not invoke *command_name*. Although the format of this string is unspecified, it indicates in which of the following categories *command_name* falls and include the information stated:

- Utilities, regular built-in utilities, and any implementation-provided functions that are found using the `PATH` variable is identified as such and include the absolute path name in the string.
- Other shell functions is identified as functions.
- Aliases are identified as aliases and their definitions are included in the string.
- Special built-in utilities are identified as special built-in utilities.
- Regular built-in utilities not associated with a `PATH` search is identified as regular built-in utilities.
- Shell reserved words are identified as reserved words.

ksh

The following options are supported by ksh command:

- p Causes a default path to be searched rather than the one defined by the value of `PATH`.
- v Equivalent to:
`whence command [argument ...]`
- V Equivalent to:
`whence -v command [argument ...]`
- x If command fails because there are too many arguments, it is invoked multiple times with a subset of the arguments on each invocation. Arguments that occur prior to the first word that expand to multiple arguments and arguments that occur after the last word that expands to multiple arguments are passed on each invocation. The exit status is the maximum invocation exit status.

操作数

The following operands are supported:

argument One of the strings treated as an argument to *command_name*.
command_name The name of a utility or a special built-in utility.

示例

示例 1 Making a Version of `cd` That Always Prints Out the New Working Directory

The following example takes a version of `cd` that always prints out the new working directory exactly once:

```
cd() {  
    command cd "$@" >/dev/null  
    pwd  
}
```

示例 2 Starting Off a secure shell script in Which the Script Avoids Being Spoofed by Its Parent

The following example starts off a secure shell script in which the script avoids being spoofed by its parent:

示例 2 Starting Off a secure shell script in Which the Script Avoids Being Spoofed by Its Parent
(续)

```
IFS='
# The preceding value should be <space><tab><newline>.
# Set IFS to its default value.
\unalias -a
# Unset all possible aliases.
# Note that unalias is escaped to prevent an alias
# being used for unalias.
unset -f command
# Ensure command is not a user function.
PATH="$(command -p getconf _CS_PATH):$PATH"
# Put on a reliable PATH prefix.
# ...
```

At this point, given correct permissions on the directories called by PATH, the script has the ability to ensure that any utility it calls is the intended one. It is being very cautious because it assumes that implementation extensions can be present that would allow user functions to exist when it is invoked. This capability is not specified by this document, but it is not prohibited as an extension. For example, the ENV variable precedes the invocation of the script with a user startup script. Such a script could define functions to spoof the application.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `command`: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

PATH Determine the search path used during the command search, except as described under the `-p` option.

退出状态

/usr/bin/command

When the `-v` or `-V` options are specified, the following exit values are returned:

- 0 Successful completion.
- >0 The *command_name* could not be found or an error occurred.

Otherwise, the following exit values are returned:

- 126 The utility specified by *command_name* was found but could not be invoked.
- 127 An error occurred in the command utility or the utility specified by *command_name* could not be found.

Otherwise, the exit status of `command` is that of the simple command specified by the arguments to `command`.

ksh If *command* is invoked, the exit status of *command* is that of *command*. Otherwise, it is one of the following:

- 0 *command_name* completed successfully.
- >0 -v or -V has been specified and an error occurred.
- 126 *command_name* was found but could not be invoked.
- 127 *command_name* could not be found.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/command

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

ksh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

另请参见 [ksh\(1\)](#), [sh\(1\)](#), [type\(1\)](#), [whence\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名	compress, uncompress, zcat – compress, uncompress files or display expanded files
用法概要	<pre>compress [-fv/] [-b <i>bits</i>] [<i>file</i>]...</pre> <pre>compress -c [-fv] [-b <i>bits</i>] [<i>file</i>]</pre> <pre>uncompress [-fv] [-c -/] [<i>file</i>]...</pre> <pre>zcat [<i>file</i>]...</pre>
描述	
compress	<p>The <code>compress</code> utility attempts to reduce the size of the named files by using adaptive Lempel-Ziv coding. Except when the output is to the standard output, each file is replaced by one with the extension <code>.Z</code>, while keeping the same ownership modes, change times and modification times, ACLs, and extended attributes. The <code>compress</code> utility also attempt to set the owner and group of <code>file.Z</code> to the owner and group of <code>file</code>, but does not fail if this cannot be done. If appending the <code>.Z</code> to the file pathname would make the pathname exceed 1023 bytes, the command fails. If no files are specified, the standard input is compressed to the standard output.</p> <p>The amount of compression obtained depends on the size of the input, the number of <i>bits</i> per code, and the distribution of common substrings. Typically, text such as source code or English is reduced by 50–60%. Compression is generally much better than that achieved by Huffman coding (as used in <code>pack(1)</code>) and it takes less time to compute. The <i>bits</i> parameter specified during compression is encoded within the compressed file, along with a magic number to ensure that neither decompression of random data nor recompression of compressed data is subsequently allowed.</p>
uncompress	<p>The <code>uncompress</code> utility restores files to their original state after they have been compressed using the <code>compress</code> utility. If no files are specified, the standard input is uncompressed to the standard output.</p> <p>This utility supports the uncompressing of any files produced by <code>compress</code>. For files produced by <code>compress</code> on other systems, <code>uncompress</code> supports 9- to 16-bit compression (see <code>-b</code>).</p>
zcat	The <code>zcat</code> utility writes to standard output the uncompressed form of files that have been compressed using <code>compress</code> . It is the equivalent of <code>uncompress -c</code> . Input files are not affected.
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> <code>-b <i>bits</i></code> Sets the upper limit (in bits) for common substring codes. <i>bits</i> must be between 9 and 16 (16 is the default). Lowering the number of bits result in larger, less compressed files. <code>-c</code> Writes to the standard output; no files are changed and no <code>.Z</code> files are created. The behavior of <code>zcat</code> is identical to that of <code>'uncompress -c'</code>. <code>-f</code> When compressing, forces compression of <i>file</i>, even if it does not actually reduce the size of the file, or if the corresponding <code>file.Z</code> file already exists.

If the `-f` option is not specified, and the process is not running in the background, prompts to verify whether an existing file should be overwritten. If the response is affirmative, the existing file is overwritten. When uncompressing, does not prompt for overwriting files. If the `-f` option is not specified, and the process is not running in the background, prompts to verify whether an existing file should be overwritten. If the standard input is not a terminal and `-f` is not specified, writes a diagnostic message to standard error and exits with a status greater than 0.

- `-v` Verbose. Writes to standard error messages concerning the percentage reduction or expansion of each file.
- `-/` When compressing or decompressing, copies any extended system attributes associated with the source file to the target file and copies any extended system attributes associated with extended attributes of the source file to the corresponding extended attributes associated with the target file. If any extended system attributes cannot be copied, the original file is retained, a diagnostic is written to `stderr`, and the final exit status is non-zero.

操作数

The following operand is supported:

file A path name of a file to be compressed by `compress`, uncompressed by `uncompress`, or whose uncompressed form is written to standard out by `zcat`. If *file* is `-`, or if no *file* is specified, the standard input is used.

用法

See [largefile\(5\)](#) for the description of the behavior of `compress`, `uncompress`, and `zcat` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `compress`, `uncompress`, and `zcat`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

Affirmative responses are processed using the extended regular expression defined for the `yesexpr` keyword in the `LC_MESSAGES` category of the user's locale. The locale specified in the `LC_COLLATE` category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for `yesexpr`. The locale specified in `LC_CTYPE` determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the `yesexpr`. See [locale\(5\)](#).

退出状态

The following error values are returned:

- 0 Successful completion.
- 1 An error occurred.
- 2 One or more files were not compressed because they would have increased in size (and the `-f` option was not specified).

>2 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[ln\(1\)](#), [pack\(1\)](#), [fgetattr\(3C\)](#), [fsetattr\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [locale\(5\)](#), [standards\(5\)](#)

诊断

Usage: `compress [-fv/] [-b bits] [file ...]`

`compress c [-fv] [-b bits] [file ...]`

Invalid options were specified on the command line.

Usage: `uncompress [-fv] [-c | -/] [file]...`

Invalid options were specified on the command line.

Missing maxbits

Maxbits must follow `-b`, or invalid maxbits, not a numeric value.

file: not in compressed format

The file specified to `uncompress` has not been compressed.

file: compressed with *xx*bits, can only handle *yy*bits

file was compressed by a program that could deal with more *bits* than the `compress` code on this machine. Recompress the file with smaller *bits*.

file: already has `.Z` suffix -- no change

The file is assumed to be already compressed. Rename the file and try again.

file: already exists; do you wish to overwrite (y or n)?

Respond `y` if you want the output file to be replaced; `n` if not.

`uncompress`: corrupt input

A SIGSEGV violation was detected, which usually means that the input file is corrupted.

Compression:*xx.xx*%

Percentage of the input saved by compression. (Relevant only for `-v`.)

-- not a regular file: unchanged

When the input file is not a regular file, (such as a directory), it is left unaltered.

-- has *xx* other links: unchanged

The input file has links; it is left unchanged. See [ln\(1\)](#) for more information.

- - file unchanged
No savings are achieved by compression. The input remains uncompressed.
- -filename too long to tack on .Z
The path name is too long to append the .Z suffix.
- -cannot preserve extended attributes. file unchanged
Extended system attributes could not be copied.

附注

Although compressed files are compatible between machines with large memory, -b 12 should be used for file transfer to architectures with a small process data space (64KB or less).

compress should be more flexible about the existence of the . Z suffix.

引用名

cp – copy files

用法概要

```

/usr/bin/cp [-fip@/] source_file target_file
/usr/bin/cp [-fip@/] source_file... target
/usr/bin/cp -r | -R [-H | -L | -P] [-fip@/] source_dir... target
/usr/bin/cp -R | -R [-H | -L | -P] [-fip@/] source_dir... target
/usr/xpg4/bin/cp [-fip@/] source_file target_file
/usr/xpg4/bin/cp [-fip@/] source_file... target
/usr/xpg4/bin/cp -r | -R [-H | -L | -P] [-fip@/] source_dir... target
/usr/xpg4/bin/cp -R | -R [-H | -L | -P] [-fip@/] source_dir... target

```

描述

In the first synopsis form, neither *source_file* nor *target_file* are directory files, nor can they have the same name. The cp utility copies the contents of *source_file* to the destination path named by *target_file*. If *target_file* exists, cp overwrites its contents, but the mode (and ACL if applicable), owner, and group associated with it are not changed. The last modification time of *target_file* and the last access time of *source_file* are set to the time the copy was made. If *target_file* does not exist, cp creates a new file named *target_file* that has the same mode as *source_file* except that the sticky bit is not set unless the user is super-user. In this case, the owner and group of *target_file* are those of the user, unless the setgid bit is set on the directory containing the newly created file. If the directory's setgid bit is set, the newly created file has the group of the containing directory rather than of the creating user. If *target_file* is a link to another file, cp overwrites the link destination with the contents of *source_file*; the link(s) from *target_file* remains.

In the second synopsis form, one or more *source_files* are copied to the directory specified by *target*. It is an error if any *source_file* is a file of type directory, if *target* either does not exist or is not a directory.

In the third or fourth synopsis forms, one or more directories specified by *source_dir* are copied to the directory specified by *target*. Either the -r or -R must be specified. For each *source_dir*, cp copies all files and subdirectories.

选项

The following options are supported for both /usr/bin/cp and /usr/xpg4/bin/cp:

- f Unlink. If a file descriptor for a destination file cannot be obtained, this option attempts to unlink the destination file and proceed.
- H Takes actions based on the type and contents of the file referenced by any symbolic link specified as a *source_file* operand.

If the *source_file* operand is a symbolic link, then cp copies the file referenced by the symbolic link for the *source_file* operand. All other symbolic links encountered during traversal of a file hierarchy are preserved.

- i Interactive. cp prompts for confirmation whenever the copy would overwrite an existing target. This is done regardless of whether the input is coming from a terminal. If the prompt for confirmation fails, it is equivalent to the user answering in the negative. An affirmative response means that the copy should proceed. Any other answer prevents cp from overwriting *target*.

- L Takes actions based on the type and contents of the file referenced by any symbolic link specified as a *source_file* operand or any symbolic links encountered during traversal of a file hierarchy.

Copies files referenced by symbolic links. Symbolic links encountered during traversal of a file hierarchy are not preserved.

- p Preserve. The cp utility duplicates not only the contents of *source_file*, but also attempts to preserve its ACL, access and modification times, extended attributes, extended system attributes, file mode, and owner and group ids.

If cp is unable to preserve the access and modification times, extended attributes, or the file mode, cp does not consider it a failure. If cp is unable to preserve the owner and group id, the copy does not fail, but cp silently clears the S_ISUID and S_ISGID bits from the file mode of the target. The copy fails if cp is unable to clear these bits. If cp is unable to preserve the ACL or extended system attributes, the copy fails. If the copy fails, then a diagnostic message is written to `stderr` and (after processing any remaining operands) cp exits with a non-zero exit status.

- P Takes actions on any symbolic link specified as a *source_file* operand or any symbolic link encountered during traversal of a file hierarchy.

Copies symbolic links. Symbolic links encountered during traversal of a file hierarchy are preserved.

- r Recursive. cp copies the directory and all its files, including any subdirectories and their files to *target*. Unless the -H, -L, or -P option is specified, the -L option is used as the default mode.

- R Same as -r, except pipes are replicated, not read from.

- @ Preserves extended attributes. cp attempts to copy all of the source file's extended attributes along with the file data to the destination file.

- / Preserves extended attributes and extended system attributes. Along with the file's data, the cp utility attempts to copy extended attributes and extended system attributes from each source file, and extended system attributes associated with extended attributes to the destination file. If cp is unable to copy extended attributes or extended system attributes, then a diagnostic message is written to `stderr` and (after processing any remaining operands) exits with a non-zero exit status.

Specifying more than one of the mutually-exclusive options `-H`, `-L`, and `-P` is not considered an error. The last option specified determines the behavior of the utility.

`/usr/bin/cp`

If the `-p` option is specified with either the `-@` option or the `-/` option, `/usr/bin/cp` behaves as follows

- When both `-p` and `-@` are specified in any order, the copy fails if extended attributes cannot be copied.
- When both `-p` and `-/` are specified in any order, the copy fails if extended system attributes cannot be copied.

`/usr/xpg4/bin/cp`

If the `-p` option is specified with either the `-@` option or the `-/` option, `/usr/xpg4/bin/cp` behaves as follows:

- When both `-p` and `-@` are specified, the last option specified determines whether the copy fails if extended attributes cannot be preserved.
- When both `-p` and `-/` are specified, the last option specified determines whether the copy fails if extended system attributes cannot be preserved.

操作数

The following operands are supported:

source_file A pathname of a regular file to be copied.

source_dir A pathname of a directory to be copied.

target_file A pathname of an existing or non-existing file, used for the output when a single file is copied.

target A pathname of a directory to contain the copied files.

用法

See [largefile\(5\)](#) for the description of the behavior of `cp` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Copying a File

The following example copies a file:

```
example% cp goodies goodies.old
```

```
example% ls goodies*
goodies goodies.old
```

示例 2 Copying a List of Files

The following example copies a list of files to a destination directory:

```
example% cp ~/src/* /tmp
```

示例 3 Copying a Directory

The following example copies a directory, first to a new, and then to an existing destination directory

```
example% ls ~/bkup
/usr/example/fred/bkup not found
```

```
example% cp -r ~/src ~/bkup
```

```
example% ls -R ~/bkup
x.c y.c z.sh
```

```
example% cp -r ~/src ~/bkup
```

```
example% ls -R ~/bkup
src x.c y.c z.sh
src:
x.c y.c z.s
```

示例 4 Copying Extended File System Attributes

The following example copies extended file system attributes:

```
$ ls -/ c file1
-rw-r--r--  1 foo  staff          0 Oct 29 20:04 file1
             {AH-----m--}
```

```
$ cp -/ file1 file2
```

```
$ ls -/c file2
-rw-r--r--  1 foo  staff          0 Oct 29 20:17 file2
             {AH-----m--}
```

示例 5 Failing to Copy Extended System Attributes

The following example fails to copy extended system attributes:

```
$ ls -/c file1
-rw-r--r--  1 foo  staff          0 Oct 29 20:04 file1
             {AH-----m--}
```

```
$ cp -/ file1 /tmp
```

```
cp: Failed to copy extended system attributes from file1 to /tmp/file1
```

```
$ ls -/c /tmp/file1
-rw-r--r--  1 foo  staff          0 Oct 29 20:09 /tmp/file1
             {}
```


环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of cp: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

Affirmative responses are processed using the extended regular expression defined for the `yesexpr` keyword in the LC_MESSAGES category of the user's locale. The locale specified in the LC_COLLATE category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for `yesexpr`. The locale specified in LC_CTYPE determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the `yesexpr`. See [locale\(5\)](#).

退出状态

The following exit values are returned:

- 0 All files were copied successfully.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/cp

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed

/usr/xpg4/bin/cp

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed

另请参见

[chmod\(1\)](#), [chown\(1\)](#), [setfacl\(1\)](#), [utime\(2\)](#), [fgetattr\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [locale\(5\)](#), [standards\(5\)](#)

附注

The permission modes of the source file are preserved in the copy.

A - - permits the user to mark the end of any command line options explicitly, thus allowing cp to recognize filename arguments that begin with a -.

引用名 cpio – copy file archives in and out

用法概要

```
cpio -i [-bBcdfkmPrsStuvV6@/] [-C bufsize] [-E file]
        [-H header] [-I [-M message]] [-R id] [pattern]. . .

cpio -o [-aABcLPvV@/] [-C bufsize] [-H header]
        [-O file [-M message]]

cpio -p [-adLLmPuvV@/] [-R id] directory
```

描述

The `cpio` command copies files into and out of a `cpio` archive. The `cpio` archive can span multiple volumes. The `-i`, `-o`, and `-p` options select the action to be performed. The following list describes each of the actions. These actions are mutually exclusive.

Copy In Mode

`cpio -i` (copy in) extracts files from the standard input, which is assumed to be the product of a previous `cpio -o` command. Only files with names that match one of the *patterns* are selected. See [sh\(1\)](#) and OPERANDS for more information about *pattern*. Extracted files are conditionally copied into the current directory tree, based on the options described below. The permissions of the files are those of the previous `cpio -o` command. The owner and group are the same as the current user, unless the current user has the `{PRIV_FILE_CHOWN_SELF}` privilege. See [chown\(2\)](#). If this is the case, owner and group are the same as those resulting from the previous `cpio -o` command. Notice that if `cpio -i` tries to create a file that already exists and the existing file is the same age or younger (newer), `cpio` outputs a warning message and not replace the file. The `-u` option can be used to unconditionally overwrite the existing file.

Copy Out Mode

`cpio -o` (copy out) reads a list of file path names from the standard input and copies those files to the standard output, together with path name and status information in the form of a `cpio` archive. Output is padded to an 8192-byte boundary by default or to the user-specified block size (with the `-B` or `-C` options) or to some device-dependent block size where necessary (as with the CTC tape).

Pass Mode

`cpio -p` (pass) reads a list of file path names from the standard input and conditionally copies those files into the destination directory tree, based on the options described below.

If the underlying file system of the source file supports detection of holes as reported by [pathconf\(2\)](#), the file is a sparse file, and the destination file is seekable, then holes in sparse files are preserved in pass mode, otherwise holes are filled with zeros.

`cpio` assumes four-byte words.

If, when writing to a character device (`-o`) or reading from a character device (`-i`), `cpio` reaches the end of a medium, and the `-O` and `-I` options are not used, `cpio` prints the following message:

```
To continue, type device/file name when ready.
```

To continue, you must replace the medium and type the character special device name and press RETURN.

选项

The following options are supported:

- i (copy in) Reads an archive from the standard input and conditionally extracts the files contained in it and places them into the current directory tree.
- o (copy out) Reads a list of file path names from the standard input and copies those files to the standard output in the form of a `cpio` archive.
- p (pass) Reads a list of file path names from the standard input and conditionally copies those files into the destination directory tree.

The following options can be appended in any sequence to the `-i`, `-o`, or `-p` options:

- 0 Reads a list of filenames terminated by a null character, instead of a NEWLINE, so that files whose names contain NEWLINESs can be archived. Using `find` with the `-print0` option is one way to produce such a list of filenames.

This option may be used in copy-out and copy-pass modes.
- a Resets access times of input files after they have been copied, making `cpio`'s access invisible. Access times are not reset for linked files when `cpio -pla` is specified.
- A Appends files to an archive. The `-A` option requires the `-0` option. Valid only with archives that are files or that are on hard disk partitions. The effect on files that are linked in the existing portion of the archive is unpredictable.
- b Reverses the order of the bytes within each word. Use only with the `-i` option.
- B Blocks input/output 5120 bytes to the record. The default buffer size is 8192 bytes when this and the `-C` options are not used. `-B` does not apply to the `-p` (pass) option.
- c Reads or writes header information in ASCII character form for portability. There are no UID or GID restrictions associated with this header format. Use this option between SVR4-based machines, or the `-H odc` option between unknown machines. The `-c` option implies the use of expanded device numbers, which are only supported on SVR4-based systems. When transferring files between SunOS 4 or Interactive UNIX and the Solaris 2.6 Operating environment or compatible versions, use `-H odc`.
- C *bufsize* Blocks input/output *bufsize* bytes to the record, where *bufsize* is replaced by a positive integer. The default buffer size is 8192 bytes when this and `-B` options are not used. `-C` does not apply to the `-p` (pass) option.
- d Creates directories as needed.
- E *file* Specifies an input file (*file*) that contains a list of filenames to be extracted from the archive (one filename per line).

-f Copies in all files except those in *patterns*. See OPERANDS for a description of *pattern*.

-H *header* Reads or writes header information in *header* format. Always use this option or the -c option when the origin and the destination machines are different types. This option is mutually exclusive with options -c and -6.

Valid values for *header* are:

bar bar head and format. Used only with the -i option (read only).

crc | CRC ASCII header with expanded device numbers and an additional per-file checksum. There are no UID or GID restrictions associated with this header format.

odc ASCII header with small device numbers. This is the IEEE/P1003 Data Interchange Standard cpio header and format. It has the widest range of portability of any of the header formats. It is the official format for transferring files between POSIX-conforming systems (see [standards\(5\)](#)). Use this format to communicate with SunOS 4 and Interactive UNIX. This header format allows UIDs and GIDs up to 262143 to be stored in the header.

tar | TAR tar header and format. This is an older tar header format that allows UIDs and GIDs up to 2097151 to be stored in the header. It is provided for the reading of legacy archives only, that is, in conjunction with option -i.

Specifying this archive format with option -o has the same effect as specifying the “ustar” format: the output archive is in ustar format, and must be read using -H ustar.

ustar | USTAR IEEE/P1003 Data Interchange Standard tar header and format. This header format allows UIDs and GIDs up to 2097151 to be stored in the header.

Files with UIDs and GIDs greater than the limit stated above are archived with the UID and GID of 60001. To transfer a large file (8 Gb — 1 byte), the header format can be tar | TAR, ustar | USTAR, or odc only.

-I *file* Reads the contents of *file* as an input archive, instead of the standard input. If *file* is a character special device, and the current medium has been completely read, replace the medium and press RETURN to continue to the next medium. This option is used only with the -i option.

-
- k Attempts to skip corrupted file headers and I/O errors that might be encountered. If you want to copy files from a medium that is corrupted or out of sequence, this option lets you read only those files with good headers. For `cpio` archives that contain other `cpio` archives, if an error is encountered, `cpio` can terminate prematurely. `cpio` finds the next good header, which can be one for a smaller archive, and terminate when the smaller archive's trailer is encountered. Use only with the `-i` option.
 - l In pass mode, makes hard links between the source and destination whenever possible. If the `-L` option is also specified, the hard link is to the file referred to by the symbolic link. Otherwise, the hard link is to the symbolic link itself. Use only with the `-p` option.
 - L Follows symbolic links. If a symbolic link to a directory is encountered, archives the directory referred to by the link, using the name of the link. Otherwise, archives the file referred to by the link, using the name of the link.
 - m Retains previous file modification time. This option is ineffective on directories that are being copied.
 - M *message* Defines a *message* to use when switching media. When you use the `-O` or `-I` options and specify a character special device, you can use this option to define the message that is printed when you reach the end of the medium. One `%d` can be placed in *message* to print the sequence number of the next medium needed to continue.
 - O *file* Directs the output of `cpio` to *file*, instead of the standard output. If *file* is a character special device and the current medium is full, replace the medium and type a carriage return to continue to the next medium. Use only with the `-o` option.
 - P Preserves ACLs. If the option is used for output, existing ACLs are written along with other attributes, except for extended attributes, to the standard output. ACLs are created as special files with a special file type. If the option is used for input, existing ACLs are extracted along with other attributes from standard input. The option recognizes the special file type. Notice that errors occurs if a `cpio` archive with ACLs is extracted by previous versions of `cpio`. This option should not be used with the `-c` option, as ACL support might not be present on all systems, and hence is not portable. Use ASCII headers for portability.
 - r Interactively renames files. If the user types a carriage return alone, the file is skipped. If the user types a ".", the original pathname is retained. Not available with `cpio -p`.
 - R *id* Reassigns ownership and group information for each file to user ID. (ID must be a valid login ID from the `passwd` database.) This option is valid only when `id` is the invoking user or the super-user. See NOTES.

- s Swaps bytes within each half word.
- S Swaps halfwords within each word.
- t Prints a table of contents of the input. If any file in the table of contents has extended attributes, these are also listed. No files are created. -t and -V are mutually exclusive.
- u Copies unconditionally. Normally, an older file is not replaced a newer file with the same name, although an older directory updates a newer directory.
- v Verbose. Prints a list of file and extended attribute names. When used with the -t option, the table of contents looks like the output of an `ls -l` command (see [ls\(1\)](#)).
- V Special verbose. Prints a dot for each file read or written. Useful to assure the user that `cpio` is working without printing out all file names.
- 6 Processes a UNIX System Sixth Edition archive format file. Use only with the -i option. This option is mutually exclusive with -c and -H.
- @ Includes extended attributes in archive. By default, `cpio` does not place extended attributes in the archive. With this flag, `cpio` looks for extended attributes on the files to be placed in the archive and add them, as regular files, to the archive. The extended attribute files go in the archive as special files with special file types. When the -@ flag is used with -i or -p, it instructs `cpio` to restore extended attribute data along with the normal file data. Extended attribute files can only be extracted from an archive as part of a normal file extract. Attempts to explicitly extract attribute records are ignored.
- / Includes extended system attributes in archive. By default, `cpio` does not place extended system attributes in the archive. With this flag, `cpio` looks for extended system attributes on the files to be placed in the archive and add them, as regular files, to the archive. The extended attribute files go in the archive as special files with special file types. When the -/ flag is used with -i or -p, it instructs `cpio` to restore extended system attribute data along with the normal file data. Extended system attribute files can only be extracted from an archive as part of a normal file extract. Attempts to explicitly extract attribute records are ignored.

操作数

The following operands are supported:

- directory* A path name of an existing directory to be used as the target of `cpio -p`.
- pattern* Expressions making use of a pattern-matching notation similar to that used by the shell (see [sh\(1\)](#)) for filename pattern matching, and similar to regular expressions. The following metacharacters are defined:
 - * Matches any string, including the empty string.

- ? Matches any single character.
- [. . .] Matches any one of the enclosed characters. A pair of characters separated by '-' matches any symbol between the pair (inclusive), as defined by the system default collating sequence. If the first character following the opening '[' is a '!', the results are unspecified.
- ! The ! (exclamation point) means *not*. For example, the !abc* pattern would exclude all files that begin with abc.

In *pattern*, metacharacters ?, *, and [. . .] match the slash (/) character, and backslash (\) is an escape character. Multiple cases of *pattern* can be specified and if no *pattern* is specified, the default for *pattern* is * (that is, select all files).

Each pattern must be enclosed in double quotes. Otherwise, the name of a file in the current directory might be used.

用法

See [largefile\(5\)](#) for the description of the behavior of `cpio` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

The following examples show three uses of `cpio`.

示例 1 Using standard input

```
example% ls | cpio -oc > ../newfile
```

When standard input is directed through a pipe to `cpio -o`, as in the example above, it groups the files so they can be directed (>) to a single file (`../newfile`). The `-c` option insures that the file is portable to other machines (as would the `-H` option). Instead of [ls\(1\)](#), you could use [find\(1\)](#), [echo\(1\)](#), [cat\(1\)](#), and so on, to pipe a list of names to `cpio`. You could direct the output to a device instead of a file.

示例 2 Extracting files into directories

```
example% cat newfile | cpio -icd "memo/a1" "memo/b*"
```

In this example, `cpio -i` uses the output file of `cpio -o` (directed through a pipe with `cat`), extracts those files that match the patterns (`memo/a1`, `memo/b*`), creates directories below the current directory as needed (`-d` option), and places the files in the appropriate directories. The `-c` option is used if the input file was created with a portable header. If no patterns were given, all files from `newfile` would be placed in the directory.

示例 3 Copying or linking files to another directory

```
example% find . -depth -print | cpio -pdlmv newdir
```

In this example, `cpio -p` takes the file names piped to it and copies or links (`-l` option) those files to another directory, `newdir`. The `-d` option says to create directories as needed. The `-m` option says to retain the modification time. (It is important to use the `-depth` option of

示例 3 Copying or linking files to another directory (续)

[find\(1\)](#) to generate path names for `cpio`. This eliminates problems that `cpio` could have trying to create files under read-only directories.) The destination directory, `newdir`, must exist.

Notice that when you use `cpio` in conjunction with `find`, if you use the `-L` option with `cpio`, you must use the `-follow` option with `find` and vice versa. Otherwise, there are undesirable results.

For multi-reel archives, dismount the old volume, mount the new one, and continue to the next tape by typing the name of the next device (probably the same as the first reel). To stop, type a RETURN and `cpio` ends.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `cpio`: `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, `TZ`, and `NLSPATH`.

`TMPDIR` `cpio` creates its temporary file in `/var/tmp` by default. Otherwise, it uses the directory specified by `TMPDIR`.

退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed

另请参见

[ar\(1\)](#), [cat\(1\)](#), [echo\(1\)](#), [find\(1\)](#), [ls\(1\)](#), [pax\(1\)](#), [setfacl\(1\)](#), [sh\(1\)](#), [tar\(1\)](#), [chown\(2\)](#), [archives.h\(3HEAD\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注

The maximum path name length allowed in a `cpio` archive is determined by the header type involved. The following table shows the proper value for each supported archive header type.

Header type	Command line options	Maximum path name length
BINARY	"-o"	256
POSIX	"-oH odc"	256

Header type	Command line options	Maximum path name length
ASCII	“-oc”	1023
CRC	“-oH crc”	1023
USTAR	“-oH ustar”	255

When the command line options “-o -H tar” are specified, the archive created is of type USTAR. This means that it is an error to read this same archive using the command line options “-i -H tar”. The archive should be read using the command line options “-i -H ustar”. The options “-i -H tar” refer to an older tar archive format.

An error message is output for files whose UID or GID are too large to fit in the selected header format. Use -H crc or -c to create archives that allow all UID or GID values.

Only the super-user can copy special files.

Blocks are reported in 512-byte quantities.

If a file has 000 permissions, contains more than 0 characters of data, and the user is not root, the file is not saved or restored.

When cpio is invoked in Copy In or Pass Mode by a user with {PRIV_FILE_CHOWN_SELF} privilege, and in particular on a system where {_POSIX_CHOWN_RESTRICTED} is not in effect (effectively granting this privilege to all users where not overridden), extracted or copied files can end up with owners and groups determined by those of the original archived files, which can differ from the invoking user's. This might not be what the user intended. The -R option can be used to retain file ownership, if desired, if you specify the user's id.

The inode number stored in the header (/usr/include/archives.h) is an unsigned short, which is 2 bytes. This limits the range of inode numbers from 0 to 65535. Files which are hard linked must fall in this inode range. This could be a problem when moving cpio archives between different vendors' machines.

You must use the same blocking factor when you retrieve or copy files from the tape to the hard disk as you did when you copied files from the hard disk to the tape. Therefore, you must specify the -B or -C option.

During -p and -o processing, cpio buffers the file list presented on stdin in a temporary file.

The new [pax\(1\)](#) format, with a command that supports it (for example, tar), should be used for large files. The cpio command is no longer part of the current POSIX standard and is deprecated in favor of pax.

引用名 cpp – the C language preprocessor

用法概要 /usr/lib/cpp [-BCHMpPRT] [-undef] [-Dname] [-Dname = def]
 [-Idirectory] [-Uname] [-Ydirectory]
 [*input-file* [*output-file*]]

描述 cpp is the C language preprocessor. cpp also used as a first-pass preprocessor for other Sun compilers.

Although cpp can be used as a macro processor, this is not normally recommended, as its output is geared toward that which would be acceptable as input to a compiler's second pass. Thus, the preferred way to invoke cpp is through a compilation command. For general-purpose macro-processing, see [m4\(1\)](#).

cpp optionally accepts two filenames as arguments. *input-file* and *output-file* are, respectively, the input and output files for the preprocessor. They default to the standard input and the standard output.

选项 The following options are supported:

- B Supports the C++ comment indicator / /. With this indicator, everything on the line after the / / is treated as a comment.
- C Passes all comments (except those that appear on cpp directive lines) through the preprocessor. By default, cpp strips out C-style comments.
- H Prints the pathnames of included files, one per line on the standard error.
- M Generates a list of makefile dependencies and write them to the standard output. This list indicates that the object file which would be generated from the input file depends on the input file as well as the include files referenced.
- p Uses only the first eight characters to distinguish preprocessor symbols, and issue a warning if extra tokens appear at the end of a line containing a directive.
- P Preprocesses the input without producing the line control information used by the next pass of the C compiler.
- R Allows recursive macros.
- T Uses only the first eight characters for distinguishing different preprocessor names. This option is included for backward compatibility with systems which always use only the first eight characters.
- undef Removes initial definitions for all predefined symbols.
- Dname Defines *name* as 1 (one). This is the same as if a -Dname=1 option appeared on the cpp command line, or as if a

 #*define name 1*

- line appeared in the source file that `cpp` is processing.
- `-Dname=def` Defines *name* as if by a `#define` directive. This is the same as if a `#define name def` line appeared in the source file that `cpp` is processing. The `-D` option has lower precedence than the `-U` option. That is, if the same name is used in both a `-U` option and a `-D` option, the name will be undefined regardless of the order of the options.
- `-Idirectory` Inserts *directory* into the search path for `#include` files with names not beginning with `./`. *directory* is inserted ahead of the standard list of `include` directories. Thus, `#include` files with names enclosed in double-quotes (") are searched for first in the directory of the file with the `#include` line, then in directories named with `-I` options, and lastly, in directories from the standard list. For `#include` files with names enclosed in angle-brackets (< >), the directory of the file with the `#include` line is not searched. See `Details` below for exact details of this search order.
- `-Uname` Removes any initial definition of *name*, where *name* is a symbol that is predefined by a particular preprocessor. Here is a partial list of symbols that may be predefined, depending upon the architecture of the system:
- | | |
|--------------------------------|---|
| Operating System: | <code>ibm</code> , <code>gcos</code> , <code>os</code> , <code>tss</code> and <code>unix</code> |
| Hardware: | <code>interdata</code> , <code>u3b20d</code> , <code>ns32000</code> , <code>i386</code> , <code>sparc</code> , and <code>sun</code> |
| UNIX system variant: | <code>RES</code> , and <code>RT</code> |
| The <code>lint</code> command: | <code>lint</code> |
- The symbols `sun`, `sparc` and `unix` are defined for all Sun systems.
- `-Ydirectory` Uses *directory* in place of the standard list of directories when searching for `#include` files.

用法

Directives

All `cpp` directives start with a hash symbol (`#`) as the first character on a line. White space (SPACE or TAB characters) can appear after the initial `#` for proper indentation.

```
#define name token-string
```

Replace subsequent instances of *name* with *token-string*.

```
#define name (argument [, argument] . . . ) token-string
```

There can be no space between *name* and the `'(`. Replace subsequent instances of *name*, followed by a parenthesized list of arguments, with *token-string*, where each occurrence of an *argument* in the *token-string* is replaced by the corresponding token in the comma-separated list. When a macro with arguments is expanded, the arguments are

placed into the expanded *token-string* unchanged. After the entire *token-string* has been expanded, cpp re-starts its scan for names to expand at the beginning of the newly created *token-string*.

#undef *name*

Remove any definition for the symbol *name*. No additional tokens are permitted on the directive line after *name*.

#include "*filename*"

#include <*filename*>

Read in the contents of *filename* at this location. This data is processed by cpp as if it were part of the current file. When the <*filename*> notation is used, *filename* is only searched for in the standard include directories. See the -I and -Y options above for more detail. No additional tokens are permitted on the directive line after the final " or >.

#line *integer-constant* "*filename*"

Generate line control information for the next pass of the C compiler. *integer-constant* is interpreted as the line number of the next line and *filename* is interpreted as the file from where it comes. If "*filename*" is not given, the current filename is unchanged. No additional tokens are permitted on the directive line after the optional *filename*.

#if *constant-expression*

Subsequent lines up to the matching **#else**, **#elif**, or **#endif** directive, appear in the output only if *constant-expression* yields a nonzero value. All binary non-assignment C operators, including **&&**, **|**, and **,**, are legal in *constant-expression*. The **?:** operator, and the unary **-**, **!**, and **~** operators, are also legal in *constant-expression*.

The precedence of these operators is the same as that for C. In addition, the unary operator **defined**, can be used in *constant-expression* in these two forms: **'defined (*name*)'** or **'defined *name*'**. This allows the effect of **#ifdef** and **#ifndef** directives (described below) in the **#if** directive. Only these operators, integer constants, and names that are known by cpp should be used within *constant-expression*. In particular, the size of operator is not available.

#ifdef *name*

Subsequent lines up to the matching **#else**, **#elif**, or **#endif** appear in the output only if *name* has been defined, either with a **#define** directive or a **-D** option, and in the absence of an intervening **#undef** directive. Additional tokens after *name* on the directive line will be silently ignored.

#ifndef *name*

Subsequent lines up to the matching **#else**, **#elif**, or **#endif** appear in the output only if *name* has *not* been defined, or if its definition has been removed with an **#undef** directive. No additional tokens are permitted on the directive line after *name*.

#elif constant-expression

Any number of `#elif` directives may appear between an `#if`, `#ifdef`, or `#ifndef` directive and a matching `#else` or `#endif` directive. The lines following the `#elif` directive appear in the output only if all of the following conditions hold:

- The *constant-expression* in the preceding `#if` directive evaluated to zero, the *name* in the preceding `#ifdef` is not defined, or the *name* in the preceding `#ifndef` directive was defined.
- The *constant-expression* in all intervening `#elif` directives evaluated to zero.
- The current *constant-expression* evaluates to non-zero.

If the *constant-expression* evaluates to non-zero, subsequent `#elif` and `#else` directives are ignored up to the matching `#endif`. Any *constant-expression* allowed in an `#if` directive is allowed in an `#elif` directive.

#else

This inverts the sense of the conditional directive otherwise in effect. If the preceding conditional would indicate that lines are to be included, then lines between the `#else` and the matching `#endif` are ignored. If the preceding conditional indicates that lines would be ignored, subsequent lines are included in the output. Conditional directives and corresponding `#else` directives can be nested.

#endif

End a section of lines begun by one of the conditional directives `#if`, `#ifdef`, or `#ifndef`. Each such directive must have a matching `#endif`.

Macros

Formal parameters for macros are recognized in `#define` directive bodies, even when they occur inside character constants and quoted strings. For instance, the output from:

```
#define abc(a) '|'|a|
abc(xyz)
```

is:

```
# 1 ""
|'|xyz |
```

The second line is a NEWLINE. The last seven characters are `|'|xyz|` (vertical-bar, back quote, vertical-bar, x, y, z, vertical-bar). Macro names are not recognized within character constants or quoted strings during the regular scan. Thus:

```
#define abc xyz
printf("abc");
```

does not expand `abc` in the second line, since it is inside a quoted string that is not part of a `#define` macro definition.

Macros are not expanded while processing a `#define` or `#undef`. Thus:

```
#define abc zingo
#define xyz abc
#undef abc
xyz
```

produces abc. The token appearing immediately after an `#ifdef` or `#ifndef` is not expanded.

Macros are not expanded during the scan which determines the actual parameters to another macro call.

```
#define reverse(first,second)second first
#define greeting hello
reverse(greeting,
#define greeting goodbye
)
```

produces

```
#define hello goodbye hello
```

Output

Output consists of a copy of the input file, with modifications, plus lines of the form:

```
#lineno " filename" "level"
```

indicating the original source line number and filename of the following output line and whether this is the first such line after an include file has been entered (*level*=1), the first such line after an include file has been exited (*level*=2), or any other such line (*level* is empty).

Details

This section contains usage details.

Directory Search Order

`#include` files are searched for in the following order:

1. The directory of the file that contains the `#include` request (that is, `#include` is relative to the file being scanned when the request is made).
2. The directories specified by `-I` options, in left-to-right order.
3. The standard directory(s) (`/usr/include` on UNIX systems).

Special Names

Two special names are understood by `cpp`. The name `__LINE__` is defined as the current line number (a decimal integer) as known by `cpp`, and `__FILE__` is defined as the current filename (a C string) as known by `cpp`. They can be used anywhere (including in macros) just as any other defined name.

Newline Characters

A NEWLINE character terminates a character constant or quoted string. An escaped NEWLINE (that is, a backslash immediately followed by a NEWLINE) may be used in the body of a `#define` statement to continue the definition onto the next line. The escaped NEWLINE is not included in the macro value.

Comments Comments are removed (unless the `-C` option is used on the command line). Comments are also ignored, except that a comment terminates a token.

退出状态 The following exit values are returned:

0 Successful completion.

non-zero An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

另请参见 [m4\(1\)](#), [attributes\(5\)](#)

诊断 The error messages produced by `cpp` are intended to be self-explanatory. The line number and filename where the error occurred are printed along with the diagnostic.

附注 When NEWLINE characters were found in argument lists for macros to be expanded, some previous versions of `cpp` put out the NEWLINE characters as they were found and expanded. The current version of `cpp` replaces them with SPACE characters.

Because the standard directory for included files may be different in different environments, this form of `#include` directive:

```
#include <file.h>
```

should be used, rather than one with an absolute path, like:

```
#include "/usr/include/file.h"
```

`cpp` warns about the use of the absolute pathname.

While the compiler allows 8-bit strings and comments, 8-bits are not allowed anywhere else.

引用名

cputrack – 使用 CPU 性能计数器监视进程和 LWP 行为

用法概要

```
cputrack -c eventspec [-c eventspec]... [-efntvD] [-N count] [-o pathname] [-T interval] command [args]
cputrack -c eventspec [-c eventspec]... -p pid [-efntvD] [-N count] [-o pathname] [-T interval]
cputrack -h
```

描述

通过 `cputrack` 实用程序，可使用 CPU 性能计数器监视系统上运行的一个或一系列进程的行为。如果使用 `-T` 选项指定 *interval*，`cputrack` 会每隔 *interval* 秒对活动进行一次抽样并一直重复。如果使用 `-N` 选项指定 *count*，则会对跟踪的每个进程重复统计 *count* 次信息。如果未指定任何参数，将使用一秒钟的间隔。如果指定 *command* 和可选 *args*，`cputrack` 会在监视指定的 CPU 性能事件时运行带有给定参数的命令。此外，可以使用 `-p` 选项指定现有进程的进程 ID。

因为 `cputrack` 是非特权程序，所以它遵循适用于 `truss(1)` 的相同限制。例如，无法跟踪 `setuid(2)` 可执行程序。

选项

支持以下选项：

-c *eventspec*

指定一组 CPU 性能计数器要监视的事件。这些事件规范的语法为：

```
[picn=eventn[,attr[n][=val]]][, [picn=eventn
    [,attr[n][=val]],...]
```

您可以使用 `-h` 选项获取可用事件和属性的列表。这会导致生成用法消息。可以不必显式分配计数器，这时 `cpustat` 会自动尝试选择适用的计数器。

可以按适合 `strtoll(3C)` 的格式以十六进制、八进制或十进制表示法表示属性值。事件规范中存在的没有显式值的属性会收到缺省值 1。没有相应计数器编号的属性将应用到规范中的所有计数器。

可通过阅读 CPU 制造商的事件文档来确定这些事件规范的语义。

可以指定多个 `-c` 选项，这时 `cputrack` 会在每个样例上的不同事件设置之间循环。

-D

启用调试模式。

-e

执行所有 `exec(2)` 或 `execve(2)` 系统调用。

-f

执行 `fork(2)`、`fork1(2)` 或 `vfork(2)` 系统调用创建的所有子项。

-h

显示一条详细帮助消息，说明如何使用实用程序、如何对依赖处理器的计数器进行编程以及从何处可以找到更多详细信息。

-n

省略所有文件头输出（在 `cputrack` 位于流水线的开头时很有用）。

- N *count*
指定在退出之前可以采集的最大 CPU 性能计数器样例数。
- o *outfile*
指定要用于 `cputrack` 输出的文件。
- p *pid*
将该参数解释为应附加和监视进程计数器上下文的现有进程的进程 ID。
- t
显示处理器周期计数的附加列（如果在当前体系结构中可用）。
- T *interval*
指定 CPU 性能计数器抽样之间的间隔（以秒为单位）。间隔太小可能会导致某些样例被跳过。请参见“警告”部分。
- v
启用更详细的输出。

用法

操作系统会在跟踪进程时强制使用某些限制。尤其是，如果命令的目标文件不能由用户读取，则该用户无法跟踪此命令；`set-uid` 和 `set-gid` 命令只能由特权用户进行跟踪。除非由特权用户运行，否则 `cputrack` 会失去对 `set-id` 或不可读目标文件执行 `exec()` 的任何进程的控制权。此类进程通常会从 `exec()` 出发继续执行，而不考虑 `cputrack`。

在使用 `-f` 选项时，系统可能用完每个用户的进程槽，因为 `cputrack` 会为每个被跟踪的进程运行一个控制进程。

当硬件计数器是实际样例时，`cputrack` 所显示的时间对应于墙上时钟时间。这与 `gethrtime(3C)` 相同的时基派生时间。

`cputrack` 实用程序会将性能计数器上下文附加到它所检查的每个进程。通过此上下文，可在系统上的不同进程之间复用性能计数器，但不能将其与 `cpustat(1M)` 实用程序同时使用。

若 `cpustat` 实用程序的某个实例正在运行中，则进一步运行 `cputrack` 的尝试将失败，直到所有 `cpustat` 实例终止为止。

有时，`cputrack` 会非常灵活并显示足够的统计信息，如此可不必将监测代码添加到应用程序中。但是，有时需要更多控制。因为应用程序本身和由 `cputrack` 注入到应用程序中的代理 LWP 使用相同的性能计数器上下文，所以应用程序可能会与计数器上下文进行交互以实现某些值得关注的功能。请参见 `cpc_enable(3CPC)`。

通过 `-t` 选项启用的处理器周期计数始终适用于用户模式和系统模式，无论应用于性能计数器寄存器的设置为何。

根据设计，通过 `nawk(1)` 和 `perl(1)` 可以很容易地解析 `cputrack` 的输出，从而允许通过在脚本中嵌入 `cputrack` 来构建性能工具。此外，也可以使用构建 `cputrack` 所依据的同一 API 或使用 `libcpc(3LIB)` 和 `libpctx(3LIB)` 的功能直接构造这些工具。请参见 `cpc(3CPC)`。

虽然 `cputrack` 使用性能计数器上下文来维护每个 LWP 的各个性能计数器值，但某些可以计数的事件会不可避免地受到系统上发生的其他活动的影响，尤其是进程间共享的受限资源（例如，高速缓存未命中率）。对于此类事件，使用 `cpustat(1M)` 监测整体系统行为也可能很值得关注。

对于 `-T interval` 选项，如果将 `interval` 指定为零，则不会执行定期抽样。仅当进程创建或销毁 LWP 或者调用 `fork(2)`、`exec(2)` 或 `exit(2)` 时，才对性能计数器进行抽样。

示例

SPARC

示例1 使用性能计数器对时钟周期进行计数

在本示例中，将在包含 UltraSPARC-III+ 处理器的计算机上使用该实用程序。计数器被设置为对处理器时钟周期和运行 `sleep(1)` 命令时在用户模式中分发的指令进行计数。

```
example% cputrack -c pic0=Cycle_cnt,pic1=Instr_cnt sleep 10
```

time	lwp	event	pic0	pic1
1.007	1	tick	765308	219233
2.007	1	tick	0	0
4.017	1	tick	0	0
6.007	1	tick	0	0
8.007	1	tick	0	0
10.007	1	tick	0	0
10.017	1	exit	844703	228058

示例2 对外部高速缓存引用和未命中进行计数

本示例显示了在 UltraSPARC 计算机上执行简单 shell 脚本的 `fork()` 和 `exec()` 时的更详细输出。计数器正在测量外部高速缓存引用和外部高速缓存未命中的数量。请注意，在没有歧义的情况下可以不必显式指定 `pic0` 和 `pic1` 名称。

```
example% cputrack -fev -c EC_ref,EC_hit /bin/ulimit -c
```

time	pid	lwp	event	pic0	pic1
0.007	101142	1	init_lwp	805286	20023
0.023	101142	1	fork		# 101143
0.026	101143	1	init_lwp	1015382	24461
0.029	101143	1	fini_lwp	1025546	25074
0.029	101143	1	exec	1025546	25074
0.000	101143	1	exec		\
					# '/usr/bin/sh /usr/bin/basename\ /bin/ulimit'
0.039	101143	1	init_lwp	1025546	25074
0.050	101143	1	fini_lwp	1140482	27806
0.050	101143	1	exec	1140482	27806
0.000	101143	1	exec		# '/usr/bin/expr \

示例2 对外部高速缓存引用和未命中进行计数 (续)

```
//bin/ulimit : \(.^[^/]\)/*$ : .*\/(..*) : \(.*\)$ | //bin/ulimi'
0.059 101143 1  init_lwp  1140482  27806
0.075 101143 1  fini_lwp  1237647  30207
0.075 101143 1  exit     1237647  30207
unlimited
0.081 101142 1  fini_lwp  953383   23814
0.081 101142 1  exit     953383   23814
```

x86

示例3 对指令进行计数

本示例显示了为了在 Pentium III 计算机上输出日期在应用程序中和内核中执行的指令数：

```
example% cputrack -c inst_retired,inst_retired,nouser1,sys1 date
```

```
time lwp      event      pic0      pic1
Fri Aug 20 20:03:08 PDT 1999
0.072 1        exit      246725    339666
```

示例4 对 TLB 命中进行计数

本示例显示了如何在 Pentium 4 计算机上使用特定于处理器的属性对 TLB 命中进行计数：

```
example% cputrack -c ITLB_reference,emask=1 date
```

```
time lwp      event      pic0
Fri Aug 20 20:03:08 PDT 1999
0.072 1        exit      246725
```

警告

通过运行 `cpustat(1M)` 实用程序的任何实例，强行使计算机上的所有现有性能计数器上下文无效。这可能会导致 `cputrack` 命令的所有调用因未知错误而过早退出。

如果在其 CPU 性能计数器不受 Solaris 支持的系统上调用 `cpustat`，将显示以下消息：

```
cputrack: cannot access performance counters - Operation not applicable
```

此错误消息表明 `cpc_open()` 已失败且已记录在 `cpc_open(3CPC)` 中。查看此文档以获取有关该问题及其可能的解决方案的更多信息。

如果请求的时间间隔较短，`cputrack` 可能无法达到所需的抽样率。在这种情况下，可能会丢弃某些样例。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	diagnostic/cpu-counters

属性类型	属性值
接口稳定性	Committed (已确定)

另请参见

[nawk\(1\)](#)、[perl\(1\)](#)、[proc\(1\)](#)、[truss\(1\)](#)、[prstat\(1M\)](#)、[cpustat\(1M\)](#)、[exec\(2\)](#)、[exit\(2\)](#)、[fork\(2\)](#)、[set](#)

引用名	crle – 配置运行时链接环境
用法概要	<pre>crle [-64] [-a name] [-A name] [-c conf] [-e env] [-E env] [-f flags] [-i name] [-I name] [-g name] [-G name] [-l dir] [-o dir] [-s dir] [-t] [-u] [-v]</pre>
描述	<p>crle 实用程序针对运行时链接配置文件的创建和显示而提供。配置文件由运行时链接程序 ld.so.1(1) 在进程启动期间读取和解释。运行时链接程序尝试为所有进程读取一个缺省配置文件。对于 32 位进程，缺省配置文件是 <code>/var/ld/ld.config</code>。对于 64 位进程，缺省配置文件是 <code>/var/ld/64/ld.config</code>。</p> <p>若不带任何参数，或仅仅只有 <code>-c</code> 选项，crle 将显示配置信息。此信息包括配置文件的内容、所有系统缺省值以及重新生成该配置文件所需的命令行。当与其他任何选项结合使用时，将会创建或更新为一个新的配置文件。</p> <p>还可通过设置 LD_CONFIG 系列的环境变量之一将运行时链接程序定向至备用配置文件。LD_CONFIG 对于 32 位和 64 位程序均适用。因为 32 位和 64 位配置文件不同，单一的配置文件无法同时用于两类目标文件。因此，在一类程序执行另一类程序时，LD_CONFIG 可能会对程序执行产生不利影响。尤其是，由 32 位版本的标准 Solaris 实用程序执行其对应的 64 位程序这种情况很常见。LD_CONFIG 在此情况下无法胜任。因此，建议使用专门针对某一类进程的 LD_CONFIG_32 和 LD_CONFIG_64 环境变量。</p> <p>在标准位置 <code>/var/ld</code> 中创建不正确的配置文件可能会阻止程序运行，进而可能难于从其中进行恢复。要防止出现此情况，建议首先在临时位置创建新的配置文件。然后，针对此新的配置文件设置适当的 LD_CONFIG 环境变量。该设置会使运行时链接程序使用新的配置文件而非任何缺省值。验证后，如果需要，可将新的配置文件移至缺省位置。任何时候，都可将环境变量 LD_NOCONFIG 设置为任意值以指示运行时链接程序忽略任何配置文件。该设置可通过实验证明其有用性。</p> <p>配置文件可以包含以下信息。</p> <p>缺省搜索路径</p> <p>运行时链接程序使用规定的搜索路径定位目标文件的动态依赖项。此搜索路径的构成部分如下所示：以任何一个 LD_LIBRARY_PATH 定义所指定的路径开始，后跟目标文件的 runpath 所指定的路径。最后使用特定于目标文件类的任何缺省搜索路径。搜索路径中这个最后的组成部分可以在配置文件中提供。通常，使用该功能时可通过任何系统缺省值进行补充。请参见 <code>-l</code> 和 <code>-u</code> 选项。</p> <p>可信目录</p> <p>在处理安全应用程序时，运行时链接程序将限制 LD_LIBRARY_PATH 搜索和 \$ORIGIN 标记扩展的使用。请参见《链接程序和库指南》中的“安全性”。此外，也会限制可从其中定位预装入和审计库的目录。与预装入和审计库关联的路径名会限制为已知可信目录。可信目录可以在配置文件中提供。通常，使用该功能时可通过任何系统缺省值进行补充。请参见 <code>-s</code> 和 <code>-u</code> 选项。</p> <p>环境变量</p> <p>任何由运行时链接程序解释的环境变量都可在配置文件中指定。</p>

目录高速缓存

已定义目录内共享目标文件的位置可以作为配置文件内的高速缓存进行维护。该目录高速缓存可以降低搜索应用程序依赖项的开销。

备用目标文件

结合目录高速缓存，共享目标文件可以指定备用目标文件以便在运行时使用。这些备用目标文件可以由用户提供。备用目标文件还可通过 `crle` 创建，作为共享目标文件的副本固定在已知存储单元。这些固定的备用目标文件在运行时所需的处理要少于其对应的原始共享目标文件。

对于希望在中心位置安装第三方软件或者更改未用适当 `runpath` 编码的应用程序搜索路径的管理员而言，定义额外的缺省搜索路径或额外的可信目录非常有用。

声明备用目标文件提供了一种替换依赖项的方法，无需使用符号链接或 `LD_LIBRARY_PATH` 设置。

声明由运行时链接程序解释的环境变量提供了一种针对所有应用程序集中控制其定义的方法。

目录高速缓存和 `crle` 生成的备用目标文件可以提供一种减少应用程序的运行时启动开销的方法。对于需要许多依赖项或重新定位其依赖项的开销极高的应用程序而言，备用目标文件非常有用。包含 *position-dependent* 代码的共享目标文件若要重新定位，通常开销都比较大。请注意，系统具有许多有助于降低开销的高速缓存工具（如负路径查找），因此，利用 `crle` 创建目录高速缓存所产生的影响可能很小（一些非常特殊的情况除外）。

如果在配置文件中指定由 `crle` 生成的备用目标文件，运行时链接程序将执行一些最小一致性验证。备用目标文件将针对其原始目标文件进行验证。此验证意在避免由于应用程序配置信息与底层系统组件失去同步而导致的应用程序故障。当此情况出现时，可能会危及到动态链接系统组件提供的灵活性。此类型的应用程序故障可能非常难于诊断。不会验证目录高速缓存信息。进程看不到对目录结构所做的任何更改，直到重新构建高速缓存。

系统共享目标文件通常都经过了良好的调优，进行高速缓存所带来的好处并不是那么明显。目录高速缓存和备用目标文件功能通常适用于用户应用程序和共享目标文件，并且可能仅在某些非常特殊的情况下才会显示出情况的改善。

使用 `-I` 和 `-G` 选项时，`crle` 将使用 `dldump(3C)` 为所发现的共享目标文件创建备用目标文件。备用目标文件在前面的 `-o` 选项所指定的目录中创建，或缺省设置为创建配置文件时所在的目录。`dldump()` 所用的标志由 `-f` 选项指定，或缺省设置为 `RTLD_REL_RELATIVE`。

选项

支持以下选项。

`-64`

指定要处理 64 位目标文件，缺省值为 32 位。使用 `-64` 创建 64 位特定配置文件。

`-a name`

为 `name` 创建备用路径名称。将备用路径名称添加到配置文件。

实际备用文件必须由用户提供。允许此选项出现多次。如果 *name* 是一个目录，则将此目录中的所有共享目标文件添加到高速缓存。如果 *name* 不存在，则在高速缓存中将 *name* 标记为不存在的文件。

通常，此选项与 `-o` 选项结合使用。

`-A name`

为 *name* 创建可选备用路径名称。将此备用路径名称添加到配置文件。

此选项类似 `-a` 选项，只是如果在运行时备用目标文件不可用，将使用原始目标文件 *name*。此模型类似辅助过滤器的用法。请参见《[链接程序和库指南](#)》中的“[生成辅助过滤器](#)”。

通常，此选项与 `-o` 选项结合使用。

`-c conf`

指定要使用配置文件名 *conf*。如果未提供此选项，则使用缺省配置文件。

`-e env`

指定一个**可替换**的环境变量 *env*。只有适用于运行时链接程序的环境变量才有意义。允许此选项出现多次。此选项类似于 `-E` 选项。然而，对于在运行时解析配置文件定义以及同名的进程环境定义的方式，这两个选项有所不同。

配置文件中设置的定义可以由进程环境定义**覆盖**，或由空值进程环境定义**抑制**。

换言之，在运行时进程环境可以替换或删除这些配置文件定义。

`-E env`

指定一个**永久**的环境变量 *env*。只有适用于运行时链接程序的环境变量才有意义。允许此选项出现多次。此选项类似于 `-e` 选项。然而，对于在运行时解析配置文件定义以及同名的进程环境定义的方式，这两个选项有所不同。

对于运行时链接程序有意义的环境变量定义属于这两种类别之一。单一定义是指诸如 `LD_NOLAZYLOAD=1` 和 `LD_DEBUG_OUTPUT=file` 之类的定义。列表定义（可以有一个或多个值）是指诸如 `LD_LIBRARY_PATH=path` 和 `LD_DEBUG=files,details` 之类的定义。

在配置文件中设置的单一定义优先于进程环境定义。在配置文件中设置的列表定义**附加**到进程环境定义。空值进程环境定义**不能抑制**配置文件中设置的任何定义。

换言之，在运行时进程环境**不能**替换或删除这些配置文件定义。

`-f flags`

向用于生成备用目标文件的 `dldump(3C)` 调用提供符号 *flags* 参数。可以使用 `/usr/include/dlfcn.h` 中定义的任何 `RTLD_REL` 标志。可以使用 `|` 字符将多个标志连接起来（逻辑 `or` 关系）。在这种情况下，字符串应以引号括起来，以避免 shell 进行扩充。如果未提供 *flags* 值，则缺省标志为 `RTLD_REL_RELATIVE`。

-i name

向配置高速缓存添加单个 *name*。允许此选项出现多次。*name* 可以是一个共享目标文件或一个目录。如果 *name* 是一个目录，则将此目录中的所有共享目标文件添加到高速缓存。如果 *name* 不存在，则在高速缓存中将 *name* 标记为不存在的目录。

-I name

类似 **-i**，此外所处理的任何共享目标文件均使用 **dlldump(3C)** 创建一个备用目标文件。如果 **-f** 标志包含 **RTLD_REL_EXEC**，则 *name* 可以是将其创建备用目标文件的动态可执行文件。以此种方式只能指定一个动态可执行文件，因为创建的高速缓存特定于此应用程序。

-g name

向配置高速缓存添加组 *name*。每个目标文件都会展开以确定其依赖项。允许此选项出现多次。*name* 可以是动态可执行文件、共享目标文件或目录。如果 *name* 是共享目标文件，则将该共享目标文件及其依赖项添加到高速缓存。如果 *name* 是目录，则将该目录中的每个共享目标文件及其依赖项添加到高速缓存。

-G name

类似 **-g** 选项，此外所处理的任何共享目标文件均使用 **dlldump(3C)** 创建一个备用目标文件。如果 *name* 是动态可执行文件，而且 **-f** 标志包含 **RTLD_REL_EXEC**，则还会为此动态可执行文件创建一个备用目标文件。以此种方式只能指定一个动态可执行文件，因为创建的高速缓存特定于此应用程序。

-l dir

为 ELF 目标文件指定新的缺省搜索目录 *dir*。允许此选项出现多次。

32 位 ELF 目标文件的缺省搜索路径是 **/lib**，后跟 **/usr/lib**。64 位 ELF 目标文件的缺省搜索路径是 **/lib/64**，后跟 **/usr/lib/64**。

使用此选项将**替换**缺省搜索路径。因此，通常需要使用 **-l** 选项来指定与任何要应用的新路径相关的原始系统缺省值。但是，如果 **-u** 选项生效，而配置文件不存在，则系统缺省值会添加到新配置文件。这些缺省值在使用 **-l** 选项指定新路径之前进行添加。

-o dir

当与 **-a** 或 **-A** 选项结合使用时，指定任何备用目标文件所在的目录 *dir*。如果备用目标文件由 **crle** 创建，此选项将指定备用目标文件的创建位置。若不使用此选项，备用目标文件将位于创建配置文件的目录。允许此选项出现多次，目录 *dir* 用于为后面的任何命令行选项定位备用目标文件。备用目标文件不允许覆盖其关联的原始目标文件。

通常，此选项与 **-a** 或 **-A** 选项结合使用。

-s dir

为 *secure* ELF 目标文件指定新的可信目录 *dir*。允许此选项出现多次。

有关安全目标文件的定义，请参见 **ld.so.1(1)** 中的“安全”部分。有关安全应用程序的运行时限制的论述，请参见《**链接程序和库指南**》中的“安全性”。

32 位安全 ELF 目标文件的缺省可信目录是 `/lib/secure`，后跟 `/usr/lib/secure`。64 位安全 ELF 目标文件的缺省可信目录是 `/lib/secure/64`，后跟 `/usr/lib/secure/64`。

使用此选项将**替换**缺省可信目录。因此，通常需要使用 `-s` 选项来指定与任何要应用的新目录相关的原始系统缺省值。但是，如果 `-u` 选项生效，而配置文件不存在，则系统缺省值会添加到新配置文件。这些缺省值在使用 `-l` 选项指定新目录之前进行添加。

-t
已过时。请参见“附注”部分。

-u
请求更新配置文件，可能是添加新信息。若无其他选项，将检查任何现有配置文件并重新计算其内容。其他参数允许将信息附加到重新计算的内容。请参见“附注”部分。

如果配置文件不存在，则会按其他参数的指示创建配置文件。对于 `-l` 和 `-s` 选项，任何系统缺省值都在使用这些选项指定目录之前应用于配置文件。

配置文件的格式可以是缺少系统标识信息（通常写在文件开头处）的较旧格式。在这种情况下，`crle` 不会将系统标识信息放入所生成的文件中，以保留文件与旧版本 Solaris 的兼容性。请参见“附注”部分。

-v
指定详细模式。创建配置文件时，对于要处理的文件的跟踪会写入标准输出。在输出配置文件的内容时，会提供更详细的目录和文件信息。

缺省情况下，运行时链接程序会为每个处理的 32 位应用程序尝试读取配置文件 `/var/ld/ld.config`。对于每个 64 位应用程序，则读取 `/var/ld/64/ld.config`。在处理备用应用程序时，运行时链接程序将使用 `$ORIGIN/ld.config.app-name` 配置文件（如果存在）。请参见“附注”部分。应用程序可以通过设置 `LD_CONFIG` 环境变量引用备用配置文件。还可通过在构建应用程序时在应用程序中记录配置文件名来指定备用配置文件。请参见 [ld\(1\)](#) 的 `-c` 选项。

示例

示例1 试用临时配置文件

以下示例将使用 ELF 目标文件的新缺省搜索路径创建一个临时配置文件。环境变量 `LD_CONFIG_32` 用于指示运行时链接程序针对所有 32 位进程使用此配置文件。

```
$ crle -c /tmp/ld.config -u -l /local/lib
$ crle -c /tmp/ld.config
```

```
Configuration file [version 4]: /tmp/ld.config
Platform:      32-bit MSB SPARC
Default Library Path (ELF): /lib:/usr/lib:/local/lib
Trusted Directories (ELF): /lib/secure:/usr/lib/secure \
                          (system default)
```

示例1 试用临时配置文件 (续)

```
Command line:
  crle -c /tmp/ld.config -l /lib:/usr/lib:/local/lib
```

```
$ LD_CONFIG_32=/tmp/ld.config date
Thu May 29 17:42:00 PDT 2008
```

示例2 为ELF目标文件更新并显示新的缺省搜索路径
以下示例将更新并显示ELF目标文件的新缺省搜索路径。

```
# crle -u -l /local/lib
# crle
```

```
Configuration file [version 4]: /var/ld/ld.config
Platform:      32-bit MSB SPARC
Default Library Path (ELF): /lib:/usr/lib:/local/lib
Trusted Directories (ELF): /lib/secure:/usr/lib/secure \
                          (system default)
```

```
Command line:
  crle -l /lib:/usr/lib:/local/lib
```

```
# crle -u -l /ISV/lib
# crle
```

```
Configuration file [version 4]: /var/ld/ld.config
Platform      32-bit MSB SPARC
Default Library Path (ELF): /lib:/usr/lib:/local/lib:/ISV/lib
Trusted Directories (ELF): /lib/secure:/usr/lib/secure \
                          (system default)
```

```
Command line:
  crle -l /lib:/usr/lib:/local/lib:/usr/local/lib
```

在本示例中，缺省配置文件最初并不存在。因此，新的搜索路径 `/local/lib` 附加到系统缺省值。下一次更新将搜索路径 `/ISV/lib` 附加到已在配置文件中设置的那些路径。

示例3 从错误配置文件进行恢复

以下示例将在缺省位置创建一个错误配置文件。可通过使用 `LD_NOCONFIG` 环境变量指示运行时链接程序忽略任何配置文件来删除该文件。请注意，建议创建临时配置文件，并使用环境变量 `LD_CONFIG` 试用这些文件。

```
# crle -l /local/lib
# date
ld.so.1: date: fatal: libc.so.1: open failed: \
  No such file or directory
```

```

Killed
# LD_NOCONFIG=yes rm /var/ld/ld.config
# date
Thu May 29 17:52:00 PDT 2008

```

此配置文件不包含系统缺省搜索路径，因此，`date` 实用程序无法找到所需的系统依赖项。在这种情况下，应使用 `-u` 选项。

示例 4 为 ELF 目标文件新建缺省搜索路径和可信目录并进行显示
以下示例将为 ELF 目标文件新建缺省搜索路径和可信目录并进行显示。

```

# crle -l /local/lib -l /lib -l /usr/lib -s /local/lib
# crle

```

```

Configuration file [version 4]: /var/ld/ld.config
Platform:      32-bit MSB SPARC
Default Library Path (ELF): /local/lib:/lib:/usr/lib
Trusted Directories (ELF): /local/lib

```

```

Command line:
crle -l /local/lib:/lib:/usr/lib -s /local/lib

```

通过此配置文件，第三方应用程序可以安装在 `/local/bin` 中，其依赖项可安装在 `/local/lib` 中。缺省搜索路径允许应用程序无需设置 `LD_LIBRARY_PATH` 即可定位其依赖项。本示例中缺省可信目录也进行了替换。

示例 5 为 ELF 目标文件创建目录高速缓存
以下示例将为 ELF 目标文件创建目录高速缓存。

```

$ crle -i /usr/dt/lib -i /usr/openwin/lib -i /lib -i /usr/lib \
-c config
$ ldd -s ./main
....
find object=libc.so.1; required by ./main
search path=/usr/dt/lib:/usr/openwin/lib (RUNPATH/RPATH ./main)
trying path=/usr/dt/lib/libc.so.1
trying path=/usr/openwin/lib/libc.so.1
search path=/lib (default)
trying path=/lib/libc.so.1
libc.so.1 => /lib/libc.so.1

$ LD_CONFIG=config ldd -s ./main
....
find object=libc.so.1; required by ./main
search path=/usr/dt/lib:/usr/openwin/lib (RUNPATH/RPATH ./main)
search path=/lib (default)
trying path=/lib/libc.so.1
libc.so.1 => /lib/libc.so.1

```

示例 5 为 ELF 目标文件创建目录高速缓存 (续)

通过此配置，高速缓存将反映出系统库 `libc.so.1` 不存在于目录 `/usr/dt/lib` 或 `/usr/openwin/lib` 中。因此，针对此系统文件的搜索将忽略这些目录，即使应用程序的 `runpath` 指示应搜索这些路径。

示例 6 为 ELF 可执行文件创建备用目标文件高速缓存

以下示例将为 ELF 可执行文件创建备用目标文件高速缓存。

```
$ crle -c /local/$HOST/.xterm/ld.config.xterm \
      -f RTLD_REL_ALL -G /usr/openwin/bin/xterm
$ ln -s /local/$HOST/.xterm/xterm /local/$HOST/xterm
$ ldd /usr/local/$HOST/xterm
    libXaw.so.5 => /local/$HOST/.xterm/libWaw.so.5 (alternate)
    libXmu.so.4 => /local/$HOST/.xterm/libXmu.so.4 (alternate)
    ....
    libc.so.1 => /local/$HOST/.xterm/libc.so.1 (alternate)
    ....
```

通过此配置，将创建新的 `xterm` 及其依赖项。这些新目标文件相互进行完全重定位，与原始目标文件相比，加快了启动速度。此应用程序的执行使用其自己特定的配置文件。通常，此模型比使用环境变量 `LD_CONFIG` 更为灵活，因为配置文件不可能由其他应用程序（如 `ldd(1)` 或 `truss(1)`）错误使用。

示例 7 创建备用目标文件高速缓存以替换 ELF 共享目标文件

以下示例将创建备用目标文件高速缓存以替换 ELF 共享目标文件。

```
$ ldd /usr/bin/vi
    libcurses.so.1 => /lib/libcurses.so.1
    ....

# crle -a /lib/libcurses.so.1 -o /usr/ucblib
# crle

Configuration file [version 4]: /var/ld/ld.config
Platform:      32-bit MSB SPARC
Default Library Path (ELF): /lib:/usr/lib (system default)
Trusted Directories (ELF): /lib/secure:/usr/lib/secure \
                          (system default)

Directory: /lib
    libcurses.so.1 (alternate: /usr/ucblib/libcurses.so.1)
    ....

$ ldd /usr/bin/vi
    libcurses.so.1 => /usr/ucblib/libcurses.so.1 (alternate)
    ....
```

示例7 创建备用目标文件高速缓存以替换ELF共享目标文件 (续)

通过此配置，任何通常解析至 `/usr/lib/libcurses.so.1` 的依赖项改为解析至 `/usr/ucblib/libcurses.so.1`。

示例8 设置可替换环境变量和永久环境变量

以下示例将设置可替换环境变量和永久环境变量。

```
# crle -e LD_LIBRARY_PATH=/local/lib \
      -E LD_PRELOAD=preload.so.1
# crle
.....
Environment Variables:
  LD_LIBRARY_PATH=/local/lib (replaceable)
  LD_PRELOAD=preload.so.1 (permanent)
.....
$ LD_DEBUG=files LD_PRELOAD=preload.so.2 ./main
.....
18764: file=preload.so.2; preloaded
18764: file=/local/lib/preload.so.2 [ ELF ]; generating link map
.....
18764: file=preload.so.1; preloaded
18764: file=/local/lib/preload.so.1 [ ELF ]; generating link map
.....
```

通过此配置文件，可替换搜索路径将与附加到进程环境定义的永久预装入目标文件一起进行指定。

退出状态

创建或显示配置文件将导致返回 0。否则，任何错误情形都会随附一条诊断消息并返回非零值。

附注

如果原始应用程序包含 *dynamic* 标记 `DT_FLAGS_1` 或 `DT_FEATURE_1` 之一，则可标记备用应用程序以使用应用程序特定的配置文件。如果没有这些条目，必须使用 `LD_CONFIG` 环境变量指定配置文件。使用后一种方法时应格外小心，因为此环境变量对任何分支应用程序均可见。

使用 `-u` 选项至少需要 `crle` 版本 2。从显示配置文件内容的角度来看，很显然需要此版本级别。

```
$ crle
Configuration file [2]: /var/ld/ld.config
.....
```

使用版本 2 配置文件，`crle` 能够构造重新生成配置文件所需的命令行参数。此命令行构造使用 `-u` 选项提供完整的更新功能。尽管版本 1 配置文件更新也是可以的，但配置文件内容可能对于 `crle` 计算整个更新需求而言并不够用。

配置文件包含平台特定的二进制数据。给定的配置文件只能由具有相同机器类和字节排序的软件进行解释。配置文件的开头是系统标识信息。此信息由 `crle` 和运行时使用以检查其与配置文件的兼容性。此信息还允许 `file(1)` 命令正确识别配置文件。为了保持向后兼容性，缺少此信息的较旧文件仍是可以接受的，尽管不具备标识和错误检查（具有这些检查也是可以的）。在针对缺少系统信息的较旧文件处理更新 (`-u`) 操作时，`crle` 不会向结果添加系统标识信息。

Oracle Solaris 11 在 SPARC 硬件上不再继续支持 SunOS 4.x AOUT 可执行文件。在做出此更改之前，`-t` 选项提供了一种表示目标文件类型（ELF 或 AOUT）的切换方式，该选项影响后面的任何 `-l` 或 `-s` 选项。`-t` 选项现在已过时，任何 AOUT 特定指令均将忽略，并给出一个警告说明此影响。同样，使用 `crle` 检查包含 AOUT 信息的旧配置文件也将生成一个警告消息，指出信息已过时。

文件

`/var/ld/ld.config`
32 位应用程序的缺省配置文件。

`/var/ld/64/ld.config`
64 位应用程序的缺省配置文件。

`/var/tmp`
临时配置文件的缺省位置。请参见 `tempnam(3C)`。

`/usr/lib/lddstub`
用于 `dldump(3C)` 32 位目标文件的桩应用程序。

`/usr/lib/64/lddstub`
用于 `dldump(3C)` 64 位目标文件的桩应用程序。

`/usr/lib/libcrle.so.1`
用于 `dldump(3C)` 32 位目标文件的审计库。

`/usr/lib/64/libcrle.so.1`
用于 `dldump(3C)` 64 位目标文件的审计库。

环境变量

没有 `crle` 引用的环境变量。不过，在处理由 `crle` 创建的配置文件时，有几个环境变量会影响运行时链接程序的行为。

`LD_CONFIG`、`LD_CONFIG_32` 和 `LD_CONFIG_64`
提供备用配置文件。

`LD_NOCONFIG`、`LD_NOCONFIG_32` 和 `LD_NOCONFIG_64`
禁用配置文件处理。

`LD_NODIRCONFIG`、`LD_NODIRCONFIG_32` 和 `LD_NODIRCONFIG_64`
从配置文件禁用目录高速缓存处理。

`LD_NOENVCONFIG`、`LD_NOENVCONFIG_32` 和 `LD_NOENVCONFIG_64`
从配置文件禁用环境变量处理。

`LD_NOOBJALTER`、`LD_NOOBJALTER_32` 和 `LD_NOOBJALTER_64`
从配置文件禁用备用目标文件处理。

属性

有关下列属性的描述，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/linker
接口稳定性	Committed（已确定）

另请参见

[file\(1\)](#)、[ld\(1\)](#)、[ld.so.1\(1\)](#)、[dldump\(3C\)](#)、[tempnam\(3C\)](#)、[attributes\(5\)](#)

《链接程序和库指南》

引用名 crontab – user crontab file

用法概要

```
/usr/bin/crontab [filename]
/usr/bin/crontab -e [username]
/usr/bin/crontab -l [username]
/usr/bin/crontab -r [username]
/usr/xpg4/bin/crontab [filename]
/usr/xpg4/bin/crontab -e [username]
/usr/xpg4/bin/crontab -l [username]
/usr/xpg4/bin/crontab -r [username]
/usr/xpg6/bin/crontab [filename]
/usr/xpg6/bin/crontab -e [username]
/usr/xpg6/bin/crontab -l [username]
/usr/xpg6/bin/crontab -r [username]
```

描述

The crontab utility manages a user's access with cron (see [cron\(1M\)](#)) by copying, creating, listing, and removing crontab files. If invoked without options, crontab copies the specified file, or the standard input if no file is specified, into a directory that holds all users' crontabs.

If crontab is invoked with *filename*, this overwrites an existing crontab entry for the user that invokes it.

crontab Access Control

Users: Access to crontab is allowed:

- if the user's name appears in `/etc/cron.d/cron.allow`.
- if `/etc/cron.d/cron.allow` does not exist and the user's name is not in `/etc/cron.d/cron.deny`.

Users: Access to crontab is denied:

- if `/etc/cron.d/cron.allow` exists and the user's name is not in it.
- if `/etc/cron.d/cron.allow` does not exist and user's name is in `/etc/cron.d/cron.deny`.
- if neither file exists, only a user with the `solaris.jobs.user` authorization is allowed to submit a job.
- if Solaris Auditing is enabled, the user's shell is not audited and the user is not the crontab owner. This can occur if the user logs in by way of a program, such as some versions of SSH, which does not set audit parameters.

The rules for `allow` and `deny` apply to root only if the `allow/deny` files exist.

The `allow/deny` files consist of one user name per line.

crontab Entry Format A crontab file consists of lines of six fields each. The fields are separated by spaces or tabs. The first five are integer patterns that specify the following:

```
minute (0-59),
hour (0-23),
day of the month (1-31),
month of the year (1-12),
day of the week (0-6 with 0=Sunday).
```

Each of these patterns can be either an asterisk (meaning all legal values) or a list of elements separated by commas. An element is either a number or two numbers separated by a minus sign (meaning an inclusive range). Time specified here is interpreted in the currently active timezone. At the top of the crontab file this is the timezone which is set system-wide in `/etc/default/init`. A user can add a line such as:

```
TZ=timezone
```

...and all subsequent entries will be interpreted using that timezone, until a new `TZ=timezone` line is encountered. The specification of days can be made by two fields (day of the month and day of the week). Both are adhered to if specified as a list of elements. See EXAMPLES.

The sixth field of a line in a crontab file is a string that is executed by the shell at the specified times. A percent character in this field (unless escaped by `\`) is translated to a NEWLINE character.

Only the first line (up to a `' % '` or end of line) of the command field is executed by the shell. Other lines are made available to the command as standard input. Any blank line or line beginning with a `' # '` is a comment and is ignored.

The shell is invoked from your `$HOME` directory. As with `$TZ`, both `$SHELL` and `$HOME` can be set by having a line such as:

```
SHELL=/usr/bin/someshell
```

...OR:

```
HOME=somedirectory
```

...which will take precedence for all the remaining entries in the crontab or until there is another HOME or SHELL entry. It is invoked with an `arg0` of the basename of the `$SHELL` that is currently in effect. A user who wants to have his `.profile` or equivalent file executed must explicitly do so in the crontab file. `cron` supplies a default environment for every shell, defining HOME, LOGNAME, SHELL, TZ, and PATH. The default PATH for user cron jobs is `/usr/bin`; while root cron jobs default to `/usr/sbin:/usr/bin`. The default PATH can be set in `/etc/default/cron` (see [cron\(1M\)](#)). The TZ, HOME, and SHELL environment variables are set to match those that are in effect in the crontab file at the time.

If you do not redirect the standard output and standard error of your commands, any generated output or errors are mailed to you.

crontab Environment Variables

The following variables are supported:

HOME

Allows the user to choose an alternative directory for cron to change directory to prior to running the command. For example:

```
HOME=/var/tmp
```

SHELL

The name of the shell to use to run subsequent commands. For example:

```
SHELL=/usr/bin/ksh
```

TZ

Allows the user to choose the timezone in which the cron entries are run. This affects both the environment of the command that is run and the timing of the entry. For example, to have your entries run using the timezone for Iceland, use:

```
TZ=Iceland
```

Each of these variables affects all of the lines that follow it in the crontab file, until it is reset by a subsequent line resetting that variable. Hence, it is possible to have multiple timezones supported within a single crontab file.

The lines that are not setting these environment variables are the same as crontab entries that conform to the UNIX standard and are described elsewhere in this man page.

Setting cron Jobs Across Timezones

The default timezone of the cron daemon sets the system-wide timezone for cron entries. This, in turn, is by set by default system-wide using `/etc/default/init`.

If some form of *daylight savings* or *summer/winter time* is in effect, then jobs scheduled during the switchover period could be executed once, twice, or not at all.

选项

The following options are supported:

- e Edits a copy of the current user's crontab file, or creates an empty file to edit if crontab does not exist. When editing is complete, the file is installed as the user's crontab file.

The environment variable EDITOR determines which editor is invoked with the -e option. All crontab jobs should be submitted using crontab. Do not add jobs by just editing the crontab file, because cron is not aware of changes made this way.

If all lines in the crontab file are deleted, the old crontab file is restored. The correct way to delete all lines is to remove the crontab file using the -r option.

If *username* is specified, the specified user's crontab file is edited, rather than the current user's crontab file. This can only be done by root or by a user with the `solaris.jobs.admin` authorization.

- l Lists the crontab file for the invoking user. Only root or a user with the `solaris.jobs.admin` authorization can specify a username following the `-l` option to list the crontab file of the specified user.
- r Removes a user's crontab from the crontab directory. Only root or a user with the `solaris.jobs.admin` authorization can specify a username following the `-r` option to remove the crontab file of the specified user.

示例

示例 1 Cleaning up Core Files

This example cleans up core files every weekday morning at 3:15 am:

```
15 3 * * 1-5 find $HOME -name core 2>/dev/null | xargs rm -f
```

示例 2 Mailing a Birthday Greeting

This example mails a birthday greeting:

```
0 12 14 2 * mailx john%Happy Birthday!%Time for lunch.
```

示例 3 Specifying Days of the Month and Week

This example runs a command on the first and fifteenth of each month, as well as on every Monday:

```
0 0 1,15 * 1
```

To specify days by only one field, the other field should be set to `*`. For example:

```
0 0 * * 1
```

would run a command only on Mondays.

示例 4 Using Environment Variables

The following entries take advantage of crontab support for certain environment variables.

```
TZ=GMT
HOME=/local/home/user
SHELL=/usr/bin/ksh
0 0 * * * echo $(date) > midnight.GMT
TZ=US/Pacific
0 0 * * * echo $(date) > midnight.PST
TZ=US/Eastern
HOME=/local/home/myuser
SHELL=/bin/csh
```

The preceding entries allow two jobs to run. The first one would run at midnight in the GMT timezone and the second would run at midnight in the PST timezone. Both would be run in the directory `/local/home/user` using the Korn shell. The file concludes with `TZ`, `HOME`, and `SHELL` entries that return those variable to their default values.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of crontab: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLS_PATH.

<code>/usr/bin/crontab</code>	EDITOR	Determine the editor to be invoked when the <code>-e</code> option is specified. This is overridden by the VISUAL environmental variable. The default editor is vi(1) .
	PATH	The PATH in crontab's environment specifies the search path used to find the editor.
	VISUAL	Determine the visual editor to be invoked when the <code>-e</code> option is specified. If VISUAL is not specified, then the environment variable EDITOR is used. If that is not set, the default is vi(1) .
<code>/usr/xpg4/bin/crontab</code>	EDITOR	Determine the editor to be invoked when the <code>-e</code> option is specified. The default editor is <code>/usr/xpg4/bin/vi</code> .
<code>/usr/xpg6/bin/crontab</code>	EDITOR	Determine the editor to be invoked when the <code>-e</code> option is specified. The default editor is <code>/usr/xpg6/bin/vi</code> .

退出状态

The following exit values are returned:

- `0` Successful completion.
- `>0` An error occurred.

文件

<code>/etc/cron.d</code>	main cron directory
<code>/etc/cron.d/cron.allow</code>	list of allowed users
<code>/etc/default/cron</code>	contains cron default settings
<code>/etc/cron.d/cron.deny</code>	list of denied users
<code>/var/cron/log</code>	accounting information
<code>/var/spool/cron/crontabs</code>	spool area for crontab

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/crontab`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

`/usr/xpg4/bin/crontab`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Standard

/usr/xpg6/bin/crontab

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu6
Interface Stability	Standard

另请参见

[atq\(1\)](#), [atrm\(1\)](#), [auths\(1\)](#), [ed\(1\)](#), [sh\(1\)](#), [vi\(1\)](#), [cron\(1M\)](#), [su\(1M\)](#), [auth_attr\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注

If you inadvertently enter the `crontab` command with no arguments, do not attempt to get out with Control-d. This removes all entries in your `crontab` file. Instead, exit with Control-c.

When updating `cron`, check first for existing `crontab` entries that can be scheduled close to the time of the update. Such entries can be lost if the update process completes after the scheduled event. This can happen because, when `cron` is notified by `crontab` to update the internal view of a user's `crontab` file, it first removes the user's existing internal `crontab` and any internal scheduled events. Then it reads the new `crontab` file and rebuilds the internal `crontab` and events. This last step takes time, especially with a large `crontab` file, and can complete *after* an existing `crontab` entry is scheduled to run if it is scheduled too close to the update. To be safe, start a new job at least 60 seconds after the current date and time.

Simultaneous modifications of the same `crontab` file may lead to unexpected results.

Care should be taken when adding `TZ`, `SHELL` and `HOME` variables to the `crontab` file when the `crontab` file could be shared with applications that do not expect those variables to be changed from the default. Resetting the values to their defaults at the bottom of the file will minimize the risk of problems.

引用名	csh – shell command interpreter with a C-like syntax
用法概要	csh [-bcefinstvVxX] [<i>argument</i>]...
描述	<p>csh, the C shell, is a command interpreter with a syntax reminiscent of the C language. It provides a number of convenient features for interactive use that are not available with the Bourne shell, including filename completion, command aliasing, history substitution, job control, and a number of built-in commands. As with the Bourne shell, the C shell provides variable, command and filename substitution.</p>
Initialization and Termination	<p>When first started, the C shell normally performs commands from the <code>.cshrc</code> file in your home directory, provided that it is readable and you either own it or your real group ID matches its group ID. If the shell is invoked with a name that starts with '-', as when started by login(1), the shell runs as a login shell.</p> <p>If the shell is a login shell, this is the sequence of invocations: First, commands in <code>/etc/.login</code> are executed. Next, commands from the <code>.cshrc</code> file your home directory are executed. Then the shell executes commands from the <code>.login</code> file in your home directory; the same permission checks as those for <code>.cshrc</code> are applied to this file. Typically, the <code>.login</code> file contains commands to specify the terminal type and environment. (For an explanation of file interpreters, see Command Execution and exec(2).)</p> <p>As a login shell terminates, it performs commands from the <code>.logout</code> file in your home directory; the same permission checks as those for <code>.cshrc</code> are applied to this file.</p>
Interactive Operation	<p>After startup processing is complete, an interactive C shell begins reading commands from the terminal, prompting with <code>hostname%</code> (or <code>hostname#</code> for the privileged user). The shell then repeatedly performs the following actions: a line of command input is read and broken into <i>words</i>. This sequence of words is placed on the history list and then parsed, as described under USAGE. Finally, the shell executes each command in the current line.</p>
Noninteractive Operation	<p>When running noninteractively, the shell does not prompt for input from the terminal. A noninteractive C shell can execute a command supplied as an <i>argument</i> on its command line, or interpret commands from a file, also known as a script.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none">-b Forced a “break” from option processing. Subsequent command line arguments are not interpreted as C shell options. This allows the passing of options to a script without confusion. The shell does not run <code>set-user-ID</code> or <code>set-group-ID</code> scripts unless this option is present.-c Executes the first <i>argument</i>, which must be present. Remaining arguments are placed in <code>argv</code>, the argument-list variable, and passed directly to <code>csh</code>.-e Exits if a command terminates abnormally or yields a nonzero exit status.-f Fast start. Reads neither the <code>.cshrc</code> file, nor the <code>.login</code> file (if a login shell) upon startup.

- i Forced interactive. Prompts for command line input, even if the standard input does not appear to be a terminal (character-special device).
- n Parses (interprets), but does not execute commands. This option can be used to check C shell scripts for syntax errors.
- s Takes commands from the standard input.
- t Reads and executes a single command line. A `\` (backslash) can be used to escape each newline for continuation of the command line onto subsequent input lines.
- v Verbose. Sets the verbose predefined variable. Command input is echoed after history substitution, but before other substitutions and before execution.
- V Sets verbose before reading `.cshrc`.
- x Echo. Sets the echo variable. Echoes commands after all substitutions and just before execution.
- X Sets echo before reading `.cshrc`.

Except with the options `-c`, `-i`, `-s`, or `-t`, the first nonoption *argument* is taken to be the name of a command or script. It is passed as argument zero, and subsequent arguments are added to the argument list for that command or script.

用法

Filename Completion When enabled by setting the variable `filec`, an interactive C shell can complete a partially typed filename or user name. When an unambiguous partial filename is followed by an ESC character on the terminal input line, the shell fills in the remaining characters of a matching filename from the working directory.

If a partial filename is followed by the EOF character (usually typed as Control-d), the shell lists all filenames that match. It then prompts once again, supplying the incomplete command line typed in so far.

When the last (partial) word begins with a tilde (`~`), the shell attempts completion with a user name, rather than a file in the working directory.

The terminal bell signals errors or multiple matches. This bell signal can be inhibited by setting the variable `nobeep`. You can exclude files with certain suffixes by listing those suffixes in the variable `ignore`. If, however, the only possible completion includes a suffix in the list, it is not ignored. `ignore` does not affect the listing of filenames by the EOF character.

Lexical Structure The shell splits input lines into words at space and tab characters, except as noted below. The characters `&`, `|`, `;`, `<`, `>`, `(`, and `)` form separate words; if paired, the pairs form single words. These shell metacharacters can be made part of other words, and their special meaning can be suppressed by preceding them with a `\` (backslash). A newline preceded by a `\` is equivalent to a space character.

In addition, a string enclosed in matched pairs of single-quotes ('), double-quotes ("), or backquotes (`), forms a partial word. Metacharacters in such a string, including any space or tab characters, do not form separate words. Within pairs of backquote (`) or double-quote (") characters, a newline preceded by a `\ `(backslash) gives a true newline character. Additional functions of each type of quote are described, below, under Variable Substitution, Command Substitution, and Filename Substitution.`

When the shell's input is not a terminal, the character # introduces a comment that continues to the end of the input line. Its special meaning is suppressed when preceded by a \ `or enclosed in matching quotes.`

Command Line Parsing A *simple command* is composed of a sequence of words. The first word (that is not part of an I/O redirection) specifies the command to be executed. A simple command, or a set of simple commands separated by | or |& characters, forms a *pipeline*. With |, the standard output of the preceding command is redirected to the standard input of the command that follows. With | &, both the standard error and the standard output are redirected through the pipeline.

Pipelines can be separated by semicolons (;), in which case they are executed sequentially. Pipelines that are separated by && or | | form conditional sequences in which the execution of pipelines on the right depends upon the success or failure, respectively, of the pipeline on the left.

A pipeline or sequence can be enclosed within parentheses `()' to form a simple command that can be a component in a pipeline or sequence.

A sequence of pipelines can be executed asynchronously or "in the background" by appending an `&'; rather than waiting for the sequence to finish before issuing a prompt, the shell displays the job number (see `Job Control`, below) and associated process IDs and prompts immediately.

History Substitution History substitution allows you to use words from previous command lines in the command line you are typing. This simplifies spelling corrections and the repetition of complicated commands or arguments. Command lines are saved in the history list, the size of which is controlled by the `history` variable. The most recent command is retained in any case. A history substitution begins with a ! (although you can change this with the `histchars` variable) and occurs anywhere on the command line; history substitutions do not nest. The ! can be escaped with \ `to suppress its special meaning.`

Input lines containing history substitutions are echoed on the terminal after being expanded, but before any other substitutions take place or the command gets executed.

Event Designators An event designator is a reference to a command line entry in the history list.

! Start a history substitution, except when followed by a space character, tab, newline, = or (.

!!	Refer to the previous command. By itself, this substitution repeats the previous command.
!n	Refer to command line <i>n</i> .
!-n	Refer to the current command line minus <i>n</i> .
!str	Refer to the most recent command starting with <i>str</i> .
!?str?	Refer to the most recent command containing <i>str</i> .
!?str? <i>additional</i>	Refer to the most recent command containing <i>str</i> and append <i>additional</i> to that referenced command.
!{ <i>command</i> } <i>additional</i>	Refer to the most recent command beginning with <i>command</i> and append <i>additional</i> to that referenced command.
^ <i>previous_word</i> ^ <i>replacement</i> ^	Repeat the previous command line replacing the string <i>previous_word</i> with the string <i>replacement</i> . This is equivalent to the history substitution: ! : s/ <i>previous_word</i> / <i>replacement</i> / . To re-execute a specific previous command AND make such a substitution, say, re-executing command #6, ! : 6s/ <i>previous_word</i> / <i>replacement</i> / .

Word Designators

A ':' (colon) separates the event specification from the word designator. It can be omitted if the word designator begins with a ^, \$, *, - or %. If the word is to be selected from the previous command, the second ! character can be omitted from the event specification. For instance, !! : 1 and ! : 1 both refer to the first word of the previous command, while !! \$ and ! \$ both refer to the last word in the previous command. Word designators include:

#	The entire command line typed so far.
0	The first input word (command).
n	The <i>n</i> 'th argument.
^	The first argument, that is, 1.
\$	The last argument.
%	The word matched by the ?s search.
x-y	A range of words; -y abbreviates 0-y.
*	All the arguments, or a null value if there is just one word in the event.
x*	Abbreviates x-\$.
x-	Like x* but omitting word \$.

Modifiers	After the optional word designator, you can add one of the following modifiers, preceded by a <code>:</code> .
<code>h</code>	Remove a trailing pathname component, leaving the head.
<code>r</code>	Remove a trailing suffix of the form <code>' .xxx'</code> , leaving the basename.
<code>e</code>	Remove all but the suffix, leaving the Extension.
<code>s/l/r/</code>	Substitute <code>r</code> for <code>l</code> .
<code>t</code>	Remove all leading pathname components, leaving the tail.
<code>&</code>	Repeat the previous substitution.
<code>g</code>	Apply the change to the first occurrence of a match in each word, by prefixing the above (for example, <code>g&</code>).
<code>p</code>	Print the new command but do not execute it.
<code>q</code>	Quote the substituted words, escaping further substitutions.
<code>x</code>	Like <code>q</code> , but break into words at each space character, tab or newline.

Unless preceded by a `g`, the modification is applied only to the first string that matches `l`; an error results if no string matches.

The left-hand side of substitutions are not regular expressions, but character strings. Any character can be used as the delimiter in place of `/`. A backslash quotes the delimiter character. The character `&`, in the right hand side, is replaced by the text from the left-hand-side. The `&` can be quoted with a backslash. A null `l` uses the previous string either from a `l` or from a contextual scan string `s` from `! ?s`. You can omit the rightmost delimiter if a newline immediately follows `r`; the rightmost `?` in a context scan can similarly be omitted.

Without an event specification, a history reference refers either to the previous command, or to a previous history reference on the command line (if any).

Quick Substitution	<code>^l^r^</code> This is equivalent to the history substitution: <code>! :s/l/r/.</code>
--------------------	---

Aliases	The C shell maintains a list of aliases that you can create, display, and modify using the <code>alias</code> and <code>unalias</code> commands. The shell checks the first word in each command to see if it matches the name of an existing alias. If it does, the command is reprocessed with the alias definition replacing its name; the history substitution mechanism is made available as though that command were the previous input line. This allows history substitutions, escaped with a backslash in the definition, to be replaced with actual command line arguments when the alias is used. If no history substitution is called for, the arguments remain unchanged.
---------	--

Aliases can be nested. That is, an alias definition can contain the name of another alias. Nested aliases are expanded before any history substitutions is applied. This is useful in pipelines such as

```
alias lm 'ls -l \!* | more'
```

which when called, pipes the output of `ls(1)` through `more(1)`.

Except for the first word, the name of the alias can not appear in its definition, nor in any alias referred to by its definition. Such loops are detected, and cause an error message.

I/O Redirection

The following metacharacters indicate that the subsequent word is the name of a file to which the command's standard input, standard output, or standard error is redirected; this word is variable, command, and filename expanded separately from the rest of the command.

< Redirect the standard input.

< < *word* Read the standard input, up to a line that is identical with *word*, and place the resulting lines in a temporary file. Unless *word* is escaped or quoted, variable and command substitutions are performed on these lines. Then, the pipeline is invoked with the temporary file as its standard input. *word* is not subjected to variable, filename, or command substitution, and each line is compared to it before any substitutions are performed by the shell.

>>! >&>&! Redirect the standard output to a file. If the file does not exist, it is created. If it does exist, it is overwritten; its previous contents are lost.

When set, the variable `noclobber` prevents destruction of existing files. It also prevents redirection to terminals and `/dev/null`, unless one of the `!` forms is used. The `&` forms redirect both standard output and the standard error (diagnostic output) to the file.

> >> >&> >! > >&! Append the standard output. Like `>`, but places output at the end of the file rather than overwriting it. If `noclobber` is set, it is an error for the file not to exist, unless one of the `!` forms is used. The `&` forms append both the standard error and standard output to the file.

Variable Substitution

The C shell maintains a set of variables, each of which is composed of a *name* and a *value*. A variable name consists of up to 128 letters and digits, and starts with a letter. An underscore (`_`) is considered a letter). A variable's value is a space-separated list of zero or more words. If the shell supports a variable name upto 128 characters the variable `SUNW_VARLEN` is defined. If a variable name of up to 128 characters is not supported, then an older version of the shell is being used, and the shell variable name length has a maximum length of 20.

To refer to a variable's value, precede its name with a `'$'`. Certain references (described below) can be used to select specific words from the value, or to display other information about the variable. Braces can be used to insulate the reference from other characters in an input-line word.

Variable substitution takes place after the input line is analyzed, aliases are resolved, and I/O redirections are applied. Exceptions to this are variable references in I/O redirections (substituted at the time the redirection is made), and backquoted strings (see Command Substitution).

Variable substitution can be suppressed by preceding the `$` with a `\`, except within double-quotes where it always occurs. Variable substitution is suppressed inside of single-quotes. A `$` is escaped if followed by a space character, tab or newline.

Variables can be created, displayed, or destroyed using the `set` and `unset` commands. Some variables are maintained or used by the shell. For instance, the `argv` variable contains an image of the shell's argument list. Of the variables used by the shell, a number are toggles; the shell does not care what their value is, only whether they are set or not.

Numerical values can be operated on as numbers (as with the `@built-in` command). With numeric operations, an empty value is considered to be zero. The second and subsequent words of multiword values are ignored. For instance, when the `verbose` variable is set to any value (including an empty value), command input is echoed on the terminal.

Command and filename substitution is subsequently applied to the words that result from the variable substitution, except when suppressed by double-quotes, when `noglob` is set (suppressing filename substitution), or when the reference is quoted with the `:q` modifier. Within double-quotes, a reference is expanded to form (a portion of) a quoted string; multiword values are expanded to a string with embedded space characters. When the `:q` modifier is applied to the reference, it is expanded to a list of space-separated words, each of which is quoted to prevent subsequent command or filename substitutions.

Except as noted below, it is an error to refer to a variable that is not set.

`$var`

`${var}` These are replaced by words from the value of *var*, each separated by a space character. If *var* is an environment variable, its value is returned (but `'`:`'` modifiers and the other forms given below are not available).

`$var[index]`

`${var[index]}` These select only the indicated words from the value of *var*. Variable substitution is applied to *index*, which can consist of (or result in) a either single number, two numbers separated by a `-`, or an asterisk. Words are indexed starting from 1; a `*` selects all words. If the first number of a range is omitted (as with `$argv[-2]`), it defaults to 1. If the last number of a range is omitted (as with `$argv[1-]`), it defaults to `$#var` (the word count). It is not an error for a range to be empty if the second argument is omitted (or within range).

`$#name`

`${#name}` These give the number of words in the variable.

`$0` This substitutes the name of the file from which command input is being read except for setuid shell scripts. An error occurs if the name is not known.

`$n`

`${n}` Equivalent to `$argv[n]`.

`$*` Equivalent to `$argv[*]`.

The modifiers `:e`, `:h`, `:q`, `:r`, `:t`, and `:x` can be applied (see [History Substitution](#)), as can `:gh`, `:gt`, and `:gr`. If `{ }` (braces) are used, then the modifiers must appear within the braces. The current implementation allows only one such modifier per expansion.

The following references can not be modified with `:` modifiers.

`$?var`

`${?var}` Substitutes the string 1 if `var` is set or 0 if it is not set.

`$?0` Substitutes 1 if the current input filename is known or 0 if it is not.

`$$` Substitutes the process number of the (parent) shell.

`$<` Substitutes a line from the standard input, with no further interpretation thereafter. It can be used to read from the keyboard in a C shell script.

Command and Filename Substitutions

Command and filename substitutions are applied selectively to the arguments of built-in commands. Portions of expressions that are not evaluated are not expanded. For non-built-in commands, filename expansion of the command name is done separately from that of the argument list; expansion occurs in a subshell, after I/O redirection is performed.

Command Substitution

A command enclosed by backquotes (`' . . . '`) is performed by a subshell. Its standard output is broken into separate words at each space character, tab and newline; null words are discarded. This text replaces the backquoted string on the current command line. Within double-quotes, only newline characters force new words; space and tab characters are preserved. However, a final newline is ignored. It is therefore possible for a command substitution to yield a partial word.

Filename Substitution

Unquoted words containing any of the characters `*`, `?`, `[` or `{`, or that begin with `~`, are expanded (also known as *globbing*) to an alphabetically sorted list of filenames, as follows:

`*` Match any (zero or more) characters.

`?` Match any single character.

`[. . .]` Match any single character in the enclosed list(s) or range(s). A list is a string of characters. A range is two characters separated by a dash (`-`), and includes all the characters in between in the ASCII collating sequence (see [ascii\(5\)](#)).

- `{str, str, ...}` Expand to each string (or filename-matching pattern) in the comma-separated list. Unlike the pattern-matching expressions above, the expansion of this construct is not sorted. For instance, `{b, a}` expands to 'b' 'a', (not 'a' 'b'). As special cases, the characters `{` and `}`, along with the string `{ }`, are passed undisturbed.
- `~[user]` Your home directory, as indicated by the value of the variable `home`, or that of `user`, as indicated by the password entry for `user`.

Only the patterns `*`, `?` and `[...]` imply pattern matching; an error results if no filename matches a pattern that contains them. The `.` (dot character), when it is the first character in a filename or pathname component, must be matched explicitly. The `/` (slash) must also be matched explicitly.

Expressions and Operators

A number of C shell built-in commands accept expressions, in which the operators are similar to those of C and have the same precedence. These expressions typically appear in the `@`, `exit`, `if`, `set` and `while` commands, and are often used to regulate the flow of control for executing commands. Components of an expression are separated by white space.

Null or missing values are considered \emptyset . The result of all expressions is a string, which can represent decimal numbers.

The following C shell operators are grouped in order of precedence:

- | | |
|------------------------------------|---|
| <code>(...)</code> | grouping |
| <code>>~</code> | one's complement |
| <code>!</code> | logical negation |
| <code>* / %</code> | multiplication, division, remainder. These are right associative, which can lead to unexpected results. Combinations should be grouped explicitly with parentheses. |
| <code>+ -</code> | addition, subtraction (also right associative) |
| <code><< >></code> | bitwise shift left, bitwise shift right |
| <code>< > <= >=</code> | less than, greater than, less than or equal to, greater than or equal to |
| <code>= = != =~ !~</code> | equal to, not equal to, filename-substitution pattern match (described below), filename-substitution pattern mismatch |
| <code>&</code> | bitwise AND |
| <code>^</code> | bitwise XOR (exclusive or) |
| <code> </code> | bitwise inclusive OR |
| <code>&&</code> | logical AND |

| | logical OR

The operators: ==, !=, =~, and !~ compare their arguments as strings; other operators use numbers. The operators =~ and !~ each check whether or not a string to the left matches a filename substitution pattern on the right. This reduces the need for `switch` statements when pattern-matching between strings is all that is required.

Also available are file inquiries:

- r*filename* Return true, or 1 if the user has read access. Otherwise it returns false, or 0.
- w*filename* True if the user has write access.
- x*filename* True if the user has execute permission (or search permission on a directory).
- ef*filename* True if *filename* exists.
- of*filename* True if the user owns *filename*.
- z *filename* True if *filename* is of zero length (empty).
- f*filename* True if *filename* is a plain file.
- d*filename* True if *filename* is a directory.

If *filename* does not exist or is inaccessible, then all inquiries return false.

An inquiry as to the success of a command is also available:

- { *command* } If *command* runs successfully, the expression evaluates to true, 1. Otherwise, it evaluates to false, 0. *Note:* Conversely, *command* itself typically returns 0 when it runs successfully, or some other value if it encounters a problem. If you want to get at the status directly, use the value of the `status` variable rather than this expression.

Control Flow

The shell contains a number of commands to regulate the flow of control in scripts and within limits, from the terminal. These commands operate by forcing the shell either to reread input (to *loop*), or to skip input under certain conditions (to *branch*).

Each occurrence of a `foreach`, `switch`, `while`, `if...then` and `else` built-in command must appear as the first word on its own input line.

If the shell's input is not seekable and a loop is being read, that input is buffered. The shell performs seeks within the internal buffer to accomplish the rereading implied by the loop. (To the extent that this allows, backward `goto` commands succeeds on nonseekable inputs.)

Command Execution

If the command is a C shell built-in command, the shell executes it directly. Otherwise, the shell searches for a file by that name with execute access. If the command name contains a /, the shell takes it as a pathname, and searches for it. If the command name does not contain a /, the shell attempts to resolve it to a pathname, searching each directory in the `path` variable for

the command. To speed the search, the shell uses its hash table (see the `rehash` built-in command) to eliminate directories that have no applicable files. This hashing can be disabled with the `-c` or `-t`, options, or the `unhash` built-in command.

As a special case, if there is no `/` in the name of the script and there is an alias for the word `shell`, the expansion of the `shell` alias is prepended (without modification) to the command line. The system attempts to execute the first word of this special (late-occurring) alias, which should be a full pathname. Remaining words of the alias's definition, along with the text of the input line, are treated as arguments.

When a pathname is found that has proper execute permissions, the shell forks a new process and passes it, along with its arguments, to the kernel using the `execve()` system call (see [exec\(2\)](#)). The kernel then attempts to overlay the new process with the desired program. If the file is an executable binary (in [a.out\(4\)](#) format) the kernel succeeds and begins executing the new process. If the file is a text file and the first line begins with `#!`, the next word is taken to be the pathname of a shell (or command) to interpret that script. Subsequent words on the first line are taken as options for that shell. The kernel invokes (overlays) the indicated shell, using the name of the script as an argument.

If neither of the above conditions holds, the kernel cannot overlay the file and the `execve()` call fails (see [exec\(2\)](#)). The C shell then attempts to execute the file by spawning a new shell, as follows:

- If the first character of the file is a `#`, a C shell is invoked.
- Otherwise, a Bourne shell is invoked.

Signal Handling

The shell normally ignores QUIT signals. Background jobs are immune to signals generated from the keyboard, including hangups (HUP). Other signals have the values that the C shell inherited from its environment. The shell's handling of interrupt and terminate signals within scripts can be controlled by the `onintr` built-in command. Login shells catch the TERM signal. Otherwise, this signal is passed on to child processes. In no case are interrupts allowed when a login shell is reading the `.logout` file.

Job Control

The shell associates a numbered *job* with each command sequence to keep track of those commands that are running in the background or have been stopped with TSTP signals (typically Control-z). When a command or command sequence (semicolon separated list) is started in the background using the `&` metacharacter, the shell displays a line with the job number in brackets and a list of associated process numbers:

```
[1] 1234
```

To see the current list of jobs, use the `jobs` built-in command. The job most recently stopped (or put into the background if none are stopped) is referred to as the *current* job and is indicated with a `'+'`. The previous job is indicated with a `'-'`. When the current job is terminated or moved to the foreground, this job takes its place (becomes the new current job).

To manipulate jobs, refer to the `bg`, `fg`, `kill`, `stop`, and `%` built-in commands.

A reference to a job begins with a `'%'`. By itself, the percent-sign refers to the current job.

- `%%+ %%` The current job.
- `%-` The previous job.
- `%j` Refer to job *j* as in: `'kill -9 %j'`. *j* can be a job number, or a string that uniquely specifies the command line by which it was started; `'fg %vi'` might bring a stopped *vi* job to the foreground, for instance.
- `%?string` Specify the job for which the command line uniquely contains *string*.

A job running in the background stops when it attempts to read from the terminal. Background jobs can normally produce output, but this can be suppressed using the `'stty tostop'` command.

Status Reporting

While running interactively, the shell tracks the status of each job and reports whenever the job finishes or becomes blocked. It normally displays a message to this effect as it issues a prompt, in order to avoid disturbing the appearance of your input. When set, the `noti fy` variable indicates that the shell is to report status changes immediately. By default, the `noti fy` command marks the current process; after starting a background job, type `noti fy` to mark it.

Commands

Built-in commands are executed within the C shell. If a built-in command occurs as any component of a pipeline except the last, it is executed in a subshell.

- `:` Null command. This command is interpreted, but performs no action.
- `alias [name [def]]` Assign *def* to the alias *name*. *def* is a list of words that can contain escaped history-substitution metasyntax. *name* is not allowed to be `alias` or `unalias`. If *def* is omitted, the current definition for the alias *name* is displayed. If both *name* and *def* are omitted, all aliases are displayed with their definitions.
- `bg [%job . . .]` Run the current or specified jobs in the background.
- `break` Resume execution after the end of the nearest enclosing `foreach` or `while` loop. The remaining commands on the current line are executed. This allows multilevel breaks to be written as a list of `break` commands, all on one line.
- `breaksw` Break from a `switch`, resuming after the `endsw`.
- `case label:` A label in a `switch` statement.
- `cd [dir]`
- `chdir [dir]` Change the shell's working directory to directory *dir*. If no argument is given, change to the home directory of the user. If *dir* is a relative

	pathname not found in the current directory, check for it in those directories listed in the <code>cdpath</code> variable. If <i>dir</i> is the name of a shell variable whose value starts with a <code>/</code> , change to the directory named by that value.
<code>continue</code>	Continue execution of the next iteration of the nearest enclosing <code>while</code> or <code>foreach</code> loop.
<code>default:</code>	Labels the default case in a <code>switch</code> statement. The default should come after all case labels. Any remaining commands on the command line are first executed.
<code>dirs [-l]</code>	Print the directory stack, most recent to the left. The first directory shown is the current directory. With the <code>-l</code> argument, produce an unabbreviated printout; use of the <code>~</code> notation is suppressed.
<code>echo [-n] list</code>	The words in <i>list</i> are written to the shell's standard output, separated by space characters. The output is terminated with a newline unless the <code>-n</code> option is used. <code>csh</code> , by default, invokes its built-in <code>echo</code> , if <code>echo</code> is called without the full pathname of a Unix command, regardless of the configuration of your <code>PATH</code> (see echo(1)).
<code>eval argument . . .</code>	Reads the arguments as input to the shell and executes the resulting command(s). This is usually used to execute commands generated as the result of command or variable substitution. See tset(1B) for an example of how to use <code>eval</code> .
<code>exec command</code>	Execute <i>command</i> in place of the current shell, which terminates.
<code>exit [(expr)]</code>	The calling shell or shell script exits, either with the value of the status variable or with the value specified by the expression <i>expr</i> .
<code>fg [%job]</code>	Bring the current or specified <i>job</i> into the foreground.
<code>foreach var(wordlist)</code>	
<code>...</code>	
<code>end</code>	The variable <i>var</i> is successively set to each member of <i>wordlist</i> . The sequence of commands between this command and the matching <code>end</code> is executed for each new value of <i>var</i> . Both <code>foreach</code> and <code>end</code> must appear alone on separate lines.
	The built-in command <code>continue</code> can be used to terminate the execution of the current iteration of the loop and the built-in command <code>break</code> can be used to terminate execution of the <code>foreach</code> command. When this command is read from the terminal, the loop is read once prompting with <code>?</code> before any statements in the loop are executed.

<code>glob <i>wordlist</i></code>	Perform filename expansion on <i>wordlist</i> . Like <code>echo</code> , but no <code>\</code> escapes are recognized. Words are delimited by NULL characters in the output.
<code>gotolabel</code>	The specified <i>label</i> is a filename and a command expanded to yield a label. The shell rewinds its input as much as possible and searches for a line of the form <i>label</i> : possibly preceded by space or tab characters. Execution continues after the indicated line. It is an error to jump to a label that occurs between a <code>while</code> or <code>for</code> built-in command and its corresponding end.
<code>hashstat</code>	Print a statistics line indicating how effective the internal hash table for the <i>path</i> variable has been at locating commands (and avoiding execs). An exec is attempted for each component of the <i>path</i> where the hash function indicates a possible hit and in each component that does not begin with a <code>'/'</code> . These statistics only reflect the effectiveness of the <i>path</i> variable, not the <i>cdpath</i> variable.
<code>history [-hr] [<i>n</i>]</code>	Display the history list; if <i>n</i> is given, display only the <i>n</i> most recent events. <ul style="list-style-type: none"> -r Reverse the order of printout to be most recent first rather than oldest first. -h Display the history list without leading numbers. This is used to produce files suitable for sourcing using the <code>-h</code> option to <i>source</i>.
<code>if (<i>expr</i>) <i>command</i></code>	If the specified expression evaluates to true, the single <i>command</i> with arguments is executed. Variable substitution on <i>command</i> happens early, at the same time it does for the rest of the <code>if</code> command. <i>command</i> must be a simple command, not a pipeline, a command list, or a parenthesized command list. <i>Note</i> : I/O redirection occurs even if <i>expr</i> is false, when <i>command</i> is <i>not</i> executed (this is a bug).
<code>if (<i>expr</i>) then</code> <code>...</code> <code>else if (<i>expr2</i>) then</code> <code>...</code> <code>else</code> <code>...</code> <code>endif</code>	If <i>expr</i> is true, commands up to the first <code>else</code> are executed. Otherwise, if <i>expr2</i> is true, the commands

between the `else if` and the second `else` are executed. Otherwise, commands between the `else` and the `endif` are executed. Any number of `else if` pairs are allowed, but only one `else`. Only one `endif` is needed, but it is required. The words `else` and `endif` must be the first nonwhite characters on a line. The `if` must appear alone on its input line or after an `else`.

`jobs [-l]`

List the active jobs under job control.

`-l` List process IDs, in addition to the normal information.

`kill [sig] [pid] [%job] . . .`
`kill -l`

Send the TERM (terminate) signal, by default, or the signal specified, to the specified process ID, the *job* indicated, or the current *job*. Signals are either given by number or by name. There is no default. Typing `kill` does not send a signal to the current job. If the signal being sent is TERM (terminate) or HUP (hangup), then the job or process is sent a CONT (continue) signal as well.

`-l` List the signal names that can be sent.

`limit [-h] [resource [max-use]]`

Limit the consumption by the current process or any process it spawns, each not to exceed *max-use* on the specified *resource*. The string `unlimited` requests that the current limit, if any, be removed. If *max-use* is omitted, print the current limit. If *resource* is omitted, display all limits. Run the `sysdef(1M)` command to obtain the maximum possible limits for your system. The values reported by `sysdef` are in hexadecimal, but can be translated into decimal numbers using the `bc(1)` command.

`-h` Use hard limits instead of the current limits. Hard limits impose a ceiling on the values of the current limits. Only the privileged user can raise the hard limits.

resource is one of:

<code>cputime</code>	Maximum CPU seconds per process.
----------------------	----------------------------------

<code>filesize</code>	Largest single file allowed. Limited by the size and capabilities of the filesystem. See df(1M) .
<code>datasize (heapsize)</code>	Maximum data size (including stack) for the process. This is the size of your virtual memory See swap(1M) .
<code>stacksize</code>	Maximum stack size for the process. The default stack size is 2^{64} bytes. You can use limit(1) to change this default within a shell.
<code>coredumpsize</code>	Maximum size of a core dump (file). This limited to the size of the filesystem.
<code>descriptors</code>	Maximum number of file descriptors. Run sysdef(1M) .
<code>memorysize</code>	Maximum size of virtual memory.

max-use is a number, with an optional scaling factor, as follows:

<i>nh</i>	Hours (for <code>cputime</code>).
<i>nk</i>	<i>n</i> kilobytes. This is the default for all but <code>cputime</code> .
<i>nm</i>	<i>n</i> megabytes or minutes (for <code>cputime</code>).
<i>mm:ss</i>	Minutes and seconds (for <code>cputime</code>).

Example of limit: To limit the size of a core file dump to 0 Megabytes, type the following:

```
limit coredumpsize 0M
```

```
login [username | -p ]
```

Terminate a login shell and invoke [login\(1\)](#). The `.logout` file is not processed. If *username* is omitted, `login` prompts for the name of a user.

`-p` Preserve the current environment (variables).

```
logout
```

Terminate a login shell.

<code>nice [+n -n] [command]</code>	<p>Increment the process priority value for the shell or for <i>command</i> by <i>n</i>. The higher the priority value, the lower the priority of a process, and the slower it runs. When given, <i>command</i> is always run in a subshell, and the restrictions placed on commands in simple <code>if</code> commands apply. If <i>command</i> is omitted, <code>nice</code> increments the value for the current shell. If no increment is specified, <code>nice</code> sets the process priority value to 4. The range of process priority values is from -20 to 20. Values of <i>n</i> outside this range set the value to the lower, or to the higher boundary, respectively.</p> <p><i>+n</i> Increment the process priority value by <i>n</i>.</p> <p><i>-n</i> Decrement by <i>n</i>. This argument can be used only by the privileged user.</p>
<code>nohup [command]</code>	<p>Run <i>command</i> with HUPs ignored. With no arguments, ignore HUPs throughout the remainder of a script. When given, <i>command</i> is always run in a subshell, and the restrictions placed on commands in simple <code>if</code> statements apply. All processes detached with <code>&</code> are effectively <code>nohup'd</code>.</p>
<code>notify [%job] ...</code>	<p>Notify the user asynchronously when the status of the current job or specified jobs changes.</p>
<code>onintr [- label]</code>	<p>Control the action of the shell on interrupts. With no arguments, <code>onintr</code> restores the default action of the shell on interrupts. (The shell terminates shell scripts and returns to the terminal command input level). With the <code>-</code> argument, the shell ignores all interrupts. With a <i>label</i> argument, the shell executes a <code>goto label</code> when an interrupt is received or a child process terminates because it was interrupted.</p>
<code>popd [+n]</code>	<p>Pop the directory stack and <code>cd</code> to the new top directory. The elements of the directory stack are numbered from 0 starting at the top.</p> <p><i>+n</i> Discard the <i>n</i>'th entry in the stack.</p>
<code>pushd [+n dir]</code>	<p>Push a directory onto the directory stack. With no arguments, exchange the top two elements.</p> <p><i>+n</i> Rotate the <i>n</i>'th entry to the top of the stack and <code>cd</code> to it.</p>

<i>dir</i>	Push the current working directory onto the stack and change to <i>dir</i> .
rehash	Recompute the internal hash table of the contents of directories listed in the <i>path</i> variable to account for new commands added. Recompute the internal hash table of the contents of directories listed in the <i>cdpath</i> variable to account for new directories added.
repeat <i>count command</i>	Repeat <i>command</i> <i>count</i> times. <i>command</i> is subject to the same restrictions as with the one-line <code>if</code> statement.
set [<i>var</i> [= <i>value</i>]] set <i>var</i> [<i>n</i>] = <i>word</i>	<p>With no arguments, set displays the values of all shell variables. Multiword values are displayed as a parenthesized list. With the <i>var</i> argument alone, set assigns an empty (null) value to the variable <i>var</i>. With arguments of the form <i>var</i> = <i>value</i> set assigns <i>value</i> to <i>var</i>, where <i>value</i> is one of:</p> <p><i>word</i> A single word (or quoted string).</p> <p>(<i>wordlist</i>) A space-separated list of words enclosed in parentheses.</p> <p>Values are command and filename expanded before being assigned. The form <code>set var[n] = word</code> replaces the <i>n</i>'th word in a multiword value with <i>word</i>.</p>
setenv [<i>VAR</i> [<i>word</i>]]	<p>With no arguments, setenv displays all environment variables. With the <i>VAR</i> argument, setenv sets the environment variable <i>VAR</i> to have an empty (null) value. (By convention, environment variables are normally given upper-case names.) With both <i>VAR</i> and <i>word</i> arguments, setenv sets the environment variable <i>NAME</i> to the value <i>word</i>, which must be either a single word or a quoted string. The most commonly used environment variables, <code>USER</code>, <code>TERM</code>, and <code>PATH</code>, are automatically imported to and exported from the csh variables <code>user</code>, <code>term</code>, and <code>path</code>. There is no need to use setenv for these. In addition, the shell sets the <code>PWD</code> environment variable from the csh variable <code>cwd</code> whenever the latter changes.</p> <p>The environment variables <code>LC_CTYPE</code>, <code>LC_MESSAGES</code>, <code>LC_TIME</code>, <code>LC_COLLATE</code>, <code>LC_NUMERIC</code>, and <code>LC_MONETARY</code> take immediate effect when changed within the C shell.</p>

If any of the LC_* variables (LC_CTYPE, LC_MESSAGES, LC_TIME, LC_COLLATE, LC_NUMERIC, and LC_MONETARY) (see [environ\(5\)](#)) are not set in the environment, the operational behavior of csh for each corresponding locale category is determined by the value of the LANG environment variable. If LC_ALL is set, its contents are used to override both the LANG and the other LC_* variables. If none of the above variables is set in the environment, the “C” (U.S. style) locale determines how csh behaves.

LC_CTYPE Determines how csh handles characters. When LC_CTYPE is set to a valid value, csh can display and handle text and filenames containing valid characters for that locale.

LC_MESSAGES Determines how diagnostic and informative messages are presented. This includes the language and style of the messages and the correct form of affirmative and negative responses. In the “C” locale, the messages are presented in the default form found in the program itself (in most cases, U.S./English).

LC_NUMERIC Determines the value of the radix character, decimal point, (.) in the “C” locale) and thousand separator, empty string (“”) in the “C” locale).

shift [*variable*]

The components of argv, or *variable*, if supplied, are shifted to the left, discarding the first component. It is an error for the variable not to be set or to have a null value.

source [-h] *name*

Reads commands from *name*. source commands can be nested, but if they are nested too deeply the shell can run out of file descriptors. An error in a sourced file at any level terminates all nested source commands.

-h Place commands from the file *name* on the history list without executing them.

stop %*jobid* . . .

Stop the current or specified background job.

stop *pid* . . .

Stop the specified process, *pid*. (see [ps\(1\)](#)).

suspend

Stop the shell in its tracks, much as if it had been sent a stop signal with `^Z`. This is most often used to stop shells started by `su`.

switch (*string*)

case *label*:

...

breaksw

...

default:

...

breaksw

endsw

Each *label* is successively matched, against the specified *string*, which is first command and filename expanded. The file metacharacters `*`, `?` and `[. . .]` can be used in the case labels, which are variable expanded. If none of the labels match before a “default” label is found, execution begins after the default label. Each case statement and the default statement must appear at the beginning of a line. The command `breaksw` continues execution after the `endsw`. Otherwise control falls through subsequent case and default statements as with `C`. If no label matches and there is no default, execution continues after the `endsw`.

time [*command*]

With no argument, print a summary of time used by this `C` shell and its children. With an optional *command*, execute *command* and print a summary of the time it uses. As of this writing, the `time` built-in command does NOT compute the last 6 fields of output, rendering the output to erroneously report the value `0` for these fields.

example `%time ls -R`

```
9.0u 11.0s 3:32 10% 0+0k 0+0io 0pf+0w
```

(See the Environment Variables and Predefined Shell Variables sub-section on the `time` variable.)

umask [*value*]

Display the file creation mask. With *value*, set the file creation mask. With *value* given in octal, the user can turn off any bits, but cannot turn on bits to allow new permissions. Common values include `077`, restricting all permissions from everyone else; `002`, giving complete access to the group, and read (and directory search) access to others; or `022`, giving read (and directory search) but not write permission to the group and others.

<code>unalias <i>pattern</i></code>	Discard aliases that match (filename substitution) <i>pattern</i> . All aliases are removed by <code>'unalias *'</code> .
<code>unhash</code>	Disable the internal hash tables for the <i>path</i> and <i>cdpath</i> variables.
<code>unlimit [-h] [<i>resource</i>]</code>	Remove a limitation on <i>resource</i> . If no <i>resource</i> is specified, then all resource limitations are removed. See the description of the <code>limit</code> command for the list of resource names. -h Remove corresponding hard limits. Only the privileged user can do this.
<code>unset <i>pattern</i></code>	Remove variables whose names match (filename substitution) <i>pattern</i> . All variables are removed by <code>'unset *'</code> ; this has noticeably distasteful side effects.
<code>unsetenv <i>variable</i></code>	Remove <i>variable</i> from the environment. As with <code>unset</code> , pattern matching is not performed.
<code>wait</code>	Wait for background jobs to finish (or for an interrupt) before prompting.
<code>while (<i>expr</i>) ... end</code>	While <i>expr</i> is true (evaluates to nonzero), repeat commands between the <code>while</code> and the matching <code>end</code> statement. <code>break</code> and <code>continue</code> can be used to terminate or continue the loop prematurely. The <code>while</code> and <code>end</code> must appear alone on their input lines. If the shell's input is a terminal, it prompts for commands with a question-mark until the <code>end</code> command is entered and then performs the commands in the loop.
<code>% [<i>job</i>] [&]</code>	Bring the current or indicated <i>job</i> to the foreground. With the ampersand, continue running <i>job</i> in the background.
<code>@ [<i>var=expr</i>] @ [<i>var</i>[<i>n</i>]=<i>expr</i>]</code>	With no arguments, display the values for all shell variables. With arguments, set the variable <i>var</i> , or the <i>n</i> 'th word in the value of <i>var</i> , to the value that <i>expr</i> evaluates to. (If [<i>n</i>] is supplied, both <i>var</i> and its <i>n</i> 'th component must already exist.)

If the expression contains the characters `>`, `<`, `&`, or `|`, then at least this part of *expr* must be placed within parentheses.

The operators `*=`, `+=`, and so forth, are available as in C. The space separating the name from the assignment operator is optional. Spaces are, however, mandatory in separating components of *expr* that would otherwise be single words.

Special postfix operators, `++` and `--`, increment or decrement *name*, respectively.

Environment Variables and Predefined Shell Variables

Unlike the Bourne shell, the C shell maintains a distinction between environment variables, which are automatically exported to processes it invokes, and shell variables, which are not. Both types of variables are treated similarly under variable substitution. The shell sets the variables `argv`, `cwd`, `home`, `path`, `prompt`, `shell`, and `status` upon initialization. The shell copies the environment variable `USER` into the shell variable `user`, `TERM` into `term`, and `HOME` into `home`, and copies each back into the respective environment variable whenever the shell variables are reset. `PATH` and `path` are similarly handled. You need only set `path` once in the `.cshrc` or `.login` file. The environment variable `PWD` is set from `cwd` whenever the latter changes. The following shell variables have predefined meanings:

<code>argv</code>	Argument list. Contains the list of command line arguments supplied to the current invocation of the shell. This variable determines the value of the positional parameters <code>\$1</code> , <code>\$2</code> , and so on.				
<code>cdpath</code>	Contains a list of directories to be searched by the <code>cd</code> , <code>chdir</code> , and <code>popd</code> commands, if the directory argument each accepts is not a subdirectory of the current directory.				
<code>cwd</code>	The full pathname of the current directory.				
<code>echo</code>	Echo commands (after substitutions) just before execution.				
<code>ignore</code>	A list of filename suffixes to ignore when attempting filename completion. Typically the single word <code>.'o'</code> .				
<code>filec</code>	Enable filename completion, in which case the Control-d character EOT and the ESC character have special significance when typed in at the end of a terminal input line: <table> <tr> <td>EOT</td> <td>Print a list of all filenames that start with the preceding string.</td> </tr> <tr> <td>ESC</td> <td>Replace the preceding string with the longest unambiguous extension.</td> </tr> </table>	EOT	Print a list of all filenames that start with the preceding string.	ESC	Replace the preceding string with the longest unambiguous extension.
EOT	Print a list of all filenames that start with the preceding string.				
ESC	Replace the preceding string with the longest unambiguous extension.				
<code>hardpaths</code>	If set, pathnames in the directory stack are resolved to contain no symbolic-link components.				

<code>histchars</code>	A two-character string. The first character replaces ! as the history-substitution character. The second replaces the caret (^) for quick substitutions.
<code>history</code>	The number of lines saved in the history list. A very large number can use up all of the C shell's memory. If not set, the C shell saves only the most recent command.
<code>home</code>	The user's home directory. The filename expansion of ~ refers to the value of this variable.
<code>ignoreeof</code>	If set, the shell ignores EOF from terminals. This protects against accidentally killing a C shell by typing a Control-d.
<code>mail</code>	A list of files where the C shell checks for mail. If the first word of the value is a number, it specifies a mail checking interval in seconds (default 5 minutes).
<code>nobeeep</code>	Suppress the bell during command completion when asking the C shell to extend an ambiguous filename.
<code>noclobber</code>	Restrict output redirection so that existing files are not destroyed by accident. > redirections can only be made to new files. >> redirections can only be made to existing files.
<code>noglob</code>	Inhibit filename substitution. This is most useful in shell scripts once filenames (if any) are obtained and no further expansion is desired.
<code>nomatch</code>	Return the filename substitution pattern, rather than an error, if the pattern is not matched. Malformed patterns still result in errors.
<code>notify</code>	If set, the shell notifies you immediately as jobs are completed, rather than waiting until just before issuing a prompt.
<code>path</code>	The list of directories in which to search for commands. <code>path</code> is initialized from the environment variable <code>PATH</code> , which the C shell updates whenever <code>path</code> changes. A null word ("") specifies the current directory. The default is typically <code>(/usr/bin .)</code> . One can override this initial search path upon <code>csh</code> start-up by setting it in <code>.cshrc</code> or <code>.login</code> (for login shells only). If <code>path</code> becomes unset, only full pathnames execute. An interactive C shell normally hashes the contents of the directories listed after reading <code>.cshrc</code> , and whenever <code>path</code> is reset. If new commands are added, use the <code>rehash</code> command to update the table.
<code>prompt</code>	The string an interactive C shell prompts with. Noninteractive shells leave the <code>prompt</code> variable unset. Aliases and other commands in the <code>.cshrc</code> file that are only useful interactively, can be placed after the following test: <code>'if (\$?prompt == 0) exit'</code> , to reduce startup time for noninteractive shells. A ! in the prompt string is replaced by the current event number. The default prompt is <code>hostname%</code> for mere mortals, or <code>hostname#</code> for the privileged user.

	The setting of <code>\$prompt</code> has three meanings:
	<code>\$prompt</code> not set non-interactive shell, test <code>\$?prompt</code> .
	<code>\$prompt</code> set but <code>== ""</code> <code>.cshrc</code> called by the <code>which(1)</code> command.
	<code>\$prompt</code> set and <code>!= ""</code> normal interactive shell.
<code>savehist</code>	The number of lines from the history list that are saved in <code>~/.history</code> when the user logs out. Large values for <code>savehist</code> slow down the C shell during startup.
<code>shell</code>	The file in which the C shell resides. This is used in forking shells to interpret files that have execute bits set, but that are not executable by the system.
<code>status</code>	The status returned by the most recent command. If that command terminated abnormally, 0200 is added to the status. Built-in commands that fail return exit status 1; all other built-in commands set status to 0.
<code>time</code>	Control automatic timing of commands. Can be supplied with one or two values. The first is the reporting threshold in CPU seconds. The second is a string of tags and text indicating which resources to report on. A tag is a percent sign (%) followed by a single upper-case letter (unrecognized tags print as text): <ul style="list-style-type: none"> <code>%D</code> Average amount of unshared data space used in Kilobytes. <code>%E</code> Elapsed (wallclock) time for the command. <code>%F</code> Page faults. <code>%I</code> Number of block input operations. <code>%K</code> Average amount of unshared stack space used in Kilobytes. <code>%M</code> Maximum real memory used during execution of the process. <code>%O</code> Number of block output operations. <code>%P</code> Total CPU time — U (user) plus S (system) — as a percentage of E (elapsed) time. <code>%S</code> Number of seconds of CPU time consumed by the kernel on behalf of the user's process. <code>%U</code> Number of seconds of CPU time devoted to the user's process. <code>%W</code> Number of swaps. <code>%X</code> Average amount of shared memory used in Kilobytes. <p>The default summary display outputs from the <code>%U</code>, <code>%S</code>, <code>%E</code>, <code>%P</code>, <code>%X</code>, <code>%D</code>, <code>%I</code>, <code>%O</code>, <code>%F</code>, and <code>%W</code> tags, in that order.</p>
<code>verbose</code>	Display each command after history substitution takes place.

Large File Behavior See [largefile\(5\)](#) for the description of the behavior of csh when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

文件

- ~/.cshrc Read at beginning of execution by each shell.
- ~/.login Read by login shells after .cshrc at login.
- ~/.logout Read by login shells at logout.
- ~/.history Saved history for use at next login.
- /usr/bin/sh The Bourne shell, for shell scripts not starting with a '#'.
- /tmp/sh* Temporary file for '<<'.
- /etc/passwd Source of home directories for '~name'.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

另请参见 [bc\(1\)](#), [echo\(1\)](#), [limit\(1\)](#), [login\(1\)](#), [ls\(1\)](#), [more\(1\)](#), [pfcsh\(1\)](#), [pfexec\(1\)](#), [ps\(1\)](#), [sh\(1\)](#), [shell_builtins\(1\)](#), [tset\(1B\)](#), [which\(1\)](#), [df\(1M\)](#), [swap\(1M\)](#), [sysdef\(1M\)](#), [access\(2\)](#), [exec\(2\)](#), [fork\(2\)](#), [pipe\(2\)](#), [a.out\(4\)](#), [ascii\(5\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [termio\(7I\)](#)

诊断 You have stopped jobs. You attempted to exit the C shell with stopped jobs under job control. An immediate second attempt to exit succeeds, terminating the stopped jobs.

警告 The use of setuid shell scripts is *strongly* discouraged.

附注 Words can be no longer than 1024 bytes. The system limits argument lists to 1,048,576 bytes. However, the maximum number of arguments to a command for which filename expansion applies is 1706. Command substitutions can expand to no more characters than are allowed in the argument list. To detect looping, the shell restricts the number of alias substitutions on a single line to 20.

When a command is restarted from a stop, the shell prints the directory it started in if this is different from the current directory; this can be misleading (that is, wrong) as the job might have changed directories internally.

Shell built-in functions are not stoppable/restartable. Command sequences of the form *a b c* are also not handled gracefully when stopping is attempted. If you suspend *b*, the shell never

executes *c*. This is especially noticeable if the expansion results from an alias. It can be avoided by placing the sequence in parentheses to force it into a subshell.

Commands within loops, prompted for by *?*, are not placed in the *history* list.

Control structures should be parsed rather than being recognized as built-in commands. This would allow control commands to be placed anywhere, to be combined with *|*, and to be used with *&* and *;* metasyntax.

It should be possible to use the *:* modifiers on the output of command substitutions. There are two problems with *:* modifier usage on variable substitutions: not all of the modifiers are available, and only one modifier per substitution is allowed.

The *g* (global) flag in history substitutions applies only to the first match in each word, rather than all matches in all words. The common text editors consistently do the latter when given the *g* flag in a substitution command.

Quoting conventions are confusing. Overriding the escape character to force variable substitutions within double quotes is counterintuitive and inconsistent with the Bourne shell.

Symbolic links can fool the shell. Setting the *hardpaths* variable alleviates this.

It is up to the user to manually remove all duplicate pathnames accrued from using built-in commands more than once, as shown below:

```
set path = pathnames
setenv PATH = pathnames
```

These often occur because a shell script or a *.cshrc* file does something like

```
'set path=(/usr/local /usr/hosts $path)'
```

to ensure that the named directories are in the pathname list.

The only way to direct the standard output and standard error separately is by invoking a subshell, as follows:

```
command > outfile ) >& errorfile
```

Although robust enough for general use, adventures into the esoteric periphery of the C shell can reveal unexpected quirks.

If you start *csh* as a login shell and you do not have a *.login* in your home directory, then the *csh* reads in the */etc/.login*.

When the shell executes a shell script that attempts to execute a non-existent command interpreter, the shell returns an erroneous diagnostic message that the shell script file does not exist.

已知问题

As of this writing, the `time` built-in command does *not* compute the last 6 fields of output, rendering the output to erroneously report the value `0` for these fields:

```
example %time ls -R
          9.0u 11.0s 3:32 10% 0+0k 0+0io 0pf+0w
```


引用名	csplit – split files based on context
用法概要	<code>csplit [-ks] [-f <i>prefix</i>] [-n <i>number</i>] <i>file arg1... argn</i></code>
描述	The <code>csplit</code> utility reads the file named by the <i>file</i> operand, writes all or part of that file into other files as directed by the <i>arg</i> operands, and writes the sizes of the files.
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -f <i>prefix</i> Names the created files <i>prefix00</i>, <i>prefix01</i>, ..., <i>prefixn</i>. The default is <i>xx00 ... xxn</i>. If the <i>prefix</i> argument would create a file name exceeding 14 bytes, an error results. In that case, <code>csplit</code> exits with a diagnostic message and no files are created. -k Leaves previously created files intact. By default, <code>csplit</code> removes created files if an error occurs. -n <i>number</i> Uses <i>number</i> decimal digits to form filenames for the file pieces. The default is 2. -s Suppresses the output of file size messages.
操作数	<p>The following operands are supported:</p> <p><i>file</i> The path name of a text file to be split. If <i>file</i> is -, the standard input will be used.</p> <p>The operands <i>arg1 ... argn</i> can be a combination of the following:</p> <ul style="list-style-type: none"> <i>/rexp/[offset]</i> Create a file using the content of the lines from the current line up to, but not including, the line that results from the evaluation of the regular expression with <i>offset</i>, if any, applied. The regular expression <i>rexp</i> must follow the rules for basic regular expressions. Regular expressions can include the use of '\/' and '\%'. These forms must be properly quoted with single quotes, since “\” is special to the shell. The optional <i>offset</i> must be a positive or negative integer value representing a number of lines. The integer value must be preceded by + or -. If the selection of lines from an offset expression of this type would create a file with zero lines, or one with greater than the number of lines left in the input file, the results are unspecified. After the section is created, the current line will be set to the line that results from the evaluation of the regular expression with any offset applied. The pattern match of <i>rexp</i> always is applied from the current line to the end of the file. <i>%rexp%[offset]</i> This operand is the same as <i>/rexp/[offset]</i>, except that no file will be created for the selected section of the input file. <i>line_no</i> Create a file from the current line up to (but not including) the line number <i>line_no</i>. Lines in the file will be numbered starting at one. The current line becomes <i>line_no</i>. <i>{num}</i> Repeat operand. This operand can follow any of the operands described previously. If it follows a <i>rexp</i> type operand, that operand will be applied <i>num</i> more times. If it follows a <i>line_no</i> operand, the file will be split every

line_no lines, *num* times, from that point.

An error will be reported if an operand does not reference a line between the current position and the end of the file.

用法 See [largefile\(5\)](#) for the description of the behavior of `csplit` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例 示例 1 Splitting and combining files

This example creates four files, `cobol00...cobol03`.

```
example% csplit -f cobol filename \  
         '/procedure division/' /par5./ /par16./
```

After editing the *split* files, they can be recombined as follows:

```
example% cat cobol0[0-3] > filename
```

This example overwrites the original file.

示例 2 Splitting a file into equal parts

This example splits the file at every 100 lines, up to 10,000 lines. The `-k` option causes the created files to be retained if there are less than 10,000 lines; however, an error message would still be printed.

```
example% csplit -k filename 100 {99}
```

示例 3 Creating a file for separate C routines

If `prog.c` follows the normal C coding convention (the last line of a routine consists only of a `}` in the first character position), this example creates a file for each separate C routine (up to 21) in `prog.c`.

```
example% csplit -k prog.c '%main(%' '/^}/+1' {20}
```

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `csplit`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态 The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[sed\(1\)](#), [split\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

诊断

The diagnostic messages are self-explanatory, except for the following:

arg – out of range The given argument did not reference a line between the current position and the end of the file.

引用名 ct – spawn login to a remote terminal

用法概要 ct [*options*] *telno*...

描述 The ct utility dials the telephone number of a modem that is attached to a terminal and spawns a login process to that terminal. The *telno* is a telephone number, with equal signs for secondary dial tones and minus signs for delays at appropriate places. (The set of legal characters for *telno* is 0 through 9, -, =, *, and #. The maximum length *telno* is 31 characters). If more than one telephone number is specified, ct will try each in succession until one answers; this is useful for specifying alternate dialing paths.

ct will try each line listed in the file /etc/uucp/Devices until it finds an available line with appropriate attributes, or runs out of entries.

After the user on the destination terminal logs out, there are two things that could occur depending on what type of port monitor is monitoring the port. In the case of no port monitor, ct prompts: Reconnect? If the response begins with the letter n, the line will be dropped; otherwise, ttymon will be started again and the login: prompt will be printed. In the second case, where a port monitor is monitoring the port, the port monitor reissues the login: prompt.

The user should log out properly before disconnecting.

选项 The following options are supported:

- h Normally, ct will hang up the current line so that it can be used to answer the incoming call. The -h option will prevent this action. The -h option will also wait for the termination of the specified ct process before returning control to the user's terminal.
- speed* The data rate may be set with the -s option. *speed* is expressed in baud rates. The default baud rate is 1200.
- v If the -v (verbose) option is used, ct will send a running narrative to the standard error output stream.
- wn* If there are no free lines ct will ask if it should wait, and for how many minutes, before it gives up. ct will continue to try to open the dialers at one-minute intervals until the specified limit is exceeded. This dialogue may be overridden by specifying the -wn option. *n* is the maximum number of minutes that ct is to wait for a line.
- xn* This option is used for debugging; it produces a detailed output of the program execution on stderr. *n* is a single number between 0 and 9. As *n* increases to 9, more detailed debugging information is given.

文件 /etc/uucp/Devices
 /var/adm/ctlog

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/uucp

另请参见 [cu\(1C\)](#), [login\(1\)](#), [uucp\(1C\)](#), [ttymon\(1M\)](#), [attributes\(5\)](#)

附注 The ct program will not work with a DATAKIT Multiplex interface.

For a shared port, one used for both dial-in and dial-out, the `ttymon` program running on the line must have the `-r` and `-b` options specified (see [ttymon\(1M\)](#)).

引用名 ctags – create a tags file for use with ex and vi

用法概要 /usr/bin/ctags [-aBFtuvwx] [-f *tagsfile*] *file*...
/usr/xpg4/bin/ctags [-aBFuvwx] [-f *tagsfile*] *file*...

描述 The ctags utility makes a tags file for [ex\(1\)](#) from the specified C, C++, Pascal, FORTRAN, [yacc\(1\)](#), and [lex\(1\)](#) sources. A tags file gives the locations of specified objects (in this case functions and typedefs) in a group of files. Each line of the tags file contains the object name, the file in which it is defined, and an address specification for the object definition. Functions are searched with a pattern, typedefs with a line number. Specifiers are given in separate fields on the line, separated by SPACE or TAB characters. Using the tags file, ex can quickly find these objects' definitions.

Normally, ctags places the tag descriptions in a file called tags; this may be overridden with the -f option.

Files with names ending in .c or .h are assumed to be either C or C++ source files and are searched for C/C++ routine and macro definitions. Files with names ending in .cc, .C, or .cxx, are assumed to be C++ source files. Files with names ending in .y are assumed to be yacc source files. Files with names ending in .l are assumed to be lex files. Others are first examined to see if they contain any Pascal or FORTRAN routine definitions; if not, they are processed again looking for C definitions.

The tag main is treated specially in C or C++ programs. The tag formed is created by prepending M to *file*, with a trailing .c, .cc .C, or .cxx removed, if any, and leading path name components also removed. This makes use of ctags practical in directories with more than one program.

选项 The precedence of the options that pertain to printing is -x, -v, then the remaining options. The following options are supported:

- a Appends output to an existing tags file.
- B Uses backward searching patterns (?..?).
- f *tagsfile* Places the tag descriptions in a file called *tagsfile* instead of tags.
- F Uses forward searching patterns (/.../) (default).
- t Creates tags for typedefs. /usr/xpg4/bin/ctags creates tags for typedefs by default.
- u Updates the specified files in tags, that is, all references to them are deleted, and the new values are appended to the file. Beware: this option is implemented in a way that is rather slow; it is usually faster to simply rebuild the tags file.
- v Produces on the standard output an index listing the function name, file name, and page number (assuming 64 line pages). Since the output will be sorted into lexicographic order, it may be desired to run the output through sort -f.

- w Suppresses warning diagnostics.
- x Produces a list of object names, the line number and file name on which each is defined, as well as the text of that line and prints this on the standard output. This is a simple index which can be printed out as an off-line readable function index.

操作数

The following *file* operands are supported:

- file.c* Files with basenames ending with the `.c` suffix are treated as C-language source code.
- file.h* Files with basenames ending with the `.h` suffix are treated as C-language source code.
- file.f* Files with basenames ending with the `.f` suffix are treated as FORTRAN-language source code.

用法

The `-v` option is mainly used with `vgrind` which will be part of the optional BSD Compatibility Package.

示例

示例 1 Producing entries in alphabetical order

Using `ctags` with the `-v` option produces entries in an order which may not always be appropriate for `vgrind`. To produce results in alphabetical order, you may want to run the output through `sort -f`.

```
example% ctags -v filename.c filename.h | sort -f > index
example% vgrind -x index
```

示例 2 Building a tags file

To build a tags file for C sources in a directory hierarchy rooted at *sourcedir*, first create an empty tags file, and then run `find(1)`

```
example% cd sourcedir ; rm -f tags ; touch tags
example% find . \( -name SCCS -prune -name \
    '*.c' -o -name '*.h' \) -exec ctags -u {} \;
```

Notice that spaces must be entered exactly as shown.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `ctags`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

文件 tags output tags file

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/ctags

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities

/usr/xpg4/bin/ctags

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [ex\(1\)](#), [lex\(1\)](#), [vgrind\(1\)](#), [vi\(1\)](#), [yacc\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注 Recognition of functions, subroutines, and procedures for FORTRAN and Pascal is done in a very simpleminded way. No attempt is made to deal with block structure; if you have two Pascal procedures in different blocks with the same name, you lose.

The method of deciding whether to look for C or Pascal and FORTRAN functions is a hack.

The ctags utility does not know about `#ifdefs`.

The ctags utility should know about Pascal types. Relies on the input being well formed to detect typedefs. Use of `-tx` shows only the last line of typedefs.

引用名	ctrun – execute command in a process contract
用法概要	<code>/usr/bin/ctrun [options] command [argument]...</code>
描述	<p>The <code>ctrun</code> utility starts a command in a newly created process contract. <code>ctrun</code> holds the contract and can be instructed to output or respond to events that occur within the contract.</p> <p>For additional information about process contracts, see contract(4) and process(4).</p>
选项	<p>The following options are supported:</p> <p><code>-A aux</code> Sets the process contract creator's auxiliary field.</p> <p><code>-i event,[event ...]</code> <code>-f event,[event ...]</code> Sets the informative and fatal events, respectively.</p> <p>The following are valid <i>events</i>:</p> <p><code>core</code> A member process dumped core. <code>core</code> events are informative by default.</p> <p><code>empty</code> The last member of the process contract exited.</p> <p><code>exit</code> A member process exited.</p> <p><code>fork</code> A process was added to the process contract.</p> <p><code>hwerr</code> A member process encountered a hardware error. <code>hwerr</code> events are fatal by default.</p> <p><code>signal</code> A member process received a fatal signal from a process in a different process contract.</p> <p>Only <code>core</code>, <code>hwerr</code>, and <code>signal</code> events can be made fatal.</p> <p>More events can be delivered than requested if <code>ctrun</code> requires them for its own purposes. For example, <code>empty</code> messages are always requested if a lifetime of contract is specified. See <code>-l</code>.</p> <p><code>-F fmri</code> Sets the process contract service FMRI field. To set this field the caller is required to have the <code>{PRIV_CONTRACT_IDENTITY}</code> in its effective set.</p> <p><code>-l lifetime</code> The following valid <i>lifetime</i> values are supported:</p> <p><code>child</code> <code>ctrun</code> exits when the command exits, regardless of whether the contract is empty.</p> <p><code>contract</code> <code>ctrun</code> exits only when the contract exits. This is the default.</p>

	none	ct run exits immediately, orphaning the contract.
-o <i>option</i> , [<i>option</i> ...]	The following <i>options</i> are supported:	
	noorphan	Kills all processes in the contract if the holder (ct run) exits. This option is invalid when a lifetime of none is specified.
	pgrponly	If a fatal error occurs, kills at most the process group of which the errant process is a member.
	regent	The contract inherits inheritable contracts when abandoned by member processes.
-r <i>count</i>	If the contract encounters a fault, this option attempts to restart the command <i>count</i> times. If <i>count</i> is 0, the attempt to restart continues indefinitely. By default, ct run does not attempt to restart the command. This option is invalid if a lifetime other than contract is specified or if the pgrponly option is used.	
-t	If the contract created by ct run inherited subcontracts from its member processes, attempts to transfer them to the new contract when restarting. This option is invalid unless -r is also specified.	
-v	Displays contract events and ct run actions as they occur.	
-V	Displays verbose contract events, as are displayed by the -v option of ctwatch. Implies -v.	

操作数

The following operands are supported:

<i>argument</i>	One of the strings treated as an argument to <i>command</i> .
<i>command</i>	The command to be passed to <code>execvp(2)</code> . See exec(2) .

示例

示例 1 Running a Shell in a New Process Contract

The following example runs a shell in a new process contract:

```
example% ctrun -l child -o pgrponly ksh
```

The `-l child` option argument is specified so that ct run won't wait until all children of the shell have exited. `-o pgrponly` is specified because an interactive ksh puts each job in a new process group, and an error in one job is unlikely to affect the others.

示例 2 Running a Simple Server

The following example runs a simple server:

```
example% ctrun -r 0 -t -f hwerr,core,signal server
```

The `-r 0` and `-t` options are specified to indicate that if the server encounters a fatal error, `ctrun` should try to restart it. The `-f` option makes “hwerr”, “core”, and “signal” fatal events.

退出状态

If *command* is specified and successfully invoked (see [exec\(2\)](#)), the exit status of `ctrun` is the exit status of *command*. Otherwise, `ctrun` exits with one of the following values:

- 123 The child process exited abnormally.
- 124 `ctrun` encountered an internal error.
- 125 Invalid arguments were provided to `ctrun`.
- 126 *command* was found but could not be invoked.
- 127 *command* could not be found.

文件

/system/contract/process/*

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

Human Readable Output is Uncommitted. Invocation is Committed.

另请参见

[ctstat\(1\)](#), [ctwatch\(1\)](#), [exec\(2\)](#), [contract\(4\)](#), [process\(4\)](#), [attributes\(5\)](#)

引用名	ctstat – display active system contracts
用法概要	<pre>/usr/bin/ctstat [-a] [-i <i>contractid...</i>] [-t <i>type...</i>] [-v] [-T u d] [<i>interval</i> [<i>count</i>]]</pre>
描述	<p>The <code>ctstat</code> utility allows a user to observe the contracts active on a system.</p> <p>Unless you specify the <code>-i</code> or <code>-t</code> option, <code>ctstat</code> displays statistics on all contracts in the system.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"><code>-a</code> Display all contracts regardless of state. By default, only those contracts which are in the owned, inherited, or orphan states are displayed.<code>-i <i>contractid...</i></code> Request status on the specified contracts, identified by their numeric contract identifier (<i>contract_id</i>). This option accepts lists as arguments . Items in the list can be separated by commas, or enclosed in quotes and separated by commas or spaces.<code>-T u d</code> Display a time stamp. Specify <code>u</code> for a printed representation of the internal representation of time. See time(2). Specify <code>d</code> for standard date format. See date(1).<code>-t <i>type...</i></code> Request status on contracts of the specified type (<i>type</i>). This option accepts lists as arguments. Items in the list can be separated by commas, or enclosed in quotes and separated by commas or spaces. The following types are supported:<ul style="list-style-type: none"><code>process</code> Process contracts<code>-v</code> Verbose output.
操作数	<p>The following operands are supported:</p> <ul style="list-style-type: none"><i>interval</i> Report once each <i>interval</i> seconds.<i>count</i> Print only <i>count</i> reports.

Output

The following list defines the column headings and the meanings of a `ctstat` report:

CTID

The contract ID of the contract.

ZONEID

The zone ID of the contract's creator.

TYPE

The contract type.

STATE

The state of the contract:

owned

Contract is owned by a process.

inherited

The contract owner has exited abnormally and the contract has been inherited by the owner's process contract.

orphan

The contract owner has abandoned the contract, the contract owner exited abnormally and the contract was not inherited by the owner's process contract, or the process contract which had inherited the contract was abandoned by its owner.

dead

The contract is no longer active. It is removed from the system automatically when all references to it (open file descriptors, contract templates, and events) have been released.

HOLDER

If the contract is in the `owned` state, the pid of the process that owns the contract. If the contract is in the `inherited` state, the id of the regent process contract.

EVENTS

The number of unacknowledged critical events pending.

QTIME

The time until quantum ends, or - if no negotiation is in progress.

NTIME

The time until negotiation ends, or - if no negotiation is in progress.

示例

示例 1 Reporting on all Contracts in the System

The following example reports on all contracts in the system:

```
example% ctstat -a
```

CTID	TYPE	STATE	HOLDER	EVENTS	QTIME	NTIME
1	process	owned	100579	0	-	-

示例 1 Reporting on all Contracts in the System (续)

```

2      process dead -      1      -      -
3      process inherit 1    3      -      -
4      process orphan -    0      -      -

```

示例 2 Obtaining a Verbose Report of All Contracts in the System

The following example obtains a verbose report of all contracts in the system:

```
example% ctstat -av
```

```

CTID   TYPE      STATE   HOLDER  EVENTS  QTIME  NTIME
1      process  owned   100579  0       -       -
      informative event set: none
      critical event set:   hwerr core
      fatal event set:     hwerr
      parameter set:       none
      member processes:    100600 100601
      inherited ctids:     none
      service fmri:       svc:/system/init:default
      svc_fmri ctid:      1
      creator:            sched
      aux:
2      process  dead    -      1      -       -
      informative event set: none
      critical event set:   none
      fatal event set:     hwerr core
      parameter set:       pgrponly
      member processes:    none
      inherited ctids:     none
      service fmri:       svc:/system/power:default
      svc_fmri ctid:      19
      creator:            svc.startd
      aux:                start

```

退出状态

The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred.
- 2 Invalid arguments.

文件 /system/contract/*

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	See below.

The human readable output is Uncommitted. The invocation is Committed.

另请参见 [ctrun\(1\)](#), [ctwatch\(1\)](#), [contract\(4\)](#), [process\(4\)](#), [attributes\(5\)](#)

- 引用名** ctwatch – watch events in a contract or group of contracts
- 用法概要** /usr/bin/ctwatch [-f] [-r] [-v] *contract-type*... | *contract-id*...
- 描述** The ctwatch utility allows a user to observe the events occurring within a set of contracts or contract types. By default, ctwatch watches all contracts.
- 选项** The following options are supported:
- f Report events starting at the front of the event queue. Normally, ctwatch reports only events which occur after it has been invoked. With the -f option, any events that still exist in the contracts' event queues when ctwatch is invoked (for example, unacknowledged critical events) are also reported.
 - r Reliably watches all messages. Normally, the system may drop informative events and acknowledged critical events at any time, so ctwatch isn't guaranteed to see them all. This option may only be used if the ctwatch is invoked with the {PRIV_CONTRACT_EVENT} privilege asserted in its effective set.
 - v Request verbose event descriptions.
- 操作数** The following operands are supported:
- contract-type* Valid contract types are:
process Process contracts.
- contract-id* A valid contract id.
- Output** The following list defines the column headings and the meanings of a ctwatch report:
- | | |
|---------|--|
| CTID | The contract ID generating the event. |
| EVID | The event ID. |
| CRIT | Whether the event is informative, critical, or initiates an exit negotiation. Values are info, crit, or neg, respectively. |
| ACK | The event has been acknowledged. Values are yes or "no". |
| CTTYPE | The contract type. |
| SUMMARY | A type-specific summary of the event. |
- 示例** 示例 1 Watching a process contract
- ```
example% ctwatch -r 1
```
- | CTID | EVID | CRIT | ACK | CTTYPE  | SUMMARY                               |
|------|------|------|-----|---------|---------------------------------------|
| 1    | 2    | crit | no  | process | pid 100569 was created                |
| 1    | 3    | info | no  | process | pid 100569 encountered hardware error |
| 1    | 4    | info | no  | process | pid 100568 exited                     |
| 1    | 5    | info | no  | process | pid 100569 exited                     |



示例 1 Watching a process contract (续)

```
1 6 crit no process contract empty
```

退出状态

The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred.
- 2 Invalid arguments.

文件

/system/contract/\*

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | system/core-os  |
| Interface Stability | See below.      |

Human Readable Output is Uncommitted. Invocation is Committed.

另请参见

[ctrun\(1\)](#), [ctstat\(1\)](#), [contract\(4\)](#), [process\(4\)](#), [attributes\(5\)](#), [privileges\(5\)](#)

附注

Ordering of events is only guaranteed within a single contract, or within a single type when a type is specified.

ctwatch can only observe those events which are generated by contracts owned or authored by processes with the same effective user ID as ctwatch, unless the {PRIV\_CONTRACT\_OBSERVER} privilege is asserted in its effective set.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | cu – call another UNIX system                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 用法概要 | cu [-c <i>device</i>   -l <i>line</i> ] [-s <i>speed</i> ] [-b <i>bits</i> ] [-h] [-n] [-t] [-d] [-o   -e] [-L] [-C] [-H] <i>telno</i>   <i>systemname</i> [ <i>local-cmd</i> ]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 描述   | The command cu calls up another UNIX system, a terminal, or possibly a non-UNIX system. It manages an interactive conversation with possible transfers of files. It is convenient to think of cu as operating in two phases. The first phase is the connection phase in which the connection is established. cu then enters the conversation phase. The -d option is the only one that applies to both phases.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 选项   | <p>cu accepts many options. The -c, -l, and -s options play a part in selecting the medium. The remaining options are used in configuring the line.</p> <ul style="list-style-type: none"> <li>-b <i>bits</i>      Forces <i>bits</i> to be the number of bits processed on the line. <i>bits</i> is either 7 or 8. This allows connection between systems with different character sizes. By default, the character size of the line is set to the same value as the current local terminal, but the character size setting is affected by LC_CTYPE also.</li> <li>-c <i>device</i>    Forces cu to use only entries in the "Type" field (the first field in the /etc/uucp/Devices file) that match the user specified <i>device</i>, usually the name of a local area network.</li> <li>-C            Runs the <i>local-cmd</i> specified at the end of the command line instead of entering interactive mode. The stdin and stdout of the command that is run refer to the remote connection.</li> <li>-d            Prints diagnostic traces.</li> <li>-e            Sets an EVEN data parity. This option designates that EVEN parity is to be generated for data sent to the remote system.</li> <li>-h            Sets communication mode to half-duplex. This option emulates local echo in order to support calls to other computer systems that expect terminals to be set to half-duplex mode.</li> <li>-H            Ignores one hangup. This allows the user to remain in cu while the remote machine disconnects and places a call back to the local machine. This option should be used when connecting to systems with callback or dialback modems. Once the callback occurs subsequent hangups will cause cu to terminate. This option can be specified more than once. For more information about dialback configuration, see <a href="#">remote(4)</a> and 《系统管理指南：IP 服务》</li> <li>-l <i>line</i>      Specifies a device name to use as the communication line. This can be used to override the search that would otherwise take place for the first available line having the right speed. When the -l option is used without the -s option, the speed of a line is taken from the /etc/uucp/Devices file record in which <i>line</i> matches the second field (the Line field). When the -l and -s options are both</li> </ul> |

used together, cu will search the `/etc/uucp/Devices` file to check if the requested speed for the requested line is available. If so, the connection will be made at the requested speed, otherwise, an error message will be printed and the call will not be made. In the general case where a specified device is a directly connected asynchronous line (for instance, `/dev/term/a`), a telephone number (*telno*) is not required. The specified device need not be in the `/dev` directory. If the specified device is associated with an auto dialer, a telephone number must be provided.

- L Goes through the login chat sequence specified in the `/etc/uucp/Systems` file. For more information about the chat sequence, see 《[系统管理指南：IP 服务](#)》
- n Requests user prompt for telephone number. For added security, this option will prompt the user to provide the telephone number to be dialed, rather than taking it from the command line.
- o Sets an ODD data parity. This option designates that ODD parity is to be generated for data sent to the remote system.
- s *speed* Specifies the transmission speed (300, 1200, 2400, 4800, 9600, 19200, 38400). The default value is "Any" speed which will depend on the order of the lines in the `/etc/uucp/Devices` file.
- t Dials a terminal which has been set to auto answer. Appropriate mapping of carriage-return to carriage-return-line-feed pairs is set.

## 操作数

The following operands are supported:

- telno* When using an automatic dialler, specifies the telephone number with equal signs for secondary dial tone or minus signs placed appropriately for delays of 4 seconds.
- systemname* Specifies a uucp system name, which can be used rather than a telephone number; in this case, cu will obtain an appropriate direct line or telephone number from a system file.

## 用法

### Connection Phase

cu uses the same mechanism that `uucp(1C)` does to establish a connection. This means that it will use the uucp control files `/etc/uucp/Devices` and `/etc/uucp/Systems`. This gives cu the ability to choose from several different media to establish the connection. The possible media include telephone lines, direct connections, and local area networks (LAN). The `/etc/uucp/Devices` file contains a list of media that are available on your system. The `/etc/uucp/Systems` file contains information for connecting to remote systems, but it is not generally readable.

*Note:* cu determines which `/etc/uucp/Systems` and `/etc/uucp/Devices` files to use based upon the name used to invoke cu. In the simple case, this name will be "cu", but you could also have created a link to cu with another name, such as "pppcu", in which case cu would then look for a "service=pppcu" entry in the `/etc/uucp/Sysfiles` file to determine which `/etc/uucp/Systems` file to use.

The *telno* or *systemname* parameter from the command line is used to tell cu what system you wish to connect to. This parameter can be blank, a telephone number, a system name, or a LAN specific address.

|                  |                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| telephone number | A telephone number is a string consisting of the tone dial characters (the digits 0 through 9, *, and #) plus the special characters = and -. The equal sign designates a secondary dial tone and the minus sign creates a 4 second delay. |
| system name      | A system name is the name of any computer that uucp can call; the <code>uname(1C)</code> command prints a list of these names.                                                                                                             |
| LAN address      | The documentation for your LAN will show the form of the LAN specific address.                                                                                                                                                             |

If cu's default behavior is invoked (not using the `-c` or `-l` options), cu will use the *telno* or *systemname* parameter to determine which medium to use. If a telephone number is specified, cu will assume that you wish to use a telephone line and it will select an automatic call unit (ACU). Otherwise, cu will assume that it is a system name. cu will follow the uucp calling mechanism and use the `/etc/uucp/Systems` and `/etc/uucp/Devices` files to obtain the best available connection. Since cu will choose a speed that is appropriate for the medium that it selects, you may not use the `-s` option when this parameter is a system name.

The `-c` and `-l` options modify this default behavior. `-c` is most often used to select a LAN by specifying a Type field from the `/etc/uucp/Devices` file. You must include either a *telno* or *systemname* value when using the `-c` option. If the connection to *systemname* fails, a connection will be attempted using *systemname* as a LAN specific address. The `-l` option is used to specify a device associated with a direct connection. If the connection is truly a direct connection to the remote machine, then there is no need to specify a *systemname*. This is the only case where a *telno* or *systemname* parameter is unnecessary. On the other hand, there may be cases in which the specified device connects to a dialer, so it is valid to specify a telephone number. The `-c` and `-l` options should not be specified on the same command line.

#### Conversation Phase

After making the connection, cu runs as two processes. The *transmit* process reads data from the standard input and, except for lines beginning with ~, passes it to the remote system. The *receive* process accepts data from the remote system and, except for lines beginning with ~, passes it to the standard output. Normally, an automatic DC3/DC1 protocol is used to control input from the remote so the buffer is not overrun. Lines beginning with ~ have special meanings.

## Commands

The *transmit* process interprets the following user initiated commands:

|                                  |                                                                                                                                                                                                         |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ~.                               | Terminates the conversation.                                                                                                                                                                            |
| ~!                               | Escapes to an interactive shell on the local system.                                                                                                                                                    |
| ~! <i>cmd</i> . . .              | Runs <i>cmd</i> on the local system (via <code>sh -c</code> ).                                                                                                                                          |
| ~\$ <i>cmd</i> . . .             | Runs <i>cmd</i> locally and send its output to the remote system.                                                                                                                                       |
| ~%cd                             | Changes the directory on the local system. Note: ~! cd will cause the command to be run by a sub-shell, probably not what was intended.                                                                 |
| ~%take <i>from</i> [ <i>to</i> ] | Copies file <i>from</i> (on the remote system) to file <i>to</i> on the local system. If <i>to</i> is omitted, the <i>from</i> argument is used in both places.                                         |
| ~%put <i>from</i> [ <i>to</i> ]  | Copies file <i>from</i> (on local system) to file <i>to</i> on remote system. If <i>to</i> is omitted, the <i>from</i> argument is used in both places.                                                 |
| ~~ <i>line</i>                   | Sends the line ~ <i>line</i> to the remote system.                                                                                                                                                      |
| ~%break                          | Transmits a BREAK to the remote system (which can also be specified as ~%b).                                                                                                                            |
| ~%debug                          | Toggles the -d debugging option on or off (which can also be specified as ~%d).                                                                                                                         |
| ~t                               | Prints the values of the termio structure variables for the user's terminal (useful for debugging).                                                                                                     |
| ~l                               | Prints the values of the termio structure variables for the remote communication line (useful for debugging).                                                                                           |
| ~%ifc                            | Toggles between DC3/DC1 input control protocol and no input control. This is useful when the remote system does not respond properly to the DC3 and DC1 characters (can also be specified as ~%nostop). |
| ~%ofc                            | Toggles the output flow control setting. When enabled, outgoing data may be flow controlled by the remote host (can also be specified as ~%nostop).                                                     |
| ~%divert                         | Allows/disallows unsolicited diversions. That is, diversions not specified by ~%take.                                                                                                                   |
| ~%old                            | Allows/disallows old style syntax for received diversions.                                                                                                                                              |
| ~%nostop                         | Same as ~%ifc.                                                                                                                                                                                          |

The *receive* process normally copies data from the remote system to the standard output of the local system. It may also direct the output to local files.

The use of `~%put` requires `stty(1)` and `cat(1)` on the remote side. It also requires that the current erase and kill characters on the remote system be identical to these current control characters on the local system. Backslashes are inserted at appropriate places.

The use of `~%take` requires the existence of `echo(1)` and `cat(1)` on the remote system, and that the remote system must be using the Bourne shell, `sh`. Also, `tabs` mode (see `stty(1)`) should be set on the remote system if tabs are to be copied without expansion to spaces.

When `cu` is used on system `X` to connect to system `Y` and subsequently used on system `Y` to connect to system `Z`, commands on system `Y` can be executed by using `~ ~`. Executing a tilde command reminds the user of the local system `uname`. For example, `uname` can be executed on `Z`, `X`, and `Y` as follows:

```
uname
Z
~[X]!uname
X
~~[Y]!uname
Y
```

In general, `~` causes the command to be executed on the original machine. `~ ~` causes the command to be executed on the next machine in the chain.

## 示例

示例 1 Dialling a system

To dial a system whose telephone number is 9 1 201 555 1234 using 1200 baud (where dialtone is expected after the 9):

```
example% cu -s 1200 9=12015551234
```

If the speed is not specified, "Any" is the default value.

示例 2 Logging in to a system on a direct line

To login to a system connected by a direct line:

```
example% cu -l /dev/term/b
```

or

```
example% cu -l term/b
```

示例 3 Dialling a system with specific line and speed

To dial a system with a specific line and speed:

```
example% cu -s 1200 -l term/b
```

示例 4 Using a system name

To use a system name:

```
example% cu systemname
```

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of cu: LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

## 退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

## 文件

/etc/uucp/Devices device file

/etc/uucp/Sysfiles system file

/etc/uucp/Systems system file

/var/spool/locks/\* lock file

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE      |
|----------------|----------------------|
| Availability   | service/network/uucp |

## 另请参见

[cat\(1\)](#), [echo\(1\)](#), [stty\(1\)](#), [tip\(1\)](#), [uname\(1\)](#), [ct\(1C\)](#), [uuname\(1C\)](#), [uucp\(1C\)](#), [remote\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#)

《系统管理指南：IP 服务》

## 附注

The cu utility takes the default action upon receipt of signals, with the exception of:

SIGHUP Close the connection and terminate.

SIGINT Forward to the remote system.

SIGQUIT Forward to the remote system.

SIGUSR1 Terminate the cu process without the normal connection closing sequence.

The cu command does not do any integrity checking on data it transfers. Data fields with special cu characters may not be transmitted properly. Depending on the interconnection hardware, it may be necessary to use a ~. to terminate the conversion, even if stty 0 has been used. Non-printing characters are not dependably transmitted using either the ~%put or ~%take commands. ~%put and ~%take cannot be used over multiple links. Files must be moved one link at a time.

There is an artificial slowing of transmission by `cu` during the `~%put` operation so that loss of data is unlikely. Files transferred using `~%take` or `~%put` must contain a trailing newline, otherwise, the operation will hang. Entering a Control-D command usually clears the hang condition.



---

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | cut – cut out selected fields of each line of a file                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 用法概要 | <pre>/usr/bin/cut -b <i>list</i> [-n] [<i>file</i>]...<br/>/usr/bin/cut -c <i>list</i> [<i>file</i>]...<br/>/usr/bin/cut -f <i>list</i> [-d <i>delim</i>] [-s] [<i>file</i>]...</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 描述   | <p>cut cuts bytes, characters, or character-delimited fields from one or more files, and concatenates them on standard output.</p> <p>The <i>option</i> argument list is a comma-separated or blank-separated list of positive numbers and ranges. Ranges can be of three forms. The first is two positive integers separated by a hyphen (low-high), which represents all fields from low to high. The second is a positive number preceded by a hyphen (<i>-high</i>), which represents all fields from field 1 to high. The last is a positive number followed by a hyphen (<i>low-</i>), which represents all fields from low to the last field, inclusive. Elements in the list can be repeated, can overlap, and can appear in any order. The order of the output is that of the input.</p> <p>One and only one of -b, -c, or -f options must be specified.</p> <p>If no file is given, or if the file is -, cut cuts from standard input. The start of the file is defined as the current offset.</p>                                                                                                                                                                                                                            |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"><li>-b <i>list</i><br/>The <i>list</i> following -b specifies byte positions (for instance, -b1-72 would pass the first 72 bytes of each line). When -b and -n are used together, <i>list</i> is adjusted so that no multi-byte character is split.</li><li>-c <i>list</i><br/>The list following -c specifies character positions (for instance, -c1-72 would pass the first 72 characters of each line).</li><li>-d <i>delim</i><br/>The character following -d is the field delimiter (-f option only). The default is TAB. Space or other characters with special meaning to the shell must be quoted. <i>delim</i> can be a multi-byte character.</li><li>-f <i>list</i><br/>The <i>list</i> following -f is a list of fields assumed to be separated in the file by a delimiter character (see -d); for instance, -f1,7 copies the first and seventh field only. Lines with no field delimiters are passed through intact (useful for table subheadings), unless -s is specified.</li><li>-n<br/>Do not split characters. When -b <i>list</i> and -n are used together, <i>list</i> is adjusted so that no multi-byte character is split.</li></ul> |

-s

Suppresses lines with no delimiter characters in case of -f option. Unless specified, lines with no delimiters is passed through untouched.

## 操作数

The following operands are supported:

*file* A path name of an input file. If no file operands are specified, or if a file operand is -, the standard input is used.

## 用法

See [largefile\(5\)](#) for the description of the behavior of cut when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

## 示例

示例 1 Mapping user IDs

A mapping of user IDs to names follows:

```
example% cut -d: -f1,5 /etc/passwd
```

示例 2 Setting the Current login name

To set name to current login name:

```
example$ name=$(who am i | cut -f1 -d' ')
```

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of cut: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

## 退出状态

The following exit values are returned:

0 All input files were output successfully.

>0 An error occurred.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTETYPE       | ATTRIBUTEVALUE                     |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

## 另请参见

[grep\(1\)](#), [paste\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | date – write the date and time                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 用法概要 | <pre> /usr/bin/date [-u] [+format] /usr/bin/date [-a [-]sss.fff] /usr/bin/date [-u] [ [mmdd] HHMM   mmddHHMM [cc] yy] [.SS] /usr/xpg4/bin/date [-u] [+format] /usr/xpg4/bin/date [-a [-]sss.fff] /usr/xpg4/bin/date [-u]     [ [mmdd] HHMM   mmddHHMM [cc] yy] [.SS] </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 描述   | <p>The date utility writes the date and time to standard output or attempts to set the system date and time. By default, the current date and time is written.</p> <p>Specifications of native language translations of month and weekday names are supported. The month and weekday names used for a language are based on the locale specified by the environment variable LC_TIME. See <a href="#">environ(5)</a>.</p> <p>The following is the default form for the C locale:</p> <pre>%a %b %e %T %Z %Y</pre> <p>For example,</p> <pre>Fri Dec 23 10:10:42 EST 1988</pre>                                                                                                                                                                                                                                        |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-a [-] <i>sss.fff</i>      Slowly adjust the time by <i>sss.fff</i> seconds (<i>fff</i> represents fractions of a second). This adjustment can be positive or negative. The system's clock is sped up or slowed down until it has drifted by the number of seconds specified. Only the super-user may adjust the time.</li> <li>-u                    Display (or set) the date in Greenwich Mean Time (GMT—universal time), bypassing the normal conversion to (or from) local time.</li> </ul>                                                                                                                                                                                                                                  |
| 操作数  | <p>The following operands are supported:</p> <ul style="list-style-type: none"> <li><i>+format</i>            If the argument begins with +, the output of date is the result of passing <i>format</i> and the current time to <code>strftime()</code>. date uses the conversion specifications listed on the <a href="#">strftime(3C)</a> manual page, with the conversion specification for %C determined by whether <code>/usr/bin/date</code> or <code>/usr/xpg4/bin/date</code> is used: <ul style="list-style-type: none"> <li><code>/usr/bin/date</code>            Locale's date and time representation. This is the default output for date.</li> <li><code>/usr/xpg4/bin/date</code>      Century (a year divided by 100 and truncated to an integer) as a decimal number [00-99].</li> </ul> </li> </ul> |

Additionally, `date` supports `%N` which represents nanosecond portion of the current time since Epoch (`00:00:00 UTC, January 1, 1970`) as a decimal number [`000000000-999999999`]. The conversion specification accepts an optional flag character, an optional field width, or both as specified in `strftime()` with a difference that, if a field width specified is less than nine, the actual date output contains only the specified amount of digits of the nanoseconds from left.

The string is always terminated with a NEWLINE. An argument containing blanks must be quoted; see the `EXAMPLES` section.

|           |                                                                                                                                                                                                                                        |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>mm</i> | Month number                                                                                                                                                                                                                           |
| <i>dd</i> | Day number in the month                                                                                                                                                                                                                |
| <i>HH</i> | Hour number (24 hour system)                                                                                                                                                                                                           |
| <i>MM</i> | Minute number                                                                                                                                                                                                                          |
| <i>SS</i> | Second number                                                                                                                                                                                                                          |
| <i>cc</i> | Century (a year divided by 100 and truncated to an integer) as a decimal number [00-99]. For example, <i>cc</i> is 19 for the year 1988 and 20 for the year 2007.                                                                      |
| <i>yy</i> | Last two digits of the year number. If century ( <i>cc</i> ) is not specified, then values in the range 69–99 shall refer to years 1969 to 1999 inclusive, and values in the range 00–68 shall refer to years 2000 to 2068, inclusive. |

The month, day, year number, and century may be omitted; the current values are applied as defaults. For example, the following entry:

```
example% date 10080045
```

sets the date to Oct 8, 12:45 a.m. The current year is the default because no year is supplied. The system operates in GMT. `date` takes care of the conversion to and from local standard and daylight time. Only the super-user may change the date. After successfully setting the date and time, `date` displays the new date according to the default format. The `date` command uses TZ to determine the correct time zone information; see [environ\(5\)](#).

## 示例

示例 1 Generating Output

The following command:

```
example% date '+DATE: %m/%d/%y%nTIME:%H:%M:%S'
```

generates as output

```
DATE: 08/01/76
```

```
TIME: 14:45:05
```

**示例 2** Setting the Current Time

The following command sets the current time to 12:34:56:

```
example# date 1234.56
```

**示例 3** Setting Another Time and Date in Greenwich Mean Time

The following command sets the date to January 1st, 12:30 am, 2000:

```
example# date -u 010100302000
```

This is displayed as:

```
Thu Jan 01 00:30:00 GMT 2000
```

**环境变量**

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `date`: LANG, LC\_ALL, LC\_CTYPE, LC\_TIME, LC\_MESSAGES, and NLSPATH.

**TZ** Determine the timezone in which the time and date are written, unless the `-u` option is specified. If the TZ variable is not set and the `-u` is not specified, the system default timezone is used.

**退出状态**

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/date

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |
| CSI            | Enabled         |

/usr/xpg4/bin/date

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/xopen/xcu4                  |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见**

[strftime\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

**诊断**

no permission You are not the super-user and you tried to change the date.

bad conversion The date set is syntactically incorrect.

## 附注

If you attempt to set the current date to one of the dates that the standard and alternate time zones change (for example, the date that daylight time is starting or ending), and you attempt to set the time to a time in the interval between the end of standard time and the beginning of the alternate time (or the end of the alternate time and the beginning of standard time), the results are unpredictable.

Using the `date` command from within windowing environments to change the date can lead to unpredictable results and is unsafe. It can also be unsafe in the multi-user mode, that is, outside of a windowing system, if the date is changed rapidly back and forth. The recommended method of changing the date is `'date -a'`.

Setting the system time or allowing the system time to progress beyond 03:14:07 UTC Jan 19, 2038 is not supported on Solaris.

引用名 dc – desk calculator

用法概要 `/usr/bin/dc [filename]`  
`/usr/xpg6/bin/dc [filename]`

描述 dc is an arbitrary precision arithmetic package. Ordinarily it operates on decimal integers, but one may specify an input base, output base, and a number of fractional digits to be maintained. The overall structure of dc is a stacking (reverse Polish) calculator. If an argument is given, input is taken from that file until its end, then from the standard input.

bc is a preprocessor for dc that provides infix notation and a C-like syntax that implements functions. bc also provides reasonable control structures for programs. See [bc\(1\)](#).

## 用法

`/usr/bin/dc`,  
`/usr/xpg6/bin/dc`

The following constructions are recognized under both `/usr/bin/dc` and `/usr/xpg6/bin/dc`:

- number* The value of the number is pushed on the stack. A number is an unbroken string of the digits 0–9. It may be preceded by an underscore (`_`) to input a negative number. Numbers may contain decimal points.
- sx* The top of the stack is popped and stored into a register named *x*, where *x* may be any character. If the *s* is capitalized, *x* is treated as a stack and the value is pushed on it.
- lx* The value in register *x* is pushed on the stack. The register *x* is not altered. All registers start with zero value. If the *l* is capitalized, register *x* is treated as a stack and its top value is popped onto the main stack.
- d* The top value on the stack is duplicated.
- p* The top value on the stack is printed. The top value remains unchanged.
- P* Interprets the top of the stack as an ASCII string, removes it, and prints it.
- f* All values on the stack are printed.
- q* Exits the program. If executing a string, the recursion level is popped by two.
- Q* Exits the program. The top value on the stack is popped and the string execution level is popped by that value.
- x* Treats the top element of the stack as a character string and executes it as a string of dc commands.
- X* Replaces the number on the top of the stack with its scale factor.
- `[ ... ]` Puts the bracketed ASCII string onto the top of the stack.
- `<x >x =x` The top two elements of the stack are popped and compared. Register *x* is evaluated if they obey the stated relation.

|     |                                                                                                                                                                                                                                                                                                                                 |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| v   | Replaces the top element on the stack by its square root. Any existing fractional part of the argument is taken into account, but otherwise the scale factor is ignored.                                                                                                                                                        |
| !   | Interprets the rest of the line as a shell command.                                                                                                                                                                                                                                                                             |
| c   | All values on the stack are popped.                                                                                                                                                                                                                                                                                             |
| i   | The top value on the stack is popped and used as the number radix for further input.                                                                                                                                                                                                                                            |
| I   | Pushes the input base on the top of the stack.                                                                                                                                                                                                                                                                                  |
| o   | The top value on the stack is popped and used as the number radix for further output.                                                                                                                                                                                                                                           |
| O   | Pushes the output base on the top of the stack.                                                                                                                                                                                                                                                                                 |
| k   | The top of the stack is popped, and that value is used as a non-negative scale factor: the appropriate number of places are printed on output, and maintained during multiplication, division, and exponentiation. The interaction of scale factor, input base, and output base will be reasonable if all are changed together. |
| K   | Pushes the current scale factor on the top of the stack.                                                                                                                                                                                                                                                                        |
| z   | The stack level is pushed onto the stack.                                                                                                                                                                                                                                                                                       |
| Z   | Replaces the number on the top of the stack with its length.                                                                                                                                                                                                                                                                    |
| ?   | A line of input is taken from the input source (usually the terminal) and executed.                                                                                                                                                                                                                                             |
| Y   | Displays dc debugging information.                                                                                                                                                                                                                                                                                              |
| ; : | Used by <code>bc(1)</code> for array operations.                                                                                                                                                                                                                                                                                |

`/usr/bin/dc`

The following construction is recognized under `/usr/bin/dc`, using the scale of whatever the result is.

`+ - / * % ^` The top two values on the stack are added (+), subtracted (-), multiplied (\*), divided (/), remaindered (%), or exponentiated (^). The two entries are popped off the stack; the result is pushed on the stack in their place. Any fractional part of an exponent is ignored.

`/usr/xpg6/bin/dc`

The following construction is recognized under `/usr/xpg6/bin/dc`. The results of division are forced to be a scale of 20.

`+ - / * % ^` The top two values on the stack are added (+), subtracted (-), multiplied (\*), divided (/), remaindered (%), or exponentiated (^). The two entries are popped off the stack. The result is pushed on the stack in their place. Any fractional part of an exponent is ignored.



Ensures that the scale set prior to division is the scale of the result.

## 示例

示例 1 Printing the first ten values of n!

This example prints the first ten values of n!:

```
[!a1+dsa*pla10>y]sy
0sa1
lyx
```

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

## 另请参见

[bc\(1\)](#), [attributes\(5\)](#)

## 诊断

|                                 |                                                                         |
|---------------------------------|-------------------------------------------------------------------------|
| <code>x</code> is unimplemented | <code>x</code> is an octal number.                                      |
| out of space                    | The free list is exhausted (too many digits).                           |
| out of stack space              | Too many pushes onto the stack (stack overflow).                        |
| empty stack                     | Too many pops from the stack (stack underflow).                         |
| nesting depth                   | Too many levels of nested execution.                                    |
| divide by 0                     | Division by zero.                                                       |
| sqrt of neg number              | Square root of a negative number is not defined (no imaginary numbers). |
| exp not an integer              | dc only processes integer exponentiation.                               |
| exp too big                     | The largest exponent allowed is 999.                                    |
| input base is too large         | The input base <code>x</code> : $2 \leq x \leq 16$ .                    |
| input base is too small         | The input base <code>x</code> : $2 \leq x \leq 16$ .                    |
| output base is too large        | The output base must be no larger than <code>BC_BASE_MAX</code> .       |
| invalid scale factor            | Scale factor cannot be less than 1.                                     |
| scale factor is too large       | A scale factor cannot be larger than <code>BC_SCALE_MAX</code> .        |
| symbol table overflow           | Too many variables have been specified.                                 |
| invalid index                   | Index cannot be less than 1.                                            |
| index is too large              | An index cannot be larger than <code>BC_DIM_MAX</code> .                |

**引用名**            deallocate – 设备解除分配

**用法概要**        deallocate [-s] [-w] [-F] [-z zonename]  
                  [-c dev-class | -g dev-type | device]  
  
deallocate [-s] [-w] [-F] [-z zonename] -I

**描述**            deallocate 命令用于释放已分配的设备。它会重置与设备关联的所有设备特殊文件的所有权和权限，同时禁止对该设备的访问。deallocate 会对 [device\\_allocate\(4\)](#) 中所指定的设备运行设备清除程序。

缺省 deallocate 操作会解除分配已分配给用户的设备。

**选项**            支持以下选项：

- c *dev-class*    取消分配指定设备类的所有设备。
- F *device*       强制解除分配与 *device* 所指定的文件关联的设备。仅允许具有 `solaris.device.revoke` 授权的用户使用此选项。
- I               强制解除分配所有可分配设备。仅允许具有 `solaris.device.revoke` 授权的用户使用此选项。此选项应仅在系统初始化时使用。
- s               无提示。抑制任何诊断信息的输出。

以下选项在系统上配置有 Trusted Extensions 时受支持：

- g *dev-type*    取消分配设备类型与 *dev-type* 匹配的设备。
- w               在窗口环境中运行设备清理程序。如果该程序的窗口版本存在，则会使用窗口版本。否则，将在终端窗口中运行标准版本。
- z *zonename*   取消分配 *zonename* 所指定的区域中的设备。

**操作数**         支持下列操作数：

*device*        取消分配指定的 *device*。

**退出状态**       将返回以下退出值：

- 0               成功完成。
- 20              对于指定的设备没有任何项。
- 其他值**       出现错误。

**文件**            /etc/security/device\_allocate  
  
                  /etc/security/device\_maps  
  
                  /etc/security/dev/\*  
  
                  /etc/security/lib/\*

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值            |
|-------|----------------|
| 可用性   | system/core-os |
| 接口稳定性 | 请参见下文。         |

调用为 "Uncommitted"（未确定）。选项为 "Uncommitted"（未确定）。输出为 "Not-an-Interface"（不是接口）。

另请参见

[allocate\(1\)](#)、[list\\_devices\(1\)](#)、[device\\_allocate\(1M\)](#)、[dminfo\(1M\)](#)、[mkdevalloc\(1M\)](#)、[mkdev](#)

“控制对设备的访问”

附注

只有 Solaris 审计功能启用后，本手册页中描述的功能才可用。

只有 [device\\_allocate\(1M\)](#) 服务启用后，本手册页中描述的功能才可用。

在配置有 Trusted Extensions 的系统上，该功能是缺省启用的。

Solaris 操作环境的将来发行版可能不再支持 `/etc/security/dev`、[mkdevalloc\(1M\)](#) 和 [mkdevmaps\(1M\)](#)。

**引用名** deroff – remove nroff/troff, tbl, and eqn constructs

**用法概要** deroff [-m [m | s | l]] [-w] [-i] [filename...]

**描述** deroff reads each of the *filenames* in sequence and removes all [troff\(1\)](#) requests, macro calls, backslash constructs, [eqn\(1\)](#) constructs (between .EQ and .EN lines, and between delimiters), and [tbl\(1\)](#) descriptions, perhaps replacing them with white space (blanks and blank lines), and writes the remainder of the file on the standard output. deroff follows chains of included files (.so and .nx troff commands); if a file has already been included, a .so naming that file is ignored and a .nx naming that file terminates execution. If no input file is given, deroff reads the standard input.

**选项** -m The -m option may be followed by an m, s, or l. The -mm option causes the macros to be interpreted so that only running text is output (that is, no text from macro lines.) The -ml option forces the -mm option and also causes deletion of lists associated with the mm macros.

-w If the -w option is given, the output is a word list, one “word” per line, with all other characters deleted. Otherwise, the output follows the original, with the deletions mentioned above. In text, a “word” is any string that *contains* at least two letters and is composed of letters, digits, ampersands (&), and apostrophes ('); in a macro call, however, a “word” is a string that *begins* with at least two letters and contains a total of at least three letters. Delimiters are any characters other than letters, digits, apostrophes, and ampersands. Trailing apostrophes and ampersands are removed from “words.”

-i The -i option causes deroff to ignore .so and .nx commands.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | text/doctools   |

**另请参见** [eqn\(1\)](#), [nroff\(1\)](#), [tbl\(1\)](#), [troff\(1\)](#), [attributes\(5\)](#)

**附注** deroff is not a complete troff interpreter, so it can be confused by subtle constructs. Most such errors result in too much rather than too little output.

The -ml option does not handle nested lists correctly.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | df – display status of disk space on file systems                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 用法概要 | <code>/usr/ucb/df [-a] [-i] [-t type] [filesystem...]<br/>[filename...]</code>                                                                                                                                                                                                                                                                                                                                                                                          |
| 描述   | <p>The <code>df</code> utility displays the amount of disk space occupied by currently mounted file systems, the amount of used and available space, and how much of the file system's total capacity has been used.</p> <p>If arguments to <code>df</code> are path names, <code>df</code> produces a report on the file system containing the named file. Thus <code>'df .'</code> shows the amount of space on the file system containing the current directory.</p> |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-a Report on all filesystems including the uninteresting ones which have zero total blocks (that is, auto-mounter).</li> <li>-i Report the number of used and free inodes. Print <code>'*'</code> if no information is available.</li> <li>-t <i>type</i> Report on filesystems of a given type (for example, <code>nfs</code> or <code>ufs</code>).</li> </ul>                      |

## 示例

示例1 Using `df`

A sample of output for `df` looks like:

```
example% df
Filesystem kbytes used avail capacity Mounted on
sparky:/ 7445 4714 1986 70% /
sparky:/usr 42277 35291 2758 93% /usr
```

Note that `used+avail` is less than the amount of space in the file system (kbytes); this is because the system reserves a fraction of the space in the file system to allow its file system allocation routines to work well. The amount reserved is typically about 10%; this can be adjusted using `tunefs` (see [tunefs\(1M\)](#)). When all the space on a file system except for this reserve is in use, only the super-user can allocate new files and data blocks to existing files. When a file system is overallocated in this way, `df` can report that the file system is more than 100% utilized.

|    |                                                                          |
|----|--------------------------------------------------------------------------|
| 文件 | <code>/etc/mnttab</code> List of file systems currently mounted          |
|    | <code>/etc/vfstab</code> List of default parameters for each file system |

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

另请参见

[du\(1\)](#), [quot\(1M\)](#), [tunefs\(1M\)](#), [mnttab\(4\)](#), [attributes\(5\)](#)

**引用名**            dhcpcinfo – 显示通过 DHCP 接收的参数的值

**用法概要**        dhcpcinfo [-c] [-i *interface*] [-n *limit*] [-v 4|6] *code*

                  dhcpcinfo [-c] [-i *interface*] [-n *limit*] [-v 4|6] *identifier*

**描述**            dhcpcinfo 实用程序可显示 DHCP 提供的命令行上所请求的参数的值。可通过 DHCP 规范中的数字代码或通过助记标识符（如 `dhcp_inittab(4)` 中所列）来标识参数。此命令旨在用于 `init(1M)` 在系统引导时调用的 shell 脚本中的命令替换。它首先会在系统引导时或在事件脚本中联系 DHCP 客户机守护进程，如 `dhcpcagent(1M)` 中所述。它首先会联系 DHCP 客户机守护进程 `dhcpcagent(1M)`，以验证 DHCP 在请求的接口上是否已成功完成。如果 DHCP 在请求的接口上已成功完成，`dhcpcinfo` 将检索所请求参数的值。对于由 `dhcpcinfo` 回显的参数值，在未检查其退出状态的情况下不应直接使用。请参见 `exit(1)`。

有关所有 DHCP 参数的助记标识符代码的列表，请参见 `dhcp_inittab(4)`。有关 DHCPv4 参数的更多详细信息，请参见《DHCP Options and BOOTP Vendor Extensions》(RFC 2132)，有关 DHCPv6 参数的更多详细信息，请参见《Dynamic Host Configuration Protocol for IPv6, DHCPv6》(RFC 3315)。

**输出格式**        来自 `dhcpcinfo` 的输出由一行或多行 ASCII 文本组成；输出的格式取决于所请求的参数。给定参数的输出中每行返回的值数和总行数分别由参数的**粒度值**和**最大值**确定，如 `dhcp_inittab(4)` 所定义。

每个值的格式由选项的数据类型确定，如 `dhcp_inittab(4)` 所确定的那样。下面列出了可能的数据类型及其格式：

| 数据类型    | 格式                     | <code>dhcp_inittab(4)</code> 类型        |
|---------|------------------------|----------------------------------------|
| 无符号数    | 一个或多个十进制数字             | UNUMBER8、UNUMBER16、UNUMBER32、UNUMBER64 |
| 带符号数    | （前面可能带有减号的）一个或多个十进制数字  | SNUMBER8、SNUMBER16、SNUMBER32、SNUMBER64 |
| IP 地址   | 点分十进制表示法               | IP                                     |
| IPv6 地址 | 以冒号分隔表示法               | IPv6                                   |
| 八位字节    | 字符串 0x 后跟一个两位数十六进制值    | OCTET                                  |
| 字符串     | 零个或多个 ASCII 字符         | ASCII                                  |
| DUID    | DHCP 唯一标识符文本           | DUID                                   |
| 域名      | 以句点分隔的标准域名，RFC 1035 格式 | DOMAIN                                 |

**选项** 支持以下选项：

`-c` 以规范格式显示输出。此格式与粒度为 1 的 OCTET 格式完全相同。

`-i interface` 指定从其检索 DHCP 参数值的接口。如果未指定此选项，则使用主接口。

如果尚未通过 `ifconfig(1M)` 为系统或通过 `-i` 为此命令选择主接口，系统将自动选择一个接口作为当前命令调用的主接口。系统会选择其名称按词汇顺序排在首位且附带有 DHCP 参数的接口。此选择不会影响系统状态。使用 `ifconfig(1M)` 设置主接口。

`dhcpageant(1M)` eventhook 脚本中的建议做法是使用 `-i` 指定所需接口，而不是依赖于主接口选择。

对于 DHCPv6，所用的接口名称应该是物理接口的名称，而不是由 `dhcpageant` 创建的某个逻辑接口。

`-n limit` 将值列表的显示行数限制到 *limit* 行。

`-v4 | 6` 指定要查询的 DHCP 版本。对 DHCPv4 使用 `-v4`，对 DHCPv6 使用 `-v6`。

**操作数** 支持下列操作数：

*code* 所请求 DHCP 参数的数字代码，如 DHCP 规范所定义。通过向 DHCPv4 的实际供应商代码添加 256，向 DHCPv6 的实际供应商代码添加 65536，指定供应商选项。

**标识符** 所请求 DHCP 参数的助记符号，如 `dhcp_inittab(4)` 中所列。

**退出状态** 将返回以下退出值：

0 操作成功。

2 操作失败。DHCP 客户机守护进程可能未在运行，接口可能配置失败，或者没有接收到满意的 DHCP 响应。

3 参数错误。

4 操作超时。

6 系统错误（永远不应发生）。

**属性** 有关下列属性的说明，请参见 `attributes(5)`：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |



| 属性类型  | 属性值             |
|-------|-----------------|
| 接口稳定性 | Committed (已确定) |

### 另请参见

[dhcagent\(1M\)](#)、[ifconfig\(1M\)](#)、[init\(1M\)](#)、[dhcp\\_inittab\(4\)](#)、[attributes\(5\)](#)

由 Alexander, S. 和 R. Droms 合著的《DHCP Options and BOOTP Vendor Extensions》，RFC 2132，Silicon Graphics, Inc. 与 Bucknell University 出版，1997 年 3 月。

由 Droms, R. 编著的《Dynamic Host Configuration Protocol for IPv6 (DHCPv6)》，RFC 3315，Cisco Systems 出版，2003 年 7 月。

由 Mockapetris, P.V. 编著的《Domain names - implementation and specification》，RFC 1035，ISI 出版，1987 年 11 月。

**引用名**

diff – compare two files

**用法概要**

```
diff [-bitw] [-c | -e | -f | -h | -n | -u] file1 file2
```

```
diff [-bitw] [-C number | -U number] file1 file2
```

```
diff [-bitw] [-D string] file1 file2
```

```
diff [-bitw] [-c | -e | -f | -h | -n | -u] [-l] [-r] [-s]
[-S name] directory1 directory2
```

**描述**

The `diff` utility compares the contents of *file1* and *file2* and write to standard output a list of changes necessary to convert *file1* into *file2*. This list should be minimal. Except in rare circumstances, `diff` finds a smallest sufficient set of file differences. No output is produced if the files are identical.

The normal output contains lines of these forms:

```
n1 a n3,n4
n1,n2 d n3
n1,n2 c n3,n4
```

where *n1* and *n2* represent lines *file1* and *n3* and *n4* represent lines in *file2*. These lines resemble `ed(1)` commands to convert *file1* to *file2*. By exchanging a `f` for `d` and reading backward, *file2* can be converted to *file1*. As in `ed`, identical pairs, where *n1*=*n2* or *n3*=*n4*, are abbreviated as a single number.

Following each of these lines come all the lines that are affected in the first file flagged by '`<`', then all the lines that are affected in the second file flagged by '`>`'.

**选项**

The following options are supported:

- b Ignores trailing blanks (spaces and tabs) and treats other strings of blanks as equivalent.
- i Ignores the case of letters. For example, '`A`' compares equal to '`a`'.
- t Expands TAB characters in output lines. Normal or `-c` output adds character(s) to the front of each line that can adversely affect the indentation of the original source lines and make the output lines difficult to interpret. This option preserves the original source's indentation.
- w Ignores all blanks (SPACE and TAB characters) and treats all other strings of blanks as equivalent. For example, '`if ( a = = b )`' compares equal to '`if(a= =b)`'.

The following options are mutually exclusive:

- c Produces a listing of differences with three lines of context. With this option, output format is modified slightly. That is, output begins with identification of the files involved and their creation dates, then each change is separated by a line with a dozen `*`'s. The lines removed from *file1* are marked with '`-`'. The lines added to *file2* are marked '`+`'. Lines that are changed from one file to the other are marked in both files with '`!`'.

- C *number*** Produces a listing of differences identical to that produced by `-c` with *number* lines of context.
- D *string*** Creates a merged version of *file1* and *file2* with C preprocessor controls included so that a compilation of the result without defining *string* is equivalent to compiling *file1*, while defining *string* yields *file2*.
- e** Produces a script of only `a`, `c`, and `d` commands for the editor `ed`, which recreates *file2* from *file1*. In connection with the `-e` option, the following shell program can help maintain multiple versions of a file. Only an ancestral file (`$1`) and a chain of version-to-version `ed` scripts (`$2,$3,...`) made by `diff` need be on hand. A “latest version” appears on the standard output.
- ```
(shift; cat $*; echo ' 1,$p') | ed - $1
```
- f** Produces a similar script, not useful with `ed`, in the opposite order.
- h** Does a fast, uninspired job.
- This option only works when changed stretches are short and well-separated. It does work on files of unlimited length.
- Only `--b` is available with `-h`.
- `diff` does not descend into directories with this option.
- n** Produces a script similar to `-e`, but in the opposite order and with a count of changed lines on each insert or delete command.
- u** Produces a listing of differences with three lines of context. The output is similar to that of the `-c` option, except that the context is “unified”. Removed and changed lines in *file1* are marked by a `'-'` while lines added or changed in *file2* are marked by a `'+'`. Both versions of changed lines appear in the output, while added, removed, and context lines appear only once. The identification of *file1* and *file2* is different, with `“——”` and `“+++”` being printed where `“***”` and `“——”` would appear with the `-c` option. Each change is separated by a line of the form
- ```
@@ -n1, n2 +n3, n4 @@
```
- U *number*** Produces a listing of differences identical to that produced by `-u` with *number* lines of context.

The following options are used for comparing directories:

- l** Produces output in long format. Before the `diff`, each text file is piped through `pr(1)` to paginate it. Other differences are remembered and summarized after all text file differences are reported.
- r** Applies `diff` recursively to common subdirectories encountered.

-s Reports files that are identical. These identical files would not otherwise be mentioned.

-S *name* Starts a directory `diff` in the middle, beginning with the file *name*.

## 操作数

The following operands are supported:

*file1*  
*file2* A path name of a file or directory to be compared. If either *file1* or *file2* is `-`, the standard input is used in its place.

*directory1*  
*directory2* A path name of a directory to be compared.

If only one of *file1* and *file2* is a directory, `diff` is applied to the non-directory file and the file contained in the directory file with a filename that is the same as the last component of the non-directory file.

## 用法

See [largefile\(5\)](#) for the description of the behavior of `diff` when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

## 示例

示例 1 Using the `diff` Command

In the following command, `dir1` is a directory containing a directory named `x`, `dir2` is a directory containing a directory named `x`, `dir1/x` and `dir2/x` both contain files named `date.out`, and `dir2/x` contains a file named `y`:

```
example% diff -r dir1 dir2
Common subdirectories: dir1/x and dir2/x
```

```
Only in dir2/x: y
```

```
diff -r dir1/x/date.out dir2/x/date.out
```

```
1c1
```

```
< Mon Jul 2 13:12:16 PDT 1990
```

```

```

```
> Tue Jun 19 21:41:39 PDT 1990
```

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `diff`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, and `NLSPATH`.

`TZ` Determines the locale for affecting the timezone used for calculating file timestamps written with the `-C` and `-c` options.

- 退出状态**           The following exit values are returned:
- 0     No differences were found.
  - 1     Differences were found.
  - >1    An error occurred.
- 文件**                /tmp/d?????        Temporary file used for comparison
- /usr/lib/diffh     Executable file for the -h option
- 属性**                See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见**           [bdiff\(1\)](#), [cmp\(1\)](#), [comm\(1\)](#), [dircmp\(1\)](#), [ed\(1\)](#), [pr\(1\)](#), [sdiff\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

**附注**                Editing scripts produced under the -e or -f options are naive about creating lines consisting of a single period (.).

Missing NEWLINE at end of file indicates that the last line of the file in question did not have a NEWLINE. If the lines are different, they are flagged and output, although the output seems to indicate they are the same.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | diff3 – 3-way differential file comparison                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 用法概要 | diff3 [-exEX3] <i>filename1 filename2 filename3</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 描述   | <p>diff3 compares three versions of a file. It publishes disagreeing ranges of text flagged with the following codes:</p> <pre>====      all three files differ ====1    <i>filename1</i> is different ====2    <i>filename2</i> is different ====3    <i>filename3</i> is different</pre> <p>The type of change suffered in converting a given range of a given file to some other is indicated in one of the following ways:</p> <pre>f : n1 a      Text is to be appended after line number n1 in file f, where f = 1, 2, or 3. f : n1 , n2 c  Text is to be changed in the range line n1 to line n2. If n1 = n2, the range                 can be abbreviated to n1.</pre> <p>The original contents of the range follows immediately after a c indication. When the contents of two files are identical, the contents of the lower-numbered file is suppressed.</p> <p>The following command applies the resulting script to <i>filename1</i>.</p> <pre>(cat script; echo `1,\$'p`)   ed - <i>filename1</i></pre>                                                                                              |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-e Produce a script for the <a href="#">ed(1)</a> editor that incorporates into <i>filename1</i> all changes between <i>filename2</i> and <i>filename3</i> (that is, the changes that normally would be flagged ==== and ====3).</li> </ul> <p>Text lines that consist of a single dot ( . ) defeat the -e option.</p> <ul style="list-style-type: none"> <li>-E Produce a script that incorporates all changes between <i>filename2</i> and <i>filename3</i>, but treat overlapping changes (that is, changes that would be flagged with ==== in the normal listing) differently. The overlapping lines from both files are inserted by the edit script, bracketed by &lt;&lt;&lt;&lt;&lt;&lt; and &gt;&gt;&gt;&gt;&gt;&gt; lines.</li> <li>-x Produce a script to incorporate only changes flagged ====.</li> <li>-X Produce a script that incorporates only changes flagged ====, but treat these changes in the manner of the -E option.</li> <li>-3 Produce a script to incorporate only changes flagged ====3.</li> </ul> |

**用法** See [largefile\(5\)](#) for the description of the behavior of `diff3` when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

**文件** `/tmp/d3*`  
`/usr/lib/diff3prog`

**退出状态** The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred. A difference was found or there was a fatal error.
- >1 A fatal error occurred.

Return values do not work the same as [diff\(1\)](#) or other vendor's versions of `diff3`.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |
| CSI            | Enabled         |

**另请参见** [diff\(1\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

**附注** Files longer than 64 Kbytes do not work.

**引用名** diffmk – mark differences between versions of a troff input file

**用法概要** diffmk *oldfile newfile markedfile*

**描述** diffmk compares two versions of a file and creates a third version that includes “change mark” (.mc) commands for [nroff\(1\)](#) and [troff\(1\)](#). *oldfile* and *newfile* are the old and new versions of the file. diffmk generates *markedfile*, which, contains the text from *newfile* with [troff\(1\)](#) “change mark” requests (.mc) inserted where *newfile* differs from *oldfile*. When *markedfile* is formatted, changed or inserted text is shown by | at the right margin of each line. The position of deleted text is shown by a single \*.

**用法** See [largefile\(5\)](#) for the description of the behavior of diffmk when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

**示例** 示例 1 An example of the diffmk command.

diffmk can also be used in conjunction with the proper troff requests to produce program listings with marked changes. In the following command line:

```
example% diffmk old.c new.c marked.c ; nroff reqs marked.c | pr
```

the file reqs contains the following troff requests:

```
.pl 1
.ll 77
.nf
.eo
.nh
```

which eliminate page breaks, adjust the line length, set no-fill mode, ignore escape characters, and turn off hyphenation, respectively.

If the characters | and \* are inappropriate, you might run *markedfile* through [sed\(1\)](#) to globally change them.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | text/doctools   |

**另请参见** [diff\(1\)](#), [nroff\(1\)](#), [sed\(1\)](#), [troff\(1\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

**已知问题** Aesthetic considerations may dictate manual adjustment of some output. File differences involving only formatting requests may produce undesirable output, that is, replacing .sp 2 will produce a “change mark” on the preceding or following line of output.



|      |                                                                                                                                                                                                                                             |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | <code>digest</code> – 计算消息摘要                                                                                                                                                                                                                |
| 用法概要 | <code>/usr/bin/digest -l   [-v] -a <i>algorithm</i> [<i>file</i>]...</code>                                                                                                                                                                 |
| 描述   | <code>digest</code> 实用程序使用指定算法计算给定文件或 <code>stdin</code> 的消息摘要。如果指定多个文件，则输出的每一行都是一个文件的摘要。                                                                                                                                                   |
| 选项   | 支持以下选项： <ul style="list-style-type: none"> <li><code>-a <i>algorithm</i></code> 指定加密或解密过程中要使用的算法的名称。有关详细信息，请参见“用法”部分的“算法”。</li> <li><code>-l</code> 显示系统上可用的算法的列表。此列表可依加密框架的配置而变化。</li> <li><code>-v</code> 详细输出。在输出中包括算法名称和文件名。</li> </ul> |

## 用法

**算法** 这些算法由加密框架提供。该命令支持的每种算法都是 PKCS #11 机制的一个别名，这样有助于简化访问。例如，`sha1` 是 `CKM_SHA_1` 的别名。

这些别名与 `-a` 选项一起使用，并且区分大小写。

## 示例

### 示例1 模拟输出

以下示例模拟通用的 `md5sum` 程序的输出：

```
example$ digest -v -a md5 /usr/bin/vi
md5 (/usr/bin/vi) = e4e3588c5212903847c66d36b1a828a5
```

### 示例2 生成文件摘要

以下示例生成文件 `/etc/motd` 的 `sha1` 摘要：

```
example$ digest -a sha1 /etc/motd
9498a4f5303d056ad3ecae826b59f41448d63790
```

### 示例3 生成目录清单

以下示例生成 `sha1` 的目录清单：

```
example$ digest -v -a sha1 /usr/lib/inet/*
sha1 (/usr/lib/inet/certdb) = f6d43e6e395d50db24d34e4af4828598c8918b16
sha1 (/usr/lib/inet/certlocal) = 7f74ba4a019b809c7023212b4bda10d9485e071d
sha1 (/usr/lib/inet/certrldb) = 1f845d30b8d02066647de04311e74549049852ed
sha1 (/usr/lib/inet/dhcp) = e3db5e4ff40a69d13f2497254526c2015d2c37b3
sha1 (/usr/lib/inet/dsvclockd) = b61aad7ed6a0f82145c3c26aedc613ab4a1f032e
sha1 (/usr/lib/inet/in.dhcpd) = 382210180c826fbb2e747236c489062bac8cc30b
sha1 (/usr/lib/inet/in.iked) = be6061fad725d37256e773dc85f8bd5248649463
sha1 (/usr/lib/inet/in.mpathd) = 5bd6bf0340fd5c4cc0c53f2df158302a0e85f9d0
sha1 (/usr/lib/inet/in.ndpd) = fdb768aebe7e5eb4465e1c1bb5e679b496f5c5c6
sha1 (/usr/lib/inet/in.ripngd) = 4f56a0df2d4a252f581a73c2e84143b920d0b66b
```

### 示例3 生成目录清单 (续)

```

sha1 (/usr/lib/inet/ncaconfd) = 7219542b5585a8d1104d7ce4a2ced07d8a260ea3
sha1 (/usr/lib/inet/ppp) = c96ee458549871a6ffdf2674a888b01d0c9e9740
sha1 (/usr/lib/inet/pppoec) = 5f022498d79dacacd947cddadc64f171822e3dee
sha1 (/usr/lib/inet/pppoed) = 252bd2f0863dbc1b05fffae72821a2a95609b8ad
sha1 (/usr/lib/inet/slpd) = dfa24cc0f0b05f790546d4f0948a9094f7089027
sha1 (/usr/lib/inet/wanboot) = a8b8c51c389c774d0be2ae43cb85d1b1439484ae
sha1 (/usr/lib/inet/ntpd) = 5b4aff102372cea801e7d08acde9655fec81f07c

```

### 示例4 显示可用算法的列表

以下示例显示可用于生成摘要的算法的列表：

```

example$ digest -l
sha1
md5
sha224
sha256
sha384
sha512

```

### 退出状态

将返回以下退出值：

- 0 成功完成。
- >0 出现错误。

### 属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值             |
|-------|-----------------|
| 可用性   | system/core-os  |
| 接口稳定性 | Committed (已确定) |

### 另请参见

[cksum\(1\)](#)、[encrypt\(1\)](#)、[mac\(1\)](#)、[bart\(1M\)](#)、[cryptoadm\(1M\)](#)、[libpkcs11\(3LIB\)](#)、[attributes\(5\)](#)、[pkcs](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | dircmp – directory comparison                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 用法概要 | dircmp [-ds] [-w <i>n</i> ] <i>dir1 dir2</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 描述   | The <code>dircmp</code> command examines <i>dir1</i> and <i>dir2</i> and generates various tabulated information about the contents of the directories. Listings of files that are unique to each directory are generated for all the options. If no option is entered, a list is output indicating whether the file names common to both directories have the same contents.                                                                                                                        |
| 选项   | The following options are supported: <ul style="list-style-type: none"> <li>-d      Compares the contents of files with the same name in both directories and output a list telling what must be changed in the two files to bring them into agreement. The list format is described in <a href="#">diff(1)</a>.</li> <li>-s      Suppresses messages about identical files.</li> <li>-w <i>n</i>   Changes the width of the output line to <i>n</i> characters. The default width is 72.</li> </ul> |
| 操作数  | The following operands are supported: <p><i>dir1</i></p> <p><i>dir2</i>    A path name of a directory to be compared.</p>                                                                                                                                                                                                                                                                                                                                                                            |
| 用法   | See <a href="#">largefile(5)</a> for the description of the behavior of <code>dircmp</code> when encountering files greater than or equal to 2 Gbyte ( $2^{31}$ bytes).                                                                                                                                                                                                                                                                                                                              |
| 环境变量 | See <a href="#">environ(5)</a> for descriptions of the following environment variables that affect the execution of <code>dircmp</code> : <code>LC_COLLATE</code> , <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>NLSPATH</code> .                                                                                                                                                                                                                                                    |
| 退出状态 | The following exit values are returned: <ul style="list-style-type: none"> <li>0      Successful completion.</li> <li>&gt;0    An error occurred. (Differences in directory contents are not considered errors.)</li> </ul>                                                                                                                                                                                                                                                                          |
| 属性   | See <a href="#">attributes(5)</a> for descriptions of the following attributes:                                                                                                                                                                                                                                                                                                                                                                                                                      |

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

另请参见 [cmp\(1\)](#), [diff\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#)

引用名           dis – object code disassembler

用法概要        dis [-onqCLV] [-d *sec*] [-D *sec*] [-F *function*]  
                  [-l *string*] [-t *sec*] *file*...

描述            The `dis` command produces an assembly language listing of *file*, which can be an object file or an archive of object files. The listing includes assembly statements and an octal or hexadecimal representation of the binary that produced those statements.

选项            Options are interpreted by the disassembler and can be specified in any order.

The following options are supported:

- C               Displays demangled C++ symbol names in the disassembly.
- d *sec*          Disassembles the named section as data, printing the offset of the data from the beginning of the section.
- D *sec*          Disassembles the named section as data, printing the actual address of the data.
- F *function*    Disassembles only the named function in each object file specified on the command line. The -F option can be specified multiple times on the command line.
- l *string*       Disassembles the archive file specified by *string*. For example, one would issue the command `dis -l x -l z` to disassemble `libx.a` and `libz.a`, which are assumed to be in `LIBDIR`.  
  
                  This option is obsolete and might be removed in a future release of Solaris.
- L               Invokes a lookup of C-language source labels in the symbol table for subsequent writing to standard output.  
  
                  This option is obsolete and might be removed in a future release of Solaris.
- n               Displays all addresses numerically. Addresses are displayed using symbolic names by default.
- o               Prints numbers in octal. The default is hexadecimal.
- q               Quiet mode. Does not print any headers or function entry labels.
- t *sec*          Disassembles the named section as text.
- V               Prints, on standard error, the version number of the disassembler being executed.  
  
                  This option is obsolete and might be removed in a future release of Solaris.

If the -d, -D, or -t options are specified, only those named sections from each user-supplied file is disassembled. Otherwise, all sections containing text is disassembled.

On output, a number enclosed in brackets at the beginning of a line, such as [5], indicates that the break-pointable line number starts with the following instruction. These line numbers is printed only if the file was compiled with additional debugging information.

### 操作数

The following operand is supported:

*file* A path name of an object file or an archive (see [ar\(1\)](#)) of object files.

### 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `dis`: `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

`LIBDIR` If this environment variable contains a value, use this as the path to search for the library. If the variable contains a null value, or is not set, it defaults to searching for the library under `/usr/lib`.

### 退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

### 文件

`/usr/lib` default `LIBDIR`

### 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | developer/base-developer-utilities |
| Interface Stability | See below.                         |

The human readable output is Uncommitted. The command line options are Committed.

### 另请参见

[ar\(1\)](#), [as\(1\)](#), [ld\(1\)](#), [a.out\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#)

### 诊断

The self-explanatory diagnostics indicate errors in the command line or problems encountered with the specified files.

|      |                                                                                                                                                                                                                                                                                                                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | disown – 用于将作业与当前 shell 解除关联的 ksh 内置函数                                                                                                                                                                                                                                                                                                                         |
| 用法概要 | disown [ <i>job ...</i> ]                                                                                                                                                                                                                                                                                                                                      |
| 描述   | <p>kshdisown 命令可阻止当前 shell 在终止登录会话时向每个指定作业发送 HUP 信号。</p> <p>如果省略 <i>job</i>，disown 将向最近启动或停止的后台作业发送 HUP 信号。</p>                                                                                                                                                                                                                                                |
| 操作数  | <p>支持下列操作数：</p> <p><i>job</i> 指定将对其执行 disown 操作的一个或多个作业。</p> <p>将 <i>job</i> 指定为以下之一：</p> <p><i>number</i> 指的是进程 ID。</p> <p><i>-number</i> 指的是进程组 ID。</p> <p><i>%number</i> 指的是作业号。</p> <p><i>%string</i> 指的是其名称以 <i>string</i> 开头的作业。</p> <p><i>%?string</i> 指的是其名称包含 <i>string</i> 的作业。</p> <p><i>%+</i> 或 <i>%%</i> 指的是当前作业。</p> <p><i>%-</i> 指的是前一个作业。</p> |
| 退出状态 | <p>0 成功完成。</p> <p>&gt;0 一个或多个指定的作业不存在。</p>                                                                                                                                                                                                                                                                                                                     |
| 示例   | <p>示例 1 对作业执行 disown 操作</p> <p>以下示例对作业 1 执行 disown 操作：</p> <pre>example% disown %1</pre>                                                                                                                                                                                                                                                                       |
| 作者   | David Korn, dgk@research.att.com                                                                                                                                                                                                                                                                                                                               |
| 属性   | 有关下列属性的说明，请参见 <a href="#">attributes(5)</a> ：                                                                                                                                                                                                                                                                                                                  |

| 属性类型  | 属性值               |
|-------|-------------------|
| 可用性   | system/core-os    |
| 接口稳定性 | Uncommitted (未确定) |

另请参见 [bg\(1\)](#)、[jobs\(1\)](#)、[ksh\(1\)](#)、[wait\(1\)](#)、[attributes\(5\)](#)

---

|      |                                               |
|------|-----------------------------------------------|
| 引用名  | dispgid – 显示所有有效组名的列表                         |
| 用法概要 | dispgid                                       |
| 描述   | dispgid 显示系统上所有组名的列表（每行显示一个组）。                |
| 退出状态 | 将返回以下退出值：<br>0 成功执行。<br>1 无法读取组文件。            |
| 属性   | 有关下列属性的说明，请参见 <a href="#">attributes(5)</a> ： |

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

另请参见 [attributes\(5\)](#)

- 引用名**            dispuid – 显示所有有效用户名的列表
- 用法概要**        dispuid
- 描述**            dispuid 显示系统上所有用户名的列表（每行显示一个名称）。
- 退出状态**        将返回以下退出值：
- 0     成功执行。
  - 1     无法读取口令文件。
- 属性**            有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

**另请参见**        [attributes\(5\)](#)



- 引用名** dos2unix – convert text file from DOS format to ISO format
- 用法概要** dos2unix [-ascii] [-iso] [-7]  
[-437 | -850 | -860 | -863 | -865] *originalfile convertedfile*
- 描述** The dos2unix utility converts characters in the DOS extended character set to the corresponding ISO standard characters.
- This command can be invoked from either DOS or SunOS. However, the filenames must conform to the conventions of the environment in which the command is invoked.
- If the original file and the converted file are the same, dos2unix will rewrite the original file after converting it.
- 选项** The following options are supported:
- ascii Removes extra carriage returns and converts end of file characters in DOS format text files to conform to SunOS requirements.
  - iso This is the default. It converts characters in the DOS extended character set to the corresponding ISO standard characters.
  - 7 Converts 8 bit DOS graphics characters to 7 bit space characters so that SunOS can read the file.
- On non-i386 systems, dos2unix will attempt to obtain the keyboard type to determine which code page to use. Otherwise, the default is US. The user may override the code page with one of the following options:
- 437 Use US code page
  - 850 Use multilingual code page
  - 860 Use Portuguese code page
  - 863 Use French Canadian code page
  - 865 Use Danish code page
- 操作数** The following operands are required:
- originalfile* The original file in DOS format that is being converted to ISO format.
- convertedfile* The new file in ISO format that has been converted from the original DOS file format.
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

另请参见

[unix2dos\(1\)](#), [ls\(1\)](#), [attributes\(5\)](#)

诊断

File *filename* not found, or no read permission

The input file you specified does not exist, or you do not have read permission. Check with the SunOS command, `ls -l` (see [ls\(1\)](#)).

Bad output filename *filename*, or no write permission

The output file you specified is either invalid, or you do not have write permission for that file or the directory that contains it. Check also that the drive is not write-protected.

Error while writing to temporary file

An error occurred while converting your file, possibly because there is not enough space on the current drive. Check the amount of space on the current drive using the DIR command. Also be certain that the default drive is write-enabled (not write-protected). Notice that when this error occurs, the original file remains intact.

Translated temporary file name = *filename*.

Could not rename temporary file to *filename*.

The program could not perform the final step in converting your file. Your converted file is stored under the name indicated on the second line of this message.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | dpost – troff postprocessor for PostScript printers                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 用法概要 | <pre>dpost [-c num] [-e num] [-m num] [-n num] [-o list]       [-w num] [-x num] [-y num] [-F dir] [-H dir]       [-L file] [-O] [-T name] [file]...</pre> <p>/usr/lib/lp/postscript/dpost</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 描述   | <p>dpost translates <i>files</i> created by <code>troff(1)</code> into PostScript and writes the results on the standard output. If no <i>files</i> are specified, or if <code>-</code> is one of the input <i>files</i>, the standard input is read.</p> <p>The <i>files</i> should be prepared by <code>troff</code>. The default font files in <code>/usr/lib/font/devpost</code> produce the best and most efficient output. They assume a resolution of 720 dpi, and can be used to format files by adding the <code>-Tpost</code> option to the <code>troff</code> call. Older versions of the <code>eqn</code> and <code>pic</code> preprocessors need to know the resolution that <code>troff</code> will be using to format the <i>files</i>. If those are the versions installed on your system, use the <code>-r720</code> option with <code>eqn</code> and <code>-T720</code> with <code>pic</code>.</p> <p>dpost makes no assumptions about resolutions. The first <code>x res</code> command sets the resolution used to translate the input <i>files</i>, the <code>DESC.out</code> file, usually <code>/usr/lib/font/devpost/DESC.out</code>, defines the resolution used in the binary font files, and the PostScript prologue is responsible for setting up an appropriate user coordinate system.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 选项   | <ul style="list-style-type: none"> <li><code>-c num</code> Print <i>num</i> copies of each page. By default only one copy is printed.</li> <li><code>-e num</code> Sets the text encoding level to <i>num</i>. The recognized choices are 0, 1, and 2. The size of the output file and print time should decrease as <i>num</i> increases. Level 2 encoding will typically be about 20 percent faster than level 0, which is the default and produces output essentially identical to previous versions of <code>dpost</code>.</li> <li><code>-m num</code> Magnify each logical page by the factor <i>num</i>. Pages are scaled uniformly about the origin, which is located near the upper left corner of each page. The default magnification is 1.0.</li> <li><code>-n num</code> Print <i>num</i> logical pages on each piece of paper, where <i>num</i> can be any positive integer. By default, <i>num</i> is set to 1.</li> <li><code>-o list</code> Print those pages for which numbers are given in the comma-separated <i>list</i>. The list contains single numbers <i>N</i> and ranges <i>N1</i>–<i>N2</i>. A missing <i>N1</i> means the lowest numbered page, a missing <i>N2</i> means the highest. The page range is an expression of logical pages rather than physical sheets of paper. For example, if you are printing two logical pages to a sheet, and you specified a range of 4, then two sheets of paper would print, containing four page layouts. If you specified a page range of 3-4, when requesting two logical pages to a sheet; then <i>only</i> page 3 and page 4 layouts would print, and they would appear on one physical sheet of paper.</li> <li><code>-p mode</code> Print <i>files</i> in either portrait or landscape <i>mode</i>. Only the first character of <i>mode</i> is significant. The default <i>mode</i> is portrait.</li> </ul> |

- w *num* Set the line width used to implement `troff` graphics commands to *num* points, where a point is approximately 1/72 of an inch. By default, *num* is set to 0.3 points.
- x *num* Translate the origin *num* inches along the positive x axis. The default coordinate system has the origin fixed near the upper left corner of the page, with positive x to the right and positive y down the page. Positive *num* moves everything right. The default offset is 0 inches.
- y *num* Translate the origin *num* inches along the positive y axis. Positive *num* moves text up the page. The default offset is 0.
- F *dir* Use *dir* as the font directory. The default *dir* is `/usr/lib/font`, and `dpost` reads binary font files from directory `/usr/lib/font/devpost`.
- H *dir* Use *dir* as the host resident font directory. Files in this directory should be complete PostScript font descriptions, and must be assigned a name that corresponds to the appropriate two-character `troff` font name. Each font file is copied to the output file only when needed and at most once during each job. There is no default directory.
- L *file* Use *file* as the PostScript prologue which, by default, is `/usr/lib/lp/postscript/dpost.ps`.
- O Disables PostScript picture inclusion. A recommended option when `dpost` is run by a spooler in a networked environment.
- T *name* Use font files for device *name* as the best description of available PostScript fonts. By default, *name* is set to `post` and `dpost` reads binary files from `/usr/lib/font/devpost`.

## 示例

示例1 Using the `dpost` Command

If the old versions of `eqn` and `pic` are installed on your system, you can obtain the best possible looking output by issuing a command line such as the following:

```
example% pic -T720 file | tbl | eqn -r720 | troff -mm -Tpost | dpost
```

Otherwise,

```
example% pic file | tbl | eqn | troff -mm -Tpost | dpost
```

should give the best results.

## 退出状态

The following exit values are returned:

0 Successful completion.

non-zero An error occurred.

文件

```

/usr/lib/font/devpost/*.out
/usr/lib/font/devpost/charlib/*
/usr/lib/lp/postscript/color.ps
/usr/lib/lp/postscript/draw.ps
/usr/lib/lp/postscript/forms.ps
/usr/lib/lp/postscript/ps.requests
/usr/lib/macros/pictures
/usr/lib/macros/color

```

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE                      |
|----------------|--------------------------------------|
| Availability   | print/lp/filter/postscript-lp-filter |

另请参见 [troff\(1\)](#), [attributes\(5\)](#)

附注 Output files often do not conform to Adobe's file structuring conventions.

Although `dpost` can handle files formatted for any device, emulation is expensive and can easily double the print time and the size of the output file. No attempt has been made to implement the character sets or fonts available on all devices supported by `troff`. Missing characters will be replaced by white space, and unrecognized fonts will usually default to one of the Times fonts (that is, R, I, B, or BI).

An `x res` command must precede the first `x init` command, and all the input *files* should have been prepared for the same output device.

Use of the `-T` option is not encouraged. Its only purpose is to enable the use of other PostScript font and device description files, that perhaps use different resolutions, character sets, or fonts.

Although level 0 encoding is the only scheme that has been thoroughly tested, level 2 is fast and may be worth a try.

**引用名**            du – summarize disk usage

**用法概要**        /usr/bin/du [-dorx] [-a | -s] [-h | -k | -m] [-H | -L]  
                  [file ...]

/usr/xpg4/bin/du [-dorx] [-a | -s] [-h | -k | -m] [-H | -L]  
                  [file ...]

**描述**            The du utility writes to standard output the size of the file space allocated to, and the size of the file space allocated to each subdirectory of, the file hierarchy rooted in each of the specified files. The size of the file space allocated to a file of type directory is defined as the sum total of space allocated to all files in the file hierarchy rooted in the directory plus the space allocated to the directory itself. This sum will include the space allocated to any extended attributes encountered.

Files with multiple links will be counted and written for only one entry. The directory entry that is selected in the report is unspecified. By default, file sizes are written in 512-byte units, rounded up to the next 512-byte unit.

/usr/xpg4/bin/du    When du cannot obtain file attributes or read directories (see [stat\(2\)](#)), it will report an error condition and the final exit status will be affected.

**选项**            The following options are supported for /usr/bin/du and /usr/xpg4/bin/du:

- a    In addition to the default output, report the size of each file not of type directory in the file hierarchy rooted in the specified file. Regardless of the presence of the -a option, non-directories given as *file* operands will always be listed.
- d    Do not cross filesystem boundaries. For example, the command, du -d / reports usage only on the root partition.
- h    All sizes are scaled to a human readable format, for example, 14K, 234M, 2.7G, or 3.0T. Scaling is done by repetitively dividing by 1024.
- H    If a symbolic link to a directory is specified on the command line, process the symbolic link by using the directory which the symbolic link references, rather than the link itself.
- k    Write the files sizes in units of 1024 bytes, rather than the default 512-byte units.
- L    Process symbolic links by using the file or directory which the symbolic link references, rather than the link itself.
- m    Write the files sizes in units of megabytes, rather than the default 512-byte units.
- o    Do not add child directories' usage to a parent's total. Without this option, the usage listed for a particular directory is the space taken by the files in that directory, as well as the files in all directories beneath it. This option does nothing if -s is used.

- r Generate diagnostic messages about unreadable directories and files whose status cannot be obtained. /usr/bin/du is silent if these conditions arise and -r is not specified. /usr/xpg4/bin/du acts as though -r is always specified.
- s Instead of the default output, report only the total sum for each of the specified files.
- x When evaluating file sizes, evaluate only those files that have the same device as the file specified by the file operand.

Specifying more than one of the options in the mutually exclusive pair, -H and -L, is not considered an error. The last option specified determines the output format.

Specifying more than one of the options in the mutually exclusive set of options -h, -k, and -m is not considered an error. The last option specified determines the output format.

### 操作数

The following operand is supported:

*file* The path name of a file whose size is to be written. If no *file* is specified, the current directory is used.

### Output

The output from du consists of the amount of the space allocated to a file and the name of the file.

### 用法

See [largefile\(5\)](#) for the description of the behavior of du when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

### 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of du: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

### 退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

### 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/du

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | system/core-os  |
| CSI                 | Enabled         |
| Interface Stability | Committed       |

/usr/xpg4/bin/du

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | system/xopen/xcu4 |

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| CSI                 | Enabled         |
| Interface Stability | Standard        |

**另请参见**

[ls\(1\)](#), [stat\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

《Oracle Solaris 管理：常见任务》

**附注**

A file with two or more links is counted only once. If, however, there are links between files in different directories where the directories are on separate branches of the file system hierarchy, du will count the excess files more than once.

Files containing holes will result in an incorrect block count.



|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | du – display the number of disk blocks used per directory or file                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 用法概要 | <code>/usr/ucb/du [-adkLr] [-o   -s] [filename]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 描述   | The <code>du</code> utility gives the number of kilobytes contained in all files and, recursively, directories within each specified directory or file <i>filename</i> . If <i>filename</i> is missing, '.' (the current directory) is used.<br><br>A file that has multiple links to it is only counted once.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 选项   | The following options are supported: <ul style="list-style-type: none"> <li>-a Generates an entry for each file.</li> <li>-d Does not cross file system boundaries. For example, <code>du -d /</code> reports usage only on the root partition.</li> <li>-k Writes the files sizes in units of 1024 bytes, rather than the default 512-byte units.</li> <li>-L Processes symbolic links by using the file or directory that the symbolic link references, rather than the link itself.</li> <li>-o Does not add child directories' usage to a parent's total. Without this option, the usage listed for a particular directory is the space taken by the files in that directory, as well as the files in all directories beneath it. This option does nothing if the <code>-s</code> option is used.</li> <li>-r Generates messages about directories that cannot be read, files that cannot be opened, and so forth, rather than being silent (the default).</li> <li>-s Only displays the grand total for each of the specified <i>filenames</i>.</li> </ul> |

Entries are generated only for each directory in the absence of options.

## 示例

示例 1 Showing usage of all subdirectories in a directory

This example uses `du` in a directory. The `pwd(1)` command was used to identify the directory, then `du` was used to show the usage of all the subdirectories in that directory. The grand total for the directory is the last entry in the display:

```
example% pwd
/usr/ralph/misc
example% du
5 ./jokes
33 ./squash
44 ./tech.papers/lpr.document
217 ./tech.papers/new.manager
401 ./tech.papers
144 ./memos
80 ./letters
388 ./window
93 ./messages
```

示例 1 Showing usage of all subdirectories in a directory (续)

```
15 ./useful.news
1211 .
```

## 环境变量

If any of the LC\_\* variables, that is, LC\_CTYPE, LC\_MESSAGES, LC\_TIME, LC\_COLLATE, LC\_NUMERIC, and LC\_MONETARY (see [environ\(5\)](#)), are not set in the environment, the operational behavior of du for each corresponding locale category is determined by the value of the LANG environment variable. If LC\_ALL is set, its contents are used to override both the LANG and the other LC\_\* variables. If none of the above variables is set in the environment, the "C" (U.S. style) locale determines how du behaves.

**LC\_CTYPE** Determines how du handles characters. When LC\_CTYPE is set to a valid value, du can display and handle text and filenames containing valid characters for that locale. du can display and handle Extended Unix Code (EUC) characters where any individual character can be 1, 2, or 3 bytes wide. du can also handle EUC characters of 1, 2, or more column widths. In the "C" locale, only characters from ISO 8859-1 are valid.

**LC\_MESSAGES** Determines how diagnostic and informative messages are presented. This includes the language and style of the messages, and the correct form of affirmative and negative responses. In the "C" locale, the messages are presented in the default form found in the program itself (in most cases, U.S. English).

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

## 另请参见

[pwd\(1\)](#), [df\(1M\)](#), [du\(1\)](#), [quot\(1M\)](#), [attributes\(5\)](#), [environ\(5\)](#)

## 附注

Filename arguments that are not directory names are ignored, unless you use -a.

If there are too many distinct linked files, du will count the excess files more than once.

---

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | dump – dump selected parts of an object file                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 用法概要 | <pre>dump [-aCcFghLorstV [-p]] [-T <i>index</i> [, <i>indexn</i>]] <i>filename</i>...</pre> <pre>dump [-afhorstL [-p] [v]] <i>filename</i>...</pre> <pre>dump [-hsr [-p] [-d <i>number</i> [, <i>numbern</i>]]] <i>filename</i>...</pre> <pre>dump [-hsrt [-p] [-n <i>name</i>]] <i>filename</i>...</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 描述   | <p>The dump utility dumps selected parts of each of its object <i>file</i> arguments.</p> <p>The dump utility is best suited for use in shell scripts, whereas the <code>elfdump(1)</code> command is recommended for more human-readable output.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 选项   | <p>This utility accepts both object files and archives of object files. It processes each file argument according to one or more of the following options:</p> <ul style="list-style-type: none"> <li>-a Dumps the archive header of each member of an archive.</li> <li>-c Dumps the string table(s).</li> <li>-C Dumps decoded C++ symbol table names.</li> <li>-f Dumps each file header.</li> <li>-g Dumps the global symbols in the symbol table of an archive.</li> <li>-h Dumps the section headers.</li> <li>-L Dumps dynamic linking information and static shared library information, if available.</li> <li>-o Dumps each program execution header.</li> <li>-r Dumps relocation information.</li> <li>-s Dumps section contents in hexadecimal.</li> <li>-t Dumps symbol table entries.</li> <li>-T <i>index</i></li> <li>-T <i>index1,index2</i> Dumps only the indexed symbol table entry defined by <i>index</i> or a range of entries defined by <i>index1</i>, <i>index2</i>.</li> <li>-V Prints version information.</li> </ul> <p>The following modifiers are used in conjunction with the options listed above to modify their capabilities.</p> <ul style="list-style-type: none"> <li>-d <i>number</i></li> <li>-d <i>number1,number2</i> Dumps the section number indicated by <i>number</i> or the range of sections starting at <i>number1</i> and ending at <i>number2</i>. This modifier can be used with -h, -s, and -r. When -d is used with -h or -s, the</li> </ul> |

argument is treated as the number of a section or range of sections. When `-d` is used with `-r`, the argument is treated as the number of the section or range of sections to which the relocation applies. For example, to print out all relocation entries associated with the `.text` section, specify the number of the section as the argument to `-d`. If `.text` is section number 2 in the file, `dump -r -d 2` prints all associated entries. To print out a specific relocation section, use `dump -s -n name` for raw data output, or `dump -sv -n name` for interpreted output.

`-n name`

Dumps information pertaining only to the named entity. This modifier can be used with `-h`, `-s`, `-r`, and `-t`. When `-n` is used with `-h` or `-s`, the argument is treated as the name of a section. When `-n` is used with `-t` or `-r`, the argument is treated as the name of a symbol. For example, `dump -t -n .text` dumps the symbol table entry associated with the symbol whose name is `.text`, where `dump -h -n .text` dumps the section header information for the `.text` section.

`-p`

Suppresses printing of the headings.

`-v`

Dumps information in symbolic representation rather than numeric. This modifier can be used with

`-a` (date, user id, group id)

`-f` (class, data, type, machine, version, flags)

`-h` (type, flags)

`-L` (value)

`-o` (type, flags)

`-r` (name, type)

`-s` (interpret section contents wherever possible)

`-t` (type, bind)

When `-v` is used with `-s`, all sections that can be interpreted, such as the string table or symbol table, is interpreted. For example, `dump -sv -n .symtab filename. . .` produces the same formatted output as `dump -tv filename. . .`, but `dump -s -n .symtab filename. . .` prints raw data in hexadecimal. Without additional modifiers, `dump -sv filename...` dumps all sections in the files, interpreting all those that it can and dumping the rest (such as `.text` or `.data`) as raw data.

---

The dump utility attempts to format the information it dumps in a meaningful way, printing certain information in character, hexadecimal, octal, or decimal representation as appropriate.

### 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE                    |
|----------------|------------------------------------|
| Availability   | developer/base-developer-utilities |

### 另请参见

[elfdump\(1\)](#), [elffile\(1\)](#), [file\(1\)](#), [nm\(1\)](#), [ar.h\(3HEAD\)](#), [a.out\(4\)](#), [attributes\(5\)](#)

**引用名** dumpcs – 显示当前语言环境的代码集表

**用法概要** dumpcs [-0123vw]

**描述** dumpcs 显示用户的当前语言环境中可列显字符及其十六进制代码值的列表。假定显示设备能够显示给定语言环境的字符。不带有任何选项的 dumpcs 会显示当前语言环境中可列显字符的完整列表。

若指定一个或多个数字选项，它将根据指定的数字按代码集编号顺序显示当前语言环境的一个或多个 EUC 代码集。每个不可列显字符由一个星号 "\*" 表示，并使用足够多的 ASCII 空格字符填满该代码集的列宽。

**选项**

- 0 显示 ASCII（或 EUC 主）代码集。
- 1 显示当前语言环境使用的 EUC 代码集 1（如果存在）。
- 2 显示当前语言环境使用的 EUC 代码集 2（如果存在）。
- 3 显示当前语言环境使用的 EUC 代码集 3（如果存在）。
- v 详细模式。通常，不可列显字符的范围会压缩显示到一行中。此选项可将每个不可列显字符显示为一行。
- w 使用相应宽度的字符值（进程代码）替换代码值。

**环境变量** 环境变量 LC\_CTYPE 和 LANG 用于控制整个 dumpcs 中的字符分类。在输入到 dumpcs 时，按该输入顺序检查这些环境变量。这表明 LANG 的新设置不会覆盖 LC\_CTYPE 的设置。当所有值都无效时，字符分类将缺省为 POSIX.1 "C" 语言环境。

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

**另请参见** [localedef\(1\)](#)、[attributes\(5\)](#)

**附注** dumpcs 仅可以处理 EUC 语言环境。

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | echo – echo arguments                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 用法概要 | <code>/usr/bin/echo [string]...</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 描述   | <p>The echo utility writes its arguments, separated by BLANKs and terminated by a NEWLINE, to the standard output. If there are no arguments, only the NEWLINE character is written.</p> <p>echo is useful for producing diagnostics in command files, for sending known data into a pipe, and for displaying the contents of environment variables.</p> <p>The C shell, the Korn shell, and the Bourne shell all have echo built-in commands, which, by default, is invoked if the user calls echo without a full pathname. See <a href="#">shell_builtins(1)</a>. sh's echo, ksh88's echo, ksh's echo, and <code>/usr/bin/echo</code> understand the back-slashed escape characters, except that sh's echo does not understand <code>\a</code> as the alert character. In addition, ksh88's and ksh's echo does not have an <code>-n</code> option. csh's echo and <code>/usr/ucb/echo</code>, on the other hand, have an <code>-n</code> option, but do not understand the back-slashed escape characters. sh and ksh88 determine whether <code>/usr/ucb/echo</code> is found first in the PATH and, if so, they adapt the behavior of the echo builtin to match <code>/usr/ucb/echo</code>.</p> |
| 操作数  | <p>The following operand is supported:</p> <p><i>string</i>    A string to be written to standard output. If any operand is “-n”, it is treated as a string, not an option. The following character sequences is recognized within any of the arguments:</p> <ul style="list-style-type: none"> <li><code>\a</code>    Alert character.</li> <li><code>\b</code>    Backspace.</li> <li><code>\c</code>    Print line without new-line. All characters following the <code>\c</code> in the argument are ignored.</li> <li><code>\f</code>    Form-feed.</li> <li><code>\n</code>    New-line.</li> <li><code>\r</code>    Carriage return.</li> <li><code>\t</code>    Tab.</li> <li><code>\v</code>    Vertical tab.</li> <li><code>\\</code>    Backslash.</li> <li><code>\0n</code>    Where <i>n</i> is the 8-bit character whose ASCII code is the 1-, 2- or 3-digit octal number representing that character.</li> </ul>                                                                                                                                                                                                                                                                     |
| 用法   | <p>Portable applications should not use <code>-n</code> (as the first argument) or escape sequences.</p> <p>The <a href="#">printf(1)</a> utility can be used portably to emulate any of the traditional behaviors of the echo utility as follows:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

- The Solaris 2.6 operating environment or compatible version's `/usr/bin/echo` is equivalent to:

```
printf "%b\n" "$*
```

- The `/usr/ucb/echo` is equivalent to:

```
if ["X$1" = "X-n"]
then
 shift
 printf "%s" "$*"
else
 printf "%s\n" "$*"
fi
```

New applications are encouraged to use `printf` instead of `echo`.

## 示例

示例 1 Finding how far below root your current directory is located

You can use `echo` to determine how many subdirectories below the root directory (`/`) is your current directory, as follows:

- Echo your current-working-directory's full pathname.
- Pipe the output through `tr` to translate the path's embedded slash-characters into space-characters.
- Pipe that output through `wc -w` for a count of the names in your path.

```
example% /usr/bin/echo $PWD | tr '/' ' ' | wc -w
```

See [tr\(1\)](#) and [wc\(1\)](#) for their functionality.

Below are the different flavors for echoing a string without a NEWLINE:

示例 2 `/usr/bin/echo`

```
example% /usr/bin/echo "$USER's current directory is $PWD\c"
```

示例 3 `sh/ksh88 shells`

```
example$ echo "$USER's current directory is $PWD\c"
```

示例 4 `csh shell`

```
example% echo -n "$USER's current directory is $PWD"
```



示例5 /usr/ucb/echo

```
example% /usr/ucb/echo -n "$USER's current directory is $PWD"
```

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `uname`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

## 退出状态

The following error values are returned:

- 0 Successful completion.
- >0 An error occurred.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

## 另请参见

[ksh\(1\)](#), [printf\(1\)](#), [shell\\_builtins\(1\)](#), [tr\(1\)](#), [wc\(1\)](#), [echo\(1B\)](#), [ascii\(5\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

## 附注

When representing an 8-bit character by using the escape convention `\0n`, the *n* must *always* be preceded by the digit zero (0).

For example, typing: `echo 'WARNING:\ 07'` prints the phrase `WARNING:` and sounds the “bell” on your terminal. The use of single (or double) quotes (or two backslashes) is required to protect the “\” that precedes the “07”.

Following the `\0`, up to three digits are used in constructing the octal output character. If, following the `\0n`, you want to echo additional digits that are not part of the octal representation, you must use the full 3-digit *n*. For example, if you want to echo “ESC 7” you must use the three digits “033” rather than just the two digits “33” after the `\ 0`.

|          |            |                                     |         |
|----------|------------|-------------------------------------|---------|
| 2 digits | Incorrect: | <code>echo "\0337"   od -xc</code>  |         |
|          | produces:  | <code>df0a</code>                   | (hex)   |
|          |            | <code>337</code>                    | (ascii) |
| 3 digits | Correct:   | <code>echo "\00337"   od -xc</code> |         |
|          | produces:  | <code>1b37 0a00</code>              | (hex)   |
|          |            | <code>033 7</code>                  | (ascii) |

For the octal equivalents of each character, see [ascii\(5\)](#).

**引用名** echo – echo arguments to standard output

**用法概要** /usr/ucb/echo [-n] [*argument*]

**描述** echo writes its arguments, separated by BLANKs and terminated by a NEWLINE, to the standard output.

echo is useful for producing diagnostics in command files and for sending known data into a pipe, and for displaying the contents of environment variables.

For example, you can use echo to determine how many subdirectories below the root directory (/) is your current directory, as follows:

- echo your current-working-directory's full pathname
- pipe the output through tr to translate the path's embedded slash-characters into space-characters
- pipe that output through wc -w for a count of the names in your path.

```
example% /usr/bin/echo "echo $PWD | tr '/' ' ' | wc -w"
```

See [tr\(1\)](#) and [wc\(1\)](#) for their functionality.

The shells [csh\(1\)](#), [ksh\(1\)](#), and [sh\(1\)](#), each have an echo built-in command, which, by default, will have precedence, and will be invoked if the user calls echo without a full pathname. /usr/ucb/echo and csh's echo() have an -n option, but do not understand back-slashed escape characters. sh's echo(), ksh's echo(), and /usr/bin/echo, on the other hand, understand the black-slashed escape characters, and ksh's echo() also understands \a as the audible bell character; however, these commands do not have an -n option.

**选项** -n Do not add the NEWLINE to the output.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

**另请参见** [csh\(1\)](#), [echo\(1\)](#), [ksh\(1\)](#), [sh\(1\)](#), [tr\(1\)](#), [wc\(1\)](#), [attributes\(5\)](#)

**附注** The -n option is a transition aid for BSD applications, and may not be supported in future releases.

|                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名                                   | ed, red – text editor                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 用法概要                                  | <pre> /usr/bin/ed [-s   -] [-p <i>string</i>] [<i>file</i>] /usr/xpg4/bin/ed [-s   -] [-p <i>string</i>] [<i>file</i>] /usr/xpg6/bin/ed [-s   -] [-p <i>string</i>] [<i>file</i>] /usr/bin/red [-s   -] [-p <i>string</i>] [<i>file</i>] </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 描述                                    | <p>The ed utility is the standard text editor. If <i>file</i> is specified, ed simulates an e command (see below) on the named file. That is, the file is read into ed's buffer so that it can be edited.</p> <p>The ed utility operates on a copy of the file it is editing. Changes made to the copy have no effect on the file until a w (write) command is given. The copy of the text being edited resides in a temporary file called the <i>buffer</i>. There is only one buffer.</p> <p>The red utility is a restricted version of ed. It will only allow editing of files in the current directory. red prohibits executing shell commands via <i>!shell command</i>. Attempts to bypass these restrictions result in an error message (<i>restricted shell</i>).</p> <p>Both ed and red support the <code>fspec(4)</code> formatting capability. The default terminal mode is either <code>stty -tabs</code> or <code>stty tab3</code>, where tab stops are set at eight columns (see <code>stty(1)</code>). If, however, the first line of <i>file</i> contains a format specification, that specification will override the default mode. For example, tab stops would be set at 5, 10, and 15, and a maximum line length of 72 would be imposed if the first line of <i>file</i> contains</p> <pre>&lt;:t5,10,15 s72:&gt;</pre> <p>Commands to ed have a simple and regular structure: zero, one, or two <i>addresses</i> followed by a single-character <i>command</i>, possibly followed by parameters to that command. These addresses specify one or more lines in the buffer. Every command that requires addresses has default addresses, so that the addresses can very often be omitted.</p> <p>In general, only one command may appear on a line. Certain commands allow the input of text. This text is placed in the appropriate place in the buffer. While ed is accepting text, it is said to be in <i>input mode</i>. In this mode, <i>no</i> commands are recognized; all input is merely collected. Leave input mode by typing a period (.) at the beginning of a line, followed immediately by a carriage return.</p> |
| /usr/bin/ed                           | If ed executes commands with arguments, it uses the default shell /usr/bin/sh (see <code>sh(1)</code> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| /usr/xpg4/bin/ed and /usr/xpg6/bin/ed | If ed executes commands with arguments, it uses /usr/xpg4/bin/sh (see <code>ksh88(1)</code> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Regular Expressions                   | The ed utility supports a limited form of <i>regular expression</i> notation. Regular expressions are used in addresses to specify lines and in some commands (for example, s) to specify portions of a line that are to be substituted. To understand addressing in ed, it is necessary to know that at any time there is a <i>current line</i> . Generally speaking, the current line is the last line affected by a command. The exact effect on the current line is discussed under the description of each command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

Internationalized Basic Regular Expressions are used for all system-supplied locales. See [regex\(5\)](#).

#### ed Commands

Commands may require zero, one, or two addresses. Commands that require no addresses regard the presence of an address as an error. Commands that accept one or two addresses assume default addresses when an insufficient number of addresses is given; if more addresses are given than such a command requires, the last one(s) are used.

Typically, addresses are separated from each other by a comma ( , ). They may also be separated by a semicolon ( ; ). In the latter case, the first address is calculated, the current line ( . ) is set to that value, and then the second address is calculated. This feature can be used to determine the starting line for forward and backward searches (see Rules 5 and 6, above). The second address of any two-address sequence must correspond to a line in the buffer that follows the line corresponding to the first address.

For `/usr/xpg6/bin/ed`, the address can be omitted on either side of the comma or semicolon separator, in which case the resulting address pairs are as follows:

| Specified | Resulting  |
|-----------|------------|
| ,         | 1, \$      |
| , addr    | 1, addr    |
| addr,     | addr, addr |
| ;         | 1; \$      |
| ; addr    | 1; addr    |
| addr;     | addr; addr |

Any *<blank>*s included between addresses, address separators, or address offsets are ignored.

In the following list of `ed` commands, the parentheses shown prior to the command are *not* part of the address. Rather, the parentheses show the default address(es) for the command.

Each address component can be preceded by zero or more blank characters. The command letter can be preceded by zero or more blank characters. If a suffix letter (l, n, or p) is given, it must immediately follow the command.

The `e`, `E`, `f`, `r`, and `w` commands take an optional *file* parameter, separated from the command letter by one or more blank characters.

If changes have been made in the buffer since the last `w` command that wrote the entire buffer, `ed` warns the user if an attempt is made to destroy the editor buffer via the `e` or `q` commands. The `ed` utility writes the string:

```
"?\n"
```

(followed by an explanatory message if *help mode* has been enabled via the H command) to standard output and continues in command mode with the current line number unchanged. If the e or q command is repeated with no intervening command, ed takes effect.

If an end-of-file is detected on standard input when a command is expected, the ed utility acts as if a q command had been entered.

It is generally illegal for more than one command to appear on a line. However, any command (except e, f, r, or w) may be suffixed by l, n, or p in which case the current line is either listed, numbered or written, respectively, as discussed below under the l, n, and p commands.

(.)a  
<text>

.

The append command accepts zero or more lines of text and appends it after the addressed line in the buffer. The current line (.) is left at the last inserted line, or, if there were none, at the addressed line. Address 0 is legal for this command: it causes the appended text to be placed at the beginning of the buffer. The maximum number of characters that may be entered from a terminal is 256 per line (including the new-line character).

(.,.)c  
<text>

.

The change command deletes the addressed lines from the buffer, then accepts zero or more lines of text that replaces these lines in the buffer. The current line (.) is left at the last line input, or, if there were none, at the first line that was not deleted. If the lines deleted were originally at the end of the buffer, the current line number will be set to the address of the new last line. If no lines remain in the buffer, the current line number will be set to 0.

/usr/xpg4/bin/ed      Address 0 is not legal for this command.

/usr/xpg6/bin/ed      Address 0 is valid for this command. It is interpreted as if the address 1 were specified.

(.,.)d

The delete command deletes the addressed lines from the buffer. The line after the last line deleted becomes the current line. If the lines deleted were originally at the end of the buffer, the new last line becomes the current line. If no lines remain in the buffer, the current line number will be set to 0.

e *file*

The edit command deletes the entire contents of the buffer and then reads the contents of *file* into the buffer. The current line (.) is set to the last line of the buffer. If *file* is not given, the currently remembered file name, if any, is used (see the f command). The number of bytes read will be written to standard output, unless the -s option was specified, in the following format:

"%d\n" <number of bytes read>

*file* is remembered for possible use as a default file name in subsequent e, E, r, and w commands. If *file* is replaced by !, the rest of the line is taken to be a shell ( `sh(1)`) command whose output is to be read. Such a shell command is *not* remembered as the current file name. See also DIAGNOSTICS below. All marks are discarded upon the completion of a successful e command. If the buffer has changed since the last time the entire buffer was written, the user is warned, as described previously.

E *file* The Edit command is like e, except that the editor does not check to see if any changes have been made to the buffer since the last w command.

f *file* If *file* is given, the f command changes the currently remembered path name to *file*. Whether the name is changed or not, the f command then writes the (possibly new) currently remembered path name to the standard output in the following format:

```
"%s\n"pathname
```

The current line number is unchanged.

(1,\$)g/*RE/command list* In the global command, the first step is to mark every line that matches the given *RE*. Then, for every such line, the given *command list* is executed with the current line (.) initially set to that line. When the g command completes, the current line number has the value assigned by the last command in the command list. If there were no matching lines, the current line number is not changed. A single command or the first of a list of commands appears on the same line as the global command. All lines of a multi-line list except the last line must be ended with a backslash (\); a, i, and c commands and associated input are permitted. The . terminating input mode may be omitted if it would be the last line of the *command list*. An empty *command list* is equivalent to the p command. The g, G, v, V, and ! commands are *not* permitted in the *command list*. See also the NOTES and the last paragraph before FILES below. Any character other than space or newline can be used instead of a slash to delimit the *RE*. Within the *RE*, the *RE* delimiter itself can be used as a literal character if it is preceded by a backslash.

(1,\$)G/*RE/* In the interactive Global command, the first step is to mark every line that matches the given *RE*. Then, for every such line, that line is written to standard output, the current line (.) is changed to that line, and any *one* command (other than one of the a, c, i, g, G, v, and V commands) may be input and is executed. After the execution of that command, the next marked line is written, and so on. A new-line acts as a null

command. An & causes the re-execution of the most recent non-null command executed within the current invocation of G. *Note:* The commands input as part of the execution of the G command may address and affect *any* lines in the buffer. The final value of the current line number is the value set by the last command successfully executed. (Notice that the last command successfully executed is the G command itself if a command fails or the null command is specified.) If there were no matching lines, the current line number is not changed. The G command can be terminated by a SIGINT signal. The G command can be terminated by an interrupt signal (ASCII DEL or BREAK). Any character other than space or newline can be used instead of a slash to delimit the *RE*. Within the *RE*, the *RE* delimiter itself can be used as a literal character if it is preceded by a backslash.

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| h                     | The <code>h</code> elp command gives a short error message that explains the reason for the most recent <code>?</code> diagnostic. The current line number is unchanged.                                                                                                                                                                                                                                                                                 |
| H                     | The <code>H</code> elp command causes <code>ed</code> to enter a mode in which error messages are written for all subsequent <code>?</code> diagnostics. It also explains the previous <code>?</code> if there was one. The <code>H</code> command alternately turns this mode on and off; it is initially off. The current line number is unchanged.                                                                                                    |
| (.,.)i<br><text><br>. | The <code>insert</code> command accepts zero or more lines of text and inserts it before the addressed line in the buffer. The current line (.) is left at the last inserted line, or, if there were none, at the addressed line. This command differs from the <code>a</code> command only in the placement of the input text. The maximum number of characters that may be entered from a terminal is 256 per line (including the new-line character). |
|                       | <pre>/usr/xpg4/bin/ed  Address 0 is not legal for this                   command.</pre> <pre>/usr/xpg6/bin/ed  Address 0 is valid for this command. It                   is interpreted as if the address 1 were                   specified.</pre>                                                                                                                                                                                                      |
| (.,.+1)j              | The <code>join</code> command joins contiguous lines by removing the appropriate new-line characters. If exactly one address is given, this command does nothing. If lines are joined, the current line                                                                                                                                                                                                                                                  |

number is set to the address of the joined line. Otherwise, the current line number is unchanged.

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (.)kx   | The mark command marks the addressed line with name <i>x</i> , which must be an ASCII lower-case letter (a-z). The address <i>x</i> then addresses this line. The current line (.) is unchanged.                                                                                                                                                                                                                                                                                                                                                                                             |
| (.,.)l  | The l command writes to standard output the addressed lines in a visually unambiguous form. The characters ( \\, \a, \b, \f, \r, \t, \v) are written as the corresponding escape sequence. The \n in that table is not applicable. Non-printable characters not in the table are written as one three-digit octal number (with a preceding backslash character) for each byte in the character, with the most significant byte first.                                                                                                                                                        |
|         | Long lines are folded, with the point of folding indicated by writing backslash/newline character. The length at which folding occurs is unspecified, but should be appropriate for the output device. The end of each line is marked with a \$. When using the /usr/xpg6/bin/ed command, the end of each line is marked with a \$ due to folding, and \$ characters within the text are written with a preceding backslash. An l command can be appended to any other command other than e, E, f, q, Q, r, w, or !. The current line number is set to the address of the last line written. |
| (.,.)ma | The move command repositions the addressed line(s) after the line addressed by <i>a</i> . Address 0 is legal for <i>a</i> and causes the addressed line(s) to be moved to the beginning of the file. It is an error if address <i>a</i> falls within the range of moved lines. The current line (.) is left at the last line moved.                                                                                                                                                                                                                                                          |
| (.,.)n  | The number command writes the addressed lines, preceding each line by its line number and a tab character. The current line (.) is left at the last line written. The n command may be appended to any command other than e, E, f, q, Q, r, w, or !.                                                                                                                                                                                                                                                                                                                                         |
| (.,.)p  | The print command writes the addressed lines to standard output. The current line (.) is left at the last line written. The p command may be appended to any command other than e, E, f, q, Q, r, w, or !. For example, dp deletes the current line and writes the new current line.                                                                                                                                                                                                                                                                                                         |
| P       | The P command causes ed to prompt with an asterisk (*) (or <i>string</i> , if -p is specified) for all subsequent commands. The P                                                                                                                                                                                                                                                                                                                                                                                                                                                            |



command alternatively turns this mode on and off; it is initially on if the `-p` option is specified, otherwise off. The current line is unchanged.

- `q` The quit command causes `ed` to exit. If the buffer has changed since the last time the entire buffer was written, the user is warned. See `DIAGNOSTICS`.
- `Q` The editor exits without checking if changes have been made in the buffer since the last `w` command.
- `( $\$$ ) r file` The read command reads the contents of *file* into the buffer. If *file* is not given, the currently remembered file name, if any, is used (see the `e` and `f` commands). The currently remembered file name is *not* changed unless *file* is the very first file name mentioned since `ed` was invoked. Address 0 is legal for `r` and causes the file to be read in at the beginning of the buffer. If the read is successful and the `-s` option was not specified, the number of characters read is written to standard output in the following format:
- ```
%d\n, <number of bytes read>
```
- The current line (`.`) is set to the last line read. If *file* is replaced by `!`, the rest of the line is taken to be a shell command (see [sh\(1\)](#)) whose output is to be read. For example, `$r !ls` appends the current directory to the end of the file being edited. Such a shell command is *not* remembered as the current file name.
- `(. . .) s/RE/replacement/`
`(. . .) s/RE/replacement/count, count=[1-2047]`
`(. . .) s/RE/replacement/g`
`(. . .) s/RE/replacement/\`
`(. . .) s/RE/replacement/n`
`(. . .) s/RE/replacement/p`

The substitute command searches each addressed line for an occurrence of the specified *RE*. Zero or more substitution commands can be specified. In each line in which a match is found, all (non-overlapped) matched strings are replaced by the *replacement* if the global replacement indicator `g` appears after the command. If the global indicator does not appear, only the first occurrence of the matched string is replaced. If a number *count* appears after the command, only the *count*-th

occurrence of the matched string on each addressed line is replaced. It is an error if the substitution fails on *all* addressed lines. Any character other than space or new-line may be used instead of the slash (/) to delimit the *RE* and the *replacement*. The current line (.) is left at the last line on which a substitution occurred. Within the *RE*, the *RE* delimiter itself can be used as a literal character if it is preceded by a backslash. See also the last paragraph before FILES below.

An ampersand (&) appearing in the *replacement* is replaced by the string matching the *RE* on the current line. The special meaning of & in this context may be suppressed by preceding it by \. As a more general feature, the characters \n, where *n* is a digit, are replaced by the text matched by the *n*-th regular subexpression of the specified *RE* enclosed between \ (and \). When nested parenthesized subexpressions are present, *n* is determined by counting occurrences of \ (starting from the left. When the character % is the only character in the *replacement*, the *replacement* used in the most recent substitute command is used as the *replacement* in the current substitute command. If there was no previous substitute command, the use of % in this manner is an error. The % loses its special meaning when it is in a replacement string of more than one character or is preceded by a \. For each backslash (\) encountered in scanning *replacement* from beginning to end, the following character loses its special meaning (if any). It is unspecified what special meaning is given to any character other than &, \, %, or digits.

A line may be split by substituting a new-line character into it. The new-line in the *replacement* must be escaped by preceding it by `\`. Such substitution cannot be done as part of a `g` or `v` command list. The current line number is set to the address of the last line on which a substitution is performed. If no substitution is performed, the current line number is unchanged. If a line is split, a substitution is considered to have been performed on each of the new lines for the purpose of determining the new current line number. A substitution is considered to have been performed even if the replacement string is identical to the string that it replaces.

The substitute command supports the following indicators:

- count* Substitute for the *count*th occurrence only of the *RE* found on each addressed line. *count* must be between 1-2047.
- `g` Globally substitute for all non-overlapping instances of the *RE* rather than just the first one. If both `g` and *count* are specified, the results are unspecified.
- `l` Write to standard output the final line in which a substitution was made. The line is written in the format specified for the `l` command.
- `n` Write to standard output the final line in which a substitution was made. The line is written in the format specified for the `n` command.

p Write to standard output the final line in which a substitution was made. The line will be written in the format specified for the p command.

(.,.)ta

This command acts just like the m command, except that a *copy* of the addressed lines is placed after address a (which may be 0). The current line (.) is left at the last line copied.

u

The undo command nullifies the effect of the most recent command that modified anything in the buffer, namely the most recent a, c, d, g, i, j, m, r, s, t, u, v, G, or V command. All changes made to the buffer by a g, G, v, or V global command is undone as a single change. If no changes were made by the global command (such as with g/ RE/p), the u command has no effect. The current line number is set to the value it had immediately before the command being undone started.

(1,\$)v/RE/command list

This command is the same as the global command g, except that the lines marked during the first step are those that do *not* match the RE.

(1,\$)V/RE/

This command is the same as the interactive global command G, except that the lines that are marked during the first step are those that do *not* match the RE.

(1,\$)w file

The write command writes the addressed lines into *file*. If *file* does not exist, it is created with mode 666 (readable and writable by everyone), unless your file creation mask dictates otherwise. See the description of the umask special command on [sh\(1\)](#). The currently remembered file name is *not* changed

unless *file* is the very first file name mentioned since *ed* was invoked. If no file name is given, the currently remembered file name, if any, is used (see the *e* and *f* commands). The current line (.) is unchanged. If the command is successful, the number of characters written is printed, unless the *-s* option is specified in the following format:

```
"%d\n", <number of bytes written>
```

If *file* is replaced by *!*, the rest of the line is taken to be a shell (see *sh(1)*) command whose standard input is the addressed lines. Such a shell command is *not* remembered as the current path name. This usage of the write command with *!* is to be considered as a "last *w* command that wrote the entire buffer".

(1, \$)W*file*

This command is the same as the write command above, except that it appends the addressed lines to the end of *file* if it exists. If *file* does not exist, it is created as described above for the *w* command.

(\$)=

The line number of the addressed line is written to standard output in the following format:

```
"%d\n" <line number>
```

The current line number is unchanged by this command.

!*shell command*

The remainder of the line after the *!* is sent to the UNIX system shell (see *sh(1)*) to be interpreted as a command. Within the text of that command, the unescaped character *%* is replaced with the remembered file name. If a *!* appears as the first character of the shell command, it is replaced with the text of the previous shell command. Thus, *!!* repeats the last shell command. If any replacements of *%* or *!* are performed, the modified line is written to the standard output before *command* is executed. The *!* command will write:

```
"!\n"
```

to standard output upon completion, unless the *-s* option is specified. The current line number is unchanged.

(. +1) <new-line> An address alone on a line causes the addressed line to be written. A new-line alone is equivalent to . +1p. It is useful for stepping forward through the buffer. The current line number will be set to the address of the written line.

If an interrupt signal (ASCII DEL or BREAK) is sent, ed writes a "?\n" and returns to *its* command level.

The ed utility takes the standard action for all signals with the following exceptions:

SIGINT The ed utility interrupts its current activity, writes the string "?\n" to standard output, and returns to command mode.

SIGHUP If the buffer is not empty and has changed since the last write, the ed utility attempts to write a copy of the buffer in a file. First, the file named ed.hup in the current directory is used. If that fails, the file named ed.hup in the directory named by the HOME environment variable is used. In any case, the ed utility exits without returning to command mode.

Some size limitations are in effect: 512 characters in a line, 256 characters in a global command list, and 255 characters in the path name of a file (counting slashes). The limit on the number of lines depends on the amount of user memory. Each line takes 1 word.

When reading a file, ed discards ASCII and NUL characters.

If a file is not terminated by a new-line character, ed adds one and puts out a message explaining what it did.

If the closing delimiter of an RE or of a replacement string (for example, /) would be the last character before a new-line, that delimiter may be omitted, in which case the addressed line is written. The following pairs of commands are equivalent:

```
s/s1/s2      s/s1/s2/p
g/s1         g/s1/p
?s1         ?s1?
```

If an invalid command is entered, ed writes the string:

```
"?\n"
```

(followed by an explanatory message if *help mode* has been enabled by the H command) to standard output and continues in command mode with the current line number unchanged.

选项

-pstring Allows the user to specify a prompt string. By default, there is no prompt string.

-s | - ; Suppresses the writing of character counts by e, r, and w commands, of diagnostics from e and q commands, and of the ! prompt after a ! *shell command*.

- 操作数** The following operand is supported:
- file* If *file* is specified, ed simulates an e command on the file named by the path name *file* before accepting commands from the standard input.
- 用法** See [largefile\(5\)](#) for the description of the behavior of ed and red when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).
- 环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of ed: HOME, LANG, LC_ALL, LC_CTYPE, LC_COLLATE, LC_MESSAGES, and NLSPATH.
- 退出状态** The following exit values are returned:
- 0 Successful completion without any file or command errors.
- >0 An error occurred.
- 文件** **\$TMPDIR** If this environment variable is not NULL, its value is used in place of /var/tmp as the directory name for the temporary work file.
- /var/tmp If /var/tmp exists, it is used as the directory name for the temporary work file.
- /tmp If the environment variable TMPDIR does not exist or is NULL, and if /var/tmp does not exist, then /tmp is used as the directory name for the temporary work file.
- ed.hup Work is saved here if the terminal is hung up.
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/ed,
/usr/bin/red

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

/usr/xpg4/bin/ed

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

/usr/xpg6/bin/ed

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu6

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Enabled
Interface Stability	Standard

另请参见

[bfs\(1\)](#), [edit\(1\)](#), [ex\(1\)](#), [grep\(1\)](#), [ksh88\(1\)](#), [sed\(1\)](#), [sh\(1\)](#), [stty\(1\)](#), [umask\(1\)](#), [vi\(1\)](#), [fspec\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#), [standards\(5\)](#)

诊断

? for command errors.

?*file* for an inaccessible file. Use the help and Help commands for detailed explanations.

If changes have been made in the buffer since the last `w` command that wrote the entire buffer, `ed` warns the user if an attempt is made to destroy `ed`'s buffer via the `e` or `q` commands. It writes `?` and allows one to continue editing. A second `e` or `q` command at this point will take effect. The `-s` command-line option inhibits this feature.

附注

The `-` option, although it continues to be supported, has been replaced in the documentation by the `-s` option that follows the Command Syntax Standard (see [Intro\(1\)](#)).

A `!` command cannot be subject to a `g` or a `v` command.

The `!` command and the `!` escape from the `e`, `r`, and `w` commands cannot be used if the editor is invoked from a restricted shell (see [sh\(1\)](#)).

The sequence `\n` in an RE does not match a new-line character.

If the editor input is coming from a command file (for example, `ed file < ed_cmd_file`), the editor exits at the first failure.

Loading an alternate `malloc()` library using the environment variable `LD_PRELOAD` can cause problems for `/usr/bin/ed`.

引用名 edit – text editor (variant of ex for casual users)

用法概要

```

/usr/bin/edit [-] [-s] [-l] [-L] [-R] [-r [filename]]
    [-t tag] [-v] [-V] [-x] [-wn] [-C]
    [+command | -c command] filename...

/usr/xpg4/bin/edit [-] [-s] [-l] [-L] [-R] [-r [filename]]
    [-t tag] [-v] [-V] [-x] [-wn] [-C]
    [+command | -c command] filename...

/usr/xpg6/bin/edit [-] [-s] [-l] [-L] [-R] [-r [filename]]
    [-t tag] [-v] [-V] [-x] [-wn] [-C]
    [+command | -c command] filename...

```

描述

The `edit` utility is a variant of the text editor `ex` recommended for new or casual users who wish to use a command-oriented editor. It operates precisely as `ex` with the following options automatically set:

```

novice      ON
report      ON
showmode    ON
magic       OFF

```

The following brief introduction should help you get started with `edit`. If you are using a CRT terminal you might want to learn about the display editor `vi`.

To edit the contents of an existing file you begin with the command `edit name` to the shell. `edit` makes a copy of the file that you can then edit, and tells you how many lines and characters are in the file. To create a new file, you also begin with the command `edit` with a filename: `edit name`; the editor tells you it is a [New File].

The `edit` command prompt is the colon (:), which you should see after starting the editor. If you are editing an existing file, then you have some lines in `edit`'s buffer (its name for the copy of the file you are editing). When you start editing, `edit` makes the last line of the file the current line. Most commands to `edit` use the current line if you do not tell them which line to use. Thus if you say `print` (which can be abbreviated `p`) and type carriage return (as you should after all `edit` commands), the current line is printed. If you `delete` (`d`) the current line, `edit` prints the new current line, which is usually the next line in the file. If you `delete` the last line, then the new last line becomes the current one.

If you start with an empty file or wish to add some new lines, then the `append` (`a`) command can be used. After you execute this command (typing a carriage return after the word `append`), `edit` reads lines from your terminal until you type a line consisting of just a dot (.); it places these lines after the current line. The last line you type then becomes the current line. The `insert` (`i`) command is like `append`, but places the lines you type before, rather than after, the current line.

The `edit` utility numbers the lines in the buffer, with the first line having number 1. If you execute the command `1`, then `edit` types the first line of the buffer. If you then execute the command `d`, `edit` deletes the first line, line 2 becomes line 1, and `edit` prints the current line (the new line 1) so you can see where you are. In general, the current line is always the last line affected by a command.

You can make a change to some text within the current line by using the `substitute (s)` command: `s/old/new/` where *old* is the string of characters you want to replace and *new* is the string of characters you want to replace *old* with.

The `filename (f)` command tells you how many lines there are in the buffer you are editing and says [Modified] if you have changed the buffer. After modifying a file, you can save the contents of the file by executing a `write (w)` command. You can leave the editor by issuing a `quit (q)` command. If you run `edit` on a file, but do not change it, it is not necessary (but does no harm) to `write` the file back. If you try to `quit` from `edit` after modifying the buffer without writing it out, you receive the message `No write since last change (:quit! overrides)`, and `edit` waits for another command. If you do not want to write the buffer out, issue the `quit` command followed by an exclamation point (`q!`). The buffer is then irretrievably discarded and you return to the shell.

By using the `d` and `a` commands and giving line numbers to see lines in the file, you can make any changes you want. You should learn at least a few more things, however, if you use `edit` more than a few times.

The `change (c)` command changes the current line to a sequence of lines you supply (as in `append`, you type lines up to a line consisting of only a dot `.`). You can tell `change` to change more than one line by giving the line numbers of the lines you want to change, that is, `3,5c`. You can print lines this way too: `1,23p` prints the first 23 lines of the file.

The `undo (u)` command reverses the effect of the last command you executed that changed the buffer. Thus if you execute a `substitute` command that does not do what you want, type `u` and the old contents of the line are restored. You can also `undo` an `undo` command. `edit` gives you a warning message when a command affects more than one line of the buffer. Note that commands such as `write` and `quit` cannot be undone.

To look at the next line in the buffer, type carriage return. To look at a number of lines, type `^D` (while holding down the control key, press `d`) rather than carriage return. This shows you a half-screen of lines on a CRT or 12 lines on a hardcopy terminal. You can look at nearby text by executing the `z` command. The current line appears in the middle of the text displayed, and the last line displayed becomes the current line; you can get back to the line where you were before you executed the `z` command by typing `'`. The `z` command has other options: `z-` prints a screen of text (or 24 lines) ending where you are; `z+` prints the next screenful. If you want less than a screenful of lines, type `z.n` to display five lines before and five lines after the current line. (Typing `z.n`, when *n* is an odd number, displays a total of *n* lines, centered about the current line; when *n* is an even number, it displays *n* - 1 lines, so that the lines displayed are

centered around the current line.) You can give counts after other commands; for example, you can delete 5 lines starting with the current line with the command `d5`.

To find things in the file, you can use line numbers if you happen to know them; since the line numbers change when you insert and delete lines this is somewhat unreliable. You can search backwards and forwards in the file for strings by giving commands of the form `/text/` to search forward for `text` or `?text?` to search backward for `text`. If a search reaches the end of the file without finding `text`, it wraps around and continues to search back to the line where you are. A useful feature here is a search of the form `/^text/` which searches for `text` at the beginning of a line. Similarly `/text$/` searches for `text` at the end of a line. You can leave off the trailing `/` or `?` in these commands.

The current line has the symbolic name `dot (.)`; this is most useful in a range of lines as in `., $p` which prints the current line plus the rest of the lines in the file. To move to the last line in the file, you can refer to it by its symbolic name `$`. Thus the command `$d` deletes the last line in the file, no matter what the current line is. Arithmetic with line references is also possible. Thus the line `$-5` is the fifth before the last and `.+20` is 20 lines after the current line.

You can find out the current line by typing `.' ='`. This is useful if you wish to move or copy a section of text within a file or between files. Find the first and last line numbers you wish to copy or move. To move lines 10 through 20, type `10, 20d a` to delete these lines from the file and place them in a buffer named `a`. `edit` has 26 such buffers named `a` through `z`. To put the contents of buffer `a` after the current line, type `put a`. If you want to move or copy these lines to another file, execute an `edit (e)` command after copying the lines; following the `e` command with the name of the other file you wish to edit, that is, `edit chapter2`. To copy lines without deleting them, use `yank (y)` in place of `d`. If the text you wish to move or copy is all within one file, it is not necessary to use named buffers. For example, to move lines 10 through 20 to the end of the file, type `10, 20m $`.

选项

These options can be turned on or off using the `set` command in [ex\(1\)](#).

- C Encryption option; same as the `-x` option, except that `vi` simulates the `C` command of `ex`. The `C` command is like the `X` command of `ex`, except that all text read in is assumed to have been encrypted.
- l Set up for editing LISP programs.
- L List the name of all files saved as the result of an editor or system crash.
- R Readonly mode; the `readonly` flag is set, preventing accidental overwriting of the file.
- r *filename* Edit *filename* after an editor or system crash. (Recovers the version of *filename* that was in the buffer when the crash occurred.)

-t <i>tag</i>	Edit the file containing the <i>tag</i> and position the editor at its definition.
-v	Start up in display editing state using vi. You can achieve the same effect by simply typing the vi command itself.
-V	Verbose. When ex commands are read by means of standard input, the input is echoed to standard error. This can be useful when processing ex commands within shell scripts.
-x	Encryption option; when used, edit simulates the X command of ex and prompts the user for a key. This key is used to encrypt and decrypt text using the algorithm of the crypt command. The X command makes an educated guess to determine whether text read in is encrypted or not. The temporary buffer file is encrypted also, using a transformed version of the key typed in for the -x option.
-wn	Set the default window size to <i>n</i> . This is useful when using the editor over a slow speed line.
+ <i>command</i> -c <i>command</i>	Begin editing by executing the specified editor command (usually a search or positioning command).
- -s	Suppress all interactive user feedback. This is useful when processing editor scripts.

The *filename* argument indicates one or more files to be edited.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/edit

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

/usr/xpg4/bin/edit

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled

/usr/xpg6/bin/edit

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu6

CSI	Enabled
-----	---------

另请参见

[ed\(1\)](#), [ex\(1\)](#), [vi\(1\)](#), [attributes\(5\)](#), [XPG4\(5\)](#)

附注

The encryption options are provided with the Security Administration Utilities package, which is available only in the United States.

引用名 egrep – search a file for a pattern using full regular expressions

用法概要

```
/usr/bin/egrep [-bchilnsv] -e pattern_list [file...]
/usr/bin/egrep [-bchilnsv] -f file [file...]
/usr/bin/egrep [-bchilnsv] pattern [file...]
/usr/xpg4/bin/egrep [-bchilnqsvx] -e pattern_list [-f file
file...]
/usr/xpg4/bin/egrep [-bchilnqsvx] [-e pattern_list] -f file
file...]
/usr/xpg4/bin/egrep [-bchilnqsvx] pattern [file...]
```

描述

The egrep (*expression grep*) utility searches files for a pattern of characters and prints all lines that contain that pattern. egrep uses full regular expressions (expressions that have string values that use the full set of alphanumeric and special characters) to match the patterns. It uses a fast deterministic algorithm that sometimes needs exponential space.

If no files are specified, egrep assumes standard input. Normally, each line found is copied to the standard output. The file name is printed before each line found if there is more than one input file.

/usr/bin/egrep

The /usr/bin/egrep utility accepts full regular expressions as described on the [regex\(5\)](#) manual page, except for \ (and \), \ (and \), \ { and \}, \ < and \ >, and \ n, and with the addition of:

1. A full regular expression followed by + that matches one or more occurrences of the full regular expression.
2. A full regular expression followed by ? that matches 0 or 1 occurrences of the full regular expression.
3. Full regular expressions separated by | or by a NEWLINE that match strings that are matched by any of the expressions.
4. A full regular expression that can be enclosed in parentheses () for grouping.

Be careful using the characters \$, *, [, ^, |, (,), and \ in *full regular expression*, because they are also meaningful to the shell. It is safest to enclose the entire *full regular expression* in single quotes (` `).

The order of precedence of operators is [], then * ? +, then concatenation, then | and NEWLINE.

/usr/xpg4/bin/egrep

The /usr/xpg4/bin/egrep utility uses the regular expressions described in the EXTENDED REGULAR EXPRESSIONS section of the [regex\(5\)](#) manual page.

选项

The following options are supported for both `/usr/bin/egrep` and `/usr/xpg4/bin/egrep`:

- b Precede each line by the block number on which it was found. This can be useful in locating block numbers by context (first block is 0).
- c Print only a count of the lines that contain the pattern.
- e *pattern_list* Search for a *pattern_list* (*full regular expression* that begins with a `-`).
- f *file* Take the list of *full regular expressions* from *file*.
- h Suppress printing of filenames when searching multiple files.
- i Ignore upper/lower case distinction during comparisons.
- l Print the names of files with matching lines once, separated by NEWLINES. Does not repeat the names of files when the pattern is found more than once.
- n Precede each line by its line number in the file (first line is 1).
- s Work silently, that is, display nothing except error messages. This is useful for checking the error status.
- v Print all lines except those that contain the pattern.

`/usr/xpg4/bin/egrep`

The following options are supported for `/usr/xpg4/bin/egrep` only:

- q Quiet. Does not write anything to the standard output, regardless of matching lines. Exits with zero status if an input line is selected.
- x Consider only input lines that use all characters in the line to match an entire fixed string or regular expression to be matching lines.

操作数

The following operands are supported:

file A path name of a file to be searched for the patterns. If no *file* operands are specified, the standard input is used.

`/usr/bin/egrep`

pattern Specify a pattern to be used during the search for input.

`/usr/xpg4/bin/egrep`

pattern Specify one or more patterns to be used during the search for input. This operand is treated as if it were specified as `-pattern_list`.

用法

See [largefile\(5\)](#) for the description of the behavior of `egrep` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `egrep`: `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

- 0 If any matches are found.

- 1 If no matches are found.
- 2 For syntax errors or inaccessible files (even if matches were found).

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/egrep

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Not Enabled

/usr/xpg4/bin/egrep

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled

另请参见

[fgrep\(1\)](#), [grep\(1\)](#), [sed\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#), [regexp\(5\)](#), [XPG4\(5\)](#)

附注

Ideally there should be only one `grep` command, but there is not a single algorithm that spans a wide enough range of space-time trade-offs.

Lines are limited only by the size of the available virtual memory.

/usr/xpg4/bin/egrep

The `/usr/xpg4/bin/egrep` utility is identical to `/usr/xpg4/bin/grep -E`. See [grep\(1\)](#). Portable applications should use `/usr/xpg4/bin/grep -E`.

引用名	eject – eject media such as CD-ROM from drive
用法概要	eject [-dflqt] [[<i>device</i> <i>nickname</i>]]
描述	<p>The <code>eject</code> utility is used for those removable media devices that do not have a manual eject button, or for those that might be locked due to, for instance, being mounted. The device may be specified by its name or by a nickname. If no device is specified, the default device is used.</p> <p>Only devices that support <code>eject</code> under program control respond to this command.</p> <p>When <code>eject</code> is used on media that can only be ejected manually, it does everything except remove the media, including unmounting the file system if it is mounted. In this case, <code>eject</code> displays a message that the media can now be manually ejected.</p> <p>Do not physically eject media from a device that contains mounted file systems. <code>eject</code> automatically searches for any mounted file systems that reside on the device, and attempts to unmount them prior to ejecting the media. See mount(1M). If the unmount operation fails, <code>eject</code> prints a warning message and exits. The <code>-f</code> option can be used to specify an eject even if the device contains mounted partitions.</p> <p>Pressing the physical media eject button located on some drives' front panel has the same effect as invoking <code>eject</code> for the respective drive. Not all drives have this capability.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -d Display the name of the default device to be ejected. -f Force the device to eject even if it is busy. -l Display paths and nicknames of ejectable devices. -q Query to see if the media is present. -t Issues the drive a CD-ROM tray close command. <p>Not all devices support this command.</p>
操作数	<p>The following operands are supported:</p> <p><i>device</i> Specifies which device to eject, by the name it appears in the directory <code>/dev</code>.</p> <p><i>nickname</i> Specifies which device to eject, by its nickname as known to this command.</p> <p> Volume label or device type (for example, <code>cdrom</code>) can be used as a nickname.</p>
示例	<p>示例 1 Ejecting Media</p> <p>The following example ejects media by its volume label:</p> <pre>example> eject 'My Pictures'</pre>

退出状态

The following exit codes are returned:

- 0 The operation was successful or, with the `-q` option, the media *is* in the drive.
- 1 The operation was unsuccessful or, with the `-q` option, the media *is not* in the drive.
- 2 Invalid options were specified.
- 3 An `ioctl()` request failed.
- 4 Manually ejectable media is now okay to remove.

文件

`/dev/sr0` default CD-ROM file (deprecated)

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[volcheck\(1\)](#), [mount\(1M\)](#), [rmmount\(1M\)](#), [ioctl\(2\)](#), [attributes\(5\)](#)

引用名 elfdump – dumps selected parts of an object file

用法概要 elfdump [-ccdegGhHiklmnPrsSuvy] [-p | -w file] [-I *index-expr*] [-N *name*] [-O *osabi*] [-T *type*] *filename...*

描述 The `elfdump` utility symbolically dumps selected parts of the specified object file(s). The options allow specific portions of the file to be displayed.

The `elfdump` utility is similar in function to the `dump(1)` utility. The `dump` utility offers an older and less user-friendly interface than `elfdump`, although `dump` might be more appropriate for certain uses such as in shell scripts.

Archive files, produced by `ar(1)`, can also be inspected with `elfdump`. In this case, each object within the archive is processed using the options supplied.

`elfdump` can display the ELF header, program header array, and section header array for any ELF object. It is also able to display the data found in the following types of sections:

Category	Option	ELF Section Type
Dynamic	-d	SHT_DYNAMIC
Global Offset Table (GOT)	-G	Special. See below.
Group	-g	SHT_GROUP
Capabilities	-H	SHT_SUNW_cap
Hash Table	-h	SHT_HASH
Interpreter	-i	Special, see below.
Move	-m	SHT_SUNW_move
Note	-n	SHT_NOTE
Relocation	-r	SHT_RELA SHT_REL
Stack Unwind/Exceptions	-u	Special. See below.
Syminfo	-y	SHT_SUNW_syminfo
Symbol Sort	-S	SHT_SUNW_symsort SHT_SUNW_tlssort
Symbol Table	-s	SHT_SYMTAB SHT_DYNSYM SHT_SUNW_LDYNSYM SHT_SUNW_versym
Versioning	-v	SHT_SUNW_verdef SHT_SUNW_verneed

Interpreter and global offset table sections do not have a special ELF section type, but are instead implemented as SHT_PROGBITS sections with well known names (`.interp` and `.got` respectively). `elfdump` is able to recognize and display these special sections.

Sections used for stack unwinding and exception handling can have the ELF section type SHT_PROGBITS, or SHT_AMD64_UNWIND, depending on the compiler and platform involved. These sections are recognized by name: `.eh_frame`, `.eh_frame_hdr`, and `.exception_ranges`.

When run without options to narrow the information displayed, `elfdump` displays all available information for each object.

For a complete description of the displayed information, refer to the 《链接程序和库指南》.

选项

The following options are supported:

- c
Dumps section header information.
- C
Demangles C++ symbol names.
- d
Dumps the contents of the `.dynamic` section.
- e
Dumps the ELF header.
- g
Dumps the contents of the `.group` section.
- G
Dumps the contents of the `.got` section.
- h
Dumps the contents of the `.hash` section.
- H
Dumps the contents of the `.SUNW_cap` capabilities section.
- i
Dumps the contents of the `.interp` section.
- I *index-expr*
Qualifies the sections or program headers to examine with a specific index or index range. For example, the third section header in a file can be displayed using:

```
example% elfdump -c -I 3 filename
```

An *index-expr* can be a single non-negative integer value that specifies a specific item, as shown in the previous example. Alternatively, an *index-expr* can consist of two such values separated by a colon (:), indicating a range of items. The following example displays the third, fourth, and fifth program headers in a file:

```
example% elfdump -p -I 3:5 filename
```

When specifying an index range, the second value can be omitted to indicate the final item in the file. For example, the following statement lists all section headers from the tenth to the end:

```
example% elfdump -c -I 10: filename
```

See Matching Options for additional information about the matching options (-I, -N, -T).

-
- k
Calculates the ELF checksum. See `gelf_checksum(3ELF)`.
 - l
Displays long section names without truncation.
 - m
Dumps the contents of the `.SUNW_move` section.
 - n
Dumps the contents of `.note` sections. By default, `elfdump` displays this data without interpretation in hexadecimal form. Core files are an exception. A subset of the core file notes described in `core(4)` are interpreted by `elfdump` and displayed in a high level format: `NT_PRSTATUS`, `NT_PRPSINFO`, `NT_PLATFORM`, `NT_AUXV`, `NT_ASRS`, `NT_PSTATUS`, `NT_PSINFO`, `NT_PRCRED`, `NT_UTSNAME`, `NT_LWPSTATUS`, `NT_LWPSINFO`, `NT_PRPRIV`, `NT_PRPRIVINFO`, `NT_CONTENT`, and `NT_ZONENAME`.
 - N *name*
Qualifies the sections or program headers to examine with a specific name. For example, in a file that contains more than one symbol table, the `.dynsym` symbol table can be displayed by itself using:

example% `elfdump -N .dynsym filename`

ELF program headers do not have names. If the `-p` option is specified, *name* refers to the program header type, and the behavior of the `-N` option is identical to that of the `-T` option. For example, the program header that identifies an interpreter can be displayed using:

example% `elfdump -p -N PT_INTERP filename`

See Matching Options for additional information about the matching options (`-I`, `-N`, `-T`).
 - O *osabi*
Specifies the Operating System ABI to apply when interpreting the object. *osabi* can be the name or value of any of the `ELFOSABI_` constants found in `/usr/include/sys/elf.h`. For convenience, the `ELFOSABI_` prefix may be omitted from these names. Two *osabi* values are fully supported: `solaris` is the native ABI of the Solaris operating system. `none` is the generic ELF ABI. Support for other operating system ABIs may be incomplete or missing. Items for which strings are unavailable are displayed in numeric form.

If `-O` is not used, and the object ELF header specifies a non-generic ABI, the ABI specified by the object is used. If the object specifies the generic ELF ABI, `elfdump` searches for a `.note.ABI-tag` section, and if found, identifies the object as having the `linux` ABI. Otherwise, an object that specifies the generic ELF ABI is assumed to conform to the `solaris` ABI.
 - p
Dumps the program headers. Individual program headers can be specified using the matching options (`-I`, `-N`, `-T`). See Matching Options for additional information.

The `-p` and `-w` options are mutually exclusive. Only one of these options can be used in a given `elfdump` invocation

`-P`

Generate and use alternative section header information based on the information from the program headers, ignoring any section header information contained in the file. If the file has no section headers a warning message is printed and this option is automatically selected. Section headers are not used by the system to execute a program. As such, a malicious program can have its section headers stripped or altered to provide misleading information. In contrast the program headers must be accurate for the program to be runnable. The use of synthetic section header information derived from the program headers allows files with altered section headers to be examined.

`-r`

Dumps the contents of the `.rel[a]` relocation sections.

`-s`

Dumps the contents of the `.SUNW_ldynsym`, `.dynsym`, and `.symtab` symbol table sections. For archives, the archive symbol table is also dumped. Individual sections can be specified with the matching options (`-I`, `-N`, `-T`). An archive symbol table can be specified using the special section name `-N ARSYM`.

In the case of core files, the `shndx` field has the value “unknown” since the field does not contain the valid values.

In addition to the standard symbol table information, the version definition index of the symbol is also provided under the `ver` heading.

See Matching Options for additional information about the matching options (`-I`, `-N`, `-T`).

`-S`

Dumps the contents of the `.SUNW_ldynsym` and `.dynsym` symbol table sections sorted in the order given by the `.SUNW_dynsym`sort and `.SUNW_dynsym`sort symbol sort sections. Thread Local Storage (TLS) symbols are sorted by offset. Regular symbols are sorted by address. Symbols not referenced by the sort sections are not displayed.

`-T type`

Qualifies the sections or program headers to examine with a specific type. For example, in a file that contains more than one symbol table, the `.dynsym` symbol table can be displayed by itself using:

```
example% elfdump -T SHT_DYNSYM filename
```

The value of `type` can be a numeric value, or any of the `SHT_` symbolic names defined in `/usr/include/sys/elf.h`. The `SHT_` prefix is optional, and `type` is case insensitive. Therefore, the above example can also be written as:

```
example% elfdump -T dynsym filename
```

If the `-p` option is specified, *type* refers to the program header type, which allows for the display of specific program headers. For example, the program header that identifies an interpreter can be displayed using:

```
example% elfdump -p -T PT_INTERP filename
```

The value of *type* can be a numeric value, or any of the `PT_` symbolic names defined in `/usr/include/sys/elf.h`. The `PT_` prefix is optional, and *type* is case insensitive. Therefore, the above example can also be written as:

```
example% elfdump -p -T interp filename
```

See Matching Options for additional information about the matching options (`-I`, `-N`, `-T`).

`-u`

Dumps the contents of sections used for stack frame unwinding and exception processing.

`-v`

Dumps the contents of the `.SUNW_version` version sections.

`-w file`

Writes the contents of sections which are specified with the matching options (`-I`, `-N`, `-T`) to the named file. For example, extracting the `.text` section of a file can be carried out with:

```
example% elfdump -w text.out -N .text filename
```

See Matching Options for additional information about the matching options (`-I`, `-N`, `-T`).

The `-p` and `-w` options are mutually exclusive. Only one of these options can be used in a given `elfdump` invocation

`-y`

Dumps the contents of the `.SUNW_syminfo` section. Symbol attributes are conveyed by character tokens.

- | | |
|---|---|
| A | Symbol definition acts as an auxiliary filter. |
| B | Assigned with D, symbol reference should be directly bound to the associated dependency definition. |
| C | Symbol definition is the result of a copy-relocation. |
| D | Symbol reference has a direct association to a dependency containing the definition. |
| F | Symbol definition acts as a standard filter. |
| I | Symbol definition acts as an interposer. |
| L | Symbol reference is to a dependency that should be lazily loaded. |
| N | External references can not directly bind to this symbol definition. |
| P | Symbol is associated with deferred (postponed) dependency. |

S Symbol is associated with capabilities.

操作数 The following operand is supported:

filename The name of the specified object file.

用法

Matching Options The options `-I`, `-N`, and `-T` are collectively referred to as the *matching options*. These options are used to narrow the range of program headers or sections to examine, by index, name, or type.

The exact interpretation of the matching options depends on the other options used:

- When used with the `-p` option, the matching options reference program headers. `-I` refers to program header indexes. `-T` refers to program header types. As program headers do not have names, the `-N` option behaves identically to `-T` for program headers.
- The matching options are used to select sections by index, name, or type when used with any of the options `-c`, `-g`, `-m`, `-n`, `-r`, `-s`, `-S`, `-u`, or `-w`.
- If matching options are used alone without any of the options `-c`, `-g`, `-m`, `-n`, `-p`, `-r`, `-s`, `-S`, `-u`, or `-w`, then `elfdump` examines each object, and displays the contents of any sections matched.

Any number and type of matching option can be mixed in a given invocation of `elfdump`. In this case, `elfdump` displays the superset of all items matched by any of the matching options used. This feature allows for the selection of complex groupings of items using the most convenient form for specifying each item.

文件 `liblddbg.so` linker debugging library

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/linker
Interface Stability	Committed

另请参见 [ar\(1\)](#), [dump\(1\)](#), [elffile\(1\)](#), [file\(1\)](#), [nm\(1\)](#), [pvs\(1\)](#), [elf\(3ELF\)](#), [core\(4\)](#), [attributes\(5\)](#)

《链接程序和库指南》

引用名	elfedit - 检查或编辑 ELF 文件
用法概要	elfedit [-adr] [-e cmd] [-L path] [-o default simple num] [infile] [outfile]
描述	elfedit 是一个用于检查或修改现有 ELF 目标文件内容的工具。具体而言，elfedit 用于修改目标文件中包含的 ELF 元数据。可以访问目标文件中包含的大多数 ELF 数据，这些数据包括 ELF 头、节头表、程序头表、动态节、硬件和软件功能、字符串表和符号表。
语法	<p>elfedit 可以处理来自命令行（-e 选项）或标准输入的命令。如果标准输入是一个终端，elfedit 可提供终端编辑功能以及涵盖大量命令的命令补齐功能。ELF 对特殊整数值和位掩码使用许多标准符号名称。elfedit 可识别此类名称的最有可能的完整形式。在输入 elfedit 命令时，您可以随时按 TAB 键，令 elfedit 显示用法消息以及当前光标处文本的任意已知完整形式。</p> <p>elfedit 功能以模块形式组织。每个模块提供一组命令，这些命令针对相关功能。通过使用冒号(:)分隔符将模块和命令名称组合到一起（中间无空格）来指定命令。例如，dyn:runpath 指的是由 dyn 模块提供的 runpath 命令。模块名称必须是唯一的。给定模块中的命令名称在该模块中必须唯一，但可在多个模块中使用相同的命令名称。</p> <p>某些模块将模块内的某个命令指定为该模块的缺省命令。用户只需指定模块名称，就可运行此命令。大多数 elfedit 模块均提供一个名为 dump 的命令，它针对模块涵盖的 ELF 文件部分生成的信息与 elfdump 实用程序所显示的信息相同。通常，模块会将 dump 指定为其缺省命令。</p> <p>用于执行 elfedit 命令的语法在设计上采用类似 UNIX 命令行实用程序的语法，这样任何会使用 UNIX 命令行实用程序的人都可以方便地使用 elfedit 命令。该语法由空格分隔的标记组成。第一个标记是命令名称。选项（即以连字符(-)开头的参数）跟在命令后面。纯参数（操作数）跟在选项后面。一个给定的命令可以有 0 个或多个选项和操作数，但是如果它们同时存在，选项始终位于纯参数前面。可使用特殊选项 --（两个连字符）来限定选项的结尾。如果遇到此选项，其余所有参数均被视为纯参数，即使它们以 - 开头。</p> <p>elfedit 标记中的字符的解释取决于所用的引用格式：</p> <p>不带引号</p> <p>单引号 (') 或双引号 (") 外面的反斜杠 (\) 充当转义符。elfedit 发现反斜杠字符时会将其忽略，并按字面意思处理反斜杠后面的字符（即使它后面的字符为反斜杠）。此功能可用于在命令的字符串参数中插入一个空白字符，从而无需将一个字符串分为两个单独的标记。同样，它可用于插入一个引号或反斜杠作为文本字符。</p> <p>单引号</p> <p>在单引号 (') 中，空白字符不用于分隔标记，且会被解释为标记内部的文本字符。双引号 (") 和反斜杠 (\) 字符会被解释为文本字符，无特殊意义。</p> <p>双引号</p> <p>在双引号 (") 中，空白字符不用于分隔标记。单引号字符会被解释为文本，不具有引用功能。反斜杠 (\) 是一个转义字符，在字符串文本中，其作用与 C 编程语言中反斜杠的作用类似：</p>

<code>\a</code>	警报 (bell)
<code>\b</code>	退格键
<code>\f</code>	换页符
<code>\n</code>	新行
<code>\r</code>	回车
<code>\t</code>	水平制表符
<code>\v</code>	垂直制表符
<code>\\</code>	反斜杠
<code>\'</code>	单引号
<code>\"</code>	双引号
<code>\ooo</code>	八进制常量，其中 ooo 是 1 到 3 个八进制位 (0...7)

在反斜杠后面跟有任何其他字符，均会出错。

核心命令均属于一个名为 `sys` 的内部模块。所有其他模块均打包为可动态装入的可共享目标文件。当执行需要某个模块的命令时或者当执行 `sys:load` 命令时，`elfedit` 会按需装入模块。由于 `sys` 模块特殊的内置状态，而且其命令使用频繁，所以 `elfedit` 命令允许在不加 `sys:` 前缀的情况下指定 `sys` 模块中的命令（例如，使用 `load` 而非 `sys:load`）。要访问任何其他模块中的命令，必须采用 `module:cmd` 完整格式指定。

`elfedit` 随以下标准模块一起提供：

<code>cap</code>	功能节
<code>dyn</code>	动态节
<code>ehdr</code>	ELF 头
<code>phdr</code>	程序头数组
<code>shdr</code>	节头数组
<code>str</code>	字符串表节
<code>sym</code>	符号表节
<code>syminfo</code>	<code>Syminfo</code> 节
<code>sys</code>	内置的核心 <code>elfedit</code> 命令

状态和命令文档

状态和命令文档

状态 (`sys:status`) 命令显示有关当前 `elfedit` 会话的信息：

- 输入和输出文件
- 选项设置
- 模块搜索路径
- 已装入的模块

每个 `elfedit` 模块均包含介绍每个命令的详细联机文档，文档格式类似于 UNIX 手册页。可使用 `help(sys:help)` 命令显示此信息。要了解有关 `elfedit` 的详细信息，请启动 `elfedit` 并使用不带有参数的 `help` 命令：

```
% elfedit
> help
```

`elfedit` 将显示一条欢迎消息，其中包含有关 `elfedit` 的详细信息和有关如何使用帮助系统的信息。

获取某个模块的摘要信息：

```
> help module
```

获取某个模块提供的某个特定命令的完整文档：

```
> help module:command
```

以 `dyn` 模块和 `dyn:runpath` 命令为例：

```
> help dyn
> help dyn:runpath
```

`help(sys:help)` 可用于获取 `help` 命令本身的帮助：

```
> help help
```

模块搜索路径

`elfedit` 模块是作为按需装入的可共享目标文件实现的。当需要某个模块时，`elfedit` 会搜索模块路径以便找到实现了该模块的可共享目标文件。路径是使用冒号 (:) 字符分界的目录名称序列。除了标准字符，路径还可以包含以下任意标记：

- %i 扩展到当前指令集体系结构 (instruction set architecture, ISA) 名称 (`sparc`、`sparcv9`、`i386`、`amd64`)。
- %I 扩展到 64 位 ISA。对于 64 位版本的 `elfedit`，此标记与 %i 相同，但对于 32 位版本，会扩展到空字符串。
- %O 展开到要修改的路径的旧值。要将目录附加到缺省路径之前或之后时，这非常有用。
- %r 包含 `elfedit` 程序的文件系统树的根（假定 `elfedit` 安装在树中的 `usr/bin/elfedit`）。在标准系统中，此标记就是标准系统根目录 (/)。在开发系统中（这种情况下，`elfedit` 副本可安装在任意位置），使用 %r 可确保使用匹配的模块集。

%% 展开到单个%字符

elfedit 的缺省模块搜索路径为：

```
%r/usr/lib/elfedit/%I
```

扩展标记，即：

```
/usr/lib/elfedit          32 位 elfedit
/usr/lib/elfedit/sparcv9  64 位 elfedit (sparc)
/usr/lib/elfedit/amd64    64 位 elfedit (x86)
```

通过设置 `ELFEDIT_PATH` 环境变量或使用 `-L` 命令行选项可更改缺省搜索路径。如果同时指定这两项，`-L` 选项将取代环境变量。

选项

支持以下选项：

-a

启用 `autoprint` 模式。启用 `autoprint` 后，`elfedit` 将输出修改 ELF 文件后所生成的修改值。此输出以当前输出样式显示，可使用 `-o` 选项更改此样式。缺省输出样式是 `elfdump(1)` 实用程序使用的样式。以交互方式使用 `elfedit` 时，`autoprint` 模式为缺省模式（当 `stdin` 和 `stdout` 为终端时）。因此，仅当在非交互的上下文中使用 `elfedit` 时，`-a` 选项才有意义。要在交互会话中禁用 `autoprint`，请使用 `elfedit` 命令：

```
> set a off
```

-d

如果已设置，此选项可使 `elfedit` 发布信息性消息，说明其内部操作和要处理的 ELF 目标文件的详细信息。这在需要深入了解所执行的操作时非常有用。

-e *cmd*

指定一个编辑命令。可以指定多个 `-e` 选项。如果在命令行上指定了多个编辑命令，`elfedit` 将在批处理模式下运行。打开文件后，`elfedit` 按给定的顺序执行每个命令，然后保存修改的文件，最后 `elfedit` 退出。从 shell 脚本和 `makefile` 执行简单操作时，批处理模式非常有用。

-L *path*

设置用于定位 `elfedit` 模块的缺省路径。本手册页的“**模块搜索路径**”部分介绍了各个模块。

-o *default* | *simple* | *num*

用于显示 ELF 数据的样式。此选项用于确立会话的当前样式。可在 `elfedit` 会话中更改此样式，方法是使用 `set (sys:set)` 命令或向会话中执行的各个命令提供 `-o` 选项。

default 缺省样式是以适合用户查看的格式显示输出。此样式与 `elfdump` 实用程序使用的样式类似。

- num** 整数值始终以整数格式显示。字符串显示为在内存中的字符串表中定位所用的整数偏移量。
- simple** 显示 ELF 文件中的字符串时，仅显示字符串。如果可能，整数值显示为符号常量，否则以整数格式显示。不显示任何标题、头或其他补充输出。
- r**
只读模式。输入文件以只读访问模式打开，编辑会话的结果不会保存。指定了 **-r** 时，**elfedit** 不允许使用 **outfile** 参数。如果不打算修改文件，强烈建议使用只读模式。除了提供额外保护以防止意外修改外，该选项还允许检查用户没有写入权限的文件。

操作数

支持下列操作数：

infile

包含要处理的 ELF 目标文件的输入文件。

此文件可以为可执行文件 (ET_EXEC)、共享目标文件 (ET_DYN) 或可重定位目标文件 (ET_REL)。不直接支持归档文件。要编辑归档中的目标文件，必须提取目标文件，编辑副本，然后将其重新插入到该归档文件。

如果未提供 *infile*，**elfedit** 将在限定模式下运行，此模式仅允许执行 **sys:** 模块中的命令。此模式主要用于访问 **help (sys:help)** 命令提供的命令文档。

如果提供了 *infile*，但未给定任何 *outfile*，**elfedit** 将就地编辑文件并将结果写入同一文件，这会导致原始文件内容被覆盖。通常，不建议以此模式使用 **elfedit**，建议指定输出文件。生成的文件经过测试和验证后，可将其移动到原始文件所在的位置。

-r 选项可用于以只读访问模式打开 *infile*。在检查不希望修改的现有文件时，此选项非常有用。

outfile

输出文件。如果同时提供了 *infile* 和 *outfile*，*infile* 将以只读访问模式打开，修改的目标文件内容会写入到 *outfile* 中。

用法

如果系统支持，**elfedit** 可作为 64 位应用程序运行，这种情况下可处理大于或等于 2 GB (2³¹ 个字节) 的文件。

启动时，**elfedit** 使用 **libelf** 打开输入文件并在内存中缓存其内容副本以供编辑。然后，**elfedit** 可执行一个或多个命令。随后，修改的目标文件写入输出文件（不一定如此），**elfedit** 退出，会话即结束。

如果未提供 *infile*，**elfedit** 将在限定模式下运行，此模式仅允许执行 **sys** 模块中的命令。此模式主要用于访问 **help (sys:help)** 命令提供的命令文档。

如果指定了一个或多个 **-e** 选项，将按给定顺序执行提供的命令。**elfedit** 会紧跟在给定命令之后添加对 **write (sys:write)** 和 **quit (sys:quit)** 的隐式调用，从而导致写入输出文件并退出 **elfedit** 进程。在 shell 脚本和 **makefile** 中使用此方式非常方便。

如果未指定 `-e` 选项，`elfedit` 将从 `stdin` 中读取命令并按给定顺序执行它们。在此模式下运行时，调用者必须显式发出 `write(sys:write)` 和 `quit(sys:quit)` 命令才能保存其工作并退出。

退出状态

将返回以下退出值：

- 0 成功完成。
- 1 发生了致命错误。
- 2 指定的命令行选项无效。

示例

在以下示例中，显示了将 `elfedit` 与 shell 提示符 (`%`) 和 `elfedit` 提示符 (`>`) 交互使用的情况。用户不应输入上述字符中的任何一个。

示例 1 更改可执行文件的 Runpath

以下示例假定名为 `prog` 的可执行文件安装在一个 `bin` 目录下，该目录的相邻目录为保存可共享目标文件的 `lib` 目录。以下命令将该可执行文件的 `runpath` 设置为 `lib` 目录：

```
elfedit -e 'dyn:runpath $ORIGIN/../lib'
```

需要对 `-e` 选项的参数使用单引号，以确保 shell 将整个命令作为一个参数传递给 `elfedit`。

另外，可在非批处理模式下使用 `elfedit` 执行相同的操作。

```
% elfedit prog
> dyn:runpath $ORIGIN/../lib
      index tag      value
      [30] RUNPATH    0x3e6      $ORIGIN/../lib
> write
> quit
```

仅当目标文件中存在 `padding` 时，才能对元素（例如 `runpath`）或所需的条目进行添加或修改。请参见“附注”部分。

示例 2 删除硬件功能位

需要满足可选硬件支持才能运行的目标文件均构建有功能节，此节包含的位掩码指定了这些目标文件所需的功能。运行时链接程序 (`ld.so.1`) 根据运行系统的属性检查此掩码，以确定当前系统是否可运行给定目标文件。如果系统未提供程序需要的功能，则会阻止这些程序运行。

该检查可防止未显式检查其所需硬件支持的不完善程序发生莫名其妙的崩溃。但是，对于编写了在运行时显式检查系统功能的程序，该检查可能会有不利影响。此类程序可能有在满足硬件支持时使用的优化代码，同时也提供了在硬件支持满足不了时可运行的通用后备版本（尽管运行速度比较慢）。在这种情况下，硬件兼容性掩码会阻止此类程序在早期的硬件上运行。此时，从掩码中删除相关位可允许程序运行。

示例 2 删除硬件功能位 (续)

以下示例从使用 SSE3 CPU 扩展的 x86 二进制文件中删除 AV_386_SSE3 硬件功能。这样就将验证能否使用 SSE3 的职责从运行时链接程序转移到程序自身：

```
elfedit -e 'cap:hw1 -and -cmp sse3' prog
```

示例 3 从目标文件中读取信息

elfedit 可用于从目标文件中提取特定目标信息。以下 shell 命令读取文件 /usr/bin/ls 中包含的节头数：

```
% SHNUM='elfedit -r -onum -e 'ehdr:e_shnum' /usr/bin/ls'
% echo $SHNUM
29
```

您可能会得到不同的值，具体取决于您使用的 Solaris 版本和计算机类型。-r 选项会使文件以只读方式打开，允许具有普通访问权限的用户打开文件并防止重要的系统可执行文件意外损坏。使用 num 输出样式是为了仅获取所需的值，而无任何多余的文本。

同样，以下示例从 C 运行时库提取 unlink 符号的符号类型：

```
% TYPE='elfedit -r -osimple -e 'sym:st_type unlink' \
      /lib/libc.so'
% echo $TYPE
STT_FUNC
```

示例 4 指定可执行文件的 ASLR 设置

可执行文件的缺省地址空间布局随机化 (Address Space Layout Randomization, ASLR) 行为是使用 DT_SUNW_AS LR 动态节条目指定的。以下演示了如何对指定程序启用或禁用 ASLR。

```
% elfedit prog
> dyn:sunw_aslr enable
      index tag          value
      [40] SUNW_AS LR    0x2      ENABLE
> dyn:sunw_aslr disable
      index tag          value
      [40] SUNW_AS LR    0x1      DISABLE
```

环境变量

ELFEDIT_PATH	更改缺省模块搜索路径。本手册页的“ 模块搜索路径 ”部分讨论了模块搜索路径。
LD_NOEXEC_64	禁止自动执行 64 位 elfedit。缺省情况下，如果系统具有 64 位功能，则会运行 64 位版本的 elfedit。
PAGER	以交互方式将输出从 elfedit 传送到屏幕。如果未设置，将使用 more。请参见 more(1) 。

- 文件** `/usr/lib/elfedit` `elfedit` 模块的缺省目录，这些模块会按需装入供编辑命令使用。
- `~/teclarc` 用于命令行编辑的 `tecla` 个人定制文件。请参见 [tecla\(5\)](#)。
- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/linker
接口稳定性	Committed（已确定）

另请参见 [dump\(1\)](#)、[elfdump\(1\)](#)、[ld.so.1\(1\)](#)、[more\(1\)](#)、[nm\(1\)](#)、[pvs\(1\)](#)、[sxadm\(1M\)](#)、[elf\(3ELF\)](#)、[libelf\(3LIB\)](#)、
 《[链接程序和库指南](#)》

警告 `elfedit` 是一个专用于测试和开发 ELF 系统的工具。通过该工具，用户几乎可检查和更改目标文件中的每一部分 ELF 元数据。对于会产生无效或不可用 ELF 文件的编辑操作，该工具也会执行，不会做出提醒。用户应当了解 ELF 格式并知晓应遵守的规则和约定。使用 `elfedit` 时，可从《[链接程序和库指南](#)》中获取帮助。

`elfedit` 允许用户更改目标文件中的 ELF 元数据，但用户无法了解或更改实际程序的代码。因此，以不符合文件实际内容的方式设置 ELF 属性（例如类型、大小、对齐方式等）可能会生成一个损坏的且不可用的输出目标文件。此类更改在测试链接程序组件时可能很有用，但在其他情况下应避免此类更改。

较高级别的操作（例如，使用 `dyn:runpath` 命令更改目标文件的 `runpath`）是安全的，在执行时不存在本节中叙述的此类风险。

附注 并非 `elfedit` 支持的每个 ELF 操作都可在每个 ELF 目标文件上成功执行。`elfedit` 受文件中现有的节制约。

特别需要指出的是，`elfedit` 可能无法修改给定目标文件的 `runpath`。要修改 `runpath`，必须满足以下条件：

- 动态字符串表中必须存在所需的字符串，或者，此节中必须有足够的预留空间可供添加新字符串使用。如果您的目标文件有一个字符串表预留区域，则 `.dynamic DT_SUNW_STRPAD` 元素的值表示该区域的大小。以下 `elfedit` 命令可用于检查此大小：


```
% elfedit -r -e 'dyn:tag DT_SUNW_STRPAD' file
```
- 动态节必须已有一个 `runpath` 元素，或者，必须存在可向其插入一个 `runpath` 元素的未使用的动态插槽。测试是否存在一个 `runpath`：


```
% elfedit -r -e 'dyn:runpath' file
```

动态节使用一个类型为 `DT_NULL` 的元素终止在该节中找到的数组。终止 `DT_NULL` 无法更改，但是如果存在多个此元素，`elfedit` 可将其中一个转换为 `runpath` 元素。测试是否有额外的动态插槽：


```
% elfedit -r -e 'dyn:tag DT_NULL' file
```

早期的目标文件不具备完成此类操作所需的额外空间。Solaris Express Community Edition 发行版中引入了执行此类操作所需的空间。

如果操作失败，使用 `-d (debug)` 选项输出的详细信息可帮助揭示失败原因。

`elfedit` 模块遵循一个约定，根据该约定，直接操作 ELF 结构中的某个字段的命令与该字段的名称相同，但实现较高级别概念的命令不使用此命名方式。例如，用于操作 ELF 头中的 `e_flags` 字段的命令名为 `ehdr:e_flags`。因此，您通常可以通过确定模块并查找具有相应字段名称的命令来查找修改 ELF 字段的命令。

引用名	elffile - 标识 ELF 文件类型
用法概要	elffile [-s basic detail summary] filename...
描述	<p>elffile 实用程序是 file 命令的专用变体，用于与 ELF 目标文件以及相关文件类型结合使用。elffile 可标识以下类型的文件：</p> <p>归档文件 除了 file 提供的信息外，elffile 还可标识归档成员的类型。</p> <p>ELF 目标文件/运行时链接程序配置文件 elffile 提供的输出与 file 相同</p> <p>其他类型的文件均报告为非 ELF 文件。不尝试对此类文件进行详细分类。建议使用 file 实用程序完成一般性的文件识别。</p>
选项	<p>支持以下选项：</p> <p><code>-s basic detail summary</code> 指定要提供的输出样式</p> <p>basic 以 file 使用的相同格式生成一行描述。</p> <p>detail 对于非归档文件，summary 的输出与 basic 的输出相同。处理归档文件时，基本输出行后显示每个归档成员的一行输出。</p> <p>summary 对于非归档文件，summary 的输出与 basic 的输出相同。处理归档文件时，会在基本输出的末尾添加归档内容的摘要描述。如果未指定 -s 选项，elffile 缺省情况下将使用 summary 样式。</p>
附注	<p>使用摘要样式时，为归档文件生成的输出取决于归档内容。如果归档文件包含同一平台的同种目标文件集合，则平台详细信息的显示格式采用显示单一目标文件的格式。否则，生成摘要描述。使用详细信息样式可获取关于各个归档成员的更具体的信息。</p> <p>摘要样式和详细信息样式均需要检查归档文件的每个成员。执行速度与归档成员数量成正比，如果归档文件非常大，速度会很慢。</p>
示例	<p>示例 1 显示归档文件的摘要输出</p> <p>以下示例显示了 elffile 针对不同内容的归档文件所生成的摘要输出。使用以下归档文件。</p> <p>same_elf.a 单个平台的 ELF 目标文件。</p> <p>mixed_elf.a 多个平台的 ELF 目标文件。</p>

示例1 显示归档文件的摘要输出 (续)

```
mixed.a
    ELF 目标文件和非 ELF 文件。

not_elf.a
    非 ELF 文件。
```

归档文件的摘要输出取决于归档成员的类型。

```
example% elffile same_elf.a mixed_elf.a mixed.a not_elf.a
same_elf.a: current ar archive, 32-bit symbol table,
           ELF 64-bit LSB relocatable AMD64 Version 1
mixed_elf.a: current ar archive, 32-bit symbol table,
           mixed ELF content
mixed.a: current ar archive, 32-bit symbol table,
         mixed ELF and non-ELF content
not_elf.a: current ar archive, non-ELF content
```

示例2 过滤归档文件的详细输出

elffile 的详细输出为归档文件生成一行输出，后面为每个归档成员显示一行输出。可以轻松地过滤此输出，以使用各种格式显示信息。以下示例使用归档文件 libCstd.a 说明了此特点，该归档文件包含 64 位 x86 系统的可重定位目标文件。此归档文件未经过滤的 elffile 详细输出如下所示：

```
example% elffile -s detail libCstd.a
libCstd.a: current ar archive, 32-bit symbol table
libCstd.a(bitset.o): ELF 64-bit LSB relocatable AMD64 Version 1 [CMOV]
libCstd.a(complex.o): ELF 64-bit LSB relocatable AMD64 Version 1 [SSE2 SSE CMOV
FPU]
libCstd.a(limits.o): ELF 64-bit LSB relocatable AMD64 Version 1 [SSE2 SSE FPU]
libCstd.a(limitsinit.o): ELF 64-bit LSB relocatable AMD64 Version 1
libCstd.a(stdexcept.o): ELF 64-bit LSB relocatable AMD64 Version 1 [SSE CMOV]
...
```

此输出显示每个目标文件均标记有运行所需的硬件功能。这些功能标记因每个目标文件中的代码而异。以下命令对 elffile 的输出进行了过滤，以标识每个唯一的功能掩码，并计算归档文件中每个掩码对应的目标文件数量。sed 命令用于从输出中删除归档成员名称，这样，每个具有相同功能掩码的归档成员的输出将会相同。sort 命令用于将这些相同的行归在一起，uniq 命令用于将每个唯一的组替换为该组中的一行，并在该行的前面显示该行在该组出现的次数。

```
example% elffile -s detail libCstd.a | sed 's,(.*),, ' | sort -f | uniq -c
  1 libCstd.a: current ar archive, 32-bit symbol table
 777 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1
   1 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [CMOV FPU]
 126 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [CMOV]
   12 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [FPU]
```

```

69 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [SSE CMOV]
 2 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [SSE2 CMOV]
 3 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [SSE2 SSE CMOV FPU]
 3 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [SSE2 SSE CMOV]
 1 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [SSE2 SSE FPU]
 2 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [SSE2 SSE]
20 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [SSE2]
 4 libCstd.a: ELF 64-bit LSB relocatable AMD64 Version 1 [SSE]

```

退出状态

将返回以下退出值：

0 成功完成
>0 出现错误

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/linker
接口稳定性	Committed（已确定）

另请参见

[ar\(1\)](#)、[dump\(1\)](#)、[elfdump\(1\)](#)、[file\(1\)](#)

《链接程序和库指南》

引用名	elfsign – 对二进制文件签名				
用法概要	<pre> /usr/bin/elfsign sign [-v] -k <i>private_key</i> -c <i>certificate_file</i> -e <i>elf_object</i> [-F <i>format</i>] [<i>file</i>]... /usr/bin/elfsign sign [-v] -c <i>certificate_file</i> -e <i>elf_object</i> -T <i>token_label</i> [-P <i>pin_file</i>] [-F <i>format</i>] [<i>file</i>]... /usr/bin/elfsign verify [-c <i>certificate_file</i>] [-v] -e <i>elf_object</i> [<i>file</i>]... /usr/bin/elfsign request -r <i>certificate_request_file</i> {-k <i>private_key</i> -T <i>token_label</i>} /usr/bin/elfsign list -f <i>field</i> -c <i>certificate_file</i> /usr/bin/elfsign list -f <i>field</i> -e <i>elf_object</i> </pre>				
描述	<p>list 列出单个证书文件或已签名的 elf 目标文件的标准输出信息。选定的字段显示在一行中。如果指定的字段不适用于指定的文件，此命令将终止，且没有标准输出。此子命令的此输出用于供脚本或其他命令使用。</p> <p>request 生成一个私钥和一个 PKCS#10 证书请求。PKCS#10 证书请求供 Solaris 加密框架使用。如果将在令牌设备中创建私钥，elfsign 会提示您输入更新令牌设备所需的 PIN。要获取证书，应将 PKCS#10 证书请求发送到以下电子邮件地址 solaris-crypto-req_ww@oracle.com。</p> <p>用户使用 elfsign 对要在 Solaris 加密框架中使用的二进制文件进行签名之前，必须首先生成证书请求并取得证书。</p> <p>sign 使用给定的私钥和证书文件为 elf 目标文件签名。</p> <p>verify 验证现有的签名目标文件。使用给定的证书，或在 /etc/crypto/certs 中搜索合适的证书（如果未指定 -c）。</p>				
选项	<p>支持以下选项：</p> <p>-c <i>certificate_file</i> 指定 PEM/PKCS#7 或 ASN.1 BER 格式的 X.509 证书所在的路径。</p> <p>-e <i>elf_object</i> 指定要签名或验证的目标文件所在的路径。</p> <p>对多个目标文件进行签名或验证时，可多次指定 -e 选项。</p> <p>-F <i>format</i> 对于 sign 子命令，指定签名的格式。有效的格式选项包括</p> <table border="0" style="margin-left: 2em;"> <tr> <td>rsa_md5_sha1</td> <td>Solaris 10 及其更新的缺省格式，rsa_md5_sha1 格式已过时。</td> </tr> <tr> <td>rsa_sha1</td> <td>此发行版的缺省格式。</td> </tr> </table>	rsa_md5_sha1	Solaris 10 及其更新的缺省格式，rsa_md5_sha1 格式已过时。	rsa_sha1	此发行版的缺省格式。
rsa_md5_sha1	Solaris 10 及其更新的缺省格式，rsa_md5_sha1 格式已过时。				
rsa_sha1	此发行版的缺省格式。				

- `rsa_md5_sha1` 以外的格式在签名中还包含一个信息时间戳，表示应用签名的时间。此时间戳没有采用加密形式来保证安全，也不用于验证。
- f *field*** 对于 `list` 子命令，指定输出中应显示的字段。
- 证书文件的有效字段说明符包括：
- subject** 主题 DN (Distinguished Name, 标识名)
- issuer** 签发者 DN
- elf 目标文件的有效字段说明符包括：
- format** 签名的格式
- signer** 目标文件签名所用证书的主题 DN
- time** 应用签名的时间，格式采用语言环境的缺省格式
- k *private_key*** 指定不使用 PKCS#11 令牌时的私钥文件位置。此文件为 RSA 私钥文件，其格式为 Solaris 特定格式。用于 `request` 子命令时，此文件是新生成的密钥的输出文件。
- 同时指定 `-k` 和 `-T` 选项会出错。
- P *pin_file*** 指定访问令牌设备所用的 PIN 保存在哪个文件中。如果未在 *pin_file* 中提供 PIN，`elfsign` 将提示您输入 PIN。
- 指定 `-P` 选项而不指定 `-T` 选项会出错。
- r *certificate_request_file*** 指定 PKCS#10 格式的证书请求文件所在的路径。
- T *token_label*** 指定存储私钥的 PKCS#11 令牌设备的标签，标签由 `pktool` 提供。
- 同时指定 `-T` 和 `-k` 选项会出错。
- v** 请求提供更为详细的信息。附加输出包括签名者和为目标文件签名的时间（如果签名格式中包含签名时间）。此输出不是稳定的可解析输出。

操作数

支持下列操作数：

- file*** 要签名或验证的一个或多个 elf 目标文件。必须至少指定一个 elf 目标文件，要么通过 `-e` 选项指定，要么在所有其他选项后指定。

示例

示例 1 使用某个文件中的密钥/证书为 ELF 目标文件签名

```
example$ elfsign sign -k myprivatekey -c mycert -e lib/libmylib.so.1
```

示例2 验证elf目标文件的签名

```
example$ elfsign verify -c mycert -e lib/libmylib.so.1
elfsign: verification of lib/libmylib.so.1 passed
```

示例3 生成证书请求

```
example$ elfsign request -k mykey -r req.pkcs10
Enter Company Name / Stock Symbol or some other globally
unique identifier.
This will be the prefix of the Certificate DN: SUNW
```

示例4 确定有关目标文件的信息

```
example$ elfsign list -f format -e lib/libmylib.so.1
rsa_md5_sha1
```

```
example$ elfsign list -f signer -e lib/libmylib.so.1
CN=VENDOR, OU=Software Development, O=Vendor Inc.
```

退出状态

将返回以下退出值：

值	含义	子命令
0	操作成功	sign/verify/request
1	参数无效	
2	无法验证 ELF 目标文件	verify
3	无法打开 ELF 目标文件	sign/verify
4	无法装入证书或证书无效	sign/verify
5	无法装入私钥，私钥无效或令牌标签无效	sign
6	无法添加签名	sign
7	尝试验证未签名的目标文件或目标文件不是 ELF 文件	verify

文件

/etc/crypto/certs 未使用 -c 标志时，供 verify 子命令搜索的目录

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	developer/base-developer-utilities
接口稳定性	请参见下文。

elfsign 命令及其子命令是 "Committed"（已确定）。尽管应用程序不应依赖于 elfsign 的输出格式，但 list 子命令的输出格式仍为 "Committed"（已确定）。

另请参见

[date\(1\)](#)、[pktool\(1\)](#)、[cryptoadm\(1M\)](#)、[libpkcs11\(3LIB\)](#)、[attributes\(5\)](#)

引用名	elfwrap – wrap data in an ELF file
用法概要	elfwrap [-64] [-o <i>relobj-file</i>] [-z target=sparc x86] <i>data-file</i> ...
描述	<p>The <code>elfwrap</code> utility creates an ELF relocatable object file from one or more data files. The relocatable object encapsulates each data file within an individual section, together with symbols that can be used to reference the section. The relocatable object is appropriate for inclusion with a subsequent <code>link-edit</code>. Users can reference the encapsulated data using the associated symbols.</p> <p>By default, a 32-bit ELF relocatable object is created that is appropriate for the machine on which <code>elfwrap</code> is executed. The <code>-64</code> option can be used to create a 64-bit ELF relocatable object. The <code>-z target</code> option can be used to create a relocatable object for a specific machine type.</p> <p>注 – Any data encapsulated with <code>elfwrap</code> must be in a format appropriate for the destination target.</p> <p>By default, the relocatable object <code>a.wrap.o</code> is created. The <code>-o</code> option can be used to specify an alternative relocatable object name.</p> <p>The <code>basename(1)</code> of each data file is used to create various pieces of ELF information. For example, if the input data file is <code>ISV/isv-data</code>, the following ELF information is created within the relocatable object.</p> <p>An ELF section named <code>.isv-data</code> This section contains the entire contents of the input data file.</p> <p>An ELF symbol named <code>isv-data_start</code> This symbol reflects the starting address of the <code>.isv-data</code> section.</p> <p>An ELF symbol named <code>isv-data_end</code> This symbol reflects the address of the first location after the <code>.isv-data</code> section.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> <code>-64</code> Create a 64-bit ELF relocatable object. <code>-o relobj-file</code> Produce a relocatable object that is named <i>relobj-file</i>. <code>-z target=sparc x86</code> Specifies the machine type for the output relocatable object. Supported targets are <code>sparc</code> and <code>x86</code>. The 32-bit machine type for the specified target is used unless the <code>-64</code> option is also present, in which case the corresponding 64-bit machine type is used. By default, the relocatable object that is generated is 32-bit for the machine one which <code>elfwrap</code> is executed.

示例 The following example encapsulates the system passwd file and the system group file within a relocatable object `passgroup.o`.

```
example% elfwrap -o passgroup.o /etc/passwd /etc/group
example% elfdump -s passgroup.o | egrep "passwd|group"
  [2] 0x00000000 0x00000000 SECT LOCL D 0 .passwd
  [3] 0x00000000 0x00000000 SECT LOCL D 0 .group
  [7] 0x00000000 0x000002f0 OBJT GLOB D 0 .passwd passwd_start
  [8] 0x000002f0 0x00000000 OBJT GLOB D 0 .passwd passwd_end
  [9] 0x00000000 0x00000121 OBJT GLOB D 0 .group group_start
 [10] 0x00000121 0x00000000 OBJT GLOB D 0 .group group_end
example% strings -N.passwd passgroup.o | head -1
root:x:0:0:Super-User:/:usr/sbin/sh
example% strings -N.group passgroup.o | head -1
root::0:
```

This relocatable object can be referenced from the following user code.

```
example% cat main.c
#include      <stdio.h>

extern char   passwd_start, passwd_end;

void main()
{
    char      *pstart = &passwd_start, *pend = &passwd_end;
    char      *str, *lstr;

    for (lstr = str = pstart; str < pend; str++) {
        if ((*str == '\n') && (str != (pend - 1))) {
            (void) printf("%.s", (++str - lstr), lstr);
            lstr = str;
        }
    }
}
example% cc -o main main.c passgroup.o
example% ./main
root:x:0:0:Super-User:/:usr/sbin/sh
....
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
```

文件 `a.wrap.o` The default relocatable object file created.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

另请参见

[elfdump\(1\)](#), [ld\(1\)](#), [strings\(1\)](#), [elf\(3ELF\)](#), [attributes\(5\)](#), [ddi_modopen\(9F\)](#)

《链接程序和库指南》

引用名 encrypt, decrypt – 加密或解密文件

用法概要

```
/usr/bin/encrypt -l

/usr/bin/encrypt -a algorithm [-v]
    [-k key_file | -K key_label [-T token_spec]]
    [-i input_file] [-o output_file]

/usr/bin/decrypt -l

/usr/bin/decrypt -a algorithm [-v]
    [-k key_file | -K key_label [-T token_spec]]
    [-i input_file] [-o output_file]
```

描述 此实用程序可使用指定算法加密或解密给定的文件或 `stdin`。如果未指定输出文件，将输出到标准输出。如果 `cryptoadm -i` 和 `-o` 选项指定同一个文件，加密输出将写入到同一文件系统中的临时工作文件，然后用于替换原始文件。

解密时，如果 `cryptoadm -i` 和 `-o` 指定同一个文件，明文将替换密文文件。

`encrypt` 的输出文件和 `decrypt` 的输入文件包含以下信息：

- 输出格式版本号，采用网络字节顺序的 4 个字节。当前版本是 1。
- 密钥生成函数中使用的迭代，采用网络字节顺序的 4 个字节。
- IV (`ivlen` 字节) [1]。iv 数据由等于一个块大小的随机字节数生成。
- 密钥生成中使用的 Salt 数据（16 字节）。
- 密文数据。

选项

支持以下选项：

- a *algorithm* 指定加密或解密过程中要使用的算法的名称。有关详细信息，请参见“用法”部分的“算法”。
- i *input_file* 指定输入文件。如果未指定 *input_file*，则缺省为 `stdin`。
- k *key_file* 指定包含用于加密算法的密钥值的文件。每种算法都具有特定的密钥材料要求，如 PKCS#11 规范中所述。如果未指定 `-k`，`encrypt` 会使用 `getpassphrase(3C)` 提示您提供密钥材料。密钥文件的大小确定了密钥长度，从终端设置的口令短语始终用于为密钥长度可变的密码生成长度为 128 位的密钥。

有关生成密钥文件的信息，请参见 `pktool(1)` 中的 `genkey` 子命令。此外，也可以使用 `dd(1M)`。
- K *key_label* 指定 PKCS#11 令牌中的对称令牌密钥的标签。
- l 显示系统上可用的算法的列表。此列表可依加密框架的配置而变化。以位为单位显示密钥大小。
- o *output_file* 指定输出文件。如果未指定 *output_file*，则缺省为 `stdout`。如果使用 `stdout` 而不重定向到文件，则终端窗口可能会因原始加密或解密数据中止终端仿真而挂起，与有时查看二进制文件所遇到的情况类似。

`-T token_spec` 指定 PKCS#11 令牌，而不使用指定 `-k` 时的缺省软令牌对象存储。

`token_spec` 的格式为：

`token_name [:manuf_id [:serial_no]]`

当令牌标签包含结尾空格时，为方便起见，此选项不要求用户键入这些空格。

使用冒号分隔令牌标识字符串。如果任一部分中包含冒号(:) 文本字符，必须使用反斜杠(\) 对其进行转义。如果未找到冒号(:)，则将整个字符串（最多 32 个字符）视为令牌标签。如果仅找到一个冒号(:)，则该字符串是令牌标签和生产商。

`-v` 显示详细信息。请参见**详细模式**。

用法

算法

在 `-l` 选项中显示支持的算法及其最小和最大密钥大小。这些算法由加密框架提供。所支持的每种算法都是 PKCS #11 机制的一个别名，对于特定的算法类型而言，PKCS #11 机制是最常用的也是限制最少的版本。例如，`des` 是 `CKM_DES_CBC_PAD` 的别名，`arcfour` 是 `CKM_RC4` 的别名。不支持没有任何填充或 ECB 的算法变体。

这些别名与 `-a` 选项一起使用，并且区分大小写。

口令短语

若在加密和解密任务期间未使用 `-k` 选项，则会提示用户提供口令短语。使用 PKCS #5 中指定的 PBKDF2 算法将该口令短语处理成更安全的密钥。

当使用口令短语进行加密和解密时，会使用 <http://www.rsasecurity.com> PKCS #5 v2.0 中定义的 PBKDF2 算法将用户输入的口令短语转换成加密密钥。

详细模式

如果向命令提供输入文件，则屏幕上会显示一个进度条。进度条会在每完成 25% 时使用一个管道符号(|) 表示。如果输入来自标准输入，每读取 40KB 后都会显示一个句点(.)。在两种输入方法都完成时，会显示 Done。

示例

示例 1 列出可用算法

以下示例列出了可用的算法：

```
example$ encrypt -l
  Algorithm      Keysize:  Min   Max
  -----
  aes             128     128
  arcfour         8        128
  des             64       64
  3des            192     192
```

示例 2 使用 AES 加密

以下示例使用 AES 加密并提示用户提供加密密钥：

```
example$ encrypt -a aes -i myfile.txt -o secretstuff
```

示例 3 对密钥文件使用 AES 加密

以下示例在已创建密钥文件后使用 AES 加密：

```
example$ pktool genkey keystore=file keytype=aes keylen=128 \  
outkey=key  
example$ encrypt -a aes -k key -i myfile.txt -o secretstuff
```

示例 4 使用输入管道提供加密的磁带备份

以下示例使用输入管道提供加密的磁带备份：

```
example$ ufsdump 0f - /var | encrypt -a arcfour \  
-k /etc/mykeys/backup.k | dd of=/dev/rmt/0
```

示例 5 使用输入管道恢复磁带备份

以下示例使用输入管道恢复磁带备份：

```
example$ decrypt -a arcfour -k /etc/mykeys/backup.k \  
-i /dev/rmt/0 | ufsrestore xvf -
```

示例 6 使用 3DES 算法加密输入文件

以下示例使用存储在 `des3key` 文件中的 192 位密钥加密 `inputfile` 文件：

```
example$ encrypt -a 3des -k des3key -i inputfile -o outputfile
```

示例 7 使用 DES 令牌密钥加密输入文件

以下示例使用软令牌密钥库中的 DES 令牌密钥加密输入文件。可使用 [pktool\(1\)](#) 生成 DES 令牌密钥：

```
example$ encrypt -a des -K mydeskey \  
-T "Sun Software PKCS#11 softtoken" -i inputfile \  
-o outputfile
```

退出状态

将返回以下退出值：

0 成功完成。

>0 出现错误。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed (已确定)

另请参见

[digest\(1\)](#)、[pktool\(1\)](#)、[mac\(1\)](#)、[dd\(1M\)](#)、[getpassphrase\(3C\)](#)、[libpkcs11\(3LIB\)](#)、[attributes\(5\)](#)

《Oracle Solaris 11.1 管理：安全服务》

RSA PKCS#11 v2.11: <http://www.rsasecurity.com>

RSA PKCS#5 v2.0: <http://www.rsasecurity.com>

引用名 enhance – 增强的命令行编辑功能

用法概要 enhance *command* [*argument*]...

描述 enhance 程序为不具有任何源代码的、第三方应用程序的用户提供了增强的命令行编辑功能。它通过在应用程序和实际终端之间放置一个伪终端来实现此功能。它使用 `tecla` 命令行编辑库从实际终端读取输入，然后通过伪终端将每个刚完成的输入行转发至应用程序。来自应用程序的所有输出将按原样转发回实际终端。

当应用程序停止生成输出的时间超出十分之一秒时，`enhance` 程序会将最后一个未完成的输出行视为提示符，并重新显示用户已在其后键入的任何未完成的输入行。用户觉察不到的微小延迟对于程序的正确操作而言是不必要的。它仅仅是一项优化，旨在阻止频繁地重新显示输入行，进而避免降低输出速率。

由 `Tecla` 库提供的用户级命令行编辑功能记录在 [tecla\(5\)](#) 手册页中。

缺点 唯一的一个尚未解决的主要问题是如何处理用于更改控制终端是否回显所键入内容的应用程序。例如，要求输入口令的程序（例如 `ftp` 和 `telnet`）临时通知其控制终端不要回显用户键入的内容。由于此请求将传至伪终端的应用程序端，所以 `enhance` 程序没有办法知道发生了此问题，并会在用户键入口令时继续将所键入的内容回显到其控制终端。

此外，在执行主机应用程序之前，`enhance` 程序最初会将伪终端设置为 `noecho` 模式，因此它发送至该程序的所有内容都不会冗余回显。如果切换到 `noecho` 模式的程序在之后显式恢复回显（而不是恢复以前的终端模式），则以后每当您输入新的输入行时，都会在其下一行显示重复内容。

文件 `/usr/lib/libtecla.so` `tecla` 库
`~/.teclarc` `tecla` 个人定制文件。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	library/libtecla
接口稳定性	Committed（已确定）

另请参见 [libtecla\(3LIB\)](#)、[attributes\(5\)](#)、[tecla\(5\)](#)

引用名	env – set environment for command invocation
用法概要	<pre>/usr/bin/env [-i -] [name=value]... [utility [arg...]]</pre> <pre>/usr/xpg4/bin/env [-i -] [name=value]... [utility [arg...]]</pre>
描述	<p>The env utility obtains the current environment, modifies it according to its arguments, then invokes the utility named by the <i>utility</i> operand with the modified environment.</p> <p>Optional arguments are passed to <i>utility</i>. If no <i>utility</i> operand is specified, the resulting environment is written to the standard output, with one <i>name=value</i> pair per line.</p>
/usr/bin	If env executes commands with arguments, it uses the default shell /usr/bin/sh (see sh(1)).
/usr/xpg4/bin	If env executes commands with arguments, it uses /usr/xpg4/bin/sh (see ksh88(1)).
选项	<p>The following options are supported:</p> <p>-i - Ignores the environment that would otherwise be inherited from the current shell. Restricts the environment for <i>utility</i> to that specified by the arguments.</p>
操作数	<p>The following operands are supported:</p> <p>name=value Arguments of the form <i>name=value</i> modify the execution environment, and are placed into the inherited environment before <i>utility</i> is invoked.</p> <p>utility The name of the utility to be invoked. If <i>utility</i> names any of the special shell built-in utilities, the results are undefined.</p> <p>arg A string to pass as an argument for the invoked utility.</p>
示例	<p>示例 1 Invoking utilities with new PATH values</p> <p>The following utility:</p> <pre>example% env -i PATH=/mybin mygrep xyz myfile</pre> <p>invokes the utility <i>mygrep</i> with a new PATH value as the only entry in its environment. In this case, PATH is used to locate <i>mygrep</i>, which then must reside in /mybin.</p>
环境变量	<p>See environ(5) for descriptions of the following environment variables that affect the execution of env: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.</p> <p>PATH Determine the location of the <i>utility</i>. If PATH is specified as a <i>name=value</i> operand to env, the value given shall be used in the search for <i>utility</i>.</p>
退出状态	<p>If <i>utility</i> is invoked, the exit status of env is the exit status of <i>utility</i>. Otherwise, the env utility returns one of the following exit values:</p> <p>0 Successful completion.</p> <p>1 - 125 An error occurred.</p>

126 *utility* was found but could not be invoked.

127 *utility* could not be found.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	enabled

/usr/xpg4/bin

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[ksh88\(1\)](#), [sh\(1\)](#), [exec\(2\)](#), [profile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名	eqn, neqn, checkeq – typeset mathematics test
用法概要	<pre>eqn [-d xy] [-f n] [-p n] [-s n] [file]... neqn [file]... checkeq [file]...</pre>
描述	<p>eqn and neqn are language processors to assist in describing equations. eqn is a preprocessor for troff(1) and is intended for devices that can print troff's output. neqn is a preprocessor for nroff(1) and is intended for use with terminals. Usage is almost always:</p> <pre>example% eqn file ... troff example% neqn file ... nroff</pre> <p>If no <i>files</i> are specified, eqn and neqn read from the standard input. A line beginning with .EQ marks the start of an equation. The end of an equation is marked by a line beginning with .EN. Neither of these lines is altered, so they may be defined in macro packages to get centering, numbering, and so on. It is also possible to set two characters as “delimiters”; subsequent text between delimiters is also treated as eqn input.</p> <p>checkeq reports missing or unbalanced delimiters and .EQ/.EN pairs.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -dxy Sets equation delimiters set to characters <i>x</i> and <i>y</i> with the command-line argument. The more common way to do this is with <code>delim xy</code> between .EQ and .EN. The left and right delimiters may be identical. Delimiters are turned off by <code>delim off</code> appearing in the text. All text that is neither between delimiters nor between .EQ and .EN is passed through untouched. -fn Changes font to <i>n</i> globally in the document. The font can also be changed globally in the body of the document by using the <code>gfont n</code> directive, where <i>n</i> is the font specification. -pn Reduces subscripts and superscripts by <i>n</i> point sizes from the previous size. In the absence of the -p option, subscripts and superscripts are reduced by 3 point sizes from the previous size. -sn Changes point size to <i>n</i> globally in the document. The point size can also be changed globally in the body of the document by using the <code>gsi size n</code> directive, where <i>n</i> is the point size.
操作数	<p>The following operands are supported:</p> <p><i>file</i> The nroff or troff file processed by eqn or neqn.</p>
Eqn Language	<p>The nroff version of this description depicts the output of neqn to the terminal screen exactly as neqn is able to display it. To see an accurate depiction of the output, view the printed version of this page.</p>

Tokens within eqn are separated by braces, double quotes, tildes, circumflexes, SPACE, TAB, or NEWLINE characters. Braces { } are used for grouping. Generally speaking, anywhere a single character like x could appear, a complicated construction enclosed in braces may be used instead. A tilde (~) represents a full SPACE in the output; a circumflex (^) half as much.

Subscripts and superscripts:

These are produced with the keywords sub and sup.

x sub i
makes x_i

a sub i sup 2
produces a_i^2

e sup {x sup 2 + y sup 2}
gives $e^{x^2+y^2}$

Fractions:

Fractions are made with over.

a over b
yields $\frac{a}{b}$

Square Roots:

These are made with sqrt

1 over sqrt {ax sup 2 +bx+c}
results in $\frac{1}{\sqrt{ax^2+bx+c}}$

Limits:

The keywords from and to introduce lower and upper limits on arbitrary things:

lim from {n→ inf } sum from 0 to n x sub i
makes $\lim_{n \rightarrow \infty} \sum_0^n x_i$

Brackets and Braces:

Left and right brackets, braces, and the like, of the right height are made with left and right.

left [x sup 2 + y sup 2 over alpha right] ~-1
produces $\left[x^2 + \frac{y^2}{\alpha} \right] = 1$

The right clause is optional. Legal characters after left and right are braces, brackets, bars, c and f for ceiling and floor, and "" for nothing at all (useful for a right-side-only

bracket).

Vertical piles:

Vertical piles of things are made with `pile`, `lpile`, `cpile`, and `rpile`.

`pile {a above b above c}`

$$\begin{array}{c} a \\ b \\ \text{produces } c \end{array}$$

There can be an arbitrary number of elements in a pile. `lpile` left-justifies, `pile` and `cpile` center, with different vertical spacing, and `rpile` right justifies.

Matrices:

Matrices are made with `matrix`.

`matrix { lcol { x sub i above y sub 2 } ccol { 1 above 2 } }`

$$\begin{array}{c} x_i \quad 1 \\ \text{produces } y_2 \quad 2 \end{array}$$

In addition, there is `rcol` for a right-justified column.

Diacritical marks:

Diacritical marks are made with `dot`, `dotdot`, `hat`, `tilde`, `bar`, `vec`, `dyad`, and `under`.

`x dot = f(t) bar`

$$\text{is } \dot{x} = \overline{f(t)}$$

`y dotdot bar ~~~ n under`

$$\text{is } \ddot{y} = \underline{n}$$

`x vec ~~~ y dyad`

$$\text{is } \vec{x} = \hat{y}$$

Sizes and Fonts:

Sizes and font can be changed with `size n` or `size ±n`, `roman`, `italic`, `bold`, and `font n`.

Size and fonts can be changed globally in a document by `gsize n` and `gfont n`, or by the command-line arguments `-sn` and `-fn`.

Successive display arguments:

Successive display arguments can be lined up. Place `mark` before the desired lineup point in the first equation; place `\lineup` at the place that is to line up vertically in subsequent equations.

Shorthands:

Shorthands may be defined or existing keywords redefined with `define`:

`define thing% replacement%` Defines a new token called *thing* which will be replaced by *replacement* whenever it appears thereafter. The % may be any character that does not occur in *replacement*.

Keywords and Shorthands:

Keywords like `sum int inf` and shorthands like `>= →` and `!=` are recognized.

Greek letters:

Greek letters are spelled out in the desired case, as in `a\lpha` or `GAMMA`.

Mathematical words:

Mathematical words like `sin`, `cos`, and `log` are made Roman automatically.

`troff(1)` four-character escapes like `\(bu (•)` can be used anywhere. Strings enclosed in double quotes `"..."` are passed through untouched; this permits keywords to be entered as text, and `troff` can be used to communicate with `troff` when all else fails.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

另请参见

[nroff\(1\)](#), [tbl\(1\)](#), [troff\(1\)](#), [attributes\(5\)](#), [ms\(5\)](#)

已知问题

To embolden characters such as digits and parentheses, it is necessary to quote them, as in `'bold "12.3"`.

引用名	error – 在恰当的源行上插入编译器错误消息
用法概要	error [-n] [-q] [-s] [-v] [-t <i>suffixlist</i>] [-I <i>ignorefile</i>] [<i>filename</i>]
描述	<p>error 分析由大量编译器和语言处理器产生的错误消息。它取代了在纸上记下错误缩写这种费力的传统方法，允许同时查看错误消息和源代码。</p> <p>error 可以从指定的文件 <i>filename</i> 或标准输入中查看错误消息，并且：</p> <ul style="list-style-type: none"> ■ 确定每个错误消息是由哪个语言处理器产生的。 ■ 确定文件名和错误行的行号。 ■ 将错误消息插入到源文件中错误行的正前方。 <p>无法按语言处理器或内容进行分类的错误消息不会插入到任何文件中，但将被发送到标准输出中。只有在读取所有输入后，error 才会触动源文件。</p> <p>error 被设计为使用其标准输入运行，其标准输入通过管道连接到错误消息源。一些语言处理器将错误消息放置在它们的标准错误文件中，而其他语言处理器将其消息放置在标准输出中。因此，这两种错误源都应通过管道连接到 error。例如，使用 csh 语法时，以下命令会分析在生成 lint 时 make(1S) 所运行的所有程序产生的所有错误消息。</p> <pre>example% make -s lint & error -q -v</pre> <p>error 了解由 as(1)、cpp(1)、ld(1)、make(1S) 和其他编译器产生的错误消息。对于除 Pascal 之外的所有语言，都将错误消息限制为一行。某些错误消息引用了多个文件中的多个行，在这种情况下，error 会复制错误消息，并将其插入到所有相应的位置。</p>
选项	<p>-n 不触动任何文件；所有错误消息将被发送到标准输出。</p> <p>-q error 询问是否应该触动文件。需要回答 "y" 或 "n" 以继续操作。不使用 -q 选项意味着将触动所有被引用的文件（那些引用了被丢弃的错误消息的文件除外）。</p> <p>-s 输出关于错误类别的统计信息。</p> <p>-v 在触动所有文件之后，打开可视化编辑器 vi，将其设置为用于编辑所触动的所有文件，并定位到所触动的第一个文件中的第一个错误。如果无法找到 vi(1)，则从标准位置尝试 ex(1) 或 ed(1)。</p> <p>-t <i>suffixlist</i> 采用以下参数作为后缀列表。不触动其后缀没有出现在后缀列表中的文件。后缀列表以点分隔，可以使用 "*" 通配符。因此，后缀列表：</p> <pre>.c.y.f*.h</pre> <p>允许 error 触动以 ".c"、".y"、".f*" 和 ".h" 结尾的文件。</p> <p>error 捕捉中断和终止信号，并以有序的方式地进行终止。</p>

示例

示例1 使用 `error` 命令

在以下 C shell (`/usr/bin/csh`) 示例中，`error` 从 FORTRAN 编译器获取输入：

```
example% f77 -c any.f |& error options
```

以下是使用 Korn shell (`/usr/bin/ksh`) 的同一示例：

```
example% f77 -c any.f 2>&1 | error options
```

用法

`error` 对错误消息执行以下六项操作之一。

<code>synchronize</code>	一些语言处理器会产生一些简短错误，描述它们正在处理的文件。 <code>error</code> 使用这些错误来为每个错误消息中不包括文件名的那些语言确定文件名。这些同步消息完全由 <code>error</code> 使用。
<code>discard</code>	丢弃来自 <code>lint</code> 的引用了两个 <code>lint</code> 库 (<code>/usr/lib/lint/llib-1c</code> 和 <code>/usr/lib/lint/llib-port</code>) 之一的错误消息，以防止意外触动这些库。同样，这些错误消息完全由 <code>error</code> 使用。
<code>nullify</code>	如果来自 <code>lint</code> 的错误消息引用了一个特定的函数，并且已知该函数生成不受关注的诊断信息，则可以废除这些错误消息。被废除的错误消息不会插入到源文件中，但会写入到标准输出中。要忽略的函数名称是从用户的起始目录中名为 <code>.errorrc</code> 的文件或从由 <code>-I</code> 选项指定的文件中获取的。如果该文件不存在，则不会废除任何错误消息。如果该文件存在，必须是每个函数名称占用一行。
<code>not file specific</code>	无法凭直觉知晓的错误消息将被分组到一起，并在触动任何文件之前写入到标准输出中。它们不会插入到任何源文件中。
<code>file specific</code>	当触动某个特定文件时，将引用该文件的错误消息写入到标准输出，但不将引用特定行的错误消息写入到标准输出。
<code>true errors</code>	可以凭直觉知晓的错误消息可供插入到它们引用的文件中。

只将真正的错误消息插入到源文件中。其他错误消息完全由 `error` 使用或被写入到标准输出中。`error` 将错误消息插入到源文件中的行上，并将放在错误消息中的行号之前。每个错误消息将被变成一行语言注释，在错误的开头和结尾分别以字符串 `###` 和 `%%` 进行内部标记。这样，可以使用编辑器更方便地以模式搜索方式查找错误，并可以方便地删除消息。此外，每个错误消息包含该消息引用的行的源行号。格式合理的源程序在重新编译后可以仍将错误消息包含在其中，并且这些错误消息本身不会导致新的错误。对于以自由格式语言（例如 C 或 Pascal）编译的格式很差的源程序，可能会将一个注释插入到另一个注释中，这会严重破坏以后的编译。要避免这种情况，应对源程序进行格式设置，使注释的结尾所在的行上不存在任何语言语句。

文件

`~/.errorrc` `lint` 错误消息中要忽略的函数名称
 设备/`dev/tty` 用户的电传打字机

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	developer/base-developer-utilities

另请参见

[as\(1\)](#)、[cpp\(1\)](#)、[csh\(1\)](#)、[ed\(1\)](#)、[ex\(1\)](#)、[make\(1S\)](#)、[ld\(1\)](#)、[vi\(1\)](#)、[attributes\(5\)](#)

已知问题

直接打开 tty 设备进行用户输入。

具有多个链接的源文件生成一个新的文件副本，该副本仅具有一个到该文件的链接。

更改语言处理器的错误消息格式可能会导致 `error` 无法理解该错误消息。

`error`，由于它是纯机械的，无法过滤出由微小的语法错误所开启的“水闸”导致的后续错误。在丢弃这些相关的错误方面，还是人类更为擅长。

Pascal 错误消息应当放置在受影响的行后，但 `error` 将它们放置在受影响的行前。`error` 还打乱了用来标记错误点的 ‘|’ 的对齐。

`error` 是为以相当高的速度在 CRT 上运行而设计的。它不太适合低速终端，也不适用于硬拷贝终端。

引用名

ex – text editor

用法概要

```

/usr/bin/ex [-| -s] [-l] [-L] [-R] [-r file] [-t tag]
            [-v] [-V] [-wn] [+command | -c command] file...

/usr/xpg4/bin/ex [-| -s] [-l] [-L] [-R] [-r file]
                [-t tag] [-v] [-V] [-wn]
                [+command | -c command] file...

/usr/xpg6/bin/ex [-| -s] [-l] [-L] [-R] [-r file]
                [-t tag] [-v] [-V] [-wn]
                [+command | -c command] file...

```

描述

The ex utility is the root of a family of editors: ex and vi. ex is a superset of [ed\(1\)](#), with the most notable extension being a display editing facility. Display based editing is the focus of vi.

If you have a CRT terminal, you can wish to use a display based editor; in this case see [vi\(1\)](#), which is a command which focuses on the display-editing portion of ex.

If you have used ed you find that, in addition to having all of the ed commands available, ex has a number of additional features useful on CRT terminals. Intelligent terminals and high speed terminals are very pleasant to use with vi. Generally, the ex editor uses far more of the capabilities of terminals than ed does, and uses the terminal capability data base (see [terminfo\(4\)](#)) and the type of the terminal you are using from the environment variable TERM to determine how to drive your terminal efficiently. The editor makes use of features such as insert and delete character and line in its visual command (which can be abbreviated vi) and which is the central mode of editing when using the vi command.

The ex utility contains a number of features for easily viewing the text of the file. The z command gives easy access to windows of text. Typing ^D (CTRL-D) causes the editor to scroll a half-window of text and is more useful for quickly stepping through a file than just typing return. Of course, the screen-oriented visual mode gives constant access to editing context.

The ex utility gives you help when you make mistakes. The undo (u) command allows you to reverse any single change which goes astray. ex gives you a lot of feedback, normally printing changed lines, and indicates when more than a few lines are affected by a command so that it is easy to detect when a command has affected more lines than it should have.

The editor also normally prevents overwriting existing files, unless you edited them, so that you do not accidentally overwrite a file other than the one you are editing. If the system (or editor) crashes, or you accidentally hang up the telephone, you can use the editor recover command (or -r *file* option) to retrieve your work. This gets you back to within a few lines of where you left off.

The ex utility has several features for dealing with more than one file at a time. You can give it a list of files on the command line and use the next (n) command to deal with each in turn. The next command can also be given a list of file names, or a pattern as used by the shell to

specify a new set of files to be dealt with. In general, file names in the editor can be formed with full shell metasyntax. The metacharacter '%' is also available in forming file names and is replaced by the name of the current file.

The editor has a group of buffers whose names are the ASCII lower-case letters (a-z). You can place text in these named buffers where it is available to be inserted elsewhere in the file. The contents of these buffers remain available when you begin editing a new file using the edit (e) command.

There is a command & in ex which repeats the last substitute command. In addition, there is a confirmed substitute command. You give a range of substitutions to be done and the editor interactively asks whether each substitution is desired.

It is possible to ignore the case of letters in searches and substitutions. ex also allows regular expressions which match words to be constructed. This is convenient, for example, in searching for the word "edit" if your document also contains the word "editor."

ex has a set of options which you can set to tailor it to your liking. One option which is very useful is the autoindent option that allows the editor to supply leading white space to align text automatically. You can then use ^D as a backtab and space or tab to move forward to align new code easily.

Miscellaneous useful features include an intelligent join (j) command that supplies white space between joined lines automatically, commands < and > which shift groups of lines, and the ability to filter portions of the buffer through commands such as sort.

选项

The following options are supported:

- -s	Suppresses all interactive user feedback. This is useful when processing editor scripts.
-l	Sets up for editing LISP programs.
-L	Lists the name of all files saved as the result of an editor or system crash.
-R	Readonly mode. The readonly flag is set, preventing accidental overwriting of the file.
-r <i>file</i>	Edits <i>file</i> after an editor or system crash. (Recovers the version of <i>file</i> that was in the buffer when the crash occurred.)
-t <i>tag</i>	Edits the file containing the <i>tag</i> and positions the editor at its definition. It is an error to specify more than one -t option.
-v	Starts up in display editing state, using vi. You can achieve the same effect by typing the vi command itself.

-V	Verbose. When ex commands are read by means of standard input, the input is echoed to standard error. This can be useful when processing ex commands within shell scripts.
-wn	Sets the default window size to <i>n</i> . This is useful when using the editor over a slow speed line.
+ <i>command</i> -c <i>command</i>	Begins editing by executing the specified editor <i>command</i> (usually a search or positioning command).
/usr/xpg4/bin/ex, /usr/xpg6/bin/ex	If both the -t <i>tag</i> and the -c <i>command</i> options are given, the -t <i>tag</i> is processed first. That is, the file containing the tag is selected by -t and then the command is executed.

操作数

The following operand is supported:

file A path name of a file to be edited.

用法

This section defines the ex states, commands, initializing options, and scanning pattern formations.

ex States

Command	Normal and initial state. Input prompted for by “:”. The line kill character cancels a partial command.
Insert	Entered by a, i, or c. Arbitrary text can be entered. Insert state normally is terminated by a line having only “.” on it, or, abnormally, with an interrupt.
Visual	Entered by typing vi. Terminated by typing Q or ^\ (Control-^).

ex Command Names and Abbreviations

Command Name	Abbreviation	Command Name	Abbreviation	Command Name	Abbreviation
abbrev	ab	map		set	se
append	a	mark	ma	shell	sh
args	ar	move	m	source	so
change	c	next	n	substitute	s
copy	co	number	nu	unabbrev	unab
delete	d	preserve	pre	undo	u
edit	e	print	p	unmap	unm
file	f	put	pu	version	ve
global	g	quit	q	visual	vi

insert	i	read	r	write	
w					
join	j	recover	rec	xit	x
list	l	rewind	rew	yank	ya
Join	[<i>range</i>] j[<i>oin</i>][!] [<i>count</i>] [<i>flags</i>]				

Join Command Arguments

If count is specified:

`/usr/bin/ex, /usr/xpg6/bin/ex`

If no address is specified, the join command behaves as if *2addr* were the current line and the current line plus *count* (`. , . + count`). If one address is specified, the join command behaves as if *2addr* were the specified address and the specified address plus *count* (`addr, addr + count`).

`/usr/xpg4/bin/ex`

If no address is specified, the join command behaves as if *2addr* were the current line and the current line plus *count - 1* (`. , . + count - 1`). If one address is specified, the join command behaves as if *2addr* were the specified address and the specified address plus *count - 1* (`addr, addr + count - 1`).

`/usr/bin/ex, /usr/xpg4/bin/ex, /usr/xpg6/bin/ex`

If two or more addresses are specified, the join command behaves as if an additional address, equal to the last address plus *count - 1* (`addr1, . . . , lastaddr, lastaddr + count - 1`), was specified. If this results in a second address greater than the last line of the edit buffer, it is corrected to be equal to the last line of the edit buffer.

If no count is specified:

`/usr/bin/ex, /usr/xpg4/bin/ex, /usr/xpg6/bin/ex`

If no address is specified, the join command behaves as if *2addr* were the current line and the next line (`. , . + 1`). If one address is specified, the join command behaves as if *2addr* were the specified address and the next line (`addr, addr + 1`).

Additional ex Command Arguments

`/usr/bin/ex, /usr/xpg6/bin/ex`

For the following ex commands, if *count* is specified, it is equivalent to specifying an additional address to the command. The additional address is equal to the last address specified to the command (either explicitly or by default) plus *count - 1*. If this results in an address greater than the last line of the edit buffer, it is corrected to equal the last line of the edit buffer.

`/usr/xpg4/bin/ex`

For the following ex commands, if both a count and a range are specified for a command that uses them, the number of lines affected is taken from the count value rather than the range. The starting line for the

command is taken to be the first line addressed by the range.

Abbreviate	ab[brev] word rhs
Append	[line]a[ppend][!]
Arguments	ar[gs]
Change	[range] c[hange][!] [count]
Change Directory	chd[ir][!] [directory]; cd[!] [directory]
Copy	[range] co[py] line [flags]; [range] t line [flags]
Delete	[range] d[etele] [buffer] [count] [flags]
Edit	e[dit][!] [+line][file]; ex[!] [+line] [file]
File	f[ile] [file]
Global	[range] g[lobal] /pattern/ [commands]; [range] v /pattern/ [commands]
Insert	[line] i[nsert][!]
List	[range] l[ist] [count] [flags]
Map	map[!] [x rhs]
Mark	[line] ma[rk] x; [line] k x
Move	[range] m[ove] line
Next	n[ext][!] [file ...]
Open	[line] o[pen] /pattern/ [flags]
Preserve	pre[serve]
Put	[line] pu[t] [buffer]
Quit	q[uit][!]
Read	[line] r[ead][!] [file]
Recover	rec[over] file
Rewind	rew[ind][!]
Set	se[t] [option[=[value]]...] [nooption...] [option?...] [all]
Shell	sh[ell]
Source	so[urce] file
Suspend	su[spend][!]; st[op][!]

Tag	ta[g][!] tagstring
Unabbreviate	una[bbrev] word
Undo	u[ndo]
Unmap	unm[ap][!] x
Visual	[line] v[isual] [type] [count] [flags]
Write	[range] w[rite][!] [>>] [file]; [range] w[rite][!] [file]; [range] wq[!] [>>] [file]
Write and Exit	[range] x[it][!] [file]
Yank	[range] ya[nk] [buffer] [count]
Adjust Window	[line] z [type] [count] [flags]
Escape	! command [range]! command
Scroll	EOF
Write Line Number	[line] = [flags]
Execute	@ buffer; * buffer

/usr/bin/ex, /usr/xpg4/bin/ex, /usr/xpg6/bin/ex

For the following *ex* commands, if *count* is specified, it is equivalent to specifying an additional address to the command. The additional address is equal to the last address specified to the command (either explicitly or by default) plus *count*-1. If this results in an address greater than the last line of the edit buffer, it is corrected to equal the last line of the edit buffer.

Number	[range] nu[mber] [count] [flags]; [range] # [count] [flags]
Print	[range] p[rint] [count] [flags]
Substitute	[range] s[ubstitute] [/pattern/repl/[options] [count] [flags]]
Shift Left	[range] < [count] [flags]
Shift Right	[range] > [count] [flags]
Resubstitute	[range] & [options] [count] [flags]; [range] s[ubstitute] [options] [count] [flags]; [range] ~ [options] [count] [flags]

ex Commands	&	resubst
	CR	print next
	>	rshift

<	lshift
^D	scroll
z	window
!	shell escape

ex Command
Addresses

<i>n</i>	line <i>n</i>
.	current
\$	last
+	next
-	previous
<i>+n</i>	<i>n</i> forward
%	1,\$
<i>/pat</i>	next with <i>pat</i>
<i>?pat</i>	previous with <i>pat</i>
<i>x-n</i>	<i>n</i> before <i>x</i>
<i>x,y</i>	<i>x</i> through <i>y</i>
' <i>x</i>	marked with <i>x</i>
"	previous context

Initializing Options

EXINIT	place set's here in environment variable
\$HOME/.exrc	editor initialization file
./exrc	editor initialization file
set <i>x</i>	enable option <i>x</i>
set no <i>x</i>	disable option <i>x</i>
set <i>x=val</i>	give value <i>val</i> to option <i>x</i>
set	show changed options
set all	show all options
set <i>x?</i>	show value of option <i>x</i>

Useful Options and Abbreviations	autoindent	ai	supply indent
	autowrite	aw	write before changing files
	directory		pathname of directory for temporary work files
	exrc	ex	allow vi/ex to read the .exrc in the current directory. This option is set in the EXINIT shell variable or in the .exrc file in the \$HOME directory.
	ignorecase	ic	ignore case of letters in scanning
	list		print ^I for tab, \$ at end
	magic		treat . [* special in patterns
	modelines		first five lines and last five lines executed as vi/ex commands if they are of the form ex:command: or vi:command:
	number	nu	number lines
	paragraphs	para	macro names that start paragraphs
	redraw		simulate smart terminal
	report		informs you if the number of lines modified by the last command is greater than the value of the report variable
	scroll		command mode lines
	sections	sect	macro names that start sections
	shiftwidth	sw	for <>, and input ^D
	showmatch	sm	to) and } as typed
	showmode	smd	show insert mode in vi
	slowopen	slow	stop updates during insert
	term		specifies to vi the type of terminal being used (the default is the value of the environment variable TERM)
	window		visual mode lines
wrapmargin	wm	automatic line splitting	
wrapscan	ws	search around end (or beginning) of buffer	
Scanning Pattern Formation	^		beginning of line
	\$		end of line
	.		any character

\<	beginning of word
\>	end of word
[<i>str</i>]	any character in <i>str</i>
[^ <i>str</i>]	any character not in <i>str</i>
[<i>xy</i>]	any character between <i>x</i> and <i>y</i>
*	any number of preceding characters

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of ex: HOME, LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, NLSPATH, PATH, SHELL, and TERM.

COLUMNS Override the system-selected horizontal screen size.

EXINIT Determine a list of ex commands that are executed on editor start-up, before reading the first file. The list can contain multiple commands by separating them using a vertical-line (|) character.

LINES Override the system-selected vertical screen size, used as the number of lines in a screenful and the vertical screen size in visual mode.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

文件

/var/tmp/Exnnnnnn	editor temporary
/var/tmp/Rxnnnnnn	named buffer temporary
/usr/lib/expreserve	preserve command
/usr/lib/exrecover	recover command
/usr/lib/exstrings	error messages
/usr/share/lib/terminfo/*	describes capabilities of terminals
/var/preserve/login	preservation directory (where login is the user's login)
\$HOME/.exrc	editor startup file
./ .exrc	editor startup file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/ex

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

/usr/xpg4/bin/ex

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

/usr/xpg6/bin/ex

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu6
CSI	Enabled
Interface Stability	Standard

另请参见

[ed\(1\)](#), [edit\(1\)](#), [grep\(1\)](#), [sed\(1\)](#), [sort\(1\)](#), [vi\(1\)](#), [curses\(3CURSES\)](#), [term\(4\)](#), [terminfo\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

《Solaris Advanced User's Guide》

Author

The `vi` and `ex` utilities are based on software developed by The University of California, Berkeley California, Computer Science Division, Department of Electrical Engineering and Computer Science.

附注

Several options, although they continue to be supported, have been replaced in the documentation by options that follow the Command Syntax Standard (see [Intro\(1\)](#)). The `-` option has been replaced by `-s`, a `-r` option that is not followed with an option-argument has been replaced by `-L`, and `+command` has been replaced by `-c command`.

The message file too large to recover with `-r` option, which is seen when a file is loaded, indicates that the file can be edited and saved successfully, but if the editing session is lost, recovery of the file with the `-r` option is not possible.

The `z` command prints the number of logical rather than physical lines. More than a screen full of output can result if long lines are present.

File input/output errors do not print a name if the command line `-s` option is used.

The editing environment defaults to certain configuration options. When an editing session is initiated, `ex` attempts to read the `EXINIT` environment variable. If it exists, the editor uses the values defined in `EXINIT`, otherwise the values set in `$HOME/.exrc` are used. If `$HOME/.exrc` does not exist, the default values are used.

To use a copy of `.exrc` located in the current directory other than `$HOME`, set the `exrc` option in `EXINIT` or `$HOME/.exrc`. Options set in `EXINIT` can be turned off in a local `.exrc` only if `exrc` is set in `EXINIT` or `$HOME/.exrc`. In order to be used, `.exrc` in `$HOME` or the current directory must fulfill these conditions:

- It must exist.
- It must be owned by the same userid as the real userid of the process, or the process has appropriate privileges.
- It is not writable by anyone other than the owner.

There is no easy way to do a single scan ignoring case.

The editor does not warn if text is placed in named buffers and not used before exiting the editor.

Null characters are discarded in input files and cannot appear in resultant files.

引用名 exec, eval, source – shell built-in functions to execute other commands

用法概要

sh exec [*argument*] ...
 eval [*argument*] ...

csh exec *command*
 eval *argument* ...
 source [-h] *name*

ksh88 *exec [*argument*] ...
 *eval [*argument*] ...

ksh +exec [-c] [-a *name*] [*command* [*argument* ...]]
 +eval [*argument*] ...

描述

sh The exec command specified by the arguments is executed in place of this shell without creating a new process. Input/output arguments and appear and, if no other arguments are specified, cause the shell input/output to be modified.

The *arguments* to the eval built-in are read as input to the shell and the resulting command(s) executed.

csh exec executes *command* in place of the current shell, which terminates.

eval reads its *arguments* as input to the shell and executes the resulting command(s). This is usually used to execute commands generated as the result of command or variable substitution.

source reads commands from *name*. source commands can be nested, but if they are nested too deeply the shell can run out of file descriptors. An error in a sourced file at any level terminates all nested source commands.

-h Place commands from the file *name* on the history list without executing them.

ksh88 With the exec built-in, if *arg* is specified, the command specified by the arguments is executed in place of this shell without creating a new process. Input/output arguments can appear and affect the current process. If no arguments are specified the effect of this command is to modify file descriptors as prescribed by the input/output redirection list. In this case, any file descriptor numbers greater than 2 that are opened with this mechanism are closed when invoking another program.

The arguments to eval are read as input to the shell and the resulting command(s) executed.

On this man page, [ksh88\(1\)](#) commands that are preceded by one or two * (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by ** that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.

ksh

exec is a special built-in command that can be used to manipulate file descriptors or to replace the current shell with a new command.

If *command* is specified, then the current shell process is replaced by *command* rather than running *command* and waiting for it to complete. There is no need to use exec to enhance performance since the shell implicitly uses the exec mechanism internally whenever possible.

If no operands are specified, exec can be used to open or close files, or to manipulate file descriptors from 0 to 9 in the current shell environment using the standard redirection mechanism available with all commands. The close-on-exec flag is set on file descriptor numbers greater than 2 that are opened this way so that they are closed when another program is invoked.

Because exec is a special command, any failure causes the script that invokes it to exit. This can be prevented by invoking exec from the command utility.

exec cannot be invoked from a restricted shell to create files or to open a file for writing or appending.

eval is a shell special built-in command that constructs a command by concatenating the *arguments* together, separating each with a space. The resulting string is taken as input to the shell and evaluated in the current environment. command words are expanded twice, once to construct *argument*, and again when the shell executes the constructed command. It is not an error if *argument* is not specified.

On this manual page, ksh commands that are preceded by one or two + symbols are special built-in commands and are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.

4. They are not valid function names.
5. Words following a command preceded by ++ that are in the format of a variable assignment are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and field splitting and file name generation are not performed.

选项

ksh

The following options are supported by ksh exec:

- a *name* *argv*[0] is set to *name* for command.
- c Clear all environment variables before executions except variable assignments that are part of the current exec command.

退出状态

ksh88

The following exit values are returned by exec:

- 0 Successful completion.
- 1 - 125 A redirection error occurred.
- 127 *command* was not found.
- 126 *command* was found, but it is not an executable utility.

ksh

The following exit values are returned by exec. If *command* is specified, exec does not return.

- 0 Successful completion. All I/O redirections were successful.
- >0 An error occurred.

The following exit values are returned by eval:

If *argument* is not specified, the exit status is 0. Otherwise, it is the exit status of the command defined by the *argument* operands.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#)

引用名 exit, return, goto – shell built-in functions to enable the execution of the shell to advance beyond its sequence of steps

用法概要

sh	exit [<i>n</i>] return [<i>n</i>]
csh	exit [(<i>expr</i>)] goto <i>label</i>
ksh88	*exit [<i>n</i>] *return [<i>n</i>]
ksh	+exit [<i>n</i>] +return [<i>n</i>]

描述

sh `exit` causes the calling shell or shell script to exit with the exit status specified by *n*. If *n* is omitted the exit status is that of the last command executed (an EOF also causes the shell to exit.)

`return` causes a function to exit with the return value specified by *n*. If *n* is omitted, the return status is that of the last command executed.

csh `exit` causes the calling shell or shell script to exit, either with the value of the status variable or with the value specified by the expression *expr*.

The `goto` built-in uses a specified *label* as a search string amongst commands. The shell rewinds its input as much as possible and searches for a line of the form *label*: possibly preceded by space or tab characters. Execution continues after the indicated line. It is an error to jump to a label that occurs between a `while` or `for` built-in command and its corresponding `end`.

ksh88 `exit` causes the calling shell or shell script to exit with the exit status specified by *n*. The value is the least significant 8 bits of the specified status. If *n* is omitted then the exit status is that of the last command executed. When `exit` occurs when executing a trap, the last command refers to the command that executed before the trap was invoked. An end-of-file also causes the shell to exit except for a shell which has the `ignoreeof` option (See `set` below) turned on.

`return` causes a shell function or a `'.'` script to return to the invoking script with the return status specified by *n*. The value is the least significant 8 bits of the specified status. If *n* is omitted then the return status is that of the last command executed. If `return` is invoked while not in a function or a `'.'` script, then it is the same as an `exit`.

On this man page, [ksh88\(1\)](#) commands that are preceded by one or two * (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by `**` that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the `=` sign and word splitting and file name generation are not performed.

ksh

`exit` is shell special built-in that causes the shell that invokes it to exit. Before exiting the shell, if the `EXIT` trap is set, it is invoked.

If n is specified, it is used to set the exit status.

`return` is a shell special built-in that causes the function or dot script that invokes it to exit. If `return` is invoked outside of a function or dot script it is equivalent to `exit`.

If `return` is invoked inside a function defined with the `function` reserved word syntax, then any `EXIT` trap set within the function is invoked in the context of the caller before the function returns.

If n is specified, it is used to set the exit status.

On this manual page, ksh commands that are preceded by one or two `+` symbols are special built-in commands and are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. They are not valid function names.
5. Words following a command preceded by `++` that are in the format of a variable assignment are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the `=` sign and field splitting and file name generation are not performed.

退出状态

ksh

If n is specified for `exit`, the exit status is the least significant eight bits of the value of n . Otherwise, the exit status is the exit status of preceding command. When invoked inside a trap, the preceding command means the command that invoked the trap.

If n is specified for `return`, the exit status is the least significant eight bits of the value of n . Otherwise, the exit status is the exit status of preceding command.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[break\(1\)](#), [csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#)

引用名	expand, unexpand – expand TAB characters to SPACE characters, and vice versa
用法概要	<pre>expand [-t <i>tablist</i>] [<i>file</i>]...</pre> <pre>expand [-<i>tabstop</i>] [-<i>tab1</i>, <i>tab2</i>, . . . , <i>tabn</i>] [<i>file</i>]...</pre> <pre>unexpand [-a] [-t <i>tablist</i>] [<i>file</i>]...</pre>
描述	<p>The expand utility copies <i>files</i> (or the standard input) to the standard output, with TAB characters expanded to SPACE characters. BACKSPACE characters are preserved into the output and decrement the column count for TAB calculations. expand is useful for pre-processing character files (before sorting, looking at specific columns, and so forth) that contain TAB characters.</p> <p>unexpand copies <i>files</i> (or the standard input) to the standard output, putting TAB characters back into the data. By default, only leading SPACE and TAB characters are converted to strings of tabs, but this can be overridden by the -a option (see the OPTIONS section below).</p>
选项	<p>The following options are supported for expand:</p> <p>-t <i>tablist</i> Specifies the tab stops. The argument <i>tablist</i> must consist of a single positive decimal integer or multiple positive decimal integers, separated by blank characters or commas, in ascending order. If a single number is given, tabs will be set <i>tablist</i> column positions apart instead of the default 8. If multiple numbers are given, the tabs will be set at those specific column positions.</p> <p style="padding-left: 100px;">Each tab-stop position <i>N</i> must be an integer value greater than zero, and the list must be in strictly ascending order. This is taken to mean that, from the start of a line of output, tabbing to position <i>N</i> causes the next character output to be in the (<i>N</i>+1)th column position on that line.</p> <p style="padding-left: 100px;">In the event of expand having to process a tab character at a position beyond the last of those specified in a multiple tab-stop list, the tab character is replaced by a single space character in the output.</p> <p>-<i>tabstop</i> Specifies as a single argument, sets TAB characters <i>tabstop</i> SPACE characters apart instead of the default 8.</p> <p>-<i>tab1</i>, <i>tab2</i>, ..., <i>tabn</i> Sets TAB characters at the columns specified by -<i>tab1</i>, <i>tab2</i>, ..., <i>tabn</i></p> <p>The following options are supported for unexpand:</p> <p>-a Inserts TAB characters when replacing a run of two or more SPACE characters would produce a smaller output file.</p> <p>-t <i>tablist</i> Specifies the tab stops. The option-argument <i>tablist</i> must be a single argument consisting of a single positive decimal integer or multiple positive decimal integers, separated by blank characters or commas, in ascending order. If a</p>

single number is given, tabs will be set *tablist* column positions apart instead of the default 8. If multiple numbers are given, the tabs will be set at those specific column positions. Each tab-stop position *N* must be an integer value greater than zero, and the list must be in strictly ascending order. This is taken to mean that, from the start of a line of output, tabbing to position *N* will cause the next character output to be in the (*N*+1)th column position on that line. When the `-t` option is not specified, the default is the equivalent of specifying `-t 8` (except for the interaction with `-a`, described below).

No space-to-tab character conversions occur for characters at positions beyond the last of those specified in a multiple tab-stop list.

When `-t` is specified, the presence or absence of the `-a` option is ignored; conversion will not be limited to the processing of leading blank characters.

操作数 The following operand is supported for `expand` and `unexpand`:

file The path name of a text file to be used as input.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `expand` and `unexpand`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态 The following exit values are returned:

0 Successful completion

>0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [tabs\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

- 引用名** exportfs – translates exportfs options to share/unshare commands
- 用法概要** /usr/sbin/exportfs [-aiuv] [-o *options*] [*pathname*]
- 描述** exportfs translates SunOS 4.x exportfs options to the corresponding share/unshare options and invokes share/unshare with the translated options.
- With no options or arguments, exportfs invokes share to print out the list of all currently shared NFS filesystems.
- exportfs is the BSD/Compatibility Package command of [share\(1M\)](#) and [unshare\(1M\)](#). Use [share\(1M\)](#)/[unshare\(1M\)](#) whenever possible.
- 选项**
- a Invokes [shareall\(1M\)](#), or if -u is specified, invokes [unshareall\(1M\)](#).
 - i Ignore options in /etc/dfs/dfstab.
 - u Invokes [unshare\(1M\)](#) on *pathname*.
 - v Verbose.
 - o *options* Specify a comma-separated list of optional characteristics for the filesystems being exported. exportfs translates *options* to share-equivalent options. (see [share\(1M\)](#) for information about individual options).
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/file-system/nfs

- 另请参见** [share\(1M\)](#), [shareall\(1M\)](#), [unshare\(1M\)](#), [unshareall\(1M\)](#), [attributes\(5\)](#)

引用名 *expr* – evaluate arguments as an expression

用法概要

```
/usr/bin/expr argument...
```

```
/usr/xpg4/bin/expr argument...
```

```
/usr/xpg6/bin/expr argument...
```

描述

/usr/bin/expr,
/usr/xpg4/bin/expr The *expr* utility evaluates the expression and writes the result to standard output. The character `0` is written to indicate a zero value and nothing is written to indicate a null string.

/usr/xpg6/bin/expr The *expr* utility evaluates the expression and writes the result to standard output followed by a NEWLINE. If there is no result from *expr* processing, a NEWLINE is written to standard output.

操作数 The *argument* operand is evaluated as an expression. Terms of the expression must be separated by blanks. Characters special to the shell must be escaped (see [sh\(1\)](#)). Strings containing blanks or other special characters should be quoted. The length of the expression is limited to `LINE_MAX` (2048 characters).

The operators and keywords are listed below. The list is in order of increasing precedence, with equal precedence operators grouped within `{ }` symbols. All of the operators are left-associative.

expr \| *expr* Returns the evaluation of the first *expr* if it is neither NULL nor `0`; otherwise, returns the evaluation of the second *expr* if it is not NULL; otherwise, `0`.

expr \& *expr* Returns the first *expr* if neither *expr* is NULL or `0`, otherwise returns `0`.

expr{ =, \>, \>=, \<, \<=, !=} *expr* Returns the result of an integer comparison if both arguments are integers, otherwise returns the result of a string comparison using the locale-specific coalition sequence. The result of each comparison is 1 if the specified relationship is TRUE, `0` if the relationship is FALSE.

expr { +, -} *expr* Addition or subtraction of integer-valued arguments.

expr { *, /, %} *expr* Multiplication, division, or remainder of the integer-valued arguments.

expr : *expr* The matching operator `:` (colon) compares the first argument with the second argument, which must be an internationalized basic regular expression (BRE), except that all patterns are anchored to the beginning of the string. That is, only sequences starting at the first

character of a string are matched by the regular expression. See [regex\(5\)](#) and NOTES. Normally, the `/usr/bin/expr` matching operator returns the number of bytes matched and the `/usr/xpg4/bin/expr` matching operator returns the number of characters matched (0 on failure). If the second argument contains at least one BRE sub-expression `[\(...\)]`, the matching operator returns the string corresponding to `\1`.

integer

An argument consisting only of an (optional) unary minus followed by digits.

string

A string argument that cannot be identified as an *integer* argument or as one of the expression operator symbols.

The following four operators: `index`, `length`, `match`, and `substr`, are all at the same precedence:

`index string character-list`

Report the first byte in *string* (counting from one) where a byte from *character-list* matches a byte from *string*. If no bytes in *character-list* appear in *string*, a 0 is returned.

`length string`

Return the length (that is, the number of bytes) of *string*. The terminating nul character is not included in that count.

`match string regular-expression`

Synonymous with the `expr : expr` matching operator.

`substr string integer-1 integer-2`

Extract the sequence of bytes from *string* (counting from one) starting at position *integer-1* and of length *integer-2* bytes. If *integer-1* has a value greater than the number of bytes in *string*, `expr` returns a null string. If you try to extract more bytes than there are in *string*, `expr` returns all the remaining bytes from *string*. Results are unspecified if either *integer-1* or *integer-2* is a negative value.

示例

示例 1 Adding an integer to a shell variable

Add 1 to the shell variable `a`:

```
example$ a='expr $a + 1'
```

示例 2 Returning a path name segment

The following example emulates [basename\(1\)](#), returning the last segment of the path name `$a`. For `$a` equal to either `/usr/abc/file` or just `file`, the example returns `file`. (Watch out for `/` alone as an argument: `expr` takes it as the division operator. See NOTES below.)

示例 2 Returning a path name segment (续)

```
example$ expr $a : '.*\/(.*\)' \| $a
```

示例 3 Using // characters to simplify the expression

Here is a better version of the previous example. The addition of the // characters eliminates any ambiguity about the division operator and simplifies the whole expression.

```
example$ expr // $a : '.*\/(.*\)'
```

/usr/bin/expr

示例 4 Returning the number of bytes in a variable

```
example$ expr "$VAR" : '.*'
```

/usr/xpg4/bin/expr

示例 5 Returning the number of characters in a variable

```
example$ expr "$VAR" : '.*'
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of expr: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

As a side effect of expression evaluation, expr returns the following exit values:

- 0 If the expression is neither NULL nor 0.
- 1 If the expression is either NULL or 0.
- 2 For invalid expressions.
- >2 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled. See Notes.
Interface Stability	See below.
Standard	See standards(5) .

The match, substr, length, and index operators are Uncommitted. Everything else is Committed.

另请参见

[basename\(1\)](#), [ed\(1\)](#), [sh\(1\)](#), [Intro\(3\)](#), [attributes\(5\)](#), [environ\(5\)](#), [regex\(5\)](#), [standards\(5\)](#)

诊断	<p>syntax error Operator and operand errors.</p> <p>non-numeric argument Arithmetic is attempted on such a string.</p>
附注	<p>The following three operators are not CSI enabled. They are also not available in /usr/xpg4/bin/expr and /usr/xpg6/bin/expr:</p> <p><i>index string character-list</i></p> <p><i>length string</i></p> <p><i>substr string integer-1 integer-2</i></p> <p>After argument processing by the shell, expr cannot tell the difference between an operator and an operand except by the value. If \$a is an =, the command:</p> <p>example\$ expr \$a = '='</p> <p>looks like:</p> <p>example\$ expr = = =</p> <p>as the arguments are passed to expr (and they are all taken as the = operator). The following works:</p> <p>example\$ expr X\$a = X=</p>
Regular Expressions	<p>Unlike some previous versions, expr uses Internationalized Basic Regular Expressions for all system-provided locales. Internationalized Regular Expressions are explained on the regex(5) manual page.</p>

引用名

expr – evaluate arguments as a logical, arithmetic, or string expression

用法概要

/usr/ucb/expr *argument*...

描述

The `expr` utility evaluates expressions as specified by its arguments. After evaluation, the result is written on the standard output. Each token of the expression is a separate argument, so terms of the expression must be separated by blanks. Characters special to the shell must be escaped. Note: `0` is returned to indicate a zero value, rather than the null string. Strings containing blanks or other special characters should be quoted. Integer-valued arguments may be preceded by a unary minus sign. Internally, integers are treated as 32-bit, two's-complement numbers.

The operators and keywords are listed below. Characters that need to be escaped are preceded by `\`. The list is in order of increasing precedence, with equal precedence operators grouped within `{ }` symbols.

`expr \| expr` Returns the evaluation of the first *expr* if it is neither NULL nor `0`; otherwise, returns the evaluation of the second *expr* if it is not NULL; otherwise, `0`.

`expr \& expr` Returns the first *expr* if neither *expr* is NULL or `0`, otherwise returns `0`.

`expr { =, \, \, \<, \<=, != } expr` Returns the result of an integer comparison if both arguments are integers, otherwise returns the result of a lexical comparison.

`expr { +, - } expr` Addition or subtraction of integer-valued arguments.

`expr { \, /, % } expr` Multiplication, division, or remainder of the integer-valued arguments.

`string : regular-expression`

`match string regular-expression`

The two forms of the matching operator above are synonymous. The matching operators `:` and `match` compare the first argument with the second argument which must be a regular expression. Regular expression syntax is the same as that of [regexp\(5\)](#), except that all patterns are “anchored” (treated as if they begin with `^`) and therefore `^` is not a special character, in that context. Normally, the matching operator returns the number of characters matched (`0` on failure). Alternatively, the `\ . . . \` pattern symbols can be used to return a portion of the first argument.

`substr string integer-1 integer-2` Extracts the substring of *string* starting at position *integer-1* and of length *integer-2* characters. If *integer-1* has a value greater than the length of *string*, `expr` returns

a null string. If you try to extract more characters than there are in *string*, *expr* returns all the remaining characters from *string*. Beware of using negative values for either *integer-1* or *integer-2* as *expr* tends to run forever in these cases.

<code>index string character-list</code>	Reports the first position in <i>string</i> at which any one of the characters in <i>character-list</i> matches a character in <i>string</i> .
<code>length string</code>	Returns the length (that is, the number of characters) of <i>string</i> .
<code>(expr)</code>	Parentheses may be used for grouping.

示例

示例 1 Adding an integer to a shell variable

Add 1 to the shell variable *a*.

```
a='expr $a + 1'
```

示例 2 Returning a path name segment

Return the last segment of a path name (that is, the filename part). Watch out for / alone as an argument: *expr* will take it as the division operator (see BUGS below).

```
# 'For $a equal to either "/usr/abc/file" or just "file"'  
expr $a : '.*\/ \ $a'
```

示例 3 Using // characters to simplify the expression

The addition of the // characters eliminates any ambiguity about the division operator and simplifies the whole expression.

```
# A better representation of example 2.  
expr // $a : '.*\/'
```

示例 4 Returning the value of a variable

Returns the number of characters in \$VAR.

```
expr $VAR : '.*'
```

退出状态

expr returns the following exit codes:

- 0 If the expression is neither NULL nor 0.
- 1 If the expression is NULL or 0.
- 2 For invalid expressions.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见

[sh\(1\)](#), [test\(1\)](#), [attributes\(5\)](#), [regex\(5\)](#)

诊断

syntax error	for operator/operand errors
non-numeric argument	if arithmetic is attempted on such a string
division by zero	if an attempt to divide by zero is made

已知问题

After argument processing by the shell, `expr` cannot tell the difference between an operator and an operand except by the value. If `$a` is an `=`, the command:

```
expr $a = '='
```

looks like:

```
expr = = =
```

as the arguments are passed to `expr` (and they will all be taken as the `=` operator). The following works:

```
expr X$a = X=
```

Note: the `match`, `substr`, `length`, and `index` operators cannot themselves be used as ordinary strings. That is, the expression:

```
example% expr index expurgatorious length
syntax error
example%
```

generates the 'syntax error' message as shown instead of the value 1 as you might expect.

引用名	exstr – extract strings from source files
用法概要	<pre>exstr filename... exstr -e filename... exstr -r [-d] filename...</pre>
描述	<p>The <code>exstr</code> utility is used to extract strings from C-language source files and replace them by calls to the message retrieval function (see gettext(3C)). This utility will extract all character strings surrounded by double quotes, not just strings used as arguments to the <code>printf</code> command or the <code>printf</code> routine. In the first form, <code>exstr</code> finds all strings in the source files and writes them on the standard output. Each string is preceded by the source file name and a colon (:).</p> <p>The first step is to use <code>exstr -e</code> to extract a list of strings and save it in a file. Next, examine this list and determine which strings can be translated and subsequently retrieved by the message retrieval function. Then, modify this file by deleting lines that can't be translated and, for lines that can be translated, by adding the message file names and the message numbers as the fourth (<i>msgfile</i>) and fifth (<i>msgnum</i>) entries on a line. The message files named must have been created by mkmsgs(1) and exist in <code>/usr/lib/locale/locale/LC_MESSAGES</code>. (The directory <code>locale</code> corresponds to the language in which the text strings are written; see setlocale(3C)). The message numbers used must correspond to the sequence numbers of strings in the message files.</p> <p>Now use this modified file as input to <code>exstr -r</code> to produce a new version of the original C-language source file in which the strings have been replaced by calls to the message retrieval function <code>gettext()</code>. The <i>msgfile</i> and <i>msgnum</i> fields are used to construct the first argument to <code>gettext()</code>. The second argument to <code>gettext()</code> is printed if the message retrieval fails at run-time. This argument is the null string, unless the <code>-d</code> option is used.</p> <p>This utility cannot replace strings in all instances. For example, a static initialized character string cannot be replaced by a function call. A second example is that a string could be in a form of an escape sequence which could not be translated. In order not to break existing code, the files created by invoking <code>exstr -e</code> must be examined and lines containing strings not replaceable by function calls must be deleted. In some cases the code may require modifications so that strings can be extracted and replaced by calls to the message retrieval function.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -e Extract a list of strings from the named C-language source files, with positional information. This list is produced on standard output in the following format: <pre>file:line:position:msgfile:msgnum:string</pre> <p>where</p> <pre>file the name of a C-language source file</pre>

<i>line</i>	line number in the file
<i>position</i>	character position in the line
<i>msgfile</i>	null
<i>msgnum</i>	null
<i>string</i>	the extracted text string

Normally you would redirect this output into a file. Then you would edit this file to add the values you want to use for *msgfile* and *msgnum*:

msgfile the file that contains the text strings that will replace *string*. A file with this name must be created and installed in the appropriate place by the [mkmsgs\(1\)](#) utility.

msgnum the sequence number of the string in *msgfile*.

The next step is to use `exstr -r` to replace *strings* in *file*.

- r Replace strings in a C-language source file with function calls to the message retrieval function `gettext()`.
- d This option is used together with the `-r` option. If the message retrieval fails when `gettext()` is invoked at run-time, then the extracted string is printed. You would use the capability provided by `exstr` on an application program that needs to run in an international environment and have messages print in more than one language. `exstr` replaces text strings with function calls that point at strings in a message data base. The data base used depends on the run-time value of the `LC_MESSAGES` environment variable (see [environ\(5\)](#)).

示例

示例 1 The following examples show uses of `exstr`

Assume that the file `example.c` contains two strings:

```
main()
{
    printf("This is an example\n");

    printf("Hello world!\n");
}
```

The `exstr` utility, invoked with the argument `example.c` extracts strings from the named file and prints them on the standard output.

```
example% exstr example.c
```

示例 1 The following examples show uses of exstr (续)

produces the following output:

```
example.c:This is an example\n
example.c:Hello world!\n
```

The `exstr` utility, invoked with the `-e` option and the argument `example.c`, and redirecting output to the file `example.stringsout`

```
example% exstr -e example.c > example.stringsout
```

produces the following output in the file `example.stringsout`

```
example.c:3:8::This is an example\n
example.c:4:8::Hello world!\n
```

You must edit `example.stringsout` to add the values you want to use for the `msgfile` and `msgnum` fields before these strings can be replaced by calls to the retrieval function. If `UX` is the name of the message file, and the numbers 1 and 2 represent the sequence number of the strings in the file, here is what `example.stringsout` looks like after you add this information:

```
example.c:3:8:UX:1:This is an example\n
example.c:4:8:UX:2:Hello world!\n
```

The `exstr` utility can now be invoked with the `-r` option to replace the strings in the source file by calls to the message retrieval function `gettext()`.

```
example% exstr -r example.c <example.stringsout >intlexample.c
```

produces the following output:

```
extern char *gettext();

main()

{
    printf(gettxt("UX:1", ""));

    printf(gettxt("UX:2", ""));
}

```

The following example:

```
example% exstr -rd example.c <example.stringsout >intlexample.c
```

uses the extracted strings as a second argument to `gettext()`:

示例 1 The following examples show uses of `exstr` (续)

```
extern char *gettext();

main()
{
    printf(gettext("UX:1", "This is an example\n"));
    printf(gettext("UX:2", "Hello world!\n"));
}
```

文件 `/usr/lib/locale/locale/LC_MESSAGES/*` files created by `mkmsgs(1)`

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale

另请参见 [gettext\(1\)](#), [mkmsgs\(1\)](#), [printf\(1\)](#), [srchtxt\(1\)](#), [gettext\(3C\)](#), [printf\(3C\)](#), [setlocale\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#)

诊断 The error messages produced by `exstr` are intended to be self-explanatory. They indicate errors in the command line or format errors encountered within the input file.

引用名	factor – obtain the prime factors of a number
用法概要	factor [<i>integer</i>]
描述	<p>factor writes to standard input all prime factors for any positive integer less than or equal to 10^{14}. The prime factors are written the proper number of times.</p> <p>If factor is used <i>without</i> an argument, it waits for an integer to be entered. After entry of the integer, it factors it, writes its prime factors the proper number of times, and then waits for another integer. factor exits if a 0 or any non-numeric character is entered.</p> <p>If factor is invoked <i>with</i> an argument (<i>integer</i>), it writes the integer, factors it and writes all the prime factors as described above, and then exits. If the argument is 0 or non-numeric, factor writes a 0 and then exits.</p> <p>The maximum time to factor an integer is proportional to \sqrt{n}, where n is the integer which is entered. factor will take this time when n is prime or the square of a prime.</p>
操作数	<i>integer</i> Any positive integer less than or equal to 10^{14} .
退出状态	<p>0 Successful completion.</p> <p>1 An error occurred.</p>
诊断	factor prints the error message Ouch! for input out of range or for garbage input.
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTE VALUE
Availability	system/extended-system-utilities

另请参见 [attributes\(5\)](#)

引用名 fastboot, fasthalt – reboot/halt the system without checking the disks

用法概要 /usr/ucb/fastboot [*boot-options*]

/usr/ucb/fasthalt [*halt-options*]

描述 fastboot and fasthalt are shell scripts that invoke reboot and halt with the proper arguments.

These commands are provided for compatibility only.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [fsck\(1M\)](#), [halt\(1M\)](#), [init\(1M\)](#), [reboot\(1M\)](#), [init.d\(4\)](#), [attributes\(5\)](#)

引用名	fgrep – search a file for a fixed-character string
用法概要	<pre> /usr/bin/fgrep [-bchilnsvx] -e <i>pattern_list</i> [<i>file...</i>] /usr/bin/fgrep [-bchilnsvx] -f <i>file</i> [<i>file...</i>] /usr/bin/fgrep [-bchilnsvx] <i>pattern</i> [<i>file...</i>] /usr/xpg4/bin/fgrep [-bchilnqsvx] -e <i>pattern_list</i> [-f <i>file</i>] [<i>file...</i>] /usr/xpg4/bin/fgrep [-bchilnqsvx] [-e <i>pattern_list</i>] -f <i>file</i> [<i>file...</i>] /usr/xpg4/bin/fgrep [-bchilnqsvx] <i>pattern</i> [<i>file...</i>] </pre>
描述	<p>The <code>fgrep</code> (fixed grep) utility searches files for a character string and prints all lines that contain that string. <code>fgrep</code> is different from <code>grep(1)</code> and from <code>egrep(1)</code> because it searches for a string, instead of searching for a pattern that matches an expression.</p> <p>The characters <code>\$</code>, <code>*</code>, <code>[</code>, <code>^</code>, <code> </code>, <code>(</code>, <code>)</code>, and <code>\</code> are interpreted literally by <code>fgrep</code>, that is, <code>fgrep</code> does not recognize full regular expressions as does <code>egrep</code>. These characters have special meaning to the shell. Therefore, to be safe, enclose the entire <i>string</i> within single quotes (<code>'</code>).</p> <p>If no files are specified, <code>fgrep</code> assumes standard input. Normally, each line that is found is copied to the standard output. The file name is printed before each line that is found if there is more than one input file.</p>
选项	<p>The following options are supported for both <code>/usr/bin/fgrep</code> and <code>/usr/xpg4/bin/fgrep</code>:</p> <ul style="list-style-type: none"> -b Precedes each line by the block number on which the line was found. This can be useful in locating block numbers by context. The first block is 0. -c Prints only a count of the lines that contain the pattern. -e <i>pattern_list</i> Searches for a <i>string</i> in <i>pattern-list</i>. This is useful when the <i>string</i> begins with a <code>-</code>. -f <i>pattern-file</i> Takes the list of patterns from <i>pattern-file</i>. -h Suppresses printing of files when searching multiple files. -i Ignores upper/lower case distinction during comparisons. -l Prints the names of files with matching lines once, separated by new-lines. Does not repeat the names of files when the pattern is found more than once. -n Precedes each line by its line number in the file. The first line is 1. -s Works silently, that is, displays nothing except error messages. This is useful for checking the error status. -v Prints all lines except those that contain the pattern.

-x Prints only lines that are matched entirely.

/usr/xpg4/bin/fgrep The following options are supported for /usr/xpg4/bin/fgrep only:

-q Quiet. Does not write anything to the standard output, regardless of matching lines. Exits with zero status if an input line is selected.

操作数 The following operands are supported:

file Specifies a path name of a file to be searched for the patterns. If no *file* operands are specified, the standard input will be used.

/usr/bin/fgrep *pattern* Specifies a pattern to be used during the search for input.

/usr/xpg4/bin/fgrep *pattern* Specifies one or more patterns to be used during the search for input. This operand is treated as if it were specified as *-e pattern_list*.

用法 See [largefile\(5\)](#) for the description of the behavior of fgrep when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of fgrep: LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态 The following exit values are returned:

- 0 If any matches are found
- 1 If no matches are found
- 2 For syntax errors or inaccessible files, even if matches were found.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/fgrep

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

/usr/xpg4/bin/fgrep

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled

另请参见 [ed\(1\)](#), [egrep\(1\)](#), [grep\(1\)](#), [sed\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [XPG4\(5\)](#)

附注 Ideally, there should be only one grep command, but there is not a single algorithm that spans a wide enough range of space-time tradeoffs.

Lines are limited only by the size of the available virtual memory.

`/usr/xpg4/bin/fgrep` The `/usr/xpg4/bin/fgrep` utility is identical to `/usr/xpg4/bin/grep -F` (see [grep\(1\)](#)). Portable applications should use `/usr/xpg4/bin/grep -F`.

引用名 file – determine file type

用法概要

```
/usr/bin/file [-dh] [-m mfile] [-M Mfile] [-f ffile] file...
/usr/bin/file [-dh] [-m mfile] [-M Mfile] -f ffile
/usr/bin/file -i [-h] [-f ffile] file...
/usr/bin/file -i [-h] -f ffile
/usr/bin/file -c [-d] [-m mfile] [-M Mfile]
/usr/xpg4/bin/file [-dh] [-m mfile] [-M Mfile] [-f ffile] file...
/usr/xpg4/bin/file [-dh] [-m mfile] [-M Mfile] -f ffile
/usr/xpg4/bin/file -i [-h] [-f ffile] file...
/usr/xpg4/bin/file -i [-h] -f ffile
/usr/xpg4/bin/file -c [-d] [-m mfile] [-M Mfile]
```

描述

The `file` utility performs a series of tests on each file supplied by *file* and, optionally, on each file listed in *ffile* in an attempt to classify it. If the file is not a regular file, its file type is identified. The file types directory, FIFO, block special, and character special are identified as such. If the file is a regular file and the file is zero-length, it is identified as an empty file.

If *file* appears to be a text file, `file` examines the first 512 bytes and tries to determine its programming language. If *file* is a symbolic link, by default the link is followed and `file` tests the file to which the symbolic link refers.

If *file* is a relocatable object, executable, or shared object, `file` prints out information about the file's execution requirements. This information includes the machine class, byte-ordering, static/dynamic linkage, and any software or hardware capability requirements. If *file* is a runtime linking configuration file, `file` prints information about the target platform, including the machine class and byte-ordering.

By default, `file` will try to use the localized magic file `/usr/lib/locale/locale/LC_MESSAGES/magic`, if it exists, to identify files that have a magic number. For example, in the Japanese locale, `file` will try to use `/usr/lib/locale/ja/LC_MESSAGES/magic`. If a localized magic file does not exist, `file` will utilize `/etc/magic`. A magic number is a numeric or string constant that indicates the file type. See [magic\(4\)](#) for an explanation of the format of `/etc/magic`.

If *file* does not exist, cannot be read, or its file status could not be determined, it is not considered an error that affects the exit status. The output will indicate that the file was processed, but that its type could not be determined.

选项

The following options are supported:

-c Checks the magic file for format errors. For reasons of efficiency, this validation is normally not carried out.

- d Applies any position-sensitive and context-sensitive default system tests to the file.
- f *ffile* *ffile* contains a list of the files to be examined.
- h When a symbolic link is encountered, this option identifies the file as a symbolic link. If -h is not specified and *file* is a symbolic link that refers to a non-existent file, the `file` utility identifies the file as a symbolic link, as if -h had been specified.
- i If a file is a regular file, this option does not attempt to classify the type of file further, but identifies the file as a “regular file”.
- m *mfile*
 - `/usr/bin/file` Uses *mfile* as an alternate magic file, instead of `/etc/magic`.
 - `/usr/xpg4/bin/file` Specifies the name of a file containing position-sensitive tests that are applied to a file in order to classify it (see [magic\(4\)](#)). If the -m option is specified without specifying the -d option or the -M option, position-sensitive default system tests are applied after the position-sensitive tests specified by the -m option.
- M *Mfile* Specifies the name of a file containing position-sensitive tests that are applied to a file in order to classify it (see [magic\(4\)](#)). No position-sensitive default system tests nor context-sensitive default system tests are applied unless the -d option is also specified.

If the -M option is specified with the -d option, the -m option, or both, or if the -m option is specified with the -d option, the concatenation of the position-sensitive tests specified by these options is applied in the order specified by the appearance of these options.

操作数

The following operands are supported:

file A path name of a file to be tested.

用法

See [largefile\(5\)](#) for the description of the behavior of `file` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Determining if an Argument is a Binary Executable Files

The following example determine if an argument is a binary executable file:

```
file "$1" | grep -Fq executable &&
printf "%s is executable.\n" "$1"
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `file`: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

文件

/etc/magic file's magic number file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[crle\(1\)](#), [elfdump\(1\)](#), [elffile\(1\)](#), [ls\(1\)](#), [magic\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注

The `file` utility cannot examine archive members unless they are first extracted from the archive into a separate file. The `elffile` utility can examine archive members in place, and is recommended for use with ELF objects and archives.

引用名	file – determine the type of a file by examining its contents
用法概要	<code>/usr/ucb/file [-f <i>ffile</i>] [-cL] [-m <i>mfile</i>] <i>filename</i>...</code>
描述	<p>file performs a series of tests on each <i>filename</i> in an attempt to determine what it contains. If the contents of a file appear to be ASCII text, file examines the first 512 bytes and tries to guess its language.</p> <p>file uses the file <code>/etc/magic</code> to identify files that have some sort of <i>magic number</i>, that is, any file containing a numeric or string constant that indicates its type.</p>
选项	<p>-c Check for format errors in the magic number file. For reasons of efficiency, this validation is not normally carried out. No file type-checking is done under -c.</p> <p>-f <i>ffile</i> Get a list of filenames to identify from <i>ffile</i>.</p> <p>-L If a file is a symbolic link, test the file the link references rather than the link itself.</p> <p>-m <i>mfile</i> Use <i>mfile</i> as the name of an alternate magic number file.</p>
示例	<p>示例 1 Using file on all the files in a specific user's directory.</p> <p>This example illustrates the use of file on all the files in a specific user's directory:</p> <pre>example% pwd /usr/blort/misc example% /usr/ucb/file *</pre> <pre>code: mc68020 demand paged executable code.c: c program text counts: ascii text doc: roff,nroff, or eqn input text empty.file: empty libz: archive random library memos: directory project: symboliclink to /usr/project script: executable shell script titles: ascii text s5.stuff: cpio archive</pre> <p>example%</p>
环境变量	<p>The environment variables <code>LC_CTYPE</code>, <code>LANG</code>, and <code>LC_default</code> control the character classification throughout file. On entry to file, these environment variables are checked in the following order: <code>LC_CTYPE</code>, <code>LANG</code>, and <code>LC_default</code>. When a valid value is found, remaining environment variables for character classification are ignored. For example, a new setting for <code>LANG</code> does not override the current valid character classification rules of <code>LC_CTYPE</code>. When none of the values is valid, the shell character classification defaults to the POSIX.1 “C” locale.</p>

文件 /etc/magic

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [magic\(4\)](#), [attributes\(5\)](#)

已知问题 `file` often makes mistakes. In particular, it often suggests that command files are C programs. `file` does not recognize Pascal or LISP.

引用名	filebench – 用于测量和比较文件系统性能的工作负荷的框架
用法概要	<pre>/usr/benchmarks/filebench/bin/filebench profile /usr/benchmarks/filebench/bin/filebench -c stats_dir...</pre>
描述	filebench 运行测量和比较文件系统性能的工作负荷。 完整的文档可以在性能社区中找到，网址为： http://hub.opensolaris.org/bin/view/Main/
选项	支持以下选项： -c <i>stats_dir...</i> 生成一个 HTML 文件 (<i>index.html</i>)，该文件是指定目录的比较文件。 <i>stats_dir</i> 指定在其中存储结果的一个或多个目录。
操作数	支持下列操作数： <i>profile</i> 指定配置文件的名称，并以 <i>.prof</i> 结尾。配置文件指定了以下内容： <ul style="list-style-type: none"> ▪ 运行的工作负荷， ▪ 运行的参数， ▪ 进行操作的目录路径，以及 ▪ 存储结果的目录路径。
退出状态	将返回以下退出值： <ol style="list-style-type: none"> 0 成功完成。 1 出现错误。 2 指定的命令行选项无效。
示例	<p>示例1 运行多流顺序读取工作负荷</p> <p>以下示例运行名为 <i>sqread.prof</i> 的配置文件中所描述的工作负荷：</p> <pre># filebench sqread</pre> <p>示例2 比较多次运行</p> <p>以下示例比较先前两次运行的结果。</p> <p>本示例假定先前两次运行 <i>filebench</i> 的结果位于以下目录中：<i>/stats/wombat-zfs-noel-Jun_27_2007-15h_45m_33s</i> 和 <i>/stats/wombat-ufs-noel-Jun_27_2007-15h_52m_11s</i>。</p> <p>本示例在当前工作目录中生成名为 <i>index.html</i> 的 HTML 文件。</p> <pre># filebench -c /stats/wombat-zfs-noel-Jun_27_2007-15h_45m_33s \ /stats/wombat-ufs-noel-Jun_27_2007-15h_52m_11s</pre>

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	benchmark/filebench
接口稳定性	Uncommitted（未确定）

另请参见 [attributes\(5\)](#)

引用名	filesync – 用于同步普通文件、目录或特殊文件
用法概要	<pre>filesync [-aehmnqvy] [-o src dst] [-f src dst old new] [-r directory]... filesync [-aehmnqvy] -s source-dir -d dest-dir filename...</pre>
描述	<p>filesync 实用程序在多个计算机系统之间（通常是服务器与便携式计算机之间）同步文件。filesync 可同步普通文件、目录或特殊文件。尽管是为在游牧式系统上使用而设计的，但 filesync 也非常适用于在数量更多的永久连接系统上执行备份和文件复制。</p> <p>如果在两个系统之间同步了文件，则每个系统上的对应文件是完全相同的。在一个或两个系统上更改某个文件将导致这些文件变得不一致（不同步）。为重新使文件完全相同，必须对文件之间的差异进行协调。有关 filesync 如何协调和同步文件的详细信息，请参见协调和同步文件。</p> <p>有两种形式的 filesync 命令。第一种形式的 filesync 在调用时不使用文件参数。这种形式的 filesync 协调在 \$HOME/.packingrules 文件中指定的文件和系统之间的差异。\$HOME/.packingrules 是 filesync 的打包规则列表，其中包含了要保持同步的文件列表。请参见 packingrules(4)。</p> <p>第二种形式的 filesync 将特定文件从源系统上的目录复制到目标系统上的目录。此外，这种形式的 filesync 会将指定为参数 (<i>filename</i>) 的一个或多个文件添加到 \$HOME/.packingrules。有关指定源系统和目标系统上的目录的信息，请参见 <code>-s</code> 和 <code>-d</code>。有关指定文件 (<i>filename</i>) 参数的详细信息，请参见“操作数”部分。</p> <p>多个 filesync 命令是累积的（也就是说，指定的文件将添加到已经存在的打包规则文件列表）。请参见多个 filesync 命令。</p>
协调和同步文件	<p>filesync 通过执行以下两个任务在计算机系统之间同步文件：</p> <ol style="list-style-type: none"> filesync 在两个系统上检查打包规则文件中指定的目录和文件，然后确定其是否完全相同。存在不同之处任何文件都需要进行协调。 filesync 还在 \$HOME/.filesync-base 文件中为所有被监视的文件维护一个基准线摘要。该文件将列出截至上次协调时所有文件的名称、类型和大小。 根据基准线文件中包含的信息和指定选项（请参见解决 filesync 冲突），filesync 确定各个副本中哪一个是正确的，并对其他系统做出相应更改。在此任务完成后，两个副本将重新成为完全相同的（同步的）。 如果源文件已更改，而目标文件未更改，则源系统上的更改将传播至目标系统。如果目标文件已更改，而对应的源文件未更改，则目标文件上的更改将传播至源系统。如果两个系统都发生了更改（而且文件还不是完全相同的），则会显示一条警告消息，要求用户手动解决冲突。请参见解决 filesync 冲突。

解决 filesync 冲突

如果两端的文件都发生了更改，filesync 将尝试确定应选择哪个版本。如果 filesync 无法自动确定应选择的版本，它将显示一条警告消息，并且将文件的两个冲突版本保留为不协调状态。

在此类情况下，必须手动解决差异，或告知 filesync 如何选择正确的版本。请使用 `-o` 和 `-f` 选项告知 filesync 如何解决冲突（请参见“选项”部分）。

另外，对于每个冲突的文件，您可以检查两个版本，确定应保留哪个，并手动使两个版本一致（通过复制、删除或更改所有权或保护以使之正确）。然后，可以重新运行 filesync 以查看是否还存在任何其他冲突。

打包规则文件

打包规则文件 `$HOME/.packingrules` 包含要保持同步的文件的列表。[packingrules\(4\)](#) 中描述了该文件的语法。

如果用户使用 `filename` 参数调用 filesync，则会自动创建 `$HOME/.packingrules` 文件。通过使用 filesync 选项，用户可以在 `$HOME/.packingrules` 中增加打包规则。

许多用户选择手动创建打包规则文件并手动进行编辑。用户可以编辑 `$HOME/.packingrules`（使用任何编辑器）以永久更改 `$HOME/.packingrules` 文件，或利用无法从命令行使用的更强大的选项（如 `IGNORE` 命令）。可以通过编辑 `$HOME/.packingrules` 来更方便地输入复杂的通配符表达式。

基准线文件

`$HOME/.filesync-base` 是 filesync 基准线摘要文件。在协调和同步过程中，filesync 使用 `$HOME/.filesync-base` 中的信息来识别文件之间的差异。用户不能创建或编辑基准线文件。它是由 filesync 自动创建的，并记录正在维护的所有文件之间上一次的已知一致状态。

多个 filesync 命令

经过一段时间后，您希望保持同步的文件集可能发生了更改。例如，在您的笔记本上，您通常希望文件仅与几个活动项目保持相关。如果继续使与您曾处理过的每个项目相关的文件保持同步，笔记本的磁盘将会充满旧文件。每个 filesync 命令将浪费的大量时间来更新您不再关注的文件。

如果从笔记本删除这些文件，filesync 将希望在服务器上执行对应的删除，而这可能不是您想要的结果。相反，您可能希望通过一种方法告知 filesync 停止同步某些文件。有两种方法可用于实现此目的：

1. 编辑 `$HOME/.packingrules`。删除用于您希望删除的文件的规则。
2. 删除 `$HOME/.packingrules`。使用 filesync 命令指定希望同步的文件。

两种方法都可以，您可以选择一种自己容易上手的方法。对于微小的更改，只是编辑 `$HOME/.packingrules` 可能会更为简单。对于大的更改，从头开始可能更为简单。

如果 filesync 不再同步某个文件集，您可以将这些文件从笔记本中删除，而不会影响到服务器。

游牧式计算机

当使用 `filesync` 在游牧式计算机与服务器之间保持文件同步时，请将打包规则和基准线文件存储在游牧式计算机上而不是存储在服务器上。登录到笔记本时，如果 `HOME` 环境变量没有像通常情况下那样指向笔记本上的目录，可使用 `FILESYNC` 环境变量为打包规则和基准线文件指定替代位置。

每台游牧式计算机都应带有其自己的打包规则和基线文件。如果服务器具有基准线文件并且多个游牧式计算机尝试根据该服务器的基准线文件进行协调，则可能会导致文件同步不正确。在此情况下，游牧式计算机使用的可能是未准确描述其文件的状态的基准线文件。这可能会导致不正确的协调。

为了防止与单个基准线文件被两台以上计算机共享相关的危险，`filesync` 向每个新的打包规则文件添加了一个缺省规则。该缺省规则阻止打包规则和基准线文件被复制。

选项

支持以下选项：

-a

针对所有新的和已更改的文件，强制检查访问控制列表 (Access Control List, ACL) 并尝试使其一致。如果不可以为某个特定文件设置 ACL，`filesync` 将停止对该文件进行 ACL 同步。

某些文件系统不支持 ACL。不能在支持 ACL 的文件系统与不支持 ACL 的文件系统之间同步 ACL，尝试同步将会生成许多错误消息。

-d *dest-dir*

指定目标系统上要将 *filename* 复制到其中的目录。与 `-s source-dir` 选项和 *filename* 操作数一起使用。请参见 `-s` 和“操作数”部分。

-e

标记出所有差异。涉及模式和所有权的冲突可能无法全部解决（除非是以 `root` 用户的特权运行 `filesync`）。如果您无法更改文件上的所有权或保护，`filesync` 通常会忽略所有权和保护方面的冲突。不过，如果您指定了 `-e`（所有内容必须一致）标志，则 `filesync` 将标记出这些差异。

-f *src* | *dst* | *old* | *new*

`-f` 选项告知 `filesync` 如何解决冲突的更改。如果文件在两个系统上均发生了更改，并且指定了 `-f` 选项，则 `filesync` 将保留首选系统上所做的更改，并丢弃非首选系统上所做的更改。

指定 `-f src` 可使源系统文件成为首选的。指定 `-f dst` 可使目标系统文件成为首选的。指定 `-f old` 可使文件的较旧版本成为首选的。指定 `-f new` 可使文件的较新版本成为首选的。

可以组合指定 `-f` 和 `-o` 选项，前提是它们指定了相同的首选项（`src` 和 `dst`）。如果 `-f` 和 `-o` 冲突，则会忽略 `-f` 选项。请参见 `-o` 选项说明。

-h

出现错误时停止。通常，如果 `filesync` 在复制文件时遇到读取或写入错误，它将记录该错误，程序将继续尝试协调其他文件。如果指定了 `-h` 选项，`filesync` 将在出现这些错误之一时立即停止，并且不会再尝试处理更多的文件。

-m

确保文件的两个副本都具有相同的修改时间。缺省情况下，新复制的文件的修改时间将设置为进行协调时的时间。文件更改按递增的修改时间排序，以便传播的文件具有与原始更改相同的相对修改时间排序。用户应注意，任何两个系统之间通常都会存在一些时间偏差，从一个系统向另一个系统传输修改时间偶尔可能会产生奇怪的结果。

例如，在某些情况下，使用 `filesync` 更新目录中的一些（并不是所有）文件将使 `make` 程序产生混乱。例如，如果 `filesync` 保持 `.c` 文件同步，但忽略 `.o` 文件，则更改的 `.c` 文件显示的修改时间可能会早于从之前版本的 `.c` 文件构建的 `.o` 文件。

-n

不实际执行更改。如果指定了 `-n` 选项，`filesync` 将确定已对文件做了哪些更改，以及需要进行哪些协调并在标准输出上显示该信息。不对文件进行任何更改，包括打包规则文件。

同时指定 `-n` 和 `-o` 选项将会使 `filesync` 分析占主导地位的系统并报告在该系统上已做出的更改。在计算机断开了连接（无法访问服务器），但您希望知道本地计算机上做出了哪些更改时，组合使用 `-n` 和 `-o` 非常有用。请参见 `-o` 选项说明。

-o src | dst

`-o` 选项将强制执行单向协调，无论首选系统是源系统 (`src`) 还是目标系统 (`dst`)。

指定 `-o src` 可仅将更改从源系统传播至目标系统。目标系统上所做的更改会被忽略。如果 `filesync` 无法访问源目录或目标目录，则会异常中止。

指定 `-o dst` 可仅将更改从目标系统传播至源系统。源系统上所做的更改会被忽略。如果 `filesync` 无法访问源目录或目标目录，则会异常中止。

组合指定 `-n` 和 `-o` 选项将会使 `filesync` 分析占主导地位的系统并报告在该系统上已做出的更改。在计算机断开了连接（无法访问服务器），但您希望知道本地计算机上做出了哪些更改时，组合使用 `-n` 和 `-o` 非常有用。请参见 `-n` 选项说明。

可以组合指定 `-o` 和 `-f` 选项，前提是它们指定了相同的首选项 (`src` 和 `dst`)。如果 `-o` 和 `-f` 选项冲突，则会忽略 `-f` 选项。请参见 `-f` 选项说明。

-q

抑制对各个协调操作的执行情况进行描述的标准 `filesync` 消息。

标准 `filesync` 消息以 UNIX shell 命令形式描述各个协调操作（例如，`mv`、`ln`、`cp`、`rm`、`chmod`、`chown`、`chgrp`、`setfacl`，等等）。

-r *directory*

将协调限制到 *directory*。通过使用多个 `-r` 可指定多个目录。

-s *source-dir*

指定源系统上要复制的 *filename* 所在的目录。与 `-d dest-dir` 选项和 *filename* 操作数一起使用。请参见 `-d` 选项说明和“操作数”部分。

-v

在标准输出中显示有关各个文件比较的执行情况的更多信息。

-y

绕过安全检查提示。游牧式计算机偶尔会在两个域之间移动，并且 `filesync` 所操作的许多文件期望可以通过 NFS 进行访问。这存在一定的风险，某天 `filesync` 可能会被要求根据错误的文件系统或服务器来协调本地更改。这可能会导致大量不恰当的复制和删除。为了防止出现这样的意外，`filesync` 会在协调之前执行一些安全检查。如果可能要删除大量文件，或高等级目录已更改其 I 节点数量，`filesync` 在进行协调之前会提示进行确认。如果您知道此可能性，并且不希望看到提示，可使用 `-y (yes)` 选项自动确认这些提示。

操作数

支持下列操作数：

filename 指定源目录 (*source-dir*) 中要同步的普通文件、目录、符号链接或特殊文件的名称。可通过使用空格分隔各个文件名来指定多个文件。请将 *filename* 操作数与 `-s` 和 `-d` 选项一起使用。请参见“选项”部分。

如果 *filename* 是一个普通文件，则会将该普通文件（以相同的 *filename*）复制到指定的目标目录 (*dest-dir*) 中。

如果 *filename* 是一个目录，则会将该目录和其下的所有文件和子目录（以递归方式）复制到指定的目标目录 (*dest-dir*) 中。

如果 *filename* 是一个符号链接，则会将该符号链接的副本复制到指定的目标目录 (*dest-dir*) 中。

如果 *filename* 是一个特殊文件，则会将一个具有相同主设备号或从设备号的特殊文件复制到指定的目标目录中。(*dest-dir*)。只有超级用户才可以使用 `filesync` 创建特殊文件。

在目标目录 (*dest-dir*) 中创建的文件将具有与源目录中的文件相同的所有者、组以及其他权限。

如果 *filename* 包含转义的 shell 通配符，则这些通配符存储在 `$HOME/.packingrules` 中并在每次运行 `filesync` 时进行评估。

例如，以下示例将确保两个指定的文件（当前在 `$RHOME` 中）被复制到 `$HOME` 中：

```
filesync -s $RHOME -d $HOME a.c b.c
```

以下示例将确保 `$RHOME` 中的所有 `*.c` 文件被复制到 `$HOME` 中，即使这些文件稍后才会创建。

```
filesync -s $RHOME -d $HOME '*.c'
```

如果任一目标文件已经存在，`filesync` 将确保它们是完全相同的，如果不是，它将发出警告。

在复制了文件之后，源文件和目标文件之间的区分将是相对任意的（除非是在 `-o` 和 `-f` 开关中使用）。

环境变量

- FILESYNC** 指定 `filesync` 打包规则和基准线文件的缺省位置。该变量的缺省值是 `$HOME`。将通过附加 `.packingrules` 和 `.filesync-base` 后缀来形成打包规则和基准线文件的名称。
- LC_MESSAGES** 确定如何显示诊断和信息性消息。在 C 语言环境中，消息以程序自身中的缺省形式显示（大多数情况下，为美式英文）。

退出状态

通常，如果所有文件已经是最新的，或所有文件已成功进行了协调，则 `filesync` 将以状态 `0` 退出。然而，如果指定了 `-n` 选项或出现了任何错误，退出状态将是以下项的逻辑“或”：

- `0` 无冲突，所有文件都是最新的。
- `1` 有一些可解决的冲突。
- `2` 有一些需要手动解决的冲突。
- `4` 某些指定的文件不存在。
- `8` 针对某些文件的权限不足。
- `16` 访问打包规则或基准线文件时出错。
- `32` 无效参数。
- `64` 无法访问指定的 `src` 和/或 `dst` 目录。
- `128` 其他故障。

文件

- `$HOME/.packingrules` 要保持同步的文件的列表
- `$HOME/.filesync-base` 基准线摘要文件

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	service/network/network-clients

另请参见

[packingrules\(4\)](#)、[attributes\(5\)](#)

引用名	find – find files
用法概要	<code>/usr/bin/find [-H -L] path... expression</code> <code>/usr/xpg4/bin/find [-H -L] path... expression</code>
描述	<p>The <code>find</code> utility recursively descends the directory hierarchy for each <i>path</i> seeking files that match a Boolean <i>expression</i> written in the primaries specified below.</p> <p><code>find</code> is able to descend to arbitrary depths in a file hierarchy and does not fail due to path length limitations (unless a <i>path</i> operand specified by the application exceeds <code>PATH_MAX</code> requirements).</p> <p><code>find</code> detects infinite loops; that is, entering a previously visited directory that is an ancestor of the last file encountered.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -H Causes the file information and file type evaluated for each symbolic link encountered on the command line to be those of the file referenced by the link, and not the link itself. If the referenced file does not exist, the file information and type is for the link itself. File information for all symbolic links not on the command line is that of the link itself. -L Causes the file information and file type evaluated for each symbolic link to be those of the file referenced by the link, and not the link itself. See NOTES. <p>Specifying more than one of the mutually-exclusive options -H and -L is not considered an error. The last option specified determines the behavior of the utility.</p>
操作数	<p>The following operands are supported:</p> <p><i>path</i> A pathname of a starting point in the directory hierarchy.</p> <p><i>expression</i> The first argument that starts with a <code>-</code>, or is a <code>!</code> or a <code>(</code>, and all subsequent arguments are interpreted as an <i>expression</i> made up of the following primaries and operators. In the descriptions, wherever <i>n</i> is used as a primary argument, it is interpreted as a decimal integer optionally preceded by a plus (+) or minus (-) sign, as follows:</p> <ul style="list-style-type: none"> <code>+n</code> more than <i>n</i> <code>n</code> exactly <i>n</i> <code>-n</code> less than <i>n</i>
Expressions	<p>Valid expressions are:</p> <ul style="list-style-type: none"> -acl True if the file have additional ACLs defined. -amin <i>n</i> File was last accessed <i>n</i> minutes ago.

- `-atime n` True if the file was accessed *n* days ago. The access time of directories in *path* is changed by `find` itself.
- `-cmin n` File's status was last changed *n* minutes ago.
- `-cpio device` Always true. Writes the current file on *device* in cpio format (5120-byte records).
- `-ctime n` True if the file's status was changed *n* days ago.
- `-depth` Always true. Causes descent of the directory hierarchy to be done so that all entries in a directory are acted on before the directory itself. This can be useful when `find` is used with `cpio(1)` to transfer files that are contained in directories without write permission.
- `-exec command` True if the executed command returns a zero value as exit status. The end of command must be punctuated by an escaped semicolon (;). A command argument {} is replaced by the current pathname. If the last argument to `-exec` is {} and you specify + rather than the semicolon (;), the command is invoked fewer times, with {} replaced by groups of pathnames. If any invocation of the command returns a non-zero value as exit status, `find` returns a non-zero exit status.
- `-follow` Always true and always evaluated no matter where it appears in *expression*. The behavior is unspecified if `-follow` is used when the `find` command is invoked with either the `-H` or the `-L` option. Causes symbolic links to be followed. When following symbolic links, `find` keeps track of the directories visited so that it can detect infinite loops. For example, such a loop would occur if a symbolic link pointed to an ancestor. This expression should not be used with the `find-type l` expression. See NOTES.
- `-fstype type` True if the filesystem to which the file belongs is of type *type*.
- `-group gname` True if the file belongs to the group *gname*. If *gname* is numeric and does not appear in the `group(4)` database, it is taken as a group ID.
- `-iname pattern` Similar to `-name`, but the match between the pattern and the base name of the current file name is case insensitive. (See EXAMPLES). Unlike the `-name` option, there is no special treatment in leading period and wildcard file name generation characters can match file names beginning with a . for both `/usr/bin/find` and `/usr/xpg4/bin/find`.
- `-inum n` True if the file has inode number *n*.
- `-links n` True if the file has *n* links.
- `-local` True if the file system type is not a remote file system type as defined in the `/etc/dfs/fstypes` file. `nfs` is used as the default remote filesystem type if the `/etc/dfs/fstypes` file is not present. The `-local` option descends the

	hierarchy of non-local directories. See <code>EXAMPLES</code> for an example of how to search for local files without descending.
<code>-ls</code>	<p>Always true. Prints current pathname together with its associated statistics. These include (respectively):</p> <ul style="list-style-type: none"> ▪ inode number ▪ size in kilobytes (1024 bytes) ▪ protection mode ▪ number of hard links ▪ user ▪ group ▪ size in bytes ▪ modification time. <p>If the file is a special file, the size field instead contains the major and minor device numbers.</p> <p>If the file is a symbolic link, the pathname of the linked-to file is printed preceded by '→'. The format is identical to that of <code>ls -gil</code>s (see <code>ls(1B)</code>).</p> <p>Formatting is done internally, without executing the <code>ls</code> program.</p>
<code>-mmin <i>n</i></code>	File's data was last modified <i>n</i> minutes ago.
<code>-mount</code>	Always true. Restricts the search to the file system containing the directory specified. Does not list mount points to other file systems.
<code>-mtime <i>n</i></code>	True if the file's data was modified <i>n</i> days ago.
<code>-name <i>pattern</i></code>	<p>True if <i>pattern</i> matches the basename of the current file name. Normal shell file name generation characters (see <code>sh(1)</code>) can be used. A backslash (\) is used as an escape character within the pattern. The pattern should be escaped or quoted when <code>find</code> is invoked from the shell.</p> <p>Unless the character '.' is explicitly specified in the beginning of <i>pattern</i>, a current file name beginning with '.' does not match <i>pattern</i> when using <code>/usr/bin/find</code>. <code>/usr/xpg4/bin/find</code> does not make this distinction; wildcard file name generation characters can match file names beginning with '.'.</p>
<code>-ncpio <i>device</i></code>	Always true. Writes the current file on <i>device</i> in <code>cpio -c</code> format (5120 byte records).
<code>-newer <i>file</i></code>	True if the current file has been modified more recently than the argument <i>file</i> .
<code>-nogroup</code>	True if the file belongs to a group not in the <code>group(4)</code> database.
<code>-nouser</code>	True if the file belongs to a user not in the <code>passwd(4)</code> database.

- `-ok command` Like `-exec`, except that the generated command line is printed with a question mark first, and is executed only if the response is affirmative.
- `-perm [-]mode` The *mode* argument is used to represent file mode bits. It is identical in format to the symbolic mode operand, *symbolic_mode_list*, described in [chmod\(1\)](#), and is interpreted as follows. To start, a template is assumed with all file mode bits cleared. An *op* symbol of:
- + Set the appropriate mode bits in the template
 - Clear the appropriate bits
 - = Set the appropriate mode bits, without regard to the contents of the file mode creation mask of the process
- The *op* symbol of – cannot be the first character of *mode*, to avoid ambiguity with the optional leading hyphen. Since the initial mode is all bits off, there are no symbolic modes that need to use – as the first character.
- If the hyphen is omitted, the primary evaluates as true when the file permission bits exactly match the value of the resulting template.
- Otherwise, if *mode* is prefixed by a hyphen, the primary evaluates as true if at least all the bits in the resulting template are set in the file permission bits.
- `-perm [-]onum` True if the file permission flags exactly match the octal number *onum* (see [chmod\(1\)](#)). If *onum* is prefixed by a minus sign (–), only the bits that are set in *onum* are compared with the file permission flags, and the expression evaluates true if they match.
- `-print` Always true. Causes the current pathname to be printed.
- `-print0` Always true. Causes the current pathname to be printed followed by a null character, rather than the NEWLINE character that `-print` uses.
- This allows file names that contain NEWLINES or other types of white space to be correctly interpreted by programs that process the `find` output. This option corresponds to the `-0` option of `cpio` and `xargs`.
- `-prune` Always yields true. Does not examine any directories or files in the directory structure below the *pattern* just matched. (See EXAMPLES). If `-depth` is specified, `-prune` has no effect.
- `-size n[c]` True if the file is *n* blocks long (512 bytes per block). If *n* is followed by a *c*, the size is in bytes.

-type <i>c</i>	True if the type of the file is <i>c</i> , where <i>c</i> is b, c, d, D, f, l, p, or s for block special file, character special file, directory, door, plain file, symbolic link, fifo (named pipe), or socket, respectively.
-user <i>uname</i>	True if the file belongs to the user <i>uname</i> . If <i>uname</i> is numeric and does not appear as a login name in the passwd(4) database, it is taken as a user ID.
-xdev	Same as the -mount primary.
-xattr	True if the file has extended attributes.

Complex Expressions The primaries can be combined using the following operators (in order of decreasing precedence):

1) (<i>expression</i>)	True if the parenthesized expression is true (parentheses are special to the shell and must be escaped).
2) ! <i>expression</i>	The negation of a primary (! is the unary <i>not</i> operator).
3) <i>expression</i> [-a] <i>expression</i>	Concatenation of primaries (the <i>and</i> operation is implied by the juxtaposition of two primaries).
4) <i>expression</i> -o <i>expression</i>	Alternation of primaries (-o is the <i>or</i> operator).

When you use `find` in conjunction with `cpio`, if you use the -L option with `cpio`, you must use the -L option or the -follow primitive with `find` and vice versa. Otherwise the results are unspecified.

If no *expression* is present, -print is used as the expression. Otherwise, if the specified expression does not contain any of the primaries -exec, -ok, -ls, or -print, the specified expression is effectively replaced by:

(*specified*) -print

The -user, -group, and -newer primaries each evaluate their respective arguments only once. Invocation of *command* specified by -exec or -ok does not affect subsequent primaries on the same file.

用法

See [largefile\(5\)](#) for the description of the behavior of `find` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Writing Out the Hierarchy Directory

The following commands are equivalent:

```
example% find .
example% find . -print
```

They both write out the entire directory hierarchy from the current directory.

示例 2 Removing Files

The following command removes all files in your home directory named `a.out` or `*.o` that have not been accessed for a week:

```
example% find $HOME \( -name a.out -o -name '*.o' \) \  
-atime +7 -exec rm {} \;
```

示例 3 Printing All File Names But Skipping SCCS Directories

The following command recursively print all file names in the current directory and below, but skipping SCCS directories:

```
example% find . -name SCCS -prune -o -print
```

示例 4 Printing all file names and the SCCS directory name

Recursively print all file names in the current directory and below, skipping the contents of SCCS directories, but printing out the SCCS directory name:

```
example% find . -print -name SCCS -prune
```

示例 5 Testing for the Newer File

The following command is basically equivalent to the `-nt` extension to [test\(1\)](#):

```
example$ if [ -n "$(find  
file1 -prune -newer file2)" ]; then  
  
printf %s\\n "file1 is newer than file2"
```

示例 6 Selecting a File Using 24-hour Mode

The descriptions of `-atime`, `-ctime`, and `-mtime` use the terminology *n* “24-hour periods”. For example, a file accessed at 23:59 is selected by:

```
example% find . -atime -1 -print
```

at 00:01 the next day (less than 24 hours later, not more than one day ago). The midnight boundary between days has no effect on the 24-hour calculation.

示例 7 Printing Files Matching a User's Permission Mode

The following command recursively print all file names whose permission mode exactly matches read, write, and execute access for user, and read and execute access for group and other:

```
example% find . -perm u=rwx,g=rx,o=rx
```

The above could alternatively be specified as follows:

```
example% find . -perm a=rwx,g-w,o-w
```


示例 8 Printing Files with Write Access for other

The following command recursively print all file names whose permission includes, but is not limited to, write access for other:

```
example% find . -perm -o+w
```

示例 9 Printing Local Files without Descending Non-local Directories

```
example% find . ! -local -prune -o -print
```

示例 10 Printing the Files in the Name Space Possessing Extended Attributes

```
example% find . -xattr
```

示例 11 Printing all PDF Filenames Regardless of Case

The following example finds all file names with an extension of .pdf, .PDF, .Pdf, and so forth.

```
example% find . -iname '*.pdf'
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `find`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

`PATH` Determine the location of the *utility_name* for the `-exec` and `-ok` primaries.

Affirmative responses are processed using the extended regular expression defined for the `yesexpr` keyword in the `LC_MESSAGES` category of the user's locale. The locale specified in the `LC_COLLATE` category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for `yesexpr`. The locale specified in `LC_CTYPE` determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the `yesexpr`. See [locale\(5\)](#).

退出状态

The following exit values are returned:

0 All *path* operands were traversed successfully.

>0 An error occurred.

文件

/etc/passwd Password file

/etc/group Group file

/etc/dfs/fstypes File that registers distributed file system packages

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[chmod\(1\)](#), [cpio\(1\)](#), [sh\(1\)](#), [test\(1\)](#), [ls\(1B\)](#), [acl\(2\)](#), [stat\(2\)](#), [umask\(2\)](#), [group\(4\)](#), [passwd\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [locale\(5\)](#), [standards\(5\)](#)

警告

The following options are obsolete and will not be supported in future releases:

- cpio *device* Always true. Writes the current file on *device* in cpio format (5120-byte records).
- ncpio *device* Always true. Writes the current file on *device* in cpio -c format (5120-byte records).

附注

When using `find` to determine files modified within a range of time, use the `-mtime` argument *before* the `-print` argument. Otherwise, `find` gives all files.

Some files that might be under the Solaris root file system are actually mount points for virtual file systems, such as `mntfs` or `namefs`. When comparing against a `ufs` file system, such files are not selected if `-mount` or `-xdev` is specified in the `find` expression.

Using the `-L` or `-follow` option is not recommended when descending a file-system hierarchy that is under the control of other users. In particular, when using `-exec`, symbolic links can lead the `find` command out of the hierarchy in which it started. Using `-type` is not sufficient to restrict the type of files on which the `-exec` command operates, because there is an inherent race condition between the type-check performed by the `find` command and the time the executed command operates on the file argument.

引用名	finger – display information about local and remote users
用法概要	<pre>finger [-bfhilmqsw] [username]... finger [-l] [username@hostname 1 [@hostname 2 .. @hostname n]...] finger [-l] [@hostname 1 [@hostname 2 .. @hostname n]...]</pre>
描述	<p>By default, the <code>finger</code> command displays in multi-column format the following information about each logged-in user:</p> <ul style="list-style-type: none"> ▪ user name ▪ user's full name ▪ terminal name (prefixed with a '*' (asterisk) if write-permission is denied) ▪ idle time ▪ login time ▪ host name, if logged in remotely <p>Idle time is in minutes if it is a single integer, in hours and minutes if a ':' (colon) is present, or in days and hours if a 'd' is present.</p> <p>When one or more <i>username</i> arguments are given, more detailed information is given for each <i>username</i> specified, whether they are logged in or not. <i>username</i> must be that of a local user, and may be a first or last name, or an account name. Information is presented in multi-line format as follows:</p> <ul style="list-style-type: none"> ▪ the user name and the user's full name ▪ the user's home directory and login shell ▪ time the user logged in if currently logged in, or the time the user last logged in; and the terminal or host from which the user logged in ▪ last time the user received mail, and the last time the user read mail ▪ the first line of the \$HOME/.project file, if it exists ▪ the contents of the \$HOME/.plan file, if it exists <p>Note: when the comment (GECOS) field in <code>/etc/passwd</code> includes a comma, <code>finger</code> does not display the information following the comma.</p> <p>If the arguments <code>username@hostname1[@hostname2 .. @hostnamen]</code> or <code>@hostname1[@hostname2 .. @hostnamen]</code> are used, the request is sent first to <code>hostnamen</code> and forwarded through each <code>hostnamen-1</code> to <code>hostname1</code>. The program uses the <code>finger</code> user information protocol (see RFC 1288) to query that remote host for information about the named user (if <i>username</i> is specified), or about each logged-in user. The information displayed is server dependent.</p>

As required by RFC 1288, `finger` passes only printable, 7-bit ASCII data. This behavior may be modified by a system administrator by using the `PASS` option in `/etc/default/finger`. Specifying `PASS=low` allows all characters less than decimal 32 ASCII. Specifying `PASS=high` allows all characters greater than decimal 126 ASCII. `PASS=low,high` or `PASS=high,low` allows both characters less than 32 and greater than 126 to pass through.

选项

The following options are supported, except that the `username@hostname` form supports only the `-l` option:

- `-b` Suppresses printing the user's home directory and shell in a long format printout.
- `-f` Suppresses printing the header that is normally printed in a non-long format printout.
- `-h` Suppresses printing of the `.project` file in a long format printout.
- `-i` Forces "idle" output format, which is similar to short format except that only the login name, terminal, login time, and idle time are printed.
- `-l` Forces long output format.
- `-m` Matches arguments only on user name (not first or last name).
- `-p` Suppresses printing of the `.plan` file in a long format printout.
- `-q` Forces quick output format, which is similar to short format except that only the login name, terminal, and login time are printed.
- `-s` Forces short output format.
- `-w` Suppresses printing the full name in a short format printout.

文件

<code>\$HOME/.plan</code>	user's plan
<code>\$HOME/.project</code>	user's projects
<code>/etc/default/finger</code>	finger options file
<code>/etc/passwd</code>	password file
<code>/var/adm/lastlog</code>	time of last login
<code>/var/adm/utmpx</code>	accounting

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

另请参见

[passwd\(1\)](#), [who\(1\)](#), [whois\(1\)](#), [passwd\(4\)](#), [attributes\(5\)](#)

Zimmerman, D., The Finger User Information Protocol, RFC 1288, Center for Discrete Mathematics and Theoretical Computer Science (DIMACS), Rutgers University, December 1991.

附注

The `finger user information protocol` limits the options that may be used with the remote form of this command.

引用名 `fmt` – simple text formatters

用法概要 `fmt [-cs] [-w width | -width] [inputfile]...`

描述 `fmt` is a simple text formatter that fills and joins lines to produce output lines of (up to) the number of characters specified in the `-w width` option. The default *width* is 72. `fmt` concatenates the *inputfiles* listed as arguments. If none are given, `fmt` formats text from the standard input.

Blank lines are preserved in the output, as is the spacing between words. `fmt` does not fill nor split lines beginning with a '.' (dot), for compatibility with [nroff\(1\)](#). Nor does it fill or split a set of contiguous non-blank lines which is determined to be a mail header, the first line of which must begin with "From".

Indentation is preserved in the output, and input lines with differing indentation are not joined (unless `-c` is used).

`fmt` can also be used as an in-line text filter for [vi\(1\)](#). The `vi` command:

```
!}fmt
```

reformats the text between the cursor location and the end of the paragraph.

选项

- `-c` Crown margin mode. Preserve the indentation of the first two lines within a paragraph, and align the left margin of each subsequent line with that of the second line. This is useful for tagged paragraphs.
- `-s` Split lines only. Do not join short lines to form longer ones. This prevents sample lines of code, and other such formatted text, from being unduly combined.
- `-w width | -width` Fill output lines to up to *width* columns.

操作数 *inputfile* Input file.

环境变量 See [environ\(5\)](#) for a description of the `LC_CTYPE` environment variable that affects the execution of `fmt`.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [nroff\(1\)](#), [vi\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注 The `-width` option is acceptable for BSD compatibility, but it may go away in future releases.

引用名	fmtmsg – display a message on stderr or system console
用法概要	fmtmsg [-c <i>class</i>] [-u <i>subclass</i>] [-l <i>label</i>] [-s <i>severity</i>] [-t <i>tag</i>] [-a <i>action</i>] <i>text</i>
描述	<p>Based on a message's classification component, the <code>fmtmsg</code> utility either writes a formatted message to <code>stderr</code> or writes a formatted message to the console.</p> <p>A formatted message consists of up to five standard components (see environment variable <code>MSGVERB</code> in the ENVIRONMENT VARIABLES section of this page). The classification and subclass components are not displayed as part of the standard message, but rather define the source of the message and direct the display of the formatted message.</p>
选项	<p>The following options are supported:</p> <p>-c <i>class</i> Describes the source of the message. Valid keywords are:</p> <ul style="list-style-type: none"> <code>hard</code> The source of the condition is hardware. <code>soft</code> The source of the condition is software. <code>firm</code> The source of the condition is firmware. <p>-u <i>subclass</i> A list of keywords (separated by commas) that further defines the message and directs the display of the message. Valid keywords are:</p> <ul style="list-style-type: none"> <code>appl</code> The condition originated in an application. This keyword should not be used in combination with either <code>util</code> or <code>opsys</code>. <code>util</code> The condition originated in a utility. This keyword should not be used in combination with either <code>appl</code> or <code>opsys</code>. <code>opsys</code> The message originated in the kernel. This keyword should not be used in combination with either <code>appl</code> or <code>util</code>. <code>recov</code> The application will recover from the condition. This keyword should not be used in combination with <code>nrecov</code>. <code>nrecov</code> The application will not recover from the condition. This keyword should not be used in combination with <code>recov</code>. <code>print</code> Print the message to the standard error stream <code>stderr</code>. <code>console</code> Write the message to the system console. <code>print</code>, <code>console</code>, or both may be used. <p>-l <i>label</i> Identifies the source of the message.</p> <p>-s <i>severity</i> Indicates the seriousness of the error. The keywords and definitions of the standard levels of <i>severity</i> are:</p> <ul style="list-style-type: none"> <code>halt</code> The application has encountered a severe fault and is halting.

	error	The application has detected a fault.
	warn	The application has detected a condition that is out of the ordinary and might be a problem.
	info	The application is providing information about a condition that is not in error.
-t <i>tag</i>		The string containing an identifier for the message.
-a <i>action</i>		A text string describing the first step in the error recovery process. This string must be written so that the entire <i>action</i> argument is interpreted as a single argument. <code>fmtmsg</code> precedes each action string with the TO FIX: prefix.
<i>text</i>		A text string describing the condition. Must be written so that the entire <i>text</i> argument is interpreted as a single argument.

示例

示例 1 Standard message format

The following example of `fmtmsg` produces a complete message in the standard message format and displays it to the standard error stream.

```
example% fmtmsg -c soft -u recov,print,appl -l UX:cat \  
-s error -t UX:cat:001 -a "refer to manual" "invalid syntax"
```

produces:

```
UX:cat: ERROR: invalid syntax  
TO FIX: refer to manual   UX:cat:138
```

示例 2 Using MSGVERB

When the environment variable `MSGVERB` is set as follows:

```
MSGVERB=severity:text:action
```

and Example 1 is used, `fmtmsg` produces:

```
ERROR: invalid syntax  
TO FIX: refer to manual
```

示例 3 Using SEV_LEVEL

When the environment variable `SEV_LEVEL` is set as follows:

```
SEV_LEVEL=note,5,NOTE
```

the following `fmtmsg` command:

```
example% fmtmsg -c soft -u print -l UX:cat -s note \  
-a "refer to manual" "invalid syntax"
```

produces:

示例 3 Using SEV_LEVEL (续)

NOTE: invalid syntax
TO FIX: refer to manual

and displays the message on stderr.

环境变量

The environment variables MSGVERB and SEV_LEVEL control the behavior of `fmtmsg`. MSGVERB is set by the administrator in the `/etc/profile` for the system. Users can override the value of MSGVERB set by the system by resetting MSGVERB in their own `.profile` files or by changing the value in their current shell session. SEV_LEVEL can be used in shell scripts.

MSGVERB tells `fmtmsg` which message components to select when writing messages to `stderr`. The value of MSGVERB is a colon-separated list of optional keywords. MSGVERB can be set as follows:

```
MSGVERB=[keyword[:keyword[...]]]
export MSGVERB
```

Valid *keywords* are: `label`, `severity`, `text`, `action`, and `tag`. If MSGVERB contains a keyword for a component and the component's value is not the component's null value, `fmtmsg` includes that component in the message when writing the message to `stderr`. If MSGVERB does not include a keyword for a message component, that component is not included in the display of the message. The keywords may appear in any order. If MSGVERB is not defined, if its value is the null string, if its value is not of the correct format, or if it contains keywords other than the valid ones listed above, `fmtmsg` selects all components.

MSGVERB affects only which message components are selected for display. All message components are included in console messages.

SEV_LEVEL defines severity levels and associates print strings with them for use by `fmtmsg`. The standard severity levels shown below cannot be modified. Additional severity levels can be defined, redefined, and removed.

- 0 (no severity is used)
- 1 HALT
- 2 ERROR
- 3 WARNING
- 4 INFO

SEV_LEVEL is set as follows:

description is a comma-separated list containing three fields:

```
SEV_LEVEL= [description[:description[:...]]]
export SEV_LEVEL
```

description=*severity_keyword*, *level*, *printstring*

severity_keyword is a character string used as the keyword with the `-s severity` option to `fmtmsg`.

level is a character string that evaluates to a positive integer (other than 0, 1, 2, 3, or 4, which are reserved for the standard severity levels). If the keyword *severity_keyword* is used, *level* is the severity value passed on to `fmtmsg(3C)`.

printstring is the character string used by `fmtmsg` in the standard message format whenever the severity value *level* is used.

If `SEV_LEVEL` is not defined, or if its value is null, no severity levels other than the defaults are available. If a *description* in the colon separated list is not a comma separated list containing three fields, or if the second field of a comma separated list does not evaluate to a positive integer, that *description* in the colon separated list is ignored.

退出状态

The following exit values are returned:

- 0 All the requested functions were executed successfully.
- 1 The command contains a syntax error, an invalid option, or an invalid argument to an option.
- 2 The function executed with partial success, however the message was not displayed on `stderr`.
- 4 The function executed with partial success; however, the message was not displayed on the system console.
- 32 No requested functions were executed successfully.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[addseverity\(3C\)](#), [fmtmsg\(3C\)](#), [attributes\(5\)](#)

引用名	fold – filter for folding lines
用法概要	fold [-bs] [-w <i>width</i> -width] [<i>file</i>]...
描述	<p>The <code>fold</code> utility is a filter that folds lines from its input files, breaking the lines to have a maximum of <i>width</i> column positions (or bytes, if the <code>-b</code> option is specified). Lines are broken by the insertion of a NEWLINE character such that each output line (referred to later in this section as a segment) is the maximum width possible that does not exceed the specified number of column positions (or bytes). A line is not broken in the middle of a character. The behavior is undefined if <i>width</i> is less than the number of columns any single character in the input would occupy.</p> <p>If the CARRIAGE-RETURN, BACKSPACE, or TAB characters are encountered in the input, and the <code>-b</code> option is not specified, they are treated specially:</p> <p>BACKSPACE The current count of line width is decremented by one, although the count never becomes negative. <code>fold</code> does not insert a NEWLINE character immediately before or after any BACKSPACE character.</p> <p>CARRIAGE-RETURN The current count of line width is set to 0. <code>fold</code> does not insert a NEWLINE character immediately before or after any CARRIAGE-RETURN character.</p> <p>TAB Each TAB character encountered advances the column position pointer to the next tab stop. Tab stops are at each column position <i>n</i> such that <i>n</i> modulo 8 equals 1.</p>
选项	<p>The following options are supported:</p> <p><code>-b</code> Counts <i>width</i> in bytes rather than column positions.</p> <p><code>-s</code> If a segment of a line contains a blank character within the first <i>width</i> column positions (or bytes), breaks the line after the last such blank character meeting the width constraints. If there is no blank character meeting the requirements, the <code>-s</code> option has no effect for that output segment of the input line.</p> <p><code>-w <i>width</i> -width</code> Specifies the maximum line length, in column positions (or bytes if <code>-b</code> is specified). If <i>width</i> is not a positive decimal number, an error is returned. The default value is 80.</p>
操作数	<p>The following operand is supported:</p> <p><i>file</i> A path name of a text file to be folded. If no <i>file</i> operands are specified, the standard input is used.</p>

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `fold`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态 The following exit values are returned:

- 0 All input files were processed successfully.
- >0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [cut\(1\)](#), [pr\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注 `fold` and [cut\(1\)](#) can be used to create text files out of files with arbitrary line lengths. `fold` should be used when the contents of long lines need to be kept contiguous. `cut` should be used when the number of lines (or records) needs to remain constant.

`fold` is frequently used to send text files to line printers that truncate, rather than fold, lines wider than the printer is able to print (usually 80 or 132 column positions).

`fold` might not work correctly if underlining is present.

引用名	from – display the sender and date of newly-arrived mail messages
用法概要	<code>/usr/ucb/from [-s <i>sender</i>] [<i>username</i>]</code>
描述	The <code>from</code> utility prints out the mail header lines in your mailbox file to show you who your mail is from. If <i>username</i> is specified, <i>username</i> 's mailbox is examined instead of your own.
选项	The following option is supported: <i>-s sender</i> Only display headers for mail sent by <i>sender</i> .
用法	See largefile(5) for the description of the behavior of <code>from</code> when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).
文件	<code>/var/mail/*</code>
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [biff\(1B\)](#), [mail\(1B\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

引用名 ftp – file transfer program

用法概要 ftp [-adfginpstvx] [-m *GSS Mech*] [-T *timeout*]
[*hostname* [*port*]]

描述 The ftp command is the user interface to the Internet standard File Transfer Protocol (FTP). ftp transfers files to and from a remote network site.

The host and optional port with which ftp is to communicate can be specified on the command line. If this is done, ftp immediately attempts to establish a connection to an FTP server on that host. Otherwise, ftp enters its command interpreter and awaits instructions from the user. When ftp is awaiting commands from the user, it displays the prompt ftp>.

选项 The following options can be specified at the command line, or to the command interpreter:

- a Uses GSSAPI authentication *only*. If the authentication fails, this option closes the connection.
- d Enables debugging.
- f Forwards local security credentials to the remote server.
- g Disables filename “globbing”.
- i Turns off interactive prompting during multiple file transfers.
- m Specifies the GSS-API mechanism to use. The default is to use the kerberos_v5 mechanism. Supported alternatives are defined in /etc/gss/mech (see [mech\(4\)](#)).
- n Does not attempt “auto-login” upon initial connection. If auto-login is not disabled, ftp checks the .netrc file in the user's home directory for an entry describing an account on the remote machine. If no entry exists, ftp prompts for the login name of the account on the remote machine (the default is the login name on the local machine), and, if necessary, prompts for a password and an account with which to login.
- p Enables passive mode for data transfers. This command is useful when connecting to a remote host from behind a connection filtering firewall.
- s Skips the SYST command that is sent by default to all remote servers upon connection. The system command is what enables the automatic use of binary mode rather than the protocol default ascii mode.

As some older servers cannot handle the ftp command, this directive is provided to allow inter-operability with these servers.
- t Enables packet tracing (unimplemented).
- T *timeout* Enables global connection timer, specified in seconds (decimal). There is a timer for the control connection that is reset when anything is sent to the

- server and disabled while the client is prompting for user input. Another independent timer is used to monitor incoming or outgoing data connections.
- v Shows all responses from the remote server, as well as report on data transfer statistics. This is turned on by default if `ftp` is running interactively with its input coming from the user's terminal.
 - x Attempts to use GSSAPI for authentication and encryption. Data and Command channel protection is set to "private".

The following commands can be specified to the command interpreter:

- ! [*command*] Runs *command* as a shell command on the local machine. If no *command* is given, invokes an interactive shell.
- \$ *macro-name* [*args*] Executes the macro *macro-name* that was defined with the `macdef` command. Arguments are passed to the macro unglobbed.
- account [*passwd*] Supplies a supplemental password required by a remote system for access to resources once a login has been successfully completed. If no argument is included, the user is prompted for an account password in a non-echoing input mode.
- append *local-file* [*remote-file*] Appends a local file to a file on the remote machine. If *remote-file* is not specified, the local file name is used, subject to alteration by any `nt rans` or `nmap` settings. File transfer uses the current settings for "representation type", "file structure", and "transfer mode".
- ascii Sets the "representation type" to "network ASCII". This is the default type.
- bell Sounds a bell after each file transfer command is completed.
- binary Sets the "representation type" to "image".
- bye Terminates the FTP session with the remote server and exit `ftp`. An EOF also terminates the session and exit.
- case Toggles remote computer file name case mapping during `mget` commands. When `case` is on (default is off), remote computer file names with all letters in upper case are written in the local directory with the letters mapped to lower case.

<i>cd remote-directory</i>	Changes the working directory on the remote machine to <i>remote-directory</i> .
<i>cdup</i>	Changes the remote machine working directory to the parent of the current remote machine working directory.
<i>clear</i>	Sets the protection level on data transfers to “clear”. If no ADAT command succeeded, then this is the default protection level.
<i>close</i>	Terminates the FTP session with the remote server, and return to the command interpreter. Any defined macros are erased.
<i>cr</i>	Toggles RETURN stripping during “network ASCII” type file retrieval. Records are denoted by a RETURN/LINEFEED sequence during “network ASCII” type file transfer. When <i>cr</i> is on (the default), RETURN characters are stripped from this sequence to conform with the UNIX system single LINEFEED record delimiter. Records on non-UNIX-system remote hosts can contain single LINEFEED characters; when an “network ASCII” type transfer is made, these LINEFEED characters can be distinguished from a record delimiter only when <i>cr</i> is off.
<i>delete remote-file</i>	Deletes the file <i>remote-file</i> on the remote machine.
<i>debug</i>	Toggles debugging mode. When debugging is on, <i>ftp</i> prints each command sent to the remote machine, preceded by the string <i>-></i> .
<i>dir [remote-directory [local-file]]</i>	Prints a listing of the directory contents in the directory, <i>remote-directory</i> , and, optionally, placing the output in <i>local-file</i> . If no directory is specified, the current working directory on the remote machine is used. If no local file is specified, or <i>local-file</i> is <i>-</i> , output is sent to the terminal.
<i>disconnect</i>	A synonym for <i>close</i> .
<i>form [format-name]</i>	Sets the carriage control format subtype of the “representation type” to <i>format-name</i> . The only valid <i>format-name</i> is <i>non-print</i> , which corresponds to the default “non-print” subtype.

<code>get remote-file [local-file]</code>	Retrieves the <i>remote-file</i> and store it on the local machine. If the local file name is not specified, it is given the same name it has on the remote machine, subject to alteration by the current case, ntrans, and nmap settings. The current settings for “representation type”, “file structure”, and “transfer mode” are used while transferring the file.
<code>glob</code>	<p>Toggles filename expansion, or “globbing”, for <code>delete</code>, <code>mget</code> and <code>mput</code>. If globbing is turned off, filenames are taken literally.</p> <p>Globbing for <code>mput</code> is done as in sh(1). For <code>delete</code> and <code>mget</code>, each remote file name is expanded separately on the remote machine, and the lists are not merged.</p> <p>Expansion of a directory name is likely to be radically different from expansion of the name of an ordinary file: the exact result depends on the remote operating system and FTP server, and can be previewed with the command, <code>mls remote-files -</code>.</p> <p><code>mget</code> and <code>mput</code> are not meant to transfer entire directory subtrees of files. You can do this by transferring a tar(1) archive of the subtree (using a “representation type” of “image” as set by the binary command).</p>
<code>hash</code>	Toggles hash-sign (#) printing for each data block transferred. The size of a data block is 8192 bytes.
<code>help [command]</code>	Prints an informative message about the meaning of <i>command</i> . If no argument is given, <code>ftp</code> prints a list of the known commands.
<code>lcd [directory]</code>	Changes the working directory on the local machine. If no <i>directory</i> is specified, the user's home directory is used.
<code>ls [-al remote-directory [local-file]]</code>	By default, prints an abbreviated listing of the contents of a directory on the remote machine. This default behavior can be changed to make <code>ls</code> a synonym of the <code>dir</code> command. This change can be achieved by setting <code>FTP_LS_SENDS_NLST</code> to 'no' in <code>/etc/default/ftp</code> or in the environment. See ftp(4) for details.

The `-a` option lists all entries, including those that begin with a dot (`.`), which are normally not listed. The `-l` option lists files in long format, giving mode, number of links, owner, group, size in bytes, and time of last modification for each file. If the file is a special file, the size field instead contains the major and minor device numbers rather than a size. If the file is a symbolic link, the filename is printed followed by “`→`” and the pathname of the referenced file.

If *remote-directory* is left unspecified, the current working directory is used.

If no local file is specified, or if *local-file* is `-`, the output is sent to the terminal.

`macdef` *macro-name*

Defines a macro. Subsequent lines are stored as the macro *macro-name*. A null line (consecutive NEWLINE characters in a file or RETURN characters from the terminal) terminates macro input mode. There is a limit of 16 macros and 4096 total characters in all defined macros. Macros remain defined until a `close` command is executed.

The macro processor interprets `$` and `\` as special characters. A `$` followed by a number (or numbers) is replaced by the corresponding argument on the macro invocation command line. A `$` followed by an `i` signals that macro processor that the executing macro is to be looped. On the first pass, `$i` is replaced by the first argument on the macro invocation command line; on the second pass, it is replaced by the second argument, and so on. A `\` followed by any character is replaced by that character. Use the `\` to prevent special treatment of the `$`.

`mdelete` *remote-files*

Deletes the *remote-files* on the remote machine.

`mdir` *remote-files local-file*

Like `dir`, except multiple remote files can be specified. If interactive prompting is on, `ftp` prompts the user to verify that the last argument is indeed the target local file for receiving `mdir` output.

<code>mget <i>remote-files</i></code>	Expands the <i>remote-files</i> on the remote machine and do a <code>get</code> for each file name thus produced. See <code>glob</code> for details on the filename expansion. Resulting file names are processed according to <code>case</code> , <code>nt rans</code> , and <code>nmap</code> settings. Files are transferred into the local working directory, which can be changed with <code>lcd <i>directory</i></code> . New local directories can be created with <code>! mkdir <i>directory</i></code> .
<code>mkdir <i>directory-name</i></code>	Makes a directory on the remote machine.
<code>m!s <i>remote-files local-file</i></code>	Like <code>!s(1)</code> , except multiple remote files can be specified. If interactive prompting is on, <code>ftp</code> prompts the user to verify that the last argument is indeed the target local file for receiving <code>m!s</code> output.
<code>mode [<i>mode-name</i>]</code>	Sets the “transfer mode” to <i>mode-name</i> . The only valid <i>mode-name</i> is <code>stream</code> , which corresponds to the default “stream” mode. This implementation only supports <code>stream</code> , and requires that it be specified.
<code>mput <i>local-files</i></code>	Expands wild cards in the list of local files given as arguments and do a <code>put</code> for each file in the resulting list. See <code>glob</code> for details of filename expansion. Resulting file names are processed according to <code>nt rans</code> and <code>nmap</code> settings.
<code>n!ist [-a! <i>remote-directory</i> [<i>local-file</i>]]</code>	Prints an abbreviated listing of the contents of a directory on the remote machine, listing only those files that can be retrieved by the <code>get</code> command, unless the <code>-a</code> or <code>-l</code> option is used. If <i>remote-directory</i> is left unspecified, the current working directory is used. The <code>-a</code> option lists all entries, including those that begin with a dot (<code>.</code>), which are normally not listed. The <code>-l</code> option lists files in long format the same way it does when used with the <code>!s</code> command.
<code>nmap [<i>inpattern outpattern</i>]</code>	Sets or unsets the filename mapping mechanism. If no arguments are specified, the filename mapping mechanism is unset. If arguments are specified, remote filenames are

mapped during `mput` commands and `put` commands issued without a specified remote target filename. If arguments are specified, local filenames are mapped during `mget` commands and `get` commands issued without a specified local target filename.

This command is useful when connecting to a non-UNIX-system remote host with different file naming conventions or practices. The mapping follows the pattern set by *inpattern* and *outpattern*. *inpattern* is a template for incoming filenames (which can have already been processed according to the `nt rans` and case settings). Variable templating is accomplished by including the sequences `$1`, `$2`, . . . , `$9` in *inpattern*. Use `\` to prevent this special treatment of the `$` character. All other characters are treated literally, and are used to determine the `nmap` *inpattern* variable values.

For example, given *inpattern* `$1.$2` and the remote file name `mydata.data`, `$1` would have the value `mydata`, and `$2` would have the value `data`.

The *outpattern* determines the resulting mapped filename. The sequences `$1`, `$2`, . . . , `$9` are replaced by any value resulting from the *inpattern* template. The sequence `$0` is replaced by the original filename. Additionally, the sequence `[seq1 , seq2]` is replaced by *seq1* if *seq1* is not a null string; otherwise it is replaced by *seq2*.

For example, the command `nmap $1.$2.$3 [$1, $2] . [$2, file]` would yield the output filename `myfile.data` for input filenames `myfile.data` and `myfile.data.old`, `myfile.file` for the input filename `myfile`, and `myfile.myfile` for the input filename `.myfile`. SPACE characters can be included in *outpattern*, as in the example `nmap $1 | sed`

<code>nt rans [<i>inchars</i> [<i>outchars</i>]]</code>	<p>"s/ *\$//" > \$1. Use the \ character to prevent special treatment of the \$, [,], and ,, characters.</p> <p>Sets or unsets the filename character translation mechanism. If no arguments are specified, the filename character translation mechanism is unset. If arguments are specified, characters in remote filenames are translated during <code>mput</code> commands and <code>put</code> commands issued without a specified remote target filename, and characters in local filenames are translated during <code>mget</code> commands and <code>get</code> commands issued without a specified local target filename.</p> <p>This command is useful when connecting to a non-UNIX-system remote host with different file naming conventions or practices. Characters in a filename matching a character in <i>inchars</i> are replaced with the corresponding character in <i>outchars</i>. If the character's position in <i>inchars</i> is longer than the length of <i>outchars</i>, the character is deleted from the file name.</p> <p>Only 16 characters can be translated when using the <code>nt rans</code> command under <code>ftp</code>. Use case (described above) if needing to convert the entire alphabet.</p>
<code>open <i>host</i> [<i>port</i>]</code>	<p>Establishes a connection to the specified <i>host</i> FTP server. An optional port number can be supplied, in which case, <code>ftp</code> attempts to contact an FTP server at that port. If the <i>auto-login</i> option is on (default setting), <code>ftp</code> also attempts to automatically log the user in to the FTP server.</p>
<code>passive</code>	<p>Toggles passive mode. When passive mode is turned on, the <code>ftp</code> client sends the <code>PASV</code> command requesting that the FTP server open a port for the data connection and return the address of that port. The remote server listens on that port and the client connects to it. When passive mode is turned off, the <code>ftp</code> client sends</p>

	<p>the PORT command to the server specifying an address for the remote server to connect back to. Passive mode is useful when the connections to the ftp client are controlled, for example, when behind a firewall. When connecting to an IPv6-enabled FTP server, EPSV can be used in place of PASV and EPRT in place of PORT.</p>
<code>private</code>	<p>Sets the protection level on data transfers to “private”. Data transmissions are confidentiality— and integrity—protected by encryption. If no ADAT command succeeded, then the only possible level is “clear”.</p>
<code>prompt</code>	<p>Toggles interactive prompting. Interactive prompting occurs during multiple file transfers to allow the user to selectively retrieve or store files. By default, prompting is turned on. If prompting is turned off, any <code>mget</code> or <code>mput</code> transfers all files, and any <code>mdelete</code> deletes all files.</p>
<code>protect <i>protection-level</i></code>	<p>Sets the protection level on data transfers to <i>protection-level</i>. The valid protection levels are “clear” for unprotected data transmissions, “safe” for data transmissions that are integrity-protected by cryptographic checksum, and “private” for data transmissions that are confidentiality— and integrity— protected by encryption. If no ADAT command succeeded, then the only possible level is “clear”. If no level is specified, the current level is printed. The default protection level is “clear”.</p>
<code>proxy <i>ftp-command</i></code>	<p>Executes an FTP command on a secondary control connection. This command allows simultaneous connection to two remote FTP servers for transferring files between the two servers. The first proxy command should be an open, to establish the secondary control connection. Enter the command <code>proxy ?</code> to see other FTP commands executable on the secondary connection.</p>

The following commands behave differently when prefaced by `proxy`: `open` does not define new macros during the auto-login process, `close` does not erase existing macro definitions, `get` and `mget` transfer files from the host on the primary control connection to the host on the secondary control connection, and `put`, `mput`, and `append` transfer files from the host on the secondary control connection to the host on the primary control connection.

Third party file transfers depend upon support of the `PASV` command by the server on the secondary control connection.

`put local-file [remote-file]`

Stores a local file on the remote machine. If *remote-file* is left unspecified, the local file name is used after processing according to any `nt rans` or `nmap` settings in naming the remote file. File transfer uses the current settings for “representation type”, “file structure”, and “transfer mode”.

`pwd`

Prints the name of the current working directory on the remote machine.

`quit`

A synonym for `bye`.

`quote arg1 arg2 ...`

Sends the arguments specified, verbatim, to the remote FTP server. A single FTP reply code is expected in return. (The `remotehelp` command displays a list of valid arguments.)

`quote` should be used only by experienced users who are familiar with the FTP protocol.

`recv remote-file [local-file]`

A synonym for `get`.

`reget remote-file [local-file]`

The `reget` command acts like `get`, except that if *local-file* exists and is smaller than *remote-file*, *local-file* is presumed to be a partially transferred copy of *remote-file* and the transfer is continued from the apparent point of failure. This command is useful when transferring large files over networks that are prone to dropping connections.

<code>remotehelp [<i>command-name</i>]</code>	Requests help from the remote FTP server. If a <i>command-name</i> is specified it is supplied to the server as well.
<code>rename <i>from to</i></code>	Renames the file <i>from</i> on the remote machine to have the name <i>to</i> .
<code>reset</code>	Clears reply queue. This command re-synchronizes command/reply sequencing with the remote FTP server. Resynchronization can be necessary following a violation of the FTP protocol by the remote server.
<code>restart [<i>marker</i>]</code>	Restarts the immediately following get or put at the indicated marker. On UNIX systems, <i>marker</i> is usually a byte offset into the file. When followed by an mget, the restart applies to the first get performed. Specifying a <i>marker</i> of 0 clears the restart marker. If no argument is specified, the current restart status is displayed.
<code>rmdir <i>directory-name</i></code>	Deletes a directory on the remote machine.
<code>runique</code>	Toggles storing of files on the local system with unique filenames. If a file already exists with a name equal to the target local filename for a get or mget command, a .1 is appended to the name. If the resulting name matches another existing file, a .2 is appended to the original name. If this process continues up to .99, an error message is printed, and the transfer does not take place. The generated unique filename is reported. runique does not affect local files generated from a shell command. The default value is off.
<code>safe</code>	Sets the protection level on data transfers to "safe". Data transmissions are integrity-protected by cryptographic checksum. If no ADAT command succeeded, then the only possible level is "clear".
<code>send <i>local-file</i> [<i>remote-file</i>]</code>	A synonym for put.

<code>sendport</code>	Toggles the use of PORT commands. By default, <code>ftp</code> attempts to use a PORT command when establishing a connection for each data transfer. The use of PORT commands can prevent delays when performing multiple file transfers. If the PORT command fails, <code>ftp</code> uses the default data port. When the use of PORT commands is disabled, no attempt is made to use PORT commands for each data transfer. This is useful when connected to certain FTP implementations that ignore PORT commands but incorrectly indicate they have been accepted.
<code>site arg1 [arg2] ...</code>	Sends the arguments specified, verbatim, to the remote FTP server as a SITE command.
<code>status</code>	Show the current status of <code>ftp</code> .
<code>struct [struct-name]</code>	Sets the file structure to <i>struct-name</i> . The only valid <i>struct-name</i> is <code>file</code> , which corresponds to the default “file” structure. The implementation only supports <code>file</code> , and requires that it be specified.
<code>sunique</code>	Toggles storing of files on remote machine under unique file names. The remote FTP server must support the STOU command for successful completion. The remote server reports the unique name. Default value is off.
<code>tcpwindow [size]</code>	Sets the TCP window size to be used for data connections. Specifying a size of 0 stops the explicit setting of the TCP window size on data connections. If no argument is specified, the current setting is displayed.
<code>tenex</code>	Sets the “representation type” to that needed to talk to TENEX machines.
<code>trace</code>	Toggles packet tracing (unimplemented).
<code>type [type-name]</code>	Sets the “representation type” to <i>type-name</i> . The valid <i>type-names</i> are <code>ascii</code> for “network ASCII”, <code>binary</code> or <code>image</code> for “image”, and <code>tenex</code> for “local byte size” with a byte size of 8 (used to talk to TENEX machines). If no type is

<code>user <i>user-name</i> [<i>password</i> [<i>account</i>]]</code>	specified, the current type is printed. The default type is “network ASCII”. Identify yourself to the remote FTP server. If the password is not specified and the server requires it, <code>ftp</code> prompts the user for it (after disabling local echo). If an account field is not specified, and the FTP server requires it, the user is prompted for it. If an account field is specified, an account command is relayed to the remote server after the login sequence is completed if the remote server did not require it for logging in. Unless <code>ftp</code> is invoked with “auto-login” disabled, this process is done automatically on initial connection to the FTP server.
<code>verbose</code>	Toggles verbose mode. In verbose mode, all responses from the FTP server are displayed to the user. In addition, if verbose mode is on, when a file transfer completes, statistics regarding the efficiency of the transfer are reported. By default, verbose mode is on if <code>ftp</code> 's commands are coming from a terminal, and off otherwise.
<code>? [<i>command</i>]</code>	A synonym for <code>help</code> .

Command arguments which have embedded spaces can be quoted with quote (") marks.

If any command argument which is not indicated as being optional is not specified, `ftp` prompts for that argument.

Aborting A File Transfer

To abort a file transfer, use the terminal interrupt key. Sending transfers is immediately halted. Receiving transfers are halted by sending an FTP protocol ABOR command to the remote server, and discarding any further data received. The speed at which this is accomplished depends upon the remote server's support for ABOR processing. If the remote server does not support the ABOR command, an `ftp>` prompt does not appear until the remote server has completed sending the requested file.

The terminal interrupt key sequence is ignored when `ftp` has completed any local processing and is awaiting a reply from the remote server. A long delay in this mode can result from the ABOR processing described above, or from unexpected behavior by the remote server, including violations of the `ftp` protocol. If the delay results from unexpected remote server behavior, the local `ftp` program must be killed by hand.

File Naming Conventions

Local files specified as arguments to `ftp` commands are processed according to the following rules.

- 1) If the file name `-` is specified, the standard input (for reading) or standard output (for writing) is used.
- 2) If the first character of the file name is `|`, the remainder of the argument is interpreted as a shell command. `ftp` then forks a shell, using `popen(3C)` with the argument supplied, and reads (writes) from the standard output (standard input) of that shell. If the shell command includes SPACE characters, the argument must be quoted; for example, `| ls -lt`. A particularly useful example of this mechanism is: `"dir | more"`.
- 3) Failing the above checks, if globbing is enabled, local file names are expanded according to the rules used in the `sh(1)`; see the `glob` command. If the `ftp` command expects a single local file (for example, `put`), only the first filename generated by the globbing operation is used.
- 4) For `mget` commands and `get` commands with unspecified local file names, the local filename is the remote filename, which can be altered by a `case`, `nt rans`, or `nmap` setting. The resulting filename can then be altered if `runique` is on.
- 5) For `mput` commands and `put` commands with unspecified remote file names, the remote filename is the local filename, which can be altered by a `nt rans` or `nmap` setting. The resulting filename can then be altered by the remote server if `sunique` is on.

File Transfer Parameters

The FTP specification specifies many parameters which can affect a file transfer.

The “representation type” can be one of “network ASCII”, “EBCDIC”, “image”, or “local byte size” with a specified byte size (for PDP-10’s and PDP-20’s mostly). The “network ASCII” and “EBCDIC” types have a further subtype which specifies whether vertical format control (NEWLINE characters, form feeds, and so on) are to be passed through (“non-print”), provided in TELNET format (“TELNET format controls”), or provided in ASA (FORTRAN) (“carriage control (ASA)”) format. `ftp` supports the “network ASCII” (subtype “non-print” only) and “image” types, plus “local byte size” with a byte size of 8 for communicating with TENEX machines.

The “file structure” can be one of `file` (no record structure), `record`, or `page`. `ftp` supports only the default value, which is `file`.

The “transfer mode” can be one of `stream`, `block`, or `compressed`. `ftp` supports only the default value, which is `stream`.

用法

See [largefile\(5\)](#) for the description of the behavior of `ftp` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

The `ftp` command is IPv6-enabled. See [ip6\(7P\)](#).

文件 ~/ .netrc

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/ftp
CSI	enabled

另请参见 [ls\(1\)](#), [rcp\(1\)](#), [sh\(1\)](#), [tar\(1\)](#), [popen\(3C\)](#), [ftp\(4\)](#), [ftputers\(4\)](#), [mech\(4\)](#), [netrc\(4\)](#), [attributes\(5\)](#), [largefile\(5\)](#), [ip6\(7P\)](#)

Allman, M., Ostermann, S., and Metz, C. RFC 2428, FTP Extensions for IPv6 and NAT's. The Internet Society. September 1998.

Lunt, S. J. RFC 2228, FTP Security Extensions. Internet Draft. November 1993.

Postel, Jon, and Joyce Reynolds. RFC 959, File Transfer Protocol (FTP). Network Information Center. October 1985.

Piscitello, D. RFC 1639, FTP Operation Over Big Address Records (FOOBAR). Network Working Group. June 1994.

附注 Failure to log in can arise from an explicit denial by the remote FTP server because the account is listed in /etc/ftputers. See [ftputers\(4\)](#).

Correct execution of many commands depends upon proper behavior by the remote server.

An error in the treatment of carriage returns in the 4.2 BSD code handling transfers with a “representation type” of “network ASCII” has been corrected. This correction can result in incorrect transfers of binary files to and from 4.2 BSD servers using a “representation type” of “network ASCII”. Avoid this problem by using the “image” type.

引用名	gcore – get core images of running processes
用法概要	gcore [-pgF] [-o <i>filename</i>] [-c <i>content</i>] <i>process-id</i> ...
描述	The gcore utility creates a core image of each specified process. By default, the name of the core image file for the process whose process ID is <i>process-id</i> is <i>core.process-id</i> .
选项	The following options are supported: <ul style="list-style-type: none"> -c <i>content</i> Produces core image files with the specified content. The content description uses the same tokens as in coreadm(1M). The -c option does not apply to cores produced due to the -p or -g flags. -F Force. Grabs the target process even if another process has control. -g Produces core image files in the global core file repository with the global content as configured by coreadm(1M). The command fails if the user does not have permissions to the global core file repository. -o <i>filename</i> Substitutes <i>filename</i> in place of core as the first part of the name of the core image files. <i>filename</i> can contain the same tokens to be expanded as the paths in coreadm(1M). -p Produces a core image file in the process-specific location with the process-specific content for each process as configured by coreadm(1M). The command fails if the user does not have permissions to the per-process core file repository.
操作数	The following operand is supported: <p><i>process-id</i> process ID</p>
用法	Caution should be exercised when using the -F flag. Imposing two controlling processes on one victim process can lead to chaos. Safety is assured only if the primary controlling process, typically a debugger, has stopped the victim process and the primary controlling process is doing nothing at the moment of application of the proc tool in question.
退出状态	The following exit values are returned: <ul style="list-style-type: none"> 0 On success. non-zero On failure, such as non-existent process ID.
文件	<i>core.process-id</i> core images
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	See below.

The command syntax is Committed. The Output Formats are Uncommitted.

另请参见

[kill\(1\)](#), [coreadm\(1M\)](#), [setrlimit\(2\)](#), [core\(4\)](#), [proc\(4\)](#), [attributes\(5\)](#)

附注

gcore is unaffected by the [setrlimit\(2\)](#) system call using the RLIMIT_CORE value.

引用名	gencat – generate a formatted message catalog
用法概要	gencat <i>catfile msgfile...</i>
描述	<p>The gencat command merges the message text source file(s) <i>msgfile</i> into a formatted message database <i>catfile</i>. The database <i>catfile</i> is created if it does not already exist. If <i>catfile</i> does exist, its messages are included in the new <i>catfile</i>. If set and message numbers collide, the new message-text defined in <i>msgfile</i> replaces the old message text currently contained in <i>catfile</i>. The message text source file (or set of files) input to gencat can contain either set and message numbers or simply message numbers, in which case the set NL_SETD (see nl_types.h(3HEAD)) is assumed.</p>
Message Text Source File Format	<p>The format of a message text source file is defined as follows. Note that the fields of a message text source line are separated by a single ASCII space or tab character. Any other ASCII spaces or tabs are considered as part of the subsequent field.</p> <p><i>\$set n comment</i> Where <i>n</i> specifies the set identifier of the following messages until the next <i>\$set</i>, <i>\$delset</i>, or end-of-file appears. <i>n</i> must be a number in the range (1–{NL_SETMAX}). Set identifiers within a single source file need not be contiguous. Any string following the set identifier is treated as a comment. If no <i>\$set</i> directive is specified in a message text source file, all messages are located in the default message set NL_SETD.</p> <p><i>\$delset n comment</i> Deletes message set <i>n</i> from an existing message catalog. Any string following the set number is treated as a comment. (<i>Note</i>: if <i>n</i> is not a valid set it is ignored.)</p> <p><i>\$comment</i> A line beginning with a dollar symbol \$ followed by an ASCII space or tab character is treated as a comment.</p> <p><i>m message-text</i> The <i>m</i> denotes the message identifier, a number in the range (1–{NL_MSGMAX}). The <i>message-text</i> is stored in the message catalog with the set identifier specified by the last <i>\$set</i> directive, and with message identifier <i>m</i>. If the <i>message-text</i> is empty, and an ASCII space or tab field separator is present, an empty string is stored in the message catalog. If a message source line has a message number, but neither a field separator nor <i>message-text</i>, the existing message with that number (if any) is deleted from the catalog. Message identifiers need not be contiguous. The length of <i>message-text</i> must be in the range (0–{NL_TEXTMAX}).</p> <p><i>\$quote c</i> This line specifies an optional quote character <i>c</i>, which can be used to surround <i>message-text</i> so that trailing spaces or null (empty) messages are visible in a message source line. By default, or if an empty <i>\$quote</i> directive is supplied, no quoting of <i>message-text</i> will be recognized.</p>

Empty lines in a message text source file are ignored.

Text strings can contain the special characters and escape sequences defined in the following table:

Description	Symbol	Sequence
newline	NL(LF)	\n
horizontal tab	HT	\t
vertical tab	VT	\v
backspace	BS	\b
carriage return	CR	\r
form feed	FF	\f
backslash	\	\\
bit pattern	ddd	\ddd

The escape sequence `\ddd` consists of backslash followed by 1, 2 or 3 octal digits, which are taken to specify the value of the desired character. If the character following a backslash is not one of those specified, the backslash is ignored.

Backslash followed by an ASCII newline character is also used to continue a string on the following line. Thus, the following two lines describe a single message string:

```
1 This line continues \
to the next line
```

which is equivalent to:

```
1 This line continues to the next line
```

操作数

The following operands are supported:

catfile A path name of the formatted message catalog. If `-` is specified, standard output is used.

msgfile A path name of a message text source file. If `-` is specified for an instance of *msgfile*, standard input is used. The format of message text source files is defined in Message Text Source File Format.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `gencat`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态 The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale
CSI	enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [mkmsgs\(1\)](#), [catgets\(3C\)](#), [catopen\(3C\)](#), [gettxt\(3C\)](#), [nl_types.h\(3HEAD\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名	geniconvtbl – 生成 iconv 代码转换表
用法概要	geniconvtbl [-fnq] [-p <i>preprocessor</i>] [-W <i>arg</i>] [-D <i>name</i>] [-D <i>name=def</i>] [-I <i>directory</i>] [-U <i>name</i>] [<i>infile</i>]...
描述	geniconvtbl 实用程序接受在纯文本文件中定义的代码转换规则并编写可用于支持用户定义的 iconv 代码转换的代码转换二进制表文件（有关 iconv 代码转换的更多详细信息，请参见 iconv(1) 和 iconv(3C) ）。
选项	支持以下选项： <ul style="list-style-type: none"> -f 如果输出文件已存在，则覆盖该输出文件。 -n 不生成输出文件。在检查输入文件的内容时这比较有用。 -p <i>preprocessor</i> 使用指定的 <i>preprocessor</i> 而不是使用缺省的预处理程序 <code>/usr/lib/cpp</code>。 -q 静默选项。此选项抑制警告和错误消息。 -W <i>arg</i> 将参数 <i>arg</i> 传递到预处理程序。如果多次指定了此选项，会将所有参数传递到预处理程序。 -D<i>name</i> -D<i>name=def</i> -I<i>directory</i> -U<i>name</i> <code>geniconvtbl</code> 可识别这些选项并将它们及其参数传递到预处理程序。
操作数	支持下列操作数： <ul style="list-style-type: none"> <i>infile</i> 输入文件的路径名称。如果未指定输入文件，<code>geniconvtbl</code> 将从标准输入流中读取输入。如果需要，用户可以指定多个输入文件。
输出	<p>如果从标准输入流中读取输入，<code>geniconvtbl</code> 会将输出写入到标准输出流中。如果指定了一个或多个输入文件，<code>geniconvtbl</code> 会从每个输入文件中读取输入并将输出写入到相应的输出文件中。每个输出文件名称将与相应的输入文件相同，且文件名后缀为 <code>.bt</code>。</p> <p>在使用 iconv(1) 和 iconv(3C) 中的代码转换之前，必须将生成的输出文件移动到以下目录中：</p> <pre style="margin-left: 2em;">/usr/lib/iconv/geniconvtbl/binarytables/</pre> <p>输出文件名称应该以一个或多个可打印的 ASCII 字符开头作为 "fromcode" 名称，后跟一个百分比符号 (%)，然后跟有一个或多个可打印的 ASCII 字符作为 "tocode" 名称，最后以后缀 ".bt" 结尾。"fromcode" 和 "tocode" 名称用于标识 iconv(1) 和 iconv_open(3C) 中的 iconv 代码转换。应将正确命名的输出文件放置在 <code>/usr/lib/iconv/geniconvtbl/binarytables/</code> 目录中。</p>

示例

示例 1 生成 iconv 代码转换二进制表

以下示例生成输出文件名称为 `convertA2B.bt` 的代码转换二进制表：

```
example% geniconvtbl convertA2B
```

示例 2 生成多个 iconv 代码转换二进制表

以下示例生成两个代码转换二进制表，其输出文件分别为 `test1.bt` 和 `test2.bt`：

```
example% geniconvtbl test1 test2
```

示例 3 使用另一个预处理程序

在指定的预处理程序处理了输入文件后，以下示例会生成代码转换二进制表：

```
example% geniconvtbl -p /opt/SUNWspro/bin/cc -W -E convertB2A
```

示例 4 放置二进制表

要使用在上面的第一个示例中创建的二进制表作为 "fromcode" ABC 到 "tocode" DEF 的转换引擎，首先需成为超级用户，然后重命名该二进制表，并将其按如下方式放置：

```
example# mv convertA2B.bt \  
      /usr/lib/iconv/geniconvtbl/binarytables/ABC%DEF.bt
```

示例 5 提供修改的 ISO8859-1 到 UTF-8 代码转换

编写定义了代码转换的 `geniconvtbl` 源文件。例如，您可以将 `/usr/lib/iconv/geniconvtbl/srcs/ISO8859-1_to_UTF-8.src` 复制到您的目录中，然后在该源文件中进行必要的更改。在进行修改后，生成二进制表：

```
example% geniconvtbl ISO8859-1_to_UTF-8.src
```

作为超级用户，将生成的具有唯一名称的二进制表放置在 `iconv_open(3C)` 可以在其中找到该二进制表的系统目录中：

```
example su  
Password:  
example% cp ISO8859-1_to_UTF-8.bt \  
      /usr/lib/iconv/geniconvtbl/binarytables/my-iso-8859-1%utf-8.bt
```

之后，您可以执行 `iconv` 代码转换。例如：

```
example% iconv -f my-iso-8859-1 -t utf-8 testfile.txt
```

环境变量

有关影响 `geniconvtbl` 执行的环境变量 `LANG` 和 `LC_CTYPE` 的描述，请参见 [environ\(5\)](#)。

退出状态

将返回以下退出值：

- 0 没有发生任何错误，输出文件已成功创建。
- 1 未正确使用命令行选项，或指定了未知的命令行选项。

- 2 指定的输入或输出文件无效。
- 3 输入文件中的转换规则未正确定义。
- 4 已经达到了输入文件的转换规则限制。请参见 [geniconvtbl\(4\)](#) 的“附注”部分。
- 5 没有更多的系统资源错误。
- 6 内部错误。

文件 `/usr/lib/iconv/geniconvtbl/binarytables/*.bt`
转换二进制表

`/usr/lib/iconv/geniconvtbl/srcs/*`
供用户参考的转换源文件

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [cpp\(1\)](#)、[iconv\(1\)](#)、[iconv\(3C\)](#)、[iconv_close\(3C\)](#)、[iconv_open\(3C\)](#)、[geniconvtbl\(4\)](#)、[attributes\(5\)](#)

《*Solaris Internationalization Guide for Developers*》

附注 所生成的并正确放置的输出文件 (`/usr/lib/iconv/geniconvtbl/binarytables/*.bt`) 可用于 32 位和 64 位环境。

引用名 genmsg – generate a message source file by extracting messages from source files

用法概要 genmsg [-abdfrintx] [-c *message-tag*] [-g *project-file*]
 [-l *project-file*] [-m *prefix*] [-M *suffix*]
 [-o *message-file*] [-p *preprocessor*] [-s *set-tags*] *file...*

描述 The genmsg utility extracts message strings with calls to [catgets\(3C\)](#) from source files and writes them in a format suitable for input to [gencat\(1\)](#).

Invocation genmsg reads one or more input files and, by default, generates a message source file whose name is composed of the first input file name with .msg. If the -o option is specified, genmsg uses the option argument for its output file.

<i>Command</i>	<i>Output File</i>
genmsg prog.c	prog.c.msg
gensmg main.c util.c tool.c	main.c.msg
genmsg -o prog.msg mail.c util.c	prog.msg

genmsg also allows you to invoke a preprocessor to solve the dependencies of macros and define statements for the [catgets\(3C\)](#) calls.

Auto Message Numbering genmsg replaces message numbers with the calculated numbers based upon the project file if the message numbers are -1, and it generates copies of the input files with the new message numbers and a copy of the project file with the new maximum message numbers.

A project file is a database that stores a list of set numbers with their maximum message numbers. Each line in a project file is composed of a set number and its maximum message number:

Set_number *Maximum_message_number*

In a project file, a line beginning with a number sign (#) or an ASCII space is considered as a comment and ignored.

genmsg also has the reverse operation to replace all message numbers with -1.

Comment Extraction genmsg allows you to comment about messages and set numbers to inform the translator how the messages should be translated. It extracts the comment, which is surrounded with the comment indicators and has the specified tag inside the comment, from the input file and writes it with a dollar (\$) prefix in the output file. genmsg supports the C and C++ comment indicators, '/*', '*/', and '//'.

Testing genmsg generates two kinds of messages for testing, prefixed messages and long messages. Prefixed messages allow you to check that your program is retrieving the messages from the message catalog. Long messages allow you to check the appearance of your window program's initial size and position.

选项

The following options are supported:

- a Append the output into the message file *message-file* that is specified by the -o option. If two different messages that have the same set and message number are found, the message in the specified message file is kept and the other message in the input file is discarded.
- b Place the extracted comment after the corresponding message in the output file. This option changes the placement behavior of the -s or -c option.
- c *message-tag* Extract message comments having *message-tag* inside them from the input files and write them with a '\$' prefix as a comment in the output file.
- d Include an original text of a message as a comment to be preserved along with its translations. With this option, the translator can see the original messages even after they are replaced with their translations.
- f Overwrite the input files and the project file when used with the -l or -r option. With the -r option, genmsg overwrites only the input files.
- g *project-file* Generate *project-file* that has a list of set numbers and their maximum message numbers in the input files.
- l *project-file* Replace message numbers with the calculated numbers based upon *project-file* if the message numbers are -1 in the input files, and then generate copies of the input files with the new message numbers and a copy of *project-file* with the new maximum message numbers. If *project-file* is not found, genmsg uses the maximum message number in the input file as a base number and generates *project-file*.
- m *prefix* Fill in the message with *prefix*. This option is useful for testing.
- M *suffix* Fill in the message with *suffix*. This option is useful for testing.
- n Add comment lines to the output file indicating the file name and line number in the input files where each extracted string is encountered.
- o *message-file* Write the output to *message-file*.
- p *preprocessor* Invoke *preprocessor* to preprocess macros and define statements for the [catgets\(3C\)](#) calls. genmsg first invokes the option argument as a preprocessor and then starts the normal process against the output from the preprocessor. genmsg initiates this process for all the input files.

-r	Replace message numbers with -1. This is the reverse operation of the -l option.
-s <i>set-tag</i>	Extract set number comments having <i>set-tag</i> inside them from the input files and write them with a '\$' prefix as a comment in the output file. If multiple comments are specified for one set number, the first one is extracted and the rest of them are discarded.
-t	Generate a message that is three times as long as the original message. This option is useful for testing.
-x	Suppress warning messages about message and set number range checks and conflicts.
<i>file</i>	An input source file.

操作数

示例

示例 1 Assigning Message Numbers and Generating New Files

Suppose that you have the following source and project files:

```
example% cat test.c
printf(catgets(catfd, 1, -1, "line too long\n"));
printf(catgets(catfd, 2, -1, "invalid code\n"));
```

```
example% cat proj
1 10
2 20
```

The command

```
example% genmsg -l proj test.c
```

would assign the calculated message numbers based upon `proj` and generate the following files:

```
test.c.msg    Message file
proj.new      Updated project file
test.c.new    New source file
```

```
example% cat test.c.msg
$quote "
$set 1
11 "line too long\n"
$set 2
21 "invalid code\n"
```

```
example% cat proj.new
1 11
2 21
```

示例 1 Assigning Message Numbers and Generating New Files (续)

```
example% cat test.c.new
printf(catgets(catfd, 1, 11, "line too long\n"));
printf(catgets(catfd, 2, 21, "invalid code\n"));
```

示例 2 Extracting Comments Into a File

The command

```
example% genmsg -s SET -c MSG test.c
example% cat test.c
/* SET: tar messages */
/* MSG: don't translate "tar". */
catgets(catfd, 1, 1, "tar: tape write error");
// MSG: don't translate "tar" and "-I".
catgets(catfd, 1, 2, "tar: missing argument for -I flag");
```

would extract the comments and write them in the following output file:

```
example% cat test.c.msg
$ /* SET: tar messages */
$set 1
$ /* MSG: don't translate "tar". */
1 "tar: tape write error"
$ // MSG: don't translate "tar" and "-I".
2 "tar: missing argument for -I flag"
```

示例 3 Generating Test Messages

The following command:

```
example% genmsg -m PRE: -M :FIX test.c
```

might generate the following messages for testing:

```
example% cat test.c.msg
1 "PRE:OK:FIX"
2 "PRE:Cancel:FIX"
```

示例 4 Parsing a Macro and Writing the Extracted Messages

Given the following input:

```
example% cat example.c
#include <nl_types.h>
#define MSG1 "message1"
#define MSG2 "message2"
#define MSG3 "message3"
#define MSG(n) catgets(catd, 1, n, MSG ## n)
```


示例 4 Parsing a Macro and Writing the Extracted Messages (续)

```
void
main(int argc, char **argv)
{
    nl_catd catd = catopen(argv[0], NL_CAT_LOCALE);
    (void) printf("%s0\n", MSG(1));
    (void) printf("%s0\n", MSG(2));
    (void) printf("%s0\n", MSG(3));
    (void) catclose(catd);
}
```

The following command:

```
example% genmsg -p "cc -E" -o example.msg example.c
```

would parse the MSG macros and write the extracted messages in `example.msg`.

示例 5 Assigning Calculated Message Numbers

Suppose that you have the following header, source, and project files:

```
example% cat ../inc/msg.h
#define WARN_SET 1
#define ERR_SET 2
#define WARN_MSG(id, msg) catgets(catd, WARN_SET, (id), (msg))
#define ERR_MSG(id, msg) catgets(catd, ERR_SET, (id), (msg))
example% example.c
#include "msg.h"
printf("%s, WARN_MSG(-1, "Warning error");
printf("%s, ERR_MSG(-1, "Fatal error");
example % proj
1 10
2 10
```

The command

```
example% genmsg -f -p "cc -E -I../inc" -l proj \
-o example.msg example.c
```

would assign each of the -1 message numbers a calculated number based upon `proj` and would overwrite the results to `example.c` and `proj`. Also, this command writes the extracted messages in `example.msg`.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `genmsg`: `LC_MESSAGES` and `NLSPATH`.

退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale

另请参见

[gencat\(1\)](#), [catgets\(3C\)](#), [catopen\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注

genmsg does not handle pointers or variables in the [catgets\(3C\)](#) call. For example:

```
const int set_num = 1;
extern int msg_num(const char *);
const char *msg = "Hello";
catgets(catd, set_num, msg_num(msg), msg);
```

When the auto message numbering is turned on with a preprocessor, if there are multiple -1's in the [catgets\(3C\)](#) line, genmsg replaces all of the -1's in the line with a calculated number. For example, given the input:

```
#define MSG(id, msg) catgets(catd, 1, (id), (msg))
if (ret == -1) printf("%s, MSG(-1, "Failed");
```

the command

```
genmsg -l proj -p "cc -E"
```

would produce:

```
#define MSG(id, msg) catgets(catd, 1, (id), (msg))
if (ret == 1) printf("%s, MSG(1, "Failed");
```

The workaround would be to split it into two lines as follows:

```
if (ret == -1)
    printf("%s, MSG(-1, "Failed");
```

引用名	getconf – get configuration values									
用法概要	<pre> /usr/bin/getconf [-v <i>specification</i>] <i>system_var</i> /usr/bin/getconf [-v <i>specification</i>] <i>path_var</i> <i>pathname</i> /usr/bin/getconf -a /usr/xpg4/bin/getconf [-v <i>specification</i>] <i>system_var</i> /usr/xpg4/bin/getconf [-v <i>specification</i>] <i>path_var</i> <i>pathname</i> /usr/xpg4/bin/getconf -a /usr/xpg6/bin/getconf [-v <i>specification</i>] <i>system_var</i> /usr/xpg6/bin/getconf [-v <i>specification</i>] <i>path_var</i> <i>pathname</i> /usr/xpg6/bin/getconf -a </pre>									
描述	<p>In the first synopsis form, the <code>getconf</code> utility writes to the standard output the value of the variable specified by <i>system_var</i>, in accordance with <i>specification</i> if the <code>-v</code> option is used.</p> <p>In the second synopsis form, <code>getconf</code> writes to the standard output the value of the variable specified by <i>path_var</i> for the path specified by <i>pathname</i>, in accordance with <i>specification</i> if the <code>-v</code> option is used.</p> <p>In the third synopsis form, <code>conf</code> writes to the standard output the names of the current system configuration variables.</p> <p>The value of each configuration variable is determined as if it were obtained by calling the function from which it is defined to be available. The value reflects conditions in the current operating environment.</p>									
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> <code>-a</code> Writes the names of the current system configuration variables to the standard output. <code>-v <i>specification</i></code> Gives the specification which governs the selection of values for configuration variables. 									
操作数	<p>The following operands are supported:</p> <p><i>path_var</i> A name of a configuration variable whose value is available from the pathconf(2) function. All of the values in the following table are supported:</p> <table border="0" style="margin-left: 20px; width: 100%;"> <tr> <td style="padding-right: 40px;">LINK_MAX</td> <td style="padding-right: 40px;">NAME_MAX</td> <td>_POSIX_CHOWN_RESTRICTED</td> </tr> <tr> <td>MAX_CANON</td> <td>PATH_MAX</td> <td>_POSIX_NO_TRUNC</td> </tr> <tr> <td>MAX_INPUT</td> <td>PIPE_BUF</td> <td>_POSIX_VDISABLE</td> </tr> </table>	LINK_MAX	NAME_MAX	_POSIX_CHOWN_RESTRICTED	MAX_CANON	PATH_MAX	_POSIX_NO_TRUNC	MAX_INPUT	PIPE_BUF	_POSIX_VDISABLE
LINK_MAX	NAME_MAX	_POSIX_CHOWN_RESTRICTED								
MAX_CANON	PATH_MAX	_POSIX_NO_TRUNC								
MAX_INPUT	PIPE_BUF	_POSIX_VDISABLE								

pathname A path name for which the variable specified by *path_var* is to be determined.

system_var A name of a configuration variable whose value is available from `confstr(3C)` or `sysconf(3C)`. All of the values in the following table are supported:

ARG_MAX	BC_BASE_MAX
BC_DIM_MAX	BC_SCALE_MAX
BC_STRING_MAX	CHAR_BIT
CHARCLASS_NAME_MAX	CHAR_MAX
CHAR_MIN	CHILD_MAX
CLK_TCK	COLL_WEIGHTS_MAX
CS_PATH	EXPR_NEST_MAX
HOST_NAME_MAX	INT_MAX
INT_MIN	LFS64_CFLAGS
LFS64_LDFLAGS	LFS64_LIBS
LFS64_LINTFLAGS	LFS_CFLAGS
LFS_LDFLAGS	LFS_LIBS
LFS_LINTFLAGS	LINE_MAX
LONG_BIT	LONG_MAX
LONG_MIN	MB_LEN_MAX
NGROUPS_MAX	NL_ARGMAX
NL_LANGMAX	NL_MSGMAX
NL_NMAX	NL_SETMAX
NL_TEXTMAX	NZERO
OPEN_MAX	POSIX2_BC_BASE_MAX
POSIX2_BC_DIM_MAX	POSIX2_BC_SCALE_MAX
POSIX2_BC_STRING_MAX	POSIX2_C_BIND
POSIX2_C_DEV	POSIX2_CHAR_TERM
POSIX2_COLL_WEIGHTS_MAX	POSIX2_C_VERSION
POSIX2_EXPR_NEST_MAX	POSIX2_FORT_DEV
POSIX2_FORT_RUN	POSIX2_LINE_MAX

POSIX2_LOCALEDEF	POSIX2_RE_DUP_MAX
POSIX2_SW_DEV	POSIX2_SYMLINKS
POSIX2_UPE	POSIX2_VERSION
POSIX_ALLOC_SIZE_MIN	POSIX_REC_INCR_XFER_SIZE
POSIX_REC_MAX_XFER_SIZE	POSIX_REC_MIN_XFER_SIZE
POSIX_REC_XFER_ALIGN	POSIX_V6_ILP32_OFF32
POSIX_V6_ILP32_OFF32_CFLAGS	POSIX_V6_ILP32_OFF32_LDFLAGS
POSIX_V6_ILP32_OFF32_LIBS	POSIX_V6_ILP32_OFFBIG
POSIX_V6_ILP32_OFFBIG_CFLAGS	POSIX_V6_ILP32_OFFBIG_LDFLAGS
POSIX_V6_ILP32_OFFBIG_LIBS	POSIX_V6_LP64_OFF64
POSIX_V6_LP64_OFF64_CFLAGS	POSIX_V6_LP64_OFF64_LDFLAGS
POSIX_V6_LP64_OFF64_LIBS	POSIX_V6_LPBIG_OFFBIG
POSIX_V6_LPBIG_OFFBIG_CFLAGS	POSIX_V6_LPBIG_OFFBIG_LDFLAGS
POSIX_V6_LPBIG_OFFBIG_LIBS	POSIX_V6_WIDTH_RESTRICTED_ENVS
SYMLINK_MAX	SYMLOOP_MAX
_POSIX2_BC_BASE_MAX	_POSIX2_BC_DIM_MAX
_POSIX2_BC_SCALE_MAX	_POSIX2_BC_STRING_MAX
_POSIX2_CHARCLASS_NAME_MAX	_POSIX2_CHAR_TERM
_POSIX2_COLL_WEIGHTS_MAX	_POSIX2_C_BIND
_POSIX2_C_DEV	_POSIX2_C_VERSION
_POSIX2_EXPR_NEST_MAX	_POSIX2_FORT_DEV
_POSIX2_FORT_RUN	_POSIX2_LINE_MAX
_POSIX2_LOCALEDEF	_POSIX2_PBS
_POSIX2_PBS_ACCOUNTING	_POSIX2_PBS_CHECKPOINT
_POSIX2_PBS_LOCATE	_POSIX2_PBS_MESSAGE
_POSIX2_PBS_TRACK	_POSIX2_RE_DUP_MAX
_POSIX2_SW_DEV	_POSIX2_UPE
_POSIX2_VERSION	_POSIX_ADVISORY_INFO
_POSIX_AIO_LISTIO_MAX	_POSIX_AIO_MAX

_POSIX_ARG_MAX	_POSIX_ASYNC_IO
_POSIX_BARRIERS	_POSIX_CHILD_MAX
_POSIX_CLOCKRES_MIN	_POSIX_CLOCK_SELECTION
_POSIX_CPUTIME	_POSIX_DELAYTIMER_MAX
_POSIX_HOST_NAME_MAX	_POSIX_IPV6
_POSIX_JOB_CONTROL	_POSIX_LINK_MAX
_POSIX_LOGIN_NAME_MAX	_POSIX_MAX_CANON
_POSIX_MAX_INPUT	_POSIX_MONOTONIC_CLOCK
_POSIX_MQ_OPEN_MAX	_POSIX_MQ_PRIO_MAX
_POSIX_NAME_MAX	_POSIX_NGROUPS_MAX
_POSIX_OPEN_MAX	_POSIX_PATH_MAX
_POSIX_PIPE_BUF	_POSIX_PRIO_IO
_POSIX_RAW_SOCKETS	_POSIX_READER_WRITER_LOCKS
_POSIX_REGEX	_POSIX_RE_DUP_MAX
_POSIX_RTSIG_MAX	_POSIX_SAVED_IDS
_POSIX_SEM_NSEMS_MAX	_POSIX_SEM_VALUE_MAX
_POSIX_SHELL	_POSIX_SIGQUEUE_MAX
_POSIX_SPAWN	_POSIX_SPIN_LOCKS
_POSIX_SPORADIC_SERVER	_POSIX_SSIZE_MAX
_POSIX_SS_REPL_MAX	_POSIX_STREAM_MAX
_POSIX_SYMLINK_MAX	_POSIX_SYMLOOP_MAX
_POSIX_SYNC_IO	_POSIX_THREAD_ATTR_STACKADDR
_POSIX_THREAD_ATTR_STACKSIZE	_POSIX_THREAD_CPUTIME
_POSIX_THREAD_DESTRUCTOR_ITERATIONS	_POSIX_THREAD_KEYS_MAX
_POSIX_THREAD_PRIORITY_SCHEDULING	_POSIX_THREAD_PRIO_INHERIT
_POSIX_THREAD_PRIO_PROTECT	_POSIX_THREAD_PROCESS_SHARED
_POSIX_THREAD_SAFE_FUNCTIONS	_POSIX_THREAD_SPORADIC_SERVER
_POSIX_THREAD_THREADS_MAX	_POSIX_TIMEOUTS
_POSIX_TIMER_MAX	_POSIX_TRT_POSIX_TIMER_MAX

_POSIX_TIMESTAMP_RESOLUTION	
_POSIX_TRACE_EVENT_FILTER	_POSIX_TRACE_EVENT_NAME_MAX
_POSIX_TRACE_INHERIT	_POSIX_TRACE_LOG
_POSIX_TRACE_NAME_MAX	_POSIX_TRACE_SYS_MAX
_POSIX_TRACE_USER_EVENT_MAX	_POSIX_TTY_NAME_MAX
_POSIX_TYPED_MEMORY_OBJECTS	_POSIX_TZNAME_MAX
_POSIX_VERSION	_POSIX_V6_ILP32_OFF32
_POSIX_V6_ILP32_OFFBIG	_POSIX_V6_LP64_OFF64
_POSIX_V6_LPBIG_OFFBIG	_V6_ILP32_OFF32
_V6_ILP32_OFFBIG	_V6_LP64_OFF64
_V6_LPBIG_OFFBIG	RE_DUP_MAX
SCHAR_MAX	SCHAR_MIN
SHRT_MAX	SHRT_MIN
SSIZE_MAX	STREAM_MAX
TMP_MAX	TZNAME_MAX
UCHAR_MAX	UINT_MAX
ULONG_MAX	USHRT_MAX
WORD_BIT	XBS5_ILP32_OFF32
XBS5_ILP32_OFF32_CFLAGS	XBS5_ILP32_OFF32_LDFLAGS
XBS5_ILP32_OFF32_LIBS	XBS5_ILP32_OFF32_LINTFLAGS
XBS5_ILP32_OFFBIG	XBS5_ILP32_OFFBIG_CFLAGS
XBS5_ILP32_OFFBIG_LDFLAGS	XBS5_ILP32_OFFBIG_LIBS
XBS5_ILP32_OFFBIG_LINTFLAGS	XBS5_LP64_OFF64
XBS5_LP64_OFF64_CFLAGS	XBS5_LP64_OFF64_LDFLAGS
XBS5_LP64_OFF64_LIBS	XBS5_LP64_OFF64_LINTFLAGS
XBS5_LPBIG_OFFBIG	XBS5_LPBIG_OFFBIG_CFLAGS
XBS5_LPBIG_OFFBIG_LDFLAGS	XBS5_LPBIG_OFFBIG_LIBS
XBS5_LPBIG_OFFBIG_LINTFLAGS	_XOPEN_CRYPT
_XOPEN_ENH_I18N	_XOPEN_IOV_MAX

<code>_XOPEN_LEGACY</code>	<code>_XOPEN_NAME_MAX</code>
<code>_XOPEN_PATH_MAX</code>	<code>_XOPEN_SHM</code>
<code>_XOPEN_STREAMS</code>	<code>_XOPEN_VERSION</code>
<code>_XOPEN_XCU_VERSION</code>	<code>_XOPEN_XPG2</code>
<code>_XOPEN_XPG3</code>	<code>_XOPEN_XPG4</code>

The symbol `PATH` also is recognized, yielding the same value as the `confstr()` name value `CS_PATH`.

用法

See [largefile\(5\)](#) for the description of the behavior of `/usr/bin/getconf` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Writing the Value of a Variable

This example illustrates the value of `{NGROUPS_MAX}`:

```
example% getconf NGROUPS_MAX
```

示例 2 Writing the Value of a Variable for a Specific Directory

This example illustrates the value of `NAME_MAX` for a specific directory:

```
example% getconf NAME_MAX /usr
```

示例 3 Dealing with Unspecified Results

This example shows how to deal more carefully with results that might be unspecified:

```
if value=$(getconf PATH_MAX /usr); then
if [ "$value" = "undefined" ]; then
echo PATH_MAX in /usr is infinite.
else
echo PATH_MAX in /usr is $value.
fi
else
echo Error in getconf.
fi
```

For example:

```
sysconf(_SC_POSIX_C_BIND);

and

system("getconf POSIX2_C_BIND");
```

in a C program could give different answers. The `sysconf` call supplies a value that corresponds to the conditions when the program was either compiled or executed, depending on the implementation. The `system` call to `getconf` always supplies a value corresponding to

示例 3 Dealing with Unspecified Results (续)

conditions when the program is executed.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `getconf`: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 The specified variable is valid and information about its current state was written successfully.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[sh\(1\)](#), [pathconf\(2\)](#), [sysinfo\(2\)](#), [confstr\(3C\)](#), [sysconf\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

引用名 getfacl – 显示自主文件信息

用法概要 getfacl [-ad] file...

描述 对于表示常规文件、特殊文件或命名管道的每个参数，getfacl 实用程序会显示所有者、组和访问控制列表 (Access Control List, ACL)。对于每个目录参数，getfacl 会显示所有者、组、ACL 和/或缺省的 ACL。只有目录包含缺省的 ACL。

getfacl 实用程序可在不支持 ACL 的文件系统上执行。它将基于基本权限位报告 ACL。

如果未指定任何选项，getfacl 会显示文件名、文件所有者、文件组所有者、ACL 和缺省 ACL（如果存在）。

选项 支持以下选项：

-a 显示文件名、文件所有者、文件组所有者和文件的 ACL。

-d 显示文件名、文件所有者、文件组所有者和文件的缺省 ACL（如果存在）。

操作数 支持下列操作数：

file 常规文件、特殊文件或命名管道的路径名。

输出 ACL 输出的格式如下所示：

```
# file: filename
# owner: uid
# group: gid
user::perm
user:uid:perm
group::perm
group:gid:perm
mask:perm
other:perm
default:user::perm
default:user:uid:perm
default:group::perm
default:group:gid:perm
default:mask:perm
default:other:perm
```

在命令行上指定多个文件时，以一个空白行分隔每个文件的 ACL。

在执行访问权限检查时，ACL 条目将按其评估顺序显示。目录上可能存在的缺省 ACL 条目不会影响访问权限检查。

前三行显示文件名、文件所有者和文件组所有者。请注意，仅当指定了 -d 选项且文件没有缺省的 ACL 时，才会显示这三行。

不具有用户 ID 的 `user` 条目指示授予文件所有者的权限。一个或多个额外的用户条目指示授予指定用户的权限。

不具有组 ID 的 `group` 条目指示授予文件组所有者的权限。一个或多个额外的组条目指示授予指定组的权限。

`mask` 条目指示 ACL 掩码权限。这些权限是允许授予所有用户条目（文件所有者除外）和所有组条目（包括文件组所有者）的最大权限。这些权限会限制在其他条目中指定的权限。

`other` 条目指示已授予其他人员的权限。

只有对于目录才会存在 `default` 条目。这些条目指示已添加到在目录中创建的文件的缺省条目。

如果系统口令文件 `/etc/passwd` 中不存在任何 `uid` 条目，则 `uid` 是一个登录名或用户 ID。如果系统组文件 `/etc/group` 中不存在任何 `gid` 条目，则 `gid` 是一个组名或组 ID。`perm` 是一个由三个字符组成的字符串，这三个字符分别是以下表示单自主访问权限的字母：`r`（读取）、`w`（写入）、`x`（执行/搜索）或占位符 `-`。`perm` 按以下顺序显示：`rwX`。如果 ACL 条目未授予某个权限，则会显示占位符。

如果您使用 `chmod(1)` 命令更改含有 ACL 条目的文件的文件组所有者权限，则会同时将文件组所有者权限和 ACL 掩码更改为新的权限。请注意，对于文件中有其 ACL 条目的附加用户和组，新的 ACL 掩码权限可能会更改其有效权限。

为了指示 ACL 掩码限制 ACL 条目，`getfacl` 在该条目后面显示一个附加制表符、井号 (`#`) 和授予的实际权限。

示例

示例1 显示文件信息

以文件 `foo` 为例，该文件的 ACL 包含六个条目，命令

```
host% getfacl foo
```

将输出：

```
# file: foo
# owner: shea
# group: staff
user::rwX
user:spy: - - -
user:mookie:r - -
group::r - -
mask::rw -
other:: - - -
```

示例2 在执行 `chmod` 命令后显示信息

继续使用上例，在 `chmod` 之后发出了 `700 foo`：

```
host% getfacl foo
```

将输出：

```
# file: foo
# owner: shea
# group: staff
user::rwx
user:spy: - - -
user:mookie:r - -      #effective: - - -
group:: - - -
mask:: - - -
other:: - - -
```

示例3 当 ACL 包含缺省条目时显示信息

以目录 `do` 为例，该目录的 ACL 包含缺省条目，命令

```
host% getfacl -d do
```

将输出：

```
# file: do
# owner: shea
# group: staff
default:user::rwx
default:user:spy: - - -
default:user:mookie:r - -
default:group::r - -
default:mask:: - - -
default:other:: - - -
```

文件 `/etc/passwd` 系统口令文件

`/etc/group` 组文件

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed (已确定)

另请参见 [chmod\(1\)](#)、[ls\(1\)](#)、[setfacl\(1\)](#)、[acl\(2\)](#)、[aclsort\(3SEC\)](#)、[group\(4\)](#)、[passwd\(4\)](#)、[attributes\(5\)](#)

附注

getfacl 的输出必须处于正确格式，才能作为 setfacl -f 命令的输入。如果 getfacl 的输出被重定向到某个文件，则该文件可用作 setfacl 的输入。这样，用户可以轻松将一个文件的 ACL 分配给另一个文件。

引用名 getlabel - 显示文件的标签

用法概要 /usr/bin/getlabel [-sS] filename...

描述 getlabel 显示与每个 *filename* 关联的标签。未指定这些选项时，标签的输出格式会以缺省格式显示。

选项 -s 以短格式显示与 *filename* 关联的标签。
 -S 以长格式显示与 *filename* 关联的标签。

退出状态 getlabel 退出时返回下列值之一：

- 0 成功完成。
- 1 由于用法错误，未成功完成。
- 2 无法转换标签。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/trusted
接口稳定性	请参见下文。

命令行是 "Committed"（已确定）。输出为 "Not-an-Interface"（不是接口）。

另请参见 [setlabel\(1\)](#)、[m_label\(3TSOL\)](#)、[label_encodings\(4\)](#)、[attributes\(5\)](#)

附注 仅当系统配置有 Trusted Extensions 时，本手册页中介绍的功能才可用。

引用名 `getopt` – 解析命令选项

用法概要 `set -- 'getopt optstring $ *'`

描述 `getopts` 命令取代了 `getopt`。有关更多信息，请参见下文的“附注”部分。

`getopt` 可用于打断命令行中的选项以便于 shell 过程进行简单解析，还可用于检查合法选项。*optstring* 是一串可识别的选项字母；请参见 [getopt\(3C\)](#)。如果某个字母后面跟有冒号(:)，则该选项需要有一个参数，可以使用空格将其与该参数隔开，也可以不使用。特殊选项 `-` 用来限定选项的结尾。如果显式使用了该选项，则 `getopt` 会识别出该选项；否则，`getopt` 将生成该特殊选项；在任一情况下，`getopt` 都会将其放置在选项的结尾。shell 的位置参数 (`$1 $2 ...`) 已重置，以便每个选项以 `-` 开头并位于自己的位置参数中；还会将每个选项参数解析到其自己的位置参数中。

示例 示例1 为命令处理参数

以下代码片段显示了如何为可以接受选项 `-a` 或 `-b`，以及选项 `-o`(该选项需要一个参数) 的命令处理参数：

```
set -- 'getopt abo: $*'
if [ $? != 0 ]
then
    echo $USAGE
    exit 2
fi
for i in $*
do
    case $i in
    -a | -b)    FLAG=$i; shift;;
    -o)        OARG=$2; shift 2;;
    --)        shift; break;;
    esac
done
```

此代码将下列各个输入视为等效的：

```
cmd -aoarg filename1 filename2
cmd -a -o arg filename1 filename2
cmd -oarg -a filename1 filename2
cmd -a -oarg -- filename1 filename2
```

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
CSI	enabled (已启用)

另请参见 [Intro\(1\)](#)、[getopts\(1\)](#)、[getoptcv\(1\)](#)、[sh\(1\)](#)、[shell_builtins\(1\)](#)、[getopt\(3C\)](#)、[attributes\(5\)](#)

诊断 如果 `getopt` 遇到不包括在 *optstring* 中的选项字母，将会在标准错误上输出一条错误消息。

附注 在下一个主要发行版中将不支持 `getopt`。对于本发行版，提供了一个转换工具，即 `getoptcv`。有关更多信息，请参见 [getopts\(1\)](#) 和 [getoptcv\(1\)](#)。

重新扫描选项时，请将 `optind` 重置为 1。

`getopt` 不支持命令语法标准的规则 8 的部分内容（请参见 [Intro\(1\)](#)） - 允许一个选项之后的选项参数组以空格分隔并以引号括起来。例如，

```
cmd -a -b -o "xxx z yy" filename
```

不能正确处理。要更正此缺陷，应使用 `getopts` 命令，而不是 `getopt`。

如果接受选项参数的选项后面跟有一个值，该值与 *optstring* 中列出的其中一个选项相同（请参见前面的“示例”部分）但使用以下命令行：

```
cmd -o -a filename
```

则 `getopt` 会始终将其视为 `-o` 的选项参数；但不能将 `-a` 识别为选项。对于这种情况，此示例的中 `for` 循环将跳过 *filename* 参数。

引用名	getoptcv – 转换到 getopt 以解析命令选项
用法概要	<pre>/usr/lib/getoptcv [-b] filename /usr/lib/getoptcv</pre>
描述	<p><code>/usr/lib/getoptcv</code> 读取 <i>filename</i> 中的 shell 脚本，将其转换为使用 <code>getopts</code> 而不是 <code>getopt</code>，然后在标准输出中写入结果。</p> <p><code>getopts</code> 是一个内置 Bourne shell 命令，用于解析位置参数和检查有效选项。请参见 sh(1)。它支持命令语法标准的所有适用规则（请参见 Intro(1) 中的规则 3 至 10）。应该使用该命令来替代 <code>getopt</code> 命令。（请参见下文的“附注”部分。）shell 的内置 <code>getopts</code> 命令的语法为：</p> <pre>getopts optstring name [argument ...]</pre> <p><i>optstring</i> 必须包含使用 <code>getopts</code> 的命令可以识别的选项字母；如果字母后面跟有冒号 (:), 则该选项需要有一个参数，或一组参数，必须使用空格将其与参数分隔。</p> <p>每次被调用时，<code>getopts</code> 会将下一个选项放在 shell 变量 <i>name</i> 中，将下一个要处理的参数的索引放在 shell 变量 <code>OPTIND</code> 中。只要调用 shell 或 shell 脚本，就会将 <code>OPTIND</code> 初始化为 1。</p> <p>如果某个选项需要选项参数，<code>getopts</code> 会将其放置在 shell 变量 <code>OPTARG</code> 中。</p> <p>如果遇到非法选项，会将 ? 放置在 <i>name</i> 中。</p> <p>当遇到选项的结尾时，<code>getopts</code> 将以非零退出状态退出。可以使用特殊选项 <code>—</code> 来限定选项的结尾。</p> <p>缺省情况下，<code>getopts</code> 解析位置参数。如果在 (<i>argument ...</i>) <code>getopts</code> 命令行上指定了额外的参数 (<i>argument ...</i>)，<code>getopts</code> 改而对它们进行解析。</p> <p>这样，所有新命令将遵循 Intro(1) 中介绍的命令语法标准，它们应该使用 <code>getopts</code> 或 <code>getopt</code> 来解析位置参数和检查对该命令有效的选项（请参见下文的“附注”部分）。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -b 使转换的脚本可移植到 UNIX 系统的早期发行版。<code>/usr/lib/getoptcv</code> 修改 <i>filename</i> 中的 shell 脚本，使得在执行所得到的 shell 脚本时，由它在运行时确定是调用 <code>getopts</code> 还是 <code>getopt</code>。
示例	<p>示例 1 为命令处理参数</p> <p>以下 shell 程序片段显示了如何为可以接受选项 <code>-a</code> 或 <code>-b</code>，以及选项 <code>-o</code>（该选项需要一个选项参数）的命令处理参数：</p> <pre>while getopt abo: c do case \$c in</pre>

示例1 为命令处理参数 (续)

```

a | b)    FLAG=$c;;
o)       OARG=$OPTARG;;
\?)      echo $USAGE
         exit 2;;

esac

done
shift `expr $OPTIND - 1`

```

示例2 等效的代码表达式

此代码将下列各个输入视为等效的：

```

cmd -a -b -o "xxx z yy" filename
cmd -a -b -o "xxx z yy" -filename
cmd -ab -o xxx,z,yy filename
cmd -ab -o "xxx z yy" filename
cmd -o xxx,z,yy b a filename

```

环境变量

有关影响 `getopts` 执行的环境变量 `LC_CTYPE`、`LC_MESSAGES` 和 `NLSPATH` 的描述，请参见 [environ\(5\)](#)。

OPTIND `getoptcvt` 将该变量用作要处理的下一个参数的索引。

OPTARG 如果某个选项使用了参数，`getoptcvt` 将使用该变量来存储参数。

退出状态

将返回以下退出值：

- 0 找到了一个选项，可能是也可能不是由 *optstring* 指定的。
- >0 遇到选项的结尾，或发生错误。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
CSI	enabled (已启用)

另请参见

[Intro\(1\)](#)、[getopts\(1\)](#)、[sh\(1\)](#)、[shell_builtins\(1\)](#)、[getopt\(3C\)](#)、[attributes\(5\)](#)

诊断

如果 `getopts` 遇到不包括在 *optstring* 中的选项字母，将会在标准错误上输出一条错误消息。

附注

尽管在当前实现中，允许以下命令语法规则（请参见 [Intro\(1\)](#)）放宽，但也不应使用这些语法规则，因为系统的将来发行版可能不支持它们。如上面的“示例”部分所示，`-a` 和 `-b` 均为选项，选项 `-o` 需要一个选项参数。以下示例违反了规则 5：具有选项参数的选项不得与其他选项组合在一起：

```
example% cmd -abxxx filename
```

以下示例违反了规则 6：在接受选项参数的选项后面必须有空格：

```
example% cmd -ab oxxx filename
```

更改 shell 变量 `OPTIND` 的值或解析不同的参数集合可能会导致意外结果。

引用名	getopts – 解析实用程序选项
用法概要	<code>/usr/bin/getopts <i>optstring</i> <i>name</i> [<i>arg</i>...]</code>
sh	<code>getopts <i>optstring</i> <i>name</i> [<i>argument</i>]...</code>
ksh88	<code>getopts <i>optstring</i> <i>name</i> [<i>arg</i>]...</code>
ksh	<code>getopts [-a <i>name</i>] <i>optstring</i> <i>name</i> [<i>arg</i>]...</code>

描述

`/usr/bin/getopts` getopts 实用程序可用于从参数列表中检索选项和选项参数。

每次被调用时，`getopts` 实用程序都会将下一选项的值放入 `name` 操作数指定的 shell 变量中，将要处理的下一参数的索引放入 shell 变量 `OPTIND` 中。每当调用 shell 时，`OPTIND` 都将初始化为 1。

如果某个选项需要选项参数，`getopts` 实用程序会将其放置在 shell 变量 `OPTARG` 中。如果未找到选项，或找到的选项没有选项参数，则取消设置 `OPTARG`。

如果在需要选项字符的位置发现了 `optstring` 操作数中未包含的选项字符，由 `name` 指定的 shell 变量将设置为问号 (?) 字符。在此情况下，如果 `optstring` 中的第一个字符是冒号 (:)，则 shell 变量 `OPTARG` 将设置为所找到的选项字符，但不会有任何输出写入到标准错误；其他情况下，将取消设置 shell 变量 `OPTARG` 并将一条诊断信息写入到标准错误。此情况将被视为在将参数提供给调用应用程序时检测到的错误，而不是 `getopts` 处理中的错误。

如果缺少某个选项参数：

- 如果 `optstring` 的第一个字符是冒号，则由 `name` 指定的 shell 变量将设置为冒号字符，shell 变量 `OPTARG` 将设置为找到的选项字符。
- 其他情况下，由 `name` 指定的 shell 变量设置为问号字符 (?)，将取消设置 shell 变量 `OPTARG`，并且会将一条诊断消息写入标准错误。此情况将被视为在将参数提供给调用应用程序时检测到的错误，而不是 `getopts` 处理中的错误；诊断消息将编写为有状态的，但退出状态为零。

当遇到选项结尾时，`getopts` 实用程序将以一个大于零的返回值退出；shell 变量 `OPTIND` 将设置为第一个非选项参数的索引（如果第一个 `--` 参数之前没有其他非选项参数出现，则会将该参数视为选项参数）或设置为值 `$# + 1`（如果没有非选项参数）；`name` 变量将设置为问号符号。以下任意项都表示选项的结尾：特殊选项 `-`，发现未以 `-` 开头的参数，或遇到错误。

shell 变量 `OPTIND` 和 `OPTARG` 对于 `getopts` 的调用者而言是本地的，且缺省情况下不会导出。

由 `name` 操作数指定的 shell 变量、`OPTIND` 和 `OPTARG` 会影响当前的 shell 执行环境。

如果应用程序将 `OPTIND` 设置为值 1，则可使用一个新的参数集：当前的位置参数或新的 `arg` 值均可。对于尝试在单个 shell 执行环境中多次调用 `getopts` 的其他情况，如果所使用的参数（位置参数或 `arg` 操作数）不是在所有调用中都相同，或者使用了修改为非 1 值的 `OPTIND` 值，则会产生意外的结果。

sh

`getopts` 是一个内置 Bourne shell 命令，用于解析位置参数和检查有效选项。请参见 [sh\(1\)](#)。它支持命令语法标准的所有适用规则（请参见 [Intro\(1\)](#) 中的规则 3 至 10）。应该使用该命令来替代 `getopt` 命令。

optstring 必须包含使用 `getopts` 的命令可以识别的选项字母。如果某个字母后面跟有冒号，则该选项需要有一个参数或一组参数，必须使用空格将其与参数隔开。

每次被调用时，`getopts` 会将下一个选项放在 shell 变量 *name* 中，将下一个要处理的参数的索引放在 shell 变量 `OPTIND` 中。只要调用 shell 或 shell 脚本，就会将 `OPTIND` 初始化为 1。

如果某个选项需要选项参数，`getopts` 会将其放置在 shell 变量 `OPTARG` 中。

如果遇到非法选项，会将 `?` 放置在 *name* 中。

当遇到选项的结尾时，`getopts` 将以非零退出状态退出。可使用特殊选项 `-` 来限定选项的结尾。

缺省情况下，`getopts` 解析位置参数。如果在 (*argument...*) `getopts` 命令行上指定了额外的参数 (*argument,...*)，`getopts` 将改而对它们进行解析。

`/usr/lib/getoptcv` 读取 *filename* 中的 shell 脚本，将其转换为使用 `getopts` 而不是 `getopt`，然后在标准输出中写入结果。

这样，所有新命令将遵循 [Intro\(1\)](#) 中介绍的命令语法标准，它们应该使用 `getopts` 或 `getopt` 来解析位置参数和检查对该命令有效的选项。

如果 `getopts` 遇到不包括在 *optstring* 中的选项字母，将会在标准错误上输出一条错误消息。

尽管在当前实现中，允许放宽以下命令语法规则（请参见 [Intro\(1\)](#)），但也不应使用这些语法规则，因为系统的将来发行版可能不支持它们。如下面的“示例”部分所示，`-a` 和 `-b` 均为选项，选项 `-o` 需要一个选项参数。

以下示例违反了规则 5：具有选项参数的选项不得与其他选项组合在一起：

```
example% cmd -abxxx filename
```

以下示例违反了规则 6：在接受选项参数的选项后面必须有空格：

```
example% cmd -ab oxxx filename
```

更改 shell 变量 `OPTIND` 的值或解析不同的参数集合可能会导致意外结果。

ksh88 检查 *arg* 是否具有合法选项。如果省略了 *arg*，则会使用位置参数。选项参数以 + 或 - 开头。未以 + 或 - 开头的选项或者参数 - 表示选项的结尾。*optstring* 包含 `getopts` 可识别的字母。如果字母后面跟有一个 `:`，则该选项需要有一个参数。选项与参数之间可以用空格隔开。

每次在有前缀 + 的情况下被调用时，如果 *arg* 以 + 开头，则 `getopts` 会将它发现的下一个选项字母放置在变量 *name* 中。下一个 *arg* 的索引存储在 `OPTIND` 中。选项参数（如果有）存储在 `OPTARG` 中。

optstring 中的前导 `:` 将导致 `getopts` 将无效选项的字母存储在 `OPTARG` 中，并针对未知选项将 *name* 设置为 `?`，而在缺少某个必需的选项时将 *name* 设置为 `:`。其他情况下，`getopts` 将显示一条错误消息。如果没有更多选项，则退出状态是**非零**的。

`getopts` 支持传统的单字符短选项和由 Sun 的命令行接口范例 (Command Line Interface Paradigm, CLIP) 定义的长选项。

每个长选项都是短选项的别名，是在紧跟在其等效短选项后的括号中指定的。例如，可以使用以下脚本行将长选项 `file` 指定为短选项 `f` 的别名：

```
getopts "f(file)" opt
```

在命令行上，请在长选项之前加上前缀 `--` 或 `++`。在上面的示例中，命令行上的 `--file` 等同于 `-f`，而命令行上的 `++file` 等同于 `+f`。

每个短选项可以有多个长选项等效体，但是这有违 CLIP 规范，应谨慎使用。必须将每个长选项等效体都放置到括号中，如下所示：

```
getopts "f:(file)(input-file)o:(output-file)"
```

在上面的示例中，`--file` 和 `--input-file` 均等同于 `-f`，而 `--output-file` 等同于 `-o`。

变量 *name* 始终设置为短选项。当在命令行上指定长选项时，*name* 将被设置为其短选项等效体。

有关 Korn shell 的 `getopts` 内置命令的进一步讨论，请参见本手册页中之前 Bourne shell (sh) 部分中的讨论。

ksh `getopts` 实用程序可用于从 *arg* 指定的参数列表或从位置参数（如果省略了 *arg*）检索选项和参数。它还可以基于 *optstring* 中的信息为命令生成用法消息和手册页。

每次被调用时，`getopts` 实用程序都会将下一选项的值放入 *name* 操作数指定的 shell 变量中，将要处理的下一参数的索引放入 shell 变量 `OPTIND` 中。当调用 shell 时，`OPTIND` 将初始化为 1。当选项需要或允许选项参数时，`getopts` 会将选项参数置于 shell 变量 `OPTARG` 中。其他情况下，`OPTARG` 将设置为 1（当设置了该选项时）或 0（当取消设置该选项时）。

optstring 字符串包含字母数字字符、特殊字符 `+`、`-`、`?`、`:` 以及空格或括在 `[...]` 中的字符组。字符组可以嵌套在 `{...}` 中。在 `[...]` 组以外，后跟零个或多个空格的单个 NEWLINE 将被忽略。一个或多个空白行将选项与命令参数概要隔开。

每个 [...] 组包含一个可选标签、由 ; 分隔的可选属性以及跟在 ? 后面的可选描述字符串。对于选项解析和简短用法消息，从 ? 到下一个] 结尾之间的字符将被忽略。它们用于生成详细帮助或手册页。: 字符不能出现在标签中。? 字符在标签中必须指定为 ??，] 字符在描述字符串中必须指定为]]。文本位于两个 \b (backspace) 字符之间表示该文本在显示时应使用粗体。文本位于两个 \a (bell) 字符之间表示该文本在显示时应进行强调或使用斜体。文本位于两个 \v (vertical tab) 字符之间表示该文本应以定宽字体显示。位于两个 \f (form feed) 字符之间的文本将由其名称是被围文本的 shell 函数的输出替换。

此接口的所有输出都将写入到标准错误。

有多种组类型：

- 格式为

```
[-[version][flag[number]]...[?text]]
```

的组，作为第一个组出现，用以启用扩展的接口。

version 指定接口版本，目前为 1。若省略版本，则将使用最新版本。将来的增强可能会递增 *version*，但所有版本均受支持。*text* 通常指定一个 SCCS 或 CVS 标识字符串。可以指定零个或多个带有可选数值的标志来控制选项解析。这些标志包括：

- c 缓存此 *optstring* 以用于多个解析。用于优化在同一进程内可能会调用多次的内置项。
- i 在生成帮助时忽略此 *optstring*。在组合来自多个传递的 *optstring* 值时使用。
- l 仅在帮助消息中显示长选项名称。
- o - 选项字符前缀是可选的。这支持过时的 **ps(1)** 选项语法。
- p 该数字指定必须在长选项名称之前添加的 - 字符的数目。缺省值为 2。0、1 或 2 都是可以接受的，例如 **p0** 用于 **dd(1M)**，**p1** 用于 **find(1)**。
- s 该数字指定手册页节编号，缺省值为 1。

- 格式为 [*option* [!] [=number] [:longname] [? *text*]] 的选项规范。在此情况下，第一个字段是选项字符，是当选项匹配时在 *name* 操作数中返回的值。如果没有选项字符，则应当指定两个或多个数字。如果长选项匹配，该数字将返回为 *name* 操作数的值。如果选项后跟有一个 !，则选项字符的意思将是 *longname* 意思的反义。对于不接受值的选项，对于 ! 反义的选项字符，OPTARG 将设置为 0，其他情况下将设置为 1。=number 是可选的，它指定要在 *name* 操作数中返回一个数字而不是返回选项字符。longname 是由 --longname 指定的，并与所有长选项的最短非二义性前缀匹配。longname 字段中的 * 表示仅截至到此点的字符需要匹配，并假定任何其他字符完全匹配。对于没有 longname 或描述性文本的选项，最外层的 [和] 可以省略。

- 选项参数规范。接受参数的选项的后面可以跟有 : (指示字符串值) 或 # (指示数字值)，以及选项参数规范。选项参数规范由选项参数名称 (如 **field 1**) 组成。其余由 : 分隔的字段是类型名和特殊属性词 **listof**、**oneof** 和 **ignorecase** 中的零个或

多个。缺省选项值可以在最后的字段中指定为 `:=default`。选项参数规范的后面可以跟有用大括号括起的选项值描述列表。接受参数的长选项将指定为 `--longname=value`。如果 `:` 或 `#` 后跟有 `?`，则选项参数是可选的。如果仅指定了选项字符格式，且下一个参数以 `-` 或 `+` 开头，则不会设置可选参数值。

- 选项值描述。
- 参数规范。可以通过将有效选项参数值括在选项参数规范后的 `{...}` 内来指定有效选项参数值的列表。每个允许的值可以使用 `[...]`（在其中包含值以及跟在值后的描述）来指定。
- `[+\ n...]` 格式的组，以定宽字体显示代表 `...` 的字符且不换行。
- `[+name ?text]` 格式的组，指定节名称与描述性文本。如果省略了 `name`，则会将 `text` 置于新段落中。
- `[-name ?text]` 格式的组，指定实现部分的条目。

如果 `optstring` 的前导字符是 `+`，则以 `+` 开头的参数也将被视为选项。

`optstring` 中的前导 `:` 字符或跟随在前导 `+` 后的 `:` 字符会影响错误的处理方式。如果在处理选项时遇到 `optstring` 中未指定的选项字符或 `longname` 参数，则其名称是 `name` 的 `shell` 变量将设置为 `?` 字符。 `shell` 变量 `OPTARG` 将设置为找到的字符。如果某个选项参数缺失或具有无效值，则 `name` 将设置为 `:` 字符， `shell` 变量 `OPTARG` 将设置为找到的选项字符。若无前导 `:`，在遇到错误时， `name` 将设置为 `?` 字符， `OPTARG` 将取消设置，且会将一条错误消息写入到标准错误。

以下任一情况都表示选项的结尾：

1. 遇到特殊参数 `--`。
2. 遇到未以 `-` 开头的参数。
3. 指定了帮助参数。
4. 遇到了错误。

如果 `OPTIND` 设置为值 `1`，则可以使用新的参数集合。

`getopts` 还可用于生成帮助消息，其中包含命令用法和详细的描述。将 `args` 指定为：

- `-?` 这用来生成用法概要。
- `--??` 这用来生成详细的用法消息。
- `--??man` 这用来生成格式化的手册页。
- `--??api` 这用来生成易于解析的用法消息。
- `--??html` 这用来生成 `html` 格式的手册页。
- `--??nroff` 这用来生成 `nroff` 格式的手册页。
- `--??使用` 这用来列出当前的 `optstring`。

--???name 这用来列出 version=*n*，其中 *n* 大于 0（如果 getopts 可以识别选项 *name*）。

当遇到选项结尾时，getopts 将以非零返回值退出，变量 OPTIND 将设置为第一个非选项参数的索引。

选项

ksh

ksh 支持以下选项：

-a *name* 使用 *name* 而不是用法消息中的命令名称。

操作数

支持下列操作数：

optstring 一个包含调用 getopts 的实用程序可识别的选项字符的字符串。如果某个字符后跟有一个冒号，则该选项需要有一个参数，该参数应作为单独的参数提供。应用程序应将选项字符及其选项参数指定为单独的参数，但 getopts 会将跟随在需要参数的选项字符后的字符解释为参数，无论实际情况是否如此。在调用 getopts 时，如果显式的空选项参数不是作为单独的参数提供的，则无需识别该参数；请参见 [getopt\(3C\)](#)。应用程序不得使用问号字符 (?) 和冒号字符 (:) 作为选项字符。使用非字母数字的其他选项字符会产生意外结果。如果选项参数不是作为与选项字符隔开的参数提供的，则 OPTARG 中的值会被除去选项字符和 -。optstring 中的第一个字符决定了当某个选项字符未知或缺少某个选项参数时 getopts 的行为。

name shell 变量的名称，由 getopts 实用程序设置为发现的选项字符。

缺省情况下，getopts 实用程序解析传递到调用方 shell 过程的位置参数。如果指定了 *arg*，则将解析这些参数而不解析位置参数。

用法

因为 getopts 会影响当前的 shell 执行环境，因此，它通常是作为 shell 常规内置项提供的。如果它是在一个子 shell 或单独的实用程序执行环境中调用的，例如下面所示的环境之一：

```
(getopts abc value "$@")
nohup getopts ...
find . -exec getopts ... \;
```

则它不会影响调用者的环境中的 shell 变量。

请注意，尽管位置参数已更改，但 shell 函数与调用方 shell 共享 OPTIND。想使用 getopts 来解析参数的函数通常希望将 OPTIND 的值保存在条目上并在返回之前对其进行还原。然而，有时候函数希望为调用方 shell 更改 OPTIND。

示例

示例 1 解析并显示参数

以下示例脚本将解析并显示其参数：

示例1 解析并显示参数 (续)

```

aflag=
bflag=
while getopts ab: name
do
    case $name in
        a)    aflag=1;;
        b)    bflag=1
              bval="$OPTARG";;
        ?)    printf "Usage: %s: [-a] [-b value] args\n" $0
              exit 2;;
    esac
done
if [ ! -z "$aflag" ]; then
    printf "Option -a specified\n"
fi
if [ ! -z "$bflag" ]; then
    printf 'Option -b "%s" specified\n' "$bval"
fi
shift $(( $OPTIND - 1 ))
printf "Remaining arguments are: %s\n" "$*"

```

示例2 为带选项的命令处理参数

以下 shell 程序片段为可以接受选项 `-a` 或 `-b` 的命令处理参数。它还将处理选项 `-o`，该选项需要一个选项参数：

```

while getopts abo: c
do
    case $c in
        a | b)  FLAG=$c;;
        o)      OARG=$OPTARG;;
        \?)     echo $USAGE
              exit 2;;
    esac
done
shift `expr $OPTIND - 1`

```

示例3 等效的代码表达式

此代码示例将下列各个输入视为等效的：

```

cmd -a -b -o "xxx z yy" filename
cmd -a -b -o "xxx z yy" -- filename
cmd -ab -o xxx,z,yy filename
cmd -ab -o "xxx z yy" filename
cmd -o xxx,z,yy -b -a filename

```

- 环境变量** 有关影响 `getopts` 执行的以下环境变量的描述，请参见 [environ\(5\)](#)：LANG、LC_ALL、LC_CTYPE、LC_MESSAGES 和 NLSPATH。
- OPTIND** `getopts` 将该变量用作要处理的下一个参数的索引。
- OPTARG** 如果某个选项使用了参数，`getopts` 将使用该变量来存储参数。
- 退出状态** 将返回以下退出值：
- 0 找到了一个选项，可能是也可能不是由 *optstring* 指定的。
 - >0 遇到选项的结尾，或发生错误。
- ksh** ksh 返回以下退出值：
- 0 找到指定的选项。
 - 1 遇到选项结尾。
 - 2 已生成用法或信息性消息。
- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

/usr/bin/getopts、sh、ksh88	属性类型	属性值
	可用性	system/core-os
	接口稳定性	Committed（已确定）
	标准	请参见 standards(5) 。

ksh	属性类型	属性值
	可用性	system/core-os
	接口稳定性	Uncommitted（未确定）

另请参见 [Intro\(1\)](#)、[getoptcvt\(1\)](#)、[ksh\(1\)](#)、[ksh88\(1\)](#)、[ps\(1\)](#)、[sh\(1\)](#)、[getopt\(3C\)](#)、[attributes\(5\)](#)、[envi](#)

诊断 无论何时检测到错误，如果 *optstring* 操作数中的第一个字符不是冒号 (:)，则都会向标准错误写入一条诊断消息，其中以未指定格式包含以下信息：

- 消息中标识了调用方程序的名称。调用方程序名称是调用 `getopts` 实用程序时 shell 特殊参数 `0` 的值。可以使用


```
basename "$0"
```

 的名称等效体。
- 如果发现了未在 *optstring* 中指定的某个选项，则会在消息中标识该错误并标识无效的选项字符。

- 如果发现了需要选项参数的某个选项，但未发现选项参数，则会在消息中标识该错误并标识无效的选项字符。

引用名	gettext – 从消息数据库检索文本字符串
用法概要	<pre>gettext [-d <i>textdomain</i> --domain=<i>textdomain</i>] [<i>textdomain</i>] <i>msgid</i> gettext -s [-e] [-n] [-d <i>textdomain</i> --domain=<i>textdomain</i>] <i>msgid</i>...</pre>
描述	<p>gettext 实用程序从 <code>msgfmt(1)</code> 生成的消息目标文件中检索与字符串 <i>msgid</i> 对应的已翻译文本字符串。如果提供了可选参数 <i>textdomain</i>，则会从该参数中派生消息目标文件名，否则会从 <code>TEXTDOMAIN</code> 环境中派生消息目标文件名。如果未指定任何域，或无法找到对应的字符串，则 <code>gettext</code> 会输出 <i>msgid</i>。</p> <p>通常，<code>gettext</code> 在 <code>/usr/lib/locale/lang/LC_MESSAGES</code> 中查找其消息目标文件，其中 <i>lang</i> 是语言环境名称。如果已提供，<code>TEXTDOMAINDIR</code> 环境变量会将路径名组件替换为 <i>lang</i>。</p> <p>此命令解释 C 语言中的转义序列（例如将 <code>\t</code> 解析为制表符 (tab)）。使用 <code>\\</code> 输出反斜杠。要一行生成一条消息，可在 <i>msgid</i> 的末尾输入 <code>\n</code>，或将此命令与 <code>printf(1)</code> 结合使用。</p> <p>与 <code>-s</code> 选项一起使用时，<code>gettext</code> 的行为与 <code>echo(1)</code> 相同。但该命令不是仅仅将其参数复制到标准输出。而是翻译在选定目录中找到的那些消息。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> <code>-d <i>textdomain</i></code> <code>--domain=<i>textdomain</i></code> 如果未将 <i>textdomain</i> 指定为操作数，则会从域 <i>textdomain</i> 中检索已翻译的消息。 <code>-e</code> 如果与 <code>-s</code> 选项一起使用，则会启用某些转义序列的扩展。 <code>-n</code> 如果与 <code>-s</code> 选项一起使用，则会隐藏尾随的换行符。 <code>-s</code> 行为与 <code>echo(1)</code> 相同（请参见上文的“描述”部分）。如果指定了 <code>-s</code> 选项，缺省情况下，不会执行 C 语言转义序列的扩展，且一个换行符会追加到输出中。
操作数	<p>支持下列操作数：</p> <ul style="list-style-type: none"> <i>textdomain</i> 用于检索消息的域名。如果提供了此操作数，它将覆盖 <code>-d</code> 或 <code>--domain</code> 选项指定的内容。 <i>msgid</i> 用来检索本地化消息的键。
环境变量	<ul style="list-style-type: none"> <code>LANG</code> 指定语言环境名称。 <code>LC_MESSAGES</code> 指定消息语言环境，如果提供，则会覆盖消息的 <code>LANG</code>。 <code>TEXTDOMAIN</code> 指定文本域名，它与不带有 <code>.mo</code> 后缀的消息目标文件名相同。

TEXTDOMAINDIR 指定消息数据库的路径名。如果提供，则会替换 `/usr/lib/locale`。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [echo\(1\)](#)、[msgfmt\(1\)](#)、[printf\(1\)](#)、[gettext\(3C\)](#)、[setlocale\(3C\)](#)、[attributes\(5\)](#)

附注 该实用程序是库例程 [gettext\(3C\)](#) 的 shell 等效项。

引用名	gettxt – retrieve a text string from a message database
用法概要	gettxt <i>msgfile</i> : <i>msgnum</i> [<i>dflt_msg</i>]
描述	<p>gettxt retrieves a text string from a message file in the directory <code>/usr/lib/locale/locale/LC_MESSAGES</code>. The directory name <i>locale</i> corresponds to the language in which the text strings are written; see setlocale(3C).</p> <p><i>msgfile</i> Name of the file in the directory <code>/usr/lib/locale/locale/LC_MESSAGES</code> to retrieve <i>msgnum</i> from. The name of <i>msgfile</i> can be up to 14 characters in length, but may not contain either <code>\0</code> (null) or the ASCII code for <code>/</code> (slash) or <code>:</code> (colon).</p> <p><i>msgnum</i> Sequence number of the string to retrieve from <i>msgfile</i>. The strings in <i>msgfile</i> are numbered sequentially from 1 to <i>n</i>, where <i>n</i> is the number of strings in the file.</p> <p><i>dflt_msg</i> Default string to be displayed if gettxt fails to retrieve <i>msgnum</i> from <i>msgfile</i>. Nongraphic characters must be represented as alphabetic escape sequences.</p> <p>The text string to be retrieved is in the file <i>msgfile</i>, created by the mkmsgs(1) utility and installed under the directory <code>/usr/lib/locale/locale/LC_MESSAGES</code>. You control which directory is searched by setting the environment variable <code>LC_MESSAGES</code>. If <code>LC_MESSAGES</code> is not set, the environment variable <code>LANG</code> will be used. If <code>LANG</code> is not set, the files containing the strings are under the directory <code>/usr/lib/locale/C/LC_MESSAGES</code>.</p> <p>If gettxt fails to retrieve a message in the requested language, it will try to retrieve the same message from <code>/usr/lib/locale/C/LC_MESSAGES/<i>msgfile</i></code>. If this also fails, and if <i>dflt_msg</i> is present and non-null, then it will display the value of <i>dflt_msg</i>; if <i>dflt_msg</i> is not present or is null, then it will display the string <code>Message not found!!</code>.</p>
示例	<p>示例 1 The environment variables <code>LANG</code> and <code>LC_MESSAGES</code>.</p> <p>If the environment variables <code>LANG</code> or <code>LC_MESSAGES</code> have not been set to other than their default values, the following example:</p> <pre>example% gettxt UX:10 "hello world\n"</pre> <p>will try to retrieve the 10th message from <code>/usr/lib/locale/C/UX/<i>msgfile</i></code>. If the retrieval fails, the message "hello world," followed by a newline, will be displayed.</p>
环境变量	<p>See environ(5) for descriptions of the following environment variables that affect the execution of gettxt: <code>LC_CTYPE</code> and <code>LC_MESSAGES</code>.</p> <p><code>LC_CTYPE</code> Determines how gettxt handles characters. When <code>LC_CTYPE</code> is set to a valid value, gettxt can display and handle text and filenames containing valid characters for that locale. gettxt can display and handle Extended Unix Code (EUC) characters where any individual character can be 1, 2, or 3 bytes wide. gettxt can also handle EUC characters of 1, 2, or more column widths. In the "C" locale, only characters from ISO 8859-1 are valid.</p>

LC_MESSAGES Determines how diagnostic and informative messages are presented. This includes the language and style of the messages, and the correct form of affirmative and negative responses. In the "C" locale, the messages are presented in the default form found in the program itself (in most cases, U.S. English).

文件 `/usr/lib/locale/C/LC_MESSAGES/*` default message files created by [mkmsgs\(1\)](#)
`/usr/lib/locale/locale/LC_MESSAGES/*` message files for different languages created by [mkmsgs\(1\)](#)

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	text/locale
CSI	Enabled

另请参见 [exstr\(1\)](#), [mkmsgs\(1\)](#), [srchtxt\(1\)](#), [gettxt\(3C\)](#), [setlocale\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#)

引用名 getzonepath – 显示与指定标签对应的区域的根路径

用法概要 /usr/bin/getzonepath {*sensitivity-label*}

描述 getzonepath 显示与指定敏感标签对应的正在运行的有标签区域的根路径名称。返回的路径名相对于调用者的根路径名，并具有指定的敏感标签。

如果调用者在全局区域中，则返回的路径名是不允许访问的，除非调用者的进程具有 `file_dac_search` 特权。

如果调用者在某个有标签区域中，则调用者的标签必须支配指定的标签。只能对返回的路径名下的文件进行只读操作。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/trusted
稳定性	Committed (已确定)

诊断 getzonepath 退出时返回下列值之一：

- 0 成功
- 1 用法错误
- 2 失败；错误消息是来自 [getzonerootbylabel\(3TSOL\)](#) 的系统错误号

另请参见 [getzonerootbylabel\(3TSOL\)](#)、[attributes\(5\)](#)

《Trusted Extensions Developer's Guide》中的“Acquiring a Sensitivity Label”

附注 仅当系统配置有 Trusted Extensions 时，本手册页中介绍的功能才可用。

引用名 glob – shell built-in function to expand a word list

用法概要

csH glob *wordlist*

描述

csH glob performs filename expansion on *wordlist*. Like [echo\(1\)](#), but no ‘\’ escapes are recognized. Words are delimited by null characters in the output.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [csH\(1\)](#), [echo\(1\)](#), [attributes\(5\)](#)

引用名 gprof – display call-graph profile data

用法概要 gprof [-abcD\sz] [-e *function-name*] [-E *function-name*]
 [-f *function-name*] [-F *function-name*]
 [*image-file* [*profile-file*...]]
 [-n *number of functions*]

描述

The gprof utility produces an execution profile of a program. The effect of called routines is incorporated in the profile of each caller. The profile data is taken from the call graph profile file that is created by programs compiled with the -xpg option of cc(1), or by the -pg option with other compilers, or by setting the LD_PROFILE environment variable for shared objects. See [ld.so.1\(1\)](#). These compiler options also link in versions of the library routines which are compiled for profiling. The symbol table in the executable image file *image-file* (a.out by default) is read and correlated with the call graph profile file *profile-file* (gmon.out by default).

First, execution times for each routine are propagated along the edges of the call graph. Cycles are discovered, and calls into a cycle are made to share the time of the cycle. The first listing shows the functions sorted according to the time they represent, including the time of their call graph descendants. Below each function entry is shown its (direct) call-graph children and how their times are propagated to this function. A similar display above the function shows how this function's time and the time of its descendants are propagated to its (direct) call-graph parents.

Cycles are also shown, with an entry for the cycle as a whole and a listing of the members of the cycle and their contributions to the time and call counts of the cycle.

Next, a flat profile is given, similar to that provided by [prof\(1\)](#). This listing gives the total execution times and call counts for each of the functions in the program, sorted by decreasing time. Finally, an index is given, which shows the correspondence between function names and call-graph profile index numbers.

A single function may be split into subfunctions for profiling by means of the MARK macro. See [prof\(5\)](#).

Beware of quantization errors. The granularity of the sampling is shown, but remains statistical at best. It is assumed that the time for each execution of a function can be expressed by the total time for the function divided by the number of times the function is called. Thus the time propagated along the call-graph arcs to parents of that function is directly proportional to the number of times that arc is traversed.

The profiled program must call [exit\(2\)](#) or return normally for the profiling information to be saved in the gmon.out file.

选项

The following options are supported:

-a Suppress printing statically declared functions. If this option is given, all relevant information about the static function (for instance, time samples, calls to other functions, calls from other functions) belongs to the function loaded just before the static function in the a.out file.

-
- b Brief. Suppress descriptions of each field in the profile.
 - c Discover the static call-graph of the program by a heuristic which examines the text space of the object file. Static-only parents or children are indicated with call counts of 0. Note that for dynamically linked executables, the linked shared objects' text segments are not examined.
 - C Demangle C++ symbol names before printing them out.
 - D Produce a profile file `gmon.sum` that represents the difference of the profile information in all specified profile files. This summary profile file may be given to subsequent executions of `gprof` (also with `-D`) to summarize profile data across several runs of an `a.out` file. See also the `-s` option.

As an example, suppose function A calls function B n times in profile file `gmon.sum`, and m times in profile file `gmon.out`. With `-D`, a new `gmon.sum` file will be created showing the number of calls from A to B as $n-m$.
 - e`function-name` Suppress printing the graph profile entry for routine `function-name` and all its descendants (unless they have other ancestors that are not suppressed). More than one `-e` option may be given. Only one `function-name` may be given with each `-e` option.
 - E`function-name` Suppress printing the graph profile entry for routine `function-name` (and its descendants) as `-e`, below, and also exclude the time spent in `function-name` (and its descendants) from the total and percentage time computations. More than one `-E` option may be given. For example:

`-E mcount -E mcleanup`

is the default.
 - f`function-name` Print the graph profile entry only for routine `function-name` and its descendants. More than one `-f` option may be given. Only one `function-name` may be given with each `-f` option.
 - F`function-name` Print the graph profile entry only for routine `function-name` and its descendants (as `-f`, below) and also use only the times of the printed routines in total time and percentage computations. More than one `-F` option may be given. Only one `function-name` may be given with each `-F` option. The `-F` option overrides the `-E` option.
 - l Suppress the reporting of graph profile entries for all local symbols. This option would be the equivalent of placing all of the local symbols for the specified executable image on the `-E` exclusion list.
 - n Limits the size of flat and graph profile listings to the top n offending functions.

- s Produce a profile file `gmon.sum` which represents the sum of the profile information in all of the specified profile files. This summary profile file may be given to subsequent executions of `gprof` (also with `-s`) to accumulate profile data across several runs of an `a.out` file. See also the `-D` option.
- z Display routines which have zero usage (as indicated by call counts and accumulated time). This is useful in conjunction with the `-c` option for discovering which routines were never called. Note that this has restricted use for dynamically linked executables, since shared object text space will not be examined by the `-c` option.

环境变量

`PROFDIR` If this environment variable contains a value, place profiling output within that directory, in a file named `pid.programname`. `pid` is the process ID and `programname` is the name of the program being profiled, as determined by removing any path prefix from the `argv[0]` with which the program was called. If the variable contains a null value, no profiling output is produced. Otherwise, profiling output is placed in the file `gmon.out`.

文件

`a.out` Executable file containing namelist

`gmon.out` Dynamic call-graph and profile

`gmon.sum` Summarized dynamic call-graph and profile

`PROFDIR/pid.programname`

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	developer/base-developer-utilities

另请参见

[cc\(1\)](#), [ld.so.1\(1\)](#), [prof\(1\)](#), [exit\(2\)](#), [pcsample\(2\)](#), [profil\(2\)](#), [malloc\(3C\)](#), [malloc\(3MALLOC\)](#), [monitor\(3C\)](#), [attributes\(5\)](#), [prof\(5\)](#)

Graham, S.L., Kessler, P.B., McKusick, M.K., `gprof`: A Call Graph Execution Profiler Proceedings of the SIGPLAN '82 Symposium on Compiler Construction, SIGPLAN Notices, Vol. 17, No. 6, pp. 120-126, June 1982.

《[链接程序和库指南](#)》

附注

If the executable image has been stripped and does not have the `.symtab` symbol table, `gprof` reads the global dynamic symbol tables `.dynam` and `.SUNW_ldynam`, if present. The symbols in the dynamic symbol tables are a subset of the symbols that are found in `.symtab`. The `.dynam` symbol table contains the global symbols used by the runtime linker. `.SUNW_ldynam` augments the information in `.dynam` with local function symbols. In the case where `.dynam`

is found and `.SUNW_ldynsym` is not, only the information for the global symbols is available. Without local symbols, the behavior is as described for the `-a` option.

`LD_LIBRARY_PATH` must not contain `/usr/lib` as a component when compiling a program for profiling. If `LD_LIBRARY_PATH` contains `/usr/lib`, the program will not be linked correctly with the profiling versions of the system libraries in `/usr/lib/libp`.

The times reported in successive identical runs may show variances because of varying cache-hit ratios that result from sharing the cache with other processes. Even if a program seems to be the only one using the machine, hidden background or asynchronous processes may blur the data. In rare cases, the clock ticks initiating recording of the program counter may *beat* with loops in a program, grossly distorting measurements. Call counts are always recorded precisely, however.

Only programs that call `exit` or return from `main` are guaranteed to produce a profile file, unless a final call to `monitor` is explicitly coded.

Functions such as `mcount()`, `_mcount()`, `moncontrol()`, `_moncontrol()`, `monitor()`, and `_monitor()` may appear in the `gprof` report. These functions are part of the profiling implementation and thus account for some amount of the runtime overhead. Since these functions are not present in an unprofiled application, time accumulated and call counts for these functions may be ignored when evaluating the performance of an application.

64-bit profiling

64-bit profiling may be used freely with dynamically linked executables, and profiling information is collected for the shared objects if the objects are compiled for profiling. Care must be applied to interpret the profile output, since it is possible for symbols from different shared objects to have the same name. If name duplication occurs in the profile output, the module id prefix before the symbol name in the symbol index listing can be used to identify the appropriate module for the symbol.

When using the `-s` or `-D` option to sum multiple profile files, care must be taken not to mix 32-bit profile files with 64-bit profile files.

32-bit profiling

32-bit profiling may be used with dynamically linked executables, but care must be applied. In 32-bit profiling, shared objects cannot be profiled with `gprof`. Thus, when a profiled, dynamically linked program is executed, only the *main* portion of the image is sampled. This means that all time spent outside of the *main* object, that is, time spent in a shared object, will not be included in the profile summary; the total time reported for the program may be less than the total time used by the program.

Because the time spent in a shared object cannot be accounted for, the use of shared objects should be minimized whenever a program is profiled with `gprof`. If desired, the program should be linked to the profiled version of a library (or to the standard archive version if no profiling version is available), instead of the shared object to get profile information on the functions of a library. Versions of profiled libraries may be supplied with the system in the `/usr/lib/libp` directory. Refer to compiler driver documentation on profiling.

Consider an extreme case. A profiled program dynamically linked with the shared C library spends 100 units of time in some libc routine, say, `malloc()`. Suppose `malloc()` is called only from routine B and B consumes only 1 unit of time. Suppose further that routine A consumes 10 units of time, more than any other routine in the *main* (profiled) portion of the image. In this case, `gprof` will conclude that most of the time is being spent in A and almost no time is being spent in B. From this it will be almost impossible to tell that the greatest improvement can be made by looking at routine B and not routine A. The value of the profiler in this case is severely degraded; the solution is to use archives as much as possible for profiling.

已知问题

Parents which are not themselves profiled will have the time of their profiled children propagated to them, but they will appear to be spontaneously invoked in the call-graph listing, and will not have their time propagated further. Similarly, signal catchers, even though profiled, will appear to be spontaneous (although for more obscure reasons). Any profiled children of signal catchers should have their times propagated properly, unless the signal catcher was invoked during the execution of the profiling routine, in which case all is lost.

引用名 `grep` – search a file for a pattern

用法概要

```
/usr/bin/grep [-c | -l | -q] [-bhinsvw] limited-regular-expression
    [filename]...

/usr/xpg4/bin/grep [-E | -F] [-c | -l | -q] [-bhinsvx] -e pattern_list...
    [-f pattern_file]... [file]...

/usr/xpg4/bin/grep [-E | -F] [-c | -l | -q] [-bhinsvx]
    [-e pattern_list]... -f pattern_file... [file]...

/usr/xpg4/bin/grep [-E | -F] [-c | -l | -q] [-bhinsvx] pattern
    [file]...
```

描述

The `grep` utility searches text files for a pattern and prints all lines that contain that pattern. It uses a compact non-deterministic algorithm.

Be careful using the characters `$`, `*`, `[`, `^`, `|`, `(`, `)`, and `\` in the *pattern_list* because they are also meaningful to the shell. It is safest to enclose the entire *pattern_list* in single quotes `'...'`.

If no files are specified, `grep` assumes standard input. Normally, each line found is copied to standard output. The file name is printed before each line found if there is more than one input file.

`/usr/bin/grep`

The `/usr/bin/grep` utility uses limited regular expressions like those described on the [regex\(5\)](#) manual page to match the patterns.

`/usr/xpg4/bin/grep`

The options `-E` and `-F` affect the way `/usr/xpg4/bin/grep` interprets *pattern_list*. If `-E` is specified, `/usr/xpg4/bin/grep` interprets *pattern_list* as a full regular expression (see `-E` for description). If `-F` is specified, `grep` interprets *pattern_list* as a fixed string. If neither are specified, `grep` interprets *pattern_list* as a basic regular expression as described on [regex\(5\)](#) manual page.

选项

The following options are supported for both `/usr/bin/grep` and `/usr/xpg4/bin/grep`:

- b Precedes each line by the block number on which it was found. This can be useful in locating block numbers by context (first block is 0).
- c Prints only a count of the lines that contain the pattern.
- h Prevents the name of the file containing the matching line from being prepended to that line. Used when searching multiple files.
- i Ignores upper/lower case distinction during comparisons.
- l Prints only the names of files with matching lines, separated by NEWLINE characters. Does not repeat the names of files when the pattern is found more than once.
- n Precedes each line by its line number in the file (first line is 1).
- q Quiet. Does not write anything to the standard output, regardless of matching lines. Exits with zero status if an input line is selected.

- s Suppresses error messages about nonexistent or unreadable files.
- v Prints all lines except those that contain the pattern.
- w Searches for the expression as a word as if surrounded by \< and \>.

/usr/xpg4/bin/grep

The following options are supported for /usr/xpg4/bin/grep only:

- e *pattern_list* Specifies one or more patterns to be used during the search for input. Patterns in *pattern_list* must be separated by a NEWLINE character. A null pattern can be specified by two adjacent newline characters in *pattern_list*. Unless the -E or -F option is also specified, each pattern is treated as a basic regular expression. Multiple -e and -f options are accepted by grep. All of the specified patterns are used when matching lines, but the order of evaluation is unspecified.
- E Matches using full regular expressions. Treats each pattern specified as a full regular expression. If any entire full regular expression pattern matches an input line, the line is matched. A null full regular expression matches every line. Each pattern is interpreted as a full regular expression as described on the [regex\(5\)](#) manual page, except for \ (and \), and including:
 1. A full regular expression followed by + that matches one or more occurrences of the full regular expression.
 2. A full regular expression followed by ? that matches 0 or 1 occurrences of the full regular expression.
 3. Full regular expressions separated by | or by a new-line that match strings that are matched by any of the expressions.
 4. A full regular expression that is enclosed in parentheses () for grouping.

The order of precedence of operators is [], then * ? +, then concatenation, then | and new-line.
- f *pattern_file* Reads one or more patterns from the file named by the path name *pattern_file*. Patterns in *pattern_file* are terminated by a NEWLINE character. A null pattern can be specified by an empty line in *pattern_file*. Unless the -E or -F option is also specified, each pattern is treated as a basic regular expression.
- F Matches using fixed strings. Treats each pattern specified as a string instead of a regular expression. If an input line contains any of the patterns as a contiguous sequence of bytes, the line is matched. A null string matches every line. See [fgrep\(1\)](#) for more information.

-x Considers only input lines that use all characters in the line to match an entire fixed string or regular expression to be matching lines.

操作数

The following operands are supported:

file A path name of a file to be searched for the patterns. If no *file* operands are specified, the standard input is used.

`/usr/bin/grep` *pattern* Specifies a pattern to be used during the search for input.

`/usr/xpg4/bin/grep` *pattern* Specifies one or more patterns to be used during the search for input. This operand is treated as if it were specified as `-e pattern_list`.

用法

The `-c`, `-l` and `-q` options are mutually exclusive. If specified together `-q` overrides `-c` which overrides `-l`.

The `-e pattern_list` option has the same effect as the *pattern_list* operand, but is useful when *pattern_list* begins with the hyphen delimiter. It is also useful when it is more convenient to provide multiple patterns as separate arguments.

Multiple `-e` and `-f` options are accepted and `grep` uses all of the patterns it is given while matching input text lines. Notice that the order of evaluation is not specified. If an implementation finds a null string as a pattern, it is allowed to use that pattern first, matching every line, and effectively ignore any other patterns.

The `-q` option provides a means of easily determining whether or not a pattern (or string) exists in a group of files. When searching several files, it provides a performance improvement (because it can quit as soon as it finds the first match) and requires less care by the user in choosing the set of files to supply as arguments (because it exits zero if it finds a match even if `grep` detected an access or read error on earlier file operands).

Large File Behavior

See [largefile\(5\)](#) for the description of the behavior of `grep` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Finding All Uses of a Word

To find all uses of the word “Posix” (in any case) in the file `text.mm`, and write with line numbers:

```
example% /usr/bin/grep -i -n posix text.mm
```

示例 2 Finding All Empty Lines

To find all empty lines in the standard input:

```
example% /usr/bin/grep ^$
```

or

```
example% /usr/bin/grep -v .
```

示例 3 Finding Lines Containing Strings

All of the following commands print all lines containing strings abc or def or both:

```
example% /usr/xpg4/bin/grep 'abc
def'
example% /usr/xpg4/bin/grep -e 'abc
def'
example% /usr/xpg4/bin/grep -e 'abc' -e 'def'
example% /usr/xpg4/bin/grep -E 'abc|def'
example% /usr/xpg4/bin/grep -E -e 'abc|def'
example% /usr/xpg4/bin/grep -E -e 'abc' -e 'def'
example% /usr/xpg4/bin/grep -E 'abc
def'
example% /usr/xpg4/bin/grep -E -e 'abc
def'
example% /usr/xpg4/bin/grep -F -e 'abc' -e 'def'
example% /usr/xpg4/bin/grep -F 'abc
def'
example% /usr/xpg4/bin/grep -F -e 'abc
def'
```

示例 4 Finding Lines with Matching Strings

Both of the following commands print all lines matching exactly abc or def:

```
example% /usr/xpg4/bin/grep -E '^abc$ ^def$'
example% /usr/xpg4/bin/grep -F -x 'abc def'
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `grep`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

- 0 One or more matches were found.
- 1 No matches were found.
- 2 Syntax errors or inaccessible files (even if matches were found).

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/grep

ATTRIBUTE	TYPE	VALUE
Availability		system/core-os
CSI		Not Enabled

/usr/xpg4/bin/grep

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[egrep\(1\)](#), [fgrep\(1\)](#), [sed\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#), [regexp\(5\)](#), [standards\(5\)](#)

附注

/usr/bin/grep

Lines are limited only by the size of the available virtual memory. If there is a line with embedded nulls, grep only matches up to the first null. If the line matches, the entire line is printed.

/usr/xpg4/bin/grep

The results are unspecified if input files contain lines longer than `LINE_MAX` bytes or contain binary data. `LINE_MAX` is defined in `/usr/include/limits.h`.

引用名 groups – print group membership of user

用法概要 groups [*user*]...

描述 The command groups prints on standard output the groups to which you or the optionally specified user belong. Each user belongs to a group specified in /etc/passwd and possibly to other groups as specified in /etc/group. Note that /etc/passwd specifies the numerical ID (gid) of the group. The groups command converts gid to the group name in the output.

示例 The output takes the following form:

```
example% groups tester01 tester02
tester01 : staff
tester02 : staff
example%
```

文件 /etc/passwd

/etc/group

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [group\(4\)](#), [passwd\(4\)](#), [attributes\(5\)](#)

引用名 groups – display a user's group memberships

用法概要 /usr/ucb/groups [user]...

描述 With no arguments, groups displays the groups to which you belong; else it displays the groups to which the user belongs. Each user belongs to a group specified in the password file /etc/passwd and possibly to other groups as specified in the file /etc/group. If you do not own a file but belong to the group which it is owned by then you are granted group access to the file.

文件 /etc/passwd

/etc/group

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [getgroups\(2\)](#), [attributes\(5\)](#)

附注 This command is obsolete.

引用名	grpck – check group database entries
用法概要	<code>/usr/sbin/grpck [filename]</code>
描述	The grpck utility checks that a file in group(4) does not contain any errors; it checks the <code>/etc/group</code> file by default.
文件	<code>/etc/group</code>
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [groups\(1\)](#), [group\(4\)](#), [passwd\(4\)](#), [attributes\(5\)](#)

诊断	Too many/few fields	An entry in the group file does not have the proper number of fields.
	No group name	The group name field of an entry is empty.
	Bad character(s) in group name	The group name in an entry contains characters other than lower-case letters and digits.
	Invalid GID	The group ID field in an entry is not numeric or is greater than 65535.
	Null login name	A login name in the list of login names in an entry is null.
	Logname not found in password file	A login name in the list of login names in an entry is not in the password file.
	Line too long	A line (including the NEWLINE character) in the group file exceeds the maximum length of 512 characters.
	Duplicate logname entry	A login name appears more than once in the list of login names for a group file entry.
	Out of memory	The program cannot allocate memory in order to continue.
	Maximum groups exceeded for logname	A login name's group membership exceeds the maximum, <code>NGROUPS_MAX</code> .

引用名 hash, rehash, unhash, hashstat – evaluate the internal hash table of the contents of directories

用法概要 /usr/bin/hash [*utility*]

/usr/bin/hash [-r]

sh hash [-r] [*name*]...

csch rehash

unhash

hashstat

ksh88 hash [*name*]...

hash [-r]

描述

/usr/bin/hash The /usr/bin/hash utility affects the way the current shell environment remembers the locations of utilities found. Depending on the arguments specified, it adds utility locations to its list of remembered locations or it purges the contents of the list. When no arguments are specified, it reports on the contents of the list. The -r option causes the shell to forget all remembered locations.

Utilities provided as built-ins to the shell are not reported by hash.

sh For each *name*, the location in the search path of the command specified by *name* is determined and remembered by the shell. The -r option to the hash built-in causes the shell to forget all remembered locations. If no arguments are given, hash provides information about remembered commands. The *Hits* column of output is the number of times a command has been invoked by the shell process. The *Cost* column of output is a measure of the work required to locate a command in the search path. If a command is found in a "relative" directory in the search path, after changing to that directory, the stored location of that command is recalculated. Commands for which this will be done are indicated by an asterisk (*) adjacent to the *Hits* information. *Cost* will be incremented when the recalculation is done.

csch rehash recomputes the internal hash table of the contents of directories listed in the path environmental variable to account for new commands added.

unhash disables the internal hash table.

hashstat prints a statistics line indicating how effective the internal hash table has been at locating commands (and avoiding execs). An exec is attempted for each component of the *path* where the hash function indicates a possible hit and in each component that does not begin with a '/ '.

ksh88 For each *name*, the location in the search path of the command specified by *name* is determined and remembered by the shell. The `-r` option to the hash built-in causes the shell to forget all remembered locations. If no arguments are given, hash provides information about remembered commands.

操作数 The following operand is supported by hash:

utility The name of a utility to be searched for and added to the list of remembered locations.

Output The standard output of hash is used when no arguments are specified. Its format is unspecified, but includes the pathname of each utility in the list of remembered locations for the current shell environment. This list consists of those utilities named in previous hash invocations that have been invoked, and may contain those invoked and found through the normal command search process.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of hash: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

PATH Determine the location of *utility*.

退出状态 The following exit values are returned by hash:

0 Successful completion.

>0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [csh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

- 引用名** head – display first few lines of files
- 用法概要** /usr/bin/head [-number | -n *number*] [*filename*]...
- 描述** The head utility copies the first *number* of lines of each *filename* to the standard output. If no *filename* is given, head copies lines from the standard input. The default value of *number* is 10 lines.
- When more than one file is specified, the start of each file looks like:
==> *filename* <==
- Thus, a common way to display a set of short files, identifying each one, is:
example% head -9999 filename1 filename2 ...
- 选项** The following options are supported:
- n *number* The first *number* lines of each input file is copied to standard output. The *number* option-argument must be a positive decimal integer.
 - number* The *number* argument is a positive decimal integer with the same effect as the -n *number* option.
- If no options are specified, head acts as if -n 10 had been specified.
- 操作数** The following operand is supported:
- filename* A path name of an input file. If no *file* operands are specified, the standard input is used.
- 用法** See [largefile\(5\)](#) for the description of the behavior of head when encountering files greater than or equal to 2 Gbyte (2³¹ bytes).
- 示例** 示例 1 Writing the First Ten Lines of All Files
- The following example writes the first ten lines of all files, except those with a leading period, in the directory:
- ```
example% head *
```
- 环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of head: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.
- 退出状态** The following exit values are returned:
- 0 Successful completion.
  - >0 An error occurred.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

## 另请参见

[cat\(1\)](#), [more\(1\)](#), [pg\(1\)](#), [tail\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

**引用名** history, fc, hist – process command history list

**用法概要** /usr/bin/fc [-r] [-e *editor*] [*first* [*last*]]  
/usr/bin/fc -l [-nr] [*first* [*last*]]  
/usr/bin/fc -s [*old=new*] [*first*]

csH history [-hr] [*n*]

ksh88 fc -e - [*old=new*] [*command*]

fc -s [*old = new*] [*command*]

fc [-e *ename*] [-n*lr*] [*first* [*last*]]

ksh hist [-lnprs] [-e *editor*][-N *num*][*first*[*last*]]

## 描述

/usr/bin/fc The fc utility lists or edits and reexecutes, commands previously entered to an interactive sh.

The command history list references commands by number. The first number in the list is selected arbitrarily. The relationship of a number to its command does not change except when the user logs in and no other process is accessing the list, at which time the system can reset the numbering to start the oldest retained command at another number (usually 1). When the number reaches the value in HISTSIZE or 32767 (whichever is greater), the shell can wrap the numbers, starting the next command with a lower number (usually 1). However, despite this optional wrapping of numbers, fc maintains the time-ordering sequence of the commands. For example, if four commands in sequence are given the numbers 32 766, 32 767, 1 (wrapped), and 2 as they are executed, command 32 767 is considered the command previous to 1, even though its number is higher.

When commands are edited (when the -l option is not specified), the resulting lines is entered at the end of the history list and then reexecuted by sh. The fc command that caused the editing is not entered into the history list. If the editor returns a non-zero exit status, this suppresses the entry into the history list and the command reexecution. Any command-line variable assignments or redirection operators used with fc affects both the fc command itself as well as the command that results, for example:

```
fc -s -- -1 2>/dev/null
```

reinvokes the previous command, suppressing standard error for both fc and the previous command.

csH Display the history list. If *n* is given, display only the *n* most recent events.

-r Reverse the order of printout to be most recent first rather than oldest first.

-h Display the history list without leading numbers. This is used to produce files suitable for sourcing using the -h option to the csH built-in command, [source\(1\)](#).

### History Substitution:

History substitution allows you to use words from previous command lines in the command line you are typing. This simplifies spelling corrections and the repetition of complicated commands or arguments. Command lines are saved in the history list, the size of which is controlled by the `history` variable. The `history` shell variable can be set to the maximum number of command lines that is saved in the history file, that is:

```
set history = 200
```

allows the history list to keep track of the most recent 200 command lines. If not set, the C shell saves only the most recent command.

A history substitution begins with a `!` (although you can change this with the `histchars` variable) and can occur anywhere on the command line; history substitutions do not nest. The `!` can be escaped with `\` to suppress its special meaning.

Input lines containing history substitutions are echoed on the terminal after being expanded, but before any other substitutions take place or the command gets executed.

### Event Designators:

An event designator is a reference to a command line entry in the history list.

|                                          |                                                                                                                                                                                                                                                                                                                                        |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>!</code>                           | Start a history substitution, except when followed by a space character, tab, newline, <code>=</code> or <code>(</code> .                                                                                                                                                                                                              |
| <code>!!</code>                          | Refer to the previous command. By itself, this substitution repeats the previous command.                                                                                                                                                                                                                                              |
| <code>!n</code>                          | Refer to command line <i>n</i> .                                                                                                                                                                                                                                                                                                       |
| <code>!-n</code>                         | Refer to the current command line minus <i>n</i> .                                                                                                                                                                                                                                                                                     |
| <code>!str</code>                        | Refer to the most recent command starting with <i>str</i> .                                                                                                                                                                                                                                                                            |
| <code>!?str?</code>                      | Refer to the most recent command containing <i>str</i> .                                                                                                                                                                                                                                                                               |
| <code>!?str? additional</code>           | Refer to the most recent command containing <i>str</i> and append <i>additional</i> to that referenced command.                                                                                                                                                                                                                        |
| <code>!{command} additional</code>       | Refer to the most recent command beginning with <i>command</i> and append <i>additional</i> to that referenced command.                                                                                                                                                                                                                |
| <code>^previous_word^replacement^</code> | Repeat the previous command line replacing the string <i>previous_word</i> with the string <i>replacement</i> . This is equivalent to the history substitution:<br><br>Repeat the previous command line replacing the string <i>previous_word</i> with the string <i>replacement</i> . This is equivalent to the history substitution: |

```
! :s/previous_word/replacement/ .
```

To re-execute a specific previous command *and* make such a substitution, say, re-executing command #6:

```
! :6s/previous_word/replacement/ .
```

### Word Designators:

A ':' (colon) separates the event specification from the word designator. It can be omitted if the word designator begins with a ^, \$, \*, - or %. If the word is to be selected from the previous command, the second ! character can be omitted from the event specification. For instance, !!:1 and !:1 both refer to the first word of the previous command, while !!\$ and !\$ both refer to the last word in the previous command. Word designators include:

|     |                                                                            |
|-----|----------------------------------------------------------------------------|
| #   | The entire command line typed so far.                                      |
| 0   | The first input word (command).                                            |
| n   | The n'th argument.                                                         |
| ^   | The first argument, that is, 1.                                            |
| \$  | The last argument.                                                         |
| %   | The word matched by (the most recent) ?s search.                           |
| x-y | A range of words; -y abbreviates 0-y.                                      |
| *   | All the arguments, or a null value if there is just one word in the event. |
| x*  | Abbreviates x-\$.                                                          |
| x-  | Like x* but omitting word \$.                                              |

### Modifiers:

After the optional word designator, you can add a sequence of one or more of the following modifiers, each preceded by a :

|                          |                                                                                                                                                                                                                       |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| h                        | Remove a trailing pathname component, leaving the head.                                                                                                                                                               |
| r                        | Remove a trailing suffix of the form '.xxx', leaving the basename.                                                                                                                                                    |
| e                        | Remove all but the suffix, leaving the extension.                                                                                                                                                                     |
| s/oldchars/replacements/ | Substitute <i>replacements</i> for <i>oldchars</i> . <i>oldchars</i> is a string that can contain embedded blank spaces, whereas <i>previous_word</i> in the event designator can not.<br><br>^oldchars^replacements^ |
| t                        | Remove all leading pathname components, leaving the tail.                                                                                                                                                             |

|   |                                                                                                             |
|---|-------------------------------------------------------------------------------------------------------------|
| & | Repeat the previous substitution.                                                                           |
| g | Apply the change to the first occurrence of a match in each word, by prefixing the above (for example, g&). |
| p | Print the new command but do not execute it.                                                                |
| q | Quote the substituted words, escaping further substitutions.                                                |
| x | Like q, but break into words at each space character, tab or newline.                                       |

Unless preceded by a g, the modification is applied only to the first string that matches *oldchars*. An error results if no string matches.

The left-hand side of substitutions are not regular expressions, but character strings. Any character can be used as the delimiter in place of /. A backslash quotes the delimiter character. The character &, in the right hand side, is replaced by the text from the left-hand-side. The & can be quoted with a backslash. A null *oldchars* uses the previous string either from a *oldchars* or from a contextual scan string *s* from !?s. You can omit the rightmost delimiter if a newline immediately follows *replacements*; the rightmost ? in a context scan can similarly be omitted.

Without an event specification, a history reference refers either to the previous command, or to a previous history reference on the command line (if any).

ksh88

Using `fc`, in the form of

```
fc -e - [old=new] [command],
```

or

```
fc -s [old=new] [command],
```

the *command* is re-executed after the substitution *old=new* is performed. If there is not a *command* argument, the most recent command typed at this terminal is executed.

Using `fc` in the form of

```
fc [-e ename] [-n|l r] [first [last]],
```

a range of commands from *first* to *last* is selected from the last HISTSIZE commands that were typed at the terminal. The arguments *first* and *last* can be specified as a number or as a string. A string is used to locate the most recent command starting with the given string. A negative number is used as an offset to the current command number. If the -l flag is selected, the commands are listed on standard output. Otherwise, the editor program -e *name* is invoked on a file containing these keyboard commands. If *ename* is not supplied, then the value of the variable FCEDIT (default /bin/ed) is used as the editor. When editing is complete, the edited command(s) is executed. If *last* is not specified, it is set to *first*. If *first* is not specified, the

default is the previous command for editing and `-16` for listing. The flag `-r` reverses the order of the commands and the flag `-n` suppresses command numbers when listing. (See [ksh88\(1\)](#) for more about command line editing.)

**HISTFILE** If this variable is set when the shell is invoked, then the value is the pathname of the file that is used to store the command history.

**HISTSIZE** If this variable is set when the shell is invoked, then the number of previously entered commands that are accessible by this shell is greater than or equal to this number. The default is 128.

### Command Re-entry:

The text of the last **HISTSIZE** (default 128) commands entered from a terminal device is saved in a `history` file. The file `$HOME/.sh_history` is used if the **HISTFILE** variable is not set or if the file it names is not writable. A shell can access the commands of all *interactive* shells which use the same named **HISTFILE**. The special command `fc` is used to list or edit a portion of this file. The portion of the file to be edited or listed can be selected by number or by giving the first character or characters of the command. A single command or range of commands can be specified. If you do not specify an editor program as an argument to `fc` then the value of the variable **FCEDIT** is used. If **FCEDIT** is not defined then `/bin/ed` is used. The edited command(s) is printed and re-executed upon leaving the editor. The editor name `-` is used to skip the editing phase and to re-execute the command. In this case a substitution parameter of the form `old=new` can be used to modify the command before execution. For example, if `r` is aliased to `'fc -e -'` then typing `'r bad=good c'` re-executes the most recent command which starts with the letter `c`, replacing the first occurrence of the string `bad` with the string `good`.

Using the `fc` built-in command within a compound command causes the whole command to disappear from the history file.

**ksh** `hist` lists, edits, or re-executes commands previously entered into the current shell environment.

The command history list references commands by number. The first number in the list is selected arbitrarily. The relationship of a number to its command does not change during a login session. When the number reaches 32767 the number wraps around to 1 but maintains the ordering.

When the `l` option is not specified, and commands are edited, the resulting lines are entered at the end of the history list and then re-executed by the current shell. The `hist` command that caused the editing is not entered into the history list. If the editor returns a non-zero exit status, this suppresses the entry into the history list and the command re-execution. Command line variable assignments and redirections affect both the `hist` command and the commands that are re-executed.

*first* and *last* define the range of commands. Specify *first* and *last* as one of the following:



- number* A positive number representing a command number. A + sign can precede *number*.
- number* A negative number representing a command that was executed *number* commands previously. For example, -1 is the previous command.
- string* *string* indicates the most recently entered command that begins with *string*. *string* should not contain an =.

If *first* is omitted, the previous command is used, unless -l is specified, in which case it defaults to -16 and last defaults to -1.

If *first* is specified and *last* is omitted, then *last* defaults to *first* unless -l is specified in which case it defaults to -1.

If no editor is specified, then the editor specified by the HISTEDIT variable is used if set, or the FCEDIT variable is used if set, otherwise, ed is used.

## 选项

The following options are supported:

- e *editor* Uses the editor named by *editor* to edit the commands. The *editor* string is a utility name, subject to search via the PATH variable. The value in the FCEDIT variable is used as a default when -e is not specified. If FCEDIT is null or unset, ed is used as the editor.
- l (The letter ell.) Lists the commands rather than invoking an editor on them. The commands is written in the sequence indicated by the *first* and *last* operands, as affected by -r, with each command preceded by the command number.
- n Suppresses command numbers when listing with -l.
- r Reverses the order of the commands listed (with -l) or edited (with neither -l nor -s).
- s Re-executes the command without invoking an editor.

## ksh

ksh supports the following options:

- e *editor* Specify the editor to use to edit the history command. A value of - for *editor* is equivalent to specifying the -s option.
- l List the commands rather than editing and re-executing them.
- N *num* Start at *num* commands back.
- n Suppress the command numbers when the commands are listed.
- p Write the result of history expansion for each operand to standard output. All other options are ignored.
- r Reverse the order of the commands.

- s Re-execute the command without invoking an editor. In this case an operand of the form *old=new* can be specified to change the first occurrence of the string *old* in the command to *new* before re-executing the command.

## 操作数

The following operands are supported:

*first*

*last*

Selects the commands to list or edit. The number of previous commands that can be accessed is determined by the value of the HISTSIZE variable. The value of *first* or *last* or both is one of the following:

[+]*number* A positive number representing a command number. Command numbers can be displayed with the -l option.

-*number* A negative decimal number representing the command that was executed *number* of commands previously. For example, -1 is the immediately previous command.

*string* A string indicating the most recently entered command that begins with that string. If the *old=new* operand is not also specified with -s, the string form of the *first* operand cannot contain an embedded equal sign.

When the synopsis form with -s is used, if *first* is omitted, the previous command is used.

For the synopsis forms without -s :

- If *last* is omitted, *last* defaults to the previous command when -l is specified; otherwise, it defaults to *first*.
- If *first* and *last* are both omitted, the previous 16 commands is listed or the previous single command is edited (based on the -l option).
- If *first* and *last* are both present, all of the commands from *first* to *last* is edited (without -l ) or listed (with -l ). Editing multiple commands is accomplished by presenting to the editor all of the commands at one time, each command starting on a new line. If *first* represents a newer command than *last*, the commands is listed or edited in reverse sequence, equivalent to using -r. For example, the following commands on the first line are equivalent to the corresponding commands on the second:

```
fc -r 10 20 fc 30 40
fc 20 10 fc -r 40 30
```

- When a range of commands is used, it is not be an error to specify *first* or *last* values that are not in the history list. `fc` substitutes the value representing the oldest or newest command in the list, as appropriate. For example, if there are only ten commands in the history list, numbered 1 to 10:

```
fc -l
fc 1 99
```

lists and edits, respectively, all ten commands.

*old=new* Replace the first occurrence of string *old* in the commands to be reexecuted by the string *new*.

## Output

When the `-l` option is used to list commands, the format of each command in the list is as follows:

```
"%d\t%s\n", <line number>, <command>
```

If both the `-l` and `-n` options are specified, the format of each command is:

```
"\t%s\n", <command>
```

If the *commandcommand* consists of more than one line, the lines after the first are displayed as:

```
"\t%s\n", <continued-command>
```

## 示例

示例 1 Using history and `fc`

| csh                          | ksh88                        |
|------------------------------|------------------------------|
| % history                    | \$ fc -l                     |
| 1 cd /etc                    | 1 cd /etc                    |
| 2 vi passwd                  | 2 vi passwd                  |
| 3 date                       | 3 date                       |
| 4 cd                         | 4 cd                         |
| 5 du .                       | 5 du .                       |
| 6 ls -t                      | 6 ls -t                      |
| 7 history                    | 7 fc -l                      |
| % !d                         | \$ fc -e - d                 |
| du .                         | du .                         |
| 262 ./SCCS                   | 262 ./SCCS                   |
| 336 .                        | 336 .                        |
| % !da                        | \$ fc -e - da                |
| Thu Jul 21 17:29:56 PDT 1994 | Thu Jul 21 17:29:56 PDT 1994 |
| %                            | \$ alias \!= 'fc -e -'       |

示例 1 Using history and fc (续)

```
% !! $!
 date alias = 'fc -e -'
 Thu Jul 21 17:29:56 PDT 1994
```

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of fc: LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

- FCEDIT** This variable, when expanded by the shell, determines the default value for the `-e editor` option's *editor* option-argument. If FCEDIT is null or unset, [ed\(1\)](#) is used as the editor.
- HISTFILE** Determine a pathname naming a command history file. If the HISTFILE variable is not set, the shell can attempt to access or create a file `.sh_history` in the user's home directory. If the shell cannot obtain both read and write access to, or create, the history file, it uses an unspecified mechanism that allows the history to operate properly. (References to history "file" in this section are understood to mean this unspecified mechanism in such cases.) fc can choose to access this variable only when initializing the history file; this initialization occurs when fc or sh first attempt to retrieve entries from, or add entries to, the file, as the result of commands issued by the user, the file named by the ENV variable, or a system startup file such as `/etc/profile`. (The initialization process for the history file can be dependent on the system startup files, in that they can contain commands that effectively preempts the user's settings of HISTFILE and HISTSIZE. For example, function definition commands are recorded in the history file, unless the `set -o noLog` option is set. If the system administrator includes function definitions in some system startup file called before the ENV file, the history file is initialized before the user gets a chance to influence its characteristics.) The variable HISTFILE is accessed initially when the shell is invoked. Any changes to HISTFILE does not take effect until another shell is invoked.
- HISTSIZE** Determine a decimal number representing the limit to the number of previous commands that are accessible. If this variable is unset, an unspecified default greater than or equal to 128 are used. The variable HISTSIZE is accessed initially when the shell is invoked. Any changes to HISTSIZE does not take effect until another shell is invoked.

## 退出状态

The following exit values are returned:

- 0 Successful completion of the listing.
- >0 An error occurred.

---

Otherwise, the exit status is that of the commands executed by `fc` or `hist`.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

## 另请参见

[csh\(1\)](#), [ed\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [set\(1\)](#), [sh\(1\)](#), [source\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#)

**引用名**                    hostid – print the numeric identifier of the current host

**用法概要**                    /usr/bin/hostid

**描述**                        The `hostid` command prints the identifier of the current host in hexadecimal. If it is executed within a non-global zone that emulates a host identifier, the emulated host identifier is printed. This numeric value is likely to differ when `hostid` is run on a different machine.

**属性**                        See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

**另请参见**                    [sysinfo\(2\)](#), [gethostid\(3C\)](#), [attributes\(5\)](#), [zones\(5\)](#)

**引用名** hostname – set or print name of current host system

**用法概要** /usr/bin/hostname [[-t] *name-of-host*]

**描述** The `hostname` command prints the name of the current host, as given before the login prompt. The super-user can set the hostname by giving an argument. The change of the hostname is permanent unless the `-t` option is specified.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

**另请参见** [uname\(1\)](#), [nodename\(4\)](#), [attributes\(5\)](#)

**引用名** iconv – code set conversion utility

**用法概要** iconv [-cs] -f *frommap* -t *tomap* [*file*]...  
iconv -f *fromcode* [-cs] [-t *tocode*] [*file*]...  
iconv -t *tocode* [-cs] [-f *fromcode*] [*file*]...  
iconv -l

**描述** The iconv utility converts the characters or sequences of characters in *file* from one code set to another and writes the results to standard output. If no conversion exists for a particular character, an implementation-defined conversion is performed on this character.

The list of supported conversions and the locations of the associated conversion tables are provided in the [iconv\(5\)](#) manual page.

**选项** The following options are supported:

-c Omits any characters that are invalid in the codeset of the input file from the output. When -c is not used, the results of encountering invalid characters in the input stream depend on the specified codesets for the conversion. Invalid characters can be either those that are not valid characters in the codeset of the input file or those that have no corresponding character in the codeset of the output file. The presence or absence of -c does not affect the exit status of iconv. When *fromcode* is specified for the *fromcodeset* of the -f option or *tocode* is specified for the *tocodeset* of the -t option, the specification of -c may be ignored.

-f *fromcodeset* Identifies the code set of the input file. The following two forms of the *fromcodeset* option-argument are recognized:

*fromcode* The *fromcode* option-argument must not contain a slash (/) character. It is interpreted as the name of one of the codeset descriptions.

*frommap* The *frommap* option-argument must contain a slash character. It is interpreted as the pathname of a charmap file as defined in [charmap\(5\)](#). If the pathname does not represent a valid, readable charmap file, the results are undefined.

If this option is omitted, the codeset of the current locale is used.

-l Writes all supported *fromcode* and *tocode* values to standard output.

-s Suppresses any messages written to standard error concerning invalid characters. When -s is not used, the results of encountering invalid characters in the input stream depend on the specified codesets for the conversion. Invalid characters can be either those that are not valid characters in the codeset of the input file or those that have no



corresponding character in the codeset of the output file. The presence or absence of `-s` does not affect the exit status of `iconv`. When `fromcode` is specified for the `fromcodeset` of the `-f` option or `tocode` is specified for the `tocodeset` of the `-t` option, the specification of `-s` may be ignored.

`-t tocodeset` Identifies the code set used for the output file. The following two forms of the `tocodeset` option-argument are recognized:

`tocode` The `tocode` option-argument must not contain a slash (`/`) character. It is interpreted as the name of one of the codeset descriptions.

`tomap` The `tomap` option-argument must contain a slash character. It is interpreted as the pathname of a charmap file as defined in [charmap\(5\)](#). If the pathname does not represent a valid, readable charmap file, the results are undefined.

If this option is omitted, the codeset of the current locale is used.

If either `-f` or `-t` represents a charmap file but the other does not, or is omitted, or if both `-f` and `-t` are omitted, `iconv` fails as an error.

## 操作数

The following operands are supported:

`file` A path name of an input file. If no file operands are specified, or if a file operand is `'-'`, the standard input is used.

## 示例

示例 1 Converting and storing files

The following example converts the contents of file `mail1` from code set `8859` to `646fr` and stores the results in file `mail.local`:

```
example% iconv -f 8859 -t 646fr mail1 > mail.local
```

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `iconv`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

## 退出状态

The following exit values are returned:

0 Successful completion.

1 An error has occurred.

## 文件

`/usr/lib/iconv/iconv_data` list of conversions supported by conversion tables

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

## 另请参见

[iconv\(3C\)](#), [iconv\\_open\(3C\)](#), [attributes\(5\)](#), [charmap\(5\)](#), [environ\(5\)](#), [iconv\(5\)](#), [iconv\\_unicode\(5\)](#), [standards\(5\)](#)

## 附注

Make sure that both charmap files use the same symbolic names for characters the two codesets have in common.

The output format of the `-l` option is unspecified. The `-l` option is not intended for shell script usage.

When *fromcode* or *tocode* is specified for the codeset conversion, `iconv` uses the [iconv\\_open\(3C\)](#) function. If `iconv_open(3C)` fails to open the specified codeset conversion, `iconv` searches for an appropriate conversion table. As for the supported codeset conversion by `iconv_open(3C)`, please refer to [iconv\(5\)](#) and [iconv\\_locale\(5\)](#).

**引用名** indxbib – create an inverted index to a bibliographic database

**用法概要** indxbib *database-file*...

**描述** indxbib makes an inverted index to the named *database-file* (which must reside within the current directory), typically for use by [lookbib\(1\)](#) and [refer\(1\)](#). A *database* contains bibliographic references (or other kinds of information) separated by blank lines.

A bibliographic reference is a set of lines, constituting fields of bibliographic information. Each field starts on a line beginning with a '%', followed by a key-letter, then a blank, and finally the contents of the field, which may continue until the next line starting with '%'.

indxbib is a shell script that calls two programs: /usr/lib/refer/mkey and /usr/lib/refer/inv. mkey truncates words to 6 characters, and maps upper case to lower case. It also discards words shorter than 3 characters, words among the 100 most common English words, and numbers (dates) < 1000 or > 2099. These parameters can be changed.

indxbib creates an entry file (with a .ia suffix), a posting file (.ib), and a tag file (.ic), in the working directory.

**文件**

/usr/lib/refer/mkey

/usr/lib/refer/inv

*x.ia* entry file

*x.ib* posting file

*x.ic* tag file

*x.ig* reference file

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | text/doctools   |

**另请参见**

[addbib\(1\)](#), [lookbib\(1\)](#), [refer\(1\)](#), [roffb\(1\)](#), [sortbib\(1\)](#), [attributes\(5\)](#)

**已知问题**

All dates should probably be indexed, since many disciplines refer to literature written in the 1800s or earlier.

indxbib does not recognize pathnames.

引用名           install – install files

用法概要

```
/usr/ucb/install [-cs] [-g group] [-m mode]
 [-o owner] filename1 filename2

/usr/ucb/install [-cs] [-g group] [-m mode]
 [-o owner] filename... directory

/usr/ucb/install -d [-g group] [-m mode]
 [-o owner] directory
```

描述

`install` is used within makefiles to copy new versions of files into a destination directory and to create the destination directory itself.

The first two forms are similar to the [cp\(1\)](#) command with the addition that executable files can be stripped during the copy and the owner, group, and mode of the installed file(s) can be given.

The third form can be used to create a destination directory with the required owner, group and permissions.

Note: `install` uses no special privileges to copy files from one place to another. The implications of this are:

- You must have permission to read the files to be installed.
- You must have permission to copy into the destination file or directory.
- You must have permission to change the modes on the final copy of the file if you want to use the `-m` option to change modes.
- You must be superuser if you want to specify the ownership of the installed file with `-o`. If you are not the super-user, or if `-o` is not in effect, the installed file will be owned by you, regardless of who owns the original.

选项

- c           Copy files. In fact `install` *always* copies files, but the `-c` option is retained for backwards compatibility with old shell scripts that might otherwise break.
- d           Create a directory. Missing parent directories are created as required as in `mkdir -p`. If the directory already exists, the owner, group and mode will be set to the values given on the command line.
- s           Strip executable files as they are copied.
- g *group*   Set the group ownership of the installed file or directory. (staff by default.)
- m *mode*    Set the mode for the installed file or directory. (0755 by default.)
- o *owner*    If run as root, set the ownership of the installed file to the user-ID of *owner*.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

---

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

另请参见

[chgrp\(1\)](#), [chmod\(1\)](#), [chown\(1\)](#), [cp\(1\)](#), [mkdir\(1\)](#), [strip\(1\)](#), [install\(1M\)](#), [attributes\(5\)](#)

**引用名** ipcrm – 删除消息队列、信号集或共享内存 ID

**用法概要** ipcrm [-z *zone*] [-m *shmid*] [-q *msqid*] [-s *semid*]  
[-M *shmkey*] [-Q *msgkey*] [-S *semkey*]

**描述** ipcrm 删除一个或多个消息、信号或共享内存标识符。

**选项** 支持以下选项：

-z *zone* 其他选项指定的键引用指定区域中的工具（请参见 [zones\(5\)](#)）。缺省值是在其中执行此命令的区域。此选项仅适用于在全局区域中执行命令的情况。

标识符由以下选项指定：

-m *shmid* 从系统中删除共享内存标识符 *shmid*。最后分离之后，与其关联的共享内存区段和数据结构将会销毁。

-q *msqid* 从系统中删除消息队列标识符 *msqid* 并销毁与其关联的消息队列和数据结构。

-s *semid* 从系统中删除信号标识符 *semid* 并销毁与其关联的信号集和数据结构。

-M *shmkey* 从系统中删除使用键 *shmkey* 创建的共享内存标识符。最后分离之后，与其关联的共享内存区段和数据结构将会销毁。

-Q *msgkey* 从系统中删除使用键 *msgkey* 创建的消息队列标识符，并销毁与其关联的消息队列和数据结构。

-S *semkey* 从系统中删除使用键 *semkey* 创建的信号标识符，并销毁与其关联的信号集和数据结构。

[msgctl\(2\)](#)、[shmctl\(2\)](#) 和 [semctl\(2\)](#) 中介绍了删除操作的详细信息。使用 `ipcs` 命令查找标识符和键。

**环境变量** 有关影响 ipcrm 执行的以下环境变量的描述，请参见 [environ\(5\)](#)：LANG、LC\_ALL、LC\_CTYPE、LC\_MESSAGES 和 NLSPATH。

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值                                |
|-------|------------------------------------|
| 可用性   | system/core-os                     |
| 接口稳定性 | Committed（已确定）                     |
| 标准    | 请参见 <a href="#">standards(5)</a> 。 |

**另请参见** [ipcs\(1\)](#)、[msgctl\(2\)](#)、[msgget\(2\)](#)、[msgrcv\(2\)](#)、[msgsnd\(2\)](#)、[semctl\(2\)](#)、[semget\(2\)](#)、[semop\(2\)](#)、[shmctl\(2\)](#)

**引用名** ipcs – report inter-process communication facilities status

**用法概要** ipcs [-aAbciJmopqstZ] [-D *mtype*] [-z *zone*]

**描述** The `ipcs` utility prints information about active inter-process communication facilities. The information that is displayed is controlled by the options supplied. Without options, information is printed in short format for message queues, shared memory, and semaphores that are currently active in the system.

**选项** The following options are supported:

- m Prints information about active shared memory segments.
- q Prints information about active message queues.
- s Prints information about active semaphores.

If `-m`, `-q`, or `-s` are specified, information about only those indicated is printed. If none of these three is specified, information about all three is printed subject to these options:

- a Uses all XCU5 print options. (This is a shorthand notation for `-b`, `-c`, `-o`, `-p`, and `-t`.)
- A Uses all print options. (This is a shorthand notation for `-b`, `-c`, `-i`, `-J`, `-o`, `-p`, and `-t`.)
- b Prints information on biggest allowable size: maximum number of bytes in messages on queue for message queues, size of segments for shared memory, and number of semaphores in each set for semaphores. See below for meaning of columns in a listing.
- c Prints creator's login name and group name. See below.
- D *mtype* Displays, in hexadecimal and ASCII, the contents of all messages of type *mtype* found on any message queue that the user invoking `ipcs` has permission to read. If *mtype* is `0`, all messages are displayed. If *mtype* is negative, all messages with type less than or equal to the absolute value of *mtype* are displayed. (See [msgrcv\(2\)](#) and [msgsnap\(2\)](#)).
- i Prints number of ISM attaches to shared memory segments.
- J Prints the creator's project.
- o Prints information on outstanding usage: number of messages on queue and total number of bytes in messages on queue for message queues and number of processes attached to shared memory segments.
- p Prints process number information: process ID of last process to send a message, process ID of last process to receive a message on message queues, process ID of creating process, and process ID of last process to attach or detach on shared memory segments. See below.

- t            Prints time information: time of the last control operation that changed the access permissions for all facilities, time of last `msgsnd(2)` and last `msgrcv(2)` on message queues, time of last `shmat(2)` and last `shmdt(2)` on shared memory (see `shmop(2)`), time of last `semop(2)` on semaphores. See below.
- z *zone*     Prints information about facilities associated with the specified zone (see `zones(5)`). The zone can be specified as either a name or a numeric id. The default is to display information about the zone in which the command is executing. Notice that this option is only useful when executing in the global zone.
- Z            When executing in the global zone, prints information about all zones. Otherwise, prints information about the zone in which the command is executing. The output includes the zone associated with each facility.

The column headings and the meaning of the columns in an `ipcs` listing are given below. The letters in parentheses indicate the options that cause the corresponding heading to appear and “all” means that the heading always appears. *Note:* These options only determine what information is provided for each facility; they do not determine which facilities are listed.

- T (all)            Type of the facility:
- q    message queue
  - m    shared memory segment
  - s    semaphore
- ID (all)            The identifier for the facility entry.
- KEY (all)           The key used as an argument to `msgget(2)`, `semget(2)`, or `shmget(2)` to create the facility entry. (*Note:* The key of a shared memory segment is changed to `IPC_PRIVATE` when the segment has been removed until all processes attached to the segment detach it.)
- MODE (all)         The facility access modes and flags: The mode consists of 11 characters that are interpreted as follows. The first two characters are:
- R    A process is waiting on a `msgrcv(2)`.
  - S    A process is waiting on a `msgsnd(2)`.
  - The corresponding special flag is not set.

The next nine characters are interpreted as three sets of three bits each. The first set refers to the owner's permissions; the next to permissions of others in the user-group of the facility entry; and the last to all others. Within each set, the first character indicates permission to read, the second character indicates permission to write or alter the facility entry, and the last character is currently unused.



The permissions are indicated as follows:

- r Read permission is granted.
- w Write permission is granted.
- a Alter permission is granted.
- The indicated permission is not granted.

|                 |                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------|
| OWNER (all)     | The login name of the owner of the facility entry.                                           |
| GROUP (all)     | The group name of the group of the owner of the facility entry.                              |
| CREATOR (a,A,c) | The login name of the creator of the facility entry.                                         |
| CGROUP (a,A,c)  | The group name of the group of the creator of the facility entry.                            |
| CBYTES (a,A,o)  | The number of bytes in messages currently outstanding on the associated message queue.       |
| QNUM (a,A,o)    | The number of messages currently outstanding on the associated message queue.                |
| QBYTES (a,A,b)  | The maximum number of bytes allowed in messages outstanding on the associated message queue. |
| LSPID (a,A,p)   | The process ID of the last process to send a message to the associated queue.                |
| LRPID (a,A,p)   | The process ID of the last process to receive a message from the associated queue.           |
| STIME (a,A,t)   | The time the last message was sent to the associated queue.                                  |
| RTIME (a,A,t)   | The time the last message was received from the associated queue.                            |
| CTIME (a,A,t)   | The time when the associated entry was created or changed.                                   |
| ISMATCH (a,i)   | The number of ISM attaches to the associated shared memory segments.                         |
| NATCH (a,A,o)   | The number of processes attached to the associated shared memory segment.                    |
| SEGSZ (a,A,b)   | The size of the associated shared memory segment.                                            |
| CPID (a,A,p)    | The process ID of the creator of the shared memory entry.                                    |
| LPID (a,A,p)    | The process ID of the last process to attach or detach the shared memory segment.            |
| ATIME (a,A,t)   | The time the last attach was completed to the associated shared memory segment.              |

|               |                                                                                                     |
|---------------|-----------------------------------------------------------------------------------------------------|
| DTIME (a,A,t) | The time the last detach was completed on the associated shared memory segment.                     |
| NSEMS (a,A,b) | The number of semaphores in the set associated with the semaphore entry.                            |
| OTIME (a,A,t) | The time the last semaphore operation was completed on the set associated with the semaphore entry. |
| PROJECT (J,A) | The project name of the creator of the facility entry.                                              |
| ZONE (Z)      | The zone with which the facility is associated.                                                     |

**环境变量**

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `ipcs`: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

TZ Determine the timezone for the time strings written by `ipcs`.

**文件**

/etc/group group names

/etc/passwd user names

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见**

[ipcrm\(1\)](#), [msgget\(2\)](#), [msgids\(2\)](#), [msgrcv\(2\)](#), [msgsnap\(2\)](#), [msgsnd\(2\)](#), [semget\(2\)](#), [semids\(2\)](#), [semop\(2\)](#), [shmctl\(2\)](#), [shmget\(2\)](#), [shmid\(2\)](#), [shmop\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#), [zones\(5\)](#)

**附注**

Things can change while `ipcs` is running. The information it gives is guaranteed to be accurate only when it was retrieved.

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>引用名</b>  | isainfo – 描述指令集合的体系结构                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>用法概要</b> | isainfo [ [-v] [-b   -n   -k]   [-x]]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>描述</b>   | <p>isainfo 实用程序用于识别当前正在运行的系统支持的指令集体系结构的各种属性。它可以回答下列问题，是否支持 64 位应用程序、正在运行的内核使用的是 32 位设备驱动程序还是 64 位设备驱动程序，等等。</p> <p>不帶有任何选项进行调用时，isainfo 会输出当前版本的操作系统所支持的应用程序的本机指令集名称。这些是 isalist(1) 返回的列表的一个子集。该子集对应于当前正在运行的系统所支持的基本应用程序环境。</p>                                                                                                                                                                                                                                                                                                                                                                    |
| <b>选项</b>   | <p>支持以下选项：</p> <ul style="list-style-type: none"> <li>-b 列显本机指令集的地址空间中的位数。</li> <li>-k 输出操作系统内核组件（例如设备驱动程序和 STREAMS 模块）所使用的指令集的名称。</li> <li>-n 输出当前版本的操作系统支持的可移植应用程序所使用的本机指令集的名称。</li> <li>-v 当与 -b、-k 或 -n 选项结合使用时，可以输出更详细的信息。</li> <li>-x 输出平台所支持的本机 ABI 的指令扩展。</li> </ul>                                                                                                                                                                                                                                                                                                                       |
| <b>示例</b>   | <p><b>示例 1</b> 在 32 位 x86 平台上调用 isainfo</p> <p>以下示例在 32 位 x86 平台上调用 isainfo：</p> <pre>example% isainfo -v 32-bit i386 applications  example% isainfo -k i386</pre> <p><b>示例 2</b> 从在 64 位 SPARC 处理器上运行 64 位操作系统的系统上调用 isainfo</p> <p>以下示例从在 64 位 SPARC 处理器上运行 64 位操作系统的系统上调用 isainfo：</p> <pre>example% isainfo sparcv9 sparc example% isainfo -n sparcv9 example% isainfo -v 64-bit sparcv9 applications 32-bit sparc applications example% isainfo -vk 64-bit sparcv9 kernel modules</pre> <p><b>示例 3</b> 在 AMD Opteron CPU 上调用 isainfo -x</p> <p>以下示例在 AMD Opteron CPU 上使用 -x 选项调用 isainfo：</p> |

示例3 在AMD Opteron CPU上调用 `isainfo -x` (续)

```
example% isainfo -x
i386: fpu tsc cx8 sep cmov mmx ammx a3dnow a3dnowx fxsr sse sse2 pause
```

退出状态

非零值 未正确指定选项，或者命令无法识别运行它的系统的属性。会将一条错误消息输出到 `stderr`。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

另请参见

[isalist\(1\)](#)、[uname\(1\)](#)、[psrinfo\(1M\)](#)、[getisax\(2\)](#)、[sysinfo\(2\)](#)、[attributes\(5\)](#)、[isalist\(5\)](#)

**引用名** isalist – 显示在此平台上可执行的本机指令集

**用法概要** isalist

**描述** isalist 在标准输出上列出在此平台上可执行的本机指令集的名称，与由 [sysinfo\(2\)](#) 的 SI\_ISALIST 命令返回的结果相同。

名称以空格分隔，并按性能从最佳到最差的顺序排列。也就是说，较早列出的指令集可能比较晚列出的指令集包含更多的指令；在此计算机上，针对较早列出的指令集编译的程序将会比针对较晚列出的指令集编译的同一程序运行速度要快。

为列表中未显示的指令集编译的程序可能会出现性能下降，或无法在此计算机上运行。

[isalist\(5\)](#) 中列出了系统已知的指令集名称。这些名称可能与 C 语言编译系统中的预定义名称或编译器选项匹配，也可能不匹配。

此命令已过时，可能会在将来版本的 Solaris 中删除。有关更好地处理指令集扩展的方式，请参见 [isainfo\(1\)](#)。

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

**另请参见** [isainfo\(1\)](#)、[optisa\(1\)](#)、[uname\(1\)](#)、[sysinfo\(2\)](#)、[attributes\(5\)](#)、[isalist\(5\)](#)

**引用名** jobs, fg, bg, stop, notify – control process execution

## 用法概要

```
sh jobs [-p | -l] [% job_id...]
 jobs -x command [arguments]
 fg [% job_id...]
 bg [% job_id...]
 stop % job_id...
 stop pid...
csh jobs [-l]
 fg [% job_id]
 bg [% job_id]...
 notify [% job_id]...
 stop % job_id...
 stop pid...
ksh88 jobs [-lnp] [% job_id...]
 fg [% job_id...]
 bg [% job_id...]
 stop % job_id...
 stop pid...
ksh jobs [-lnp] [job_id...]
 fg [job_id...]
 bg [job_id...]
```

## 描述

sh When Job Control is enabled, the Bourne shell built-in jobs reports all jobs that are stopped or executing in the background. If *job\_id* is omitted, all jobs that are stopped or running in the background is reported. The following options modify or enhance the output of jobs:

- l Reports the process group ID and working directory of the jobs.
- p Reports only the process group ID of the jobs.
- x Replaces any *job\_id* found in *command* or *arguments* with the corresponding process group ID, and then executes *command* passing it *arguments*.

When the shell is invoked as `jsh`, Job Control is enabled in addition to all of the functionality described previously for `sh`. Typically Job Control is enabled for the interactive shell only. Non-interactive shells typically do not benefit from the added functionality of Job Control.

With Job Control enabled every command or pipeline the user enters at the terminal is called a *job\_id*. All jobs exist in one of the following states: foreground, background or stopped. These terms are defined as follows:

1. A job in the *foreground* has read and write access to the controlling terminal.
2. A job in the *background* is denied read access and has conditional write access to the controlling terminal (see [stty\(1\)](#))
3. A *stopped* job is a job that has been placed in a suspended state, usually as a result of a SIGTSTP signal (see [signal.h\(3HEAD\)](#)).

Every job that the shell starts is assigned a positive integer, called a *job\_id number* which is tracked by the shell and are used as an identifier to indicate a specific job. Additionally, the shell keeps track of the *current* and *previous* jobs. The *current job* is the most recent job to be started or restarted. The *previous job* is the first non-current job.

The acceptable syntax for a Job Identifier is of the form:

`%job_id`

where *job\_id* can be specified in any of the following formats:

- `%` or `+`        for the current job
- `-`                for the previous job
- `?<string>`       specify the job for which the command line uniquely contains *string*.
- `n`                for job number *n*, where *n* is a job number
- `pref`            where *pref* is a unique prefix of the command name (for example, if the command `ls -l` name were running in the background, it could be referred to as `%ls`); *pref* cannot contain blanks unless it is quoted.

When Job Control is enabled, `fg` resumes the execution of a stopped job in the foreground, also moves an executing background job into the foreground. If `%job_id` is omitted the current job is assumed.

When Job Control is enabled, `bg` resumes the execution of a stopped job in the background. If `%job_id` is omitted the current job is assumed.

`stop` stops the execution of a background job(s) by using its *job\_id*, or of any process by using its *pid*; see [ps\(1\)](#).

csH The C shell built-in, `jobs`, without an argument, lists the active jobs under job control.

-l List process IDs, in addition to the normal information.

The shell associates a numbered *job\_id* with each command sequence to keep track of those commands that are running in the background or have been stopped with TSTP signals (typically Control-Z). When a command or command sequence (semicolon-separated list) is started in the background using the `&` metacharacter, the shell displays a line with the job number in brackets and a list of associated process numbers:

```
[1] 1234
```

To see the current list of jobs, use the `jobs` built-in command. The job most recently stopped (or put into the background if none are stopped) is referred to as the *current* job and is indicated with a '+'. The previous job is indicated with a '-'; when the current job is terminated or moved to the foreground, this job takes its place (becomes the new current job).

To manipulate jobs, refer to the `bg`, `fg`, `kill`, `stop`, and `%` built-in commands.

A reference to a job begins with a '%'. By itself, the percent sign refers to the current job.

%%+ %% The current job.

%- The previous job.

%j Refer to job *j* as in: 'kill -9 %j'. *j* can be a job number, or a string that uniquely specifies the command line by which it was started; 'fg %vi' might bring a stopped *vi* job to the foreground, for instance.

??string Specify the job for which the command line uniquely contains *string*.

A job running in the background stops when it attempts to read from the terminal. Background jobs can normally produce output, but this can be suppressed using the 'stty tostop' command.

`fg` brings the current or specified *job\_id* into the foreground.

`bg` runs the current or specified jobs in the background.

`stop` stops the execution of a background job(s) by using its *job\_id*, or of any process by using its *pid*; see [ps\(1\)](#).

`notify` notifies the user asynchronously when the status of the current job or specified jobs changes.

ksh88 `jobs` displays the status of the jobs that were started in the current shell environment. When `jobs` reports the termination status of a job, the shell removes its process ID from the list of those known in the current shell execution environment.



*job\_id* specifies the jobs for which the status is to be displayed. If no *job\_id* is specified, the status information for all jobs are displayed.

The following options modify or enhance the output of jobs:

- l (The letter ell.) Provides more information about each job listed. This information includes the job number, current job, process group ID, state and the command that formed the job.
- n Displays only jobs that have stopped or exited since last notified.
- p Displays only the process IDs for the process group leaders of the selected jobs.

By default, jobs displays the status of all the stopped jobs, running background jobs, and all jobs whose status has changed and have not been reported by the shell.

If the `monitor` option of the `set` command is turned on, an interactive shell associates a job with each pipeline. It keeps a table of current jobs, printed by the `jobs` command, and assigns them small integer numbers. When a job is started asynchronously with `&`, the shell prints a line which looks like:

```
[1] 1234
```

indicating that the job, which was started asynchronously, was job number 1 and had one (top-level) process, whose process id was 1234.

If you are running a job and wish to do something else you can hit the key `^Z` (Control-Z) which sends a `STOP` signal to the current job. The shell then normally indicates that the job has been “Stopped” (see `OUTPUT` below), and print another prompt. You can then manipulate the state of this job, putting it in the background with the `bg` command, or run some other commands and then eventually bring the job back into the foreground with the foreground command `fg`. A `^Z` takes effect immediately and is like an interrupt, in that pending output and unread input are discarded when it is typed.

There are several ways to refer to jobs in the shell. A job can be referred to by the process id of any process of the job or by one of the following:

- %number* The job with the specified number.
- %string* Any job whose command line begins with *string*; works only in the interactive mode when the history file is active.
- %?string* Any job whose command line contains *string*; works only in the interactive mode when the history file is active.
- %* Current job.
- %+* Equivalent to *%*.
- %-* Previous job.

The shell learns immediately whenever a process changes state. It normally informs you whenever a job becomes blocked so that no further progress is possible, but only just before it prints a prompt. This is done so that it does not otherwise disturb your work. When the monitor mode is on, each background job that completes triggers any trap set for CHLD. When you try to leave the shell while jobs are running or stopped, you are warned that 'You have stopped (running) jobs.' You can use the `jobs` command to see what they are. If you do this or immediately try to exit again, the shell does not warn you a second time, and the stopped jobs are terminated.

`fg` moves a background job from the current environment into the foreground. Using `fg` to place a job in the foreground removes its process ID from the list of those known in the current shell execution environment. The `fg` command is available only on systems that support job control. If *job\_id* is not specified, the current job is brought into the foreground.

`bg` resumes suspended jobs from the current environment by running them as background jobs. If the job specified by *job\_id* is already a running background job, `bg` has no effect and exits successfully. Using `bg` to place a job into the background causes its process ID to become 'known in the current shell execution environment, as if it had been started as an asynchronous list. The `bg` command is available only on systems that support job control. If *job\_id* is not specified, the current job is placed in the background.

`stop` stops the execution of a background job(s) by using its *job\_id*, or of any process by using its *pid*. See [ps\(1\)](#).

`ksh` `jobs` displays information about specified jobs that were started by the current shell environment on standard output. The information contains the job number enclosed in `[ . . . ]`, the status, and the command line that started the job.

If *job\_id* is omitted, `jobs` displays the status of all stopped jobs, background jobs, and all jobs whose status has changed since last reported by the shell.

When `jobs` reports the termination status of a job, the shell removes the job from the list of known jobs in the current shell environment.

The following options modify or enhances the output of `jobs`:

- l     Displays process IDs after the job number in addition to the usual information.
- n     Displays only the jobs whose status has changed since the last prompt was displayed.
- p     Displays the process group leader IDs for the specified jobs.

*job\_id* can be specified to `jobs`, `fg`, and `bg` as one of the following:

- number*     The process id of job.
- number*     The process group id of job.

|                 |                                                |
|-----------------|------------------------------------------------|
| <i>%number</i>  | The job number.                                |
| <i>%string</i>  | The job whose name begins with <i>string</i> . |
| <i>%?string</i> | The job whose name contains <i>string</i> .    |
| <i>%+</i>       |                                                |
| <i>%%</i>       | The current job.                               |
| <i>%-</i>       | The previous job.                              |

*fg* places the specified jobs into the foreground in sequence and sends a `CONT` signal to start each running. If *job\_id* is omitted, the most recently started or stopped background job is moved to the foreground.

*bg* places the specified jobs into the background and sends a `CONT` signal to start them running. If *job\_id* is omitted, the most recently started or stopped background job is resumed or continued in the background.

## Output

If the `-p` option is specified, the output consists of one line for each process ID:

```
"%d\n", "process ID"
```

Otherwise, if the `-l` option is not specified, the output is a series of lines of the form:

```
"[%d] %c %s %s\n", job-number, current, state, command
```

where the fields are as follows:

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| <i>current</i>    | The character <code>+</code> identifies the job that would be used as a default for the <code>fg</code> or <code>bg</code> commands. This job can also be specified using the <i>job_id</i> <code>%%+</code> or <i>%%</i> . The character <code>-</code> identifies the job that would become the default if the current default job were to exit; this job can also be specified using the <i>job_id</i> <code>%%-</code> . For other jobs, this field is a space character. At most, one job can be identified with <code>+</code> and at most one job can be identified with <code>-</code> . If there is any suspended job, then the current job is a suspended job. If there are at least two suspended jobs, then the previous job is also a suspended job. |                                                                               |
| <i>job-number</i> | A number that can be used to identify the process group to the <code>wait</code> , <code>fg</code> , <code>bg</code> , and <code>kill</code> utilities. Using these utilities, the job can be identified by prefixing the job number with <code>%</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                                               |
| <i>state</i>      | One of the following strings in the POSIX Locale:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                               |
|                   | Running                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Indicates that the job has not been suspended by a signal and has not exited. |
|                   | Done                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Indicates that the job completed and returned exit status zero.               |

|                     |                                                                                                                                                    |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Done( <i>code</i> ) | Indicates that the job completed normally and that it exited with the specified non-zero exit status, <i>code</i> , expressed as a decimal number. |
| Stopped             | Indicates that the job was stopped.                                                                                                                |
| Stopped (SIGTSTP)   | Indicates that the job was suspended by the SIGTSTP signal.                                                                                        |
| Stopped (SIGSTOP)   | Indicates that the job was suspended by the SIGSTOP signal.                                                                                        |
| Stopped (SIGTTIN)   | Indicates that the job was suspended by the SIGTTIN signal.                                                                                        |
| Stopped (SIGTTOU)   | Indicates that the job was suspended by the SIGTTOU signal.                                                                                        |

The implementation can substitute the string Suspended in place of Stopped. If the job was terminated by a signal, the format of *state* is unspecified, but it is visibly distinct from all of the other *state* formats shown here and indicates the name or description of the signal causing the termination.

*command* The associated command that was specified to the shell.

If the `-l` option is specified, a field containing the process group ID is inserted before the *state* field. Also, more processes in a process group can be output on separate lines, using only the process ID and *command* fields.

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `jobs`, `fg`, and `bg`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLS_PATH`.

## 退出状态

sh, csh, ksh88

The following exit values are returned for `jobs`, `fg`, and `bg`:

- 0 Successful completion.
- >0 An error occurred.

ksh

The following exit values are returned for `jobs`:

- 0 The information for each job is written to standard output.
- >0 One or more jobs does not exist.

The following exit values are returned for `fg`:

- exit status of last job One or more jobs has been brought into the foreground.
- non-zero One or more jobs does not exist or has completed.

The following exit values are returned for `bg`:

- 0 All background jobs are started.
- >0 One more jobs does not exist or there are no background jobs.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

`cs`, `sh`, `ksh88`

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

`ksh`

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | system/core-os  |
| Interface Stability | Uncommitted     |

## 另请参见

[csh\(1\)](#), [kill\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [ps\(1\)](#), [sh\(1\)](#), [stop\(1\)](#), [shell\\_builtins\(1\)](#), [stty\(1\)](#), [wait\(1\)](#), [signal.h\(3HEAD\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

**引用名** join – relational database operator

**用法概要**

```

/usr/bin/join [-a filenumber] [-v filenumber] [-1 fieldnumber]
 [-2 fieldnumber] [-o list] [-e string][-t char] file1 file2

/usr/bin/join [-a filenumber] [-j fieldnumber] [-j1 fieldnumber]
 [-j2 fieldnumber] [-o list] [-e string][-t char] file1 file2

```

**描述** join performs an equality join on the files *file1* and *file2* and writes the resulting joined files to standard output. By default, a field is delimited by one or more spaces and tabs with leading spaces and/or tabs ignored. The -t option can be used to change the field delimiter.

The join field is a field in each file on which files are compared. By default join writes one line in the output for each pair of lines in *files1* and *files2* that have identical join fields. The default output line consists of the join field, then the remaining fields from *file1*, then the remaining fields from *file2*, but this can be changed with the -o option. The -a option can be used to add unmatched lines to the output. The -v option can be used to output only unmatched lines.

The files *file1* and *file2* must be ordered in the collating sequence of sort -b on the fields on which they are to be joined otherwise the results are unspecified.

If either *file1* or *file2* is -, join uses standard input starting at the current location.

**选项** Some of the options below use the argument *filenumber*. This argument should be a 1 or a 2 referring to either *file1* or *file2*, respectively.

- a *filenumber*      In addition to the normal output, produce a line for each unpairable line in file *filenumber*, where *filenumber* is 1 or 2. If both -a 1 and -a 2 are specified, all unpairable lines are output.
- e *string*          Replace empty output fields in the list selected by option -o with the string *string*.
- j *fieldnumber*      Equivalent to -1 *fieldnumber* -2*fieldnumber*. Fields are numbered starting with 1.
- j1 *fieldnumber*     Equivalent to -1 *fieldnumber*. Fields are numbered starting with 1.
- j2 *fieldnumber*     Equivalent to -2 *fieldnumber*. Fields are numbered starting with 1.
- o *list*             Each output line includes the fields specified in list. Fields selected by list that do not appear in the input are treated as empty output fields. (See the -e option.) Each element of which has the either the form *filenumber.fieldnumber*, or  $\emptyset$ , which represents the join field. The common field is not printed unless specifically requested.
- t *char*             Use character *char* as a separator. Every appearance of *char* in a line is significant. The character *char* is used as the field separator for both input and output. With this option specified, the collating term should be the same as sort without the -b option.

- `-v filenumber`      Instead of the default output, produce a line only for each unpairable line in *filenumber*, where *filenumber* is 1 or 2. If both `-v 1` and `-v 2` are specified, all unpairable lines are output.
- `-1 fieldnumber`      Join on the *fieldnumber*-th field of file 1. Fields are decimal integers starting with 1.
- `-2 fieldnumber`      Join on the *fieldnumber*-th field of file 2. Fields are decimal integers starting with 1.

**操作数**

The following operands are supported:

- file1*      A path name of a file to be joined. If either of the *file1* or *file2* operands is `-`, the standard input is used in its place.
- file2*      A path name of a file to be joined. If either of the *file1* or *file2* operands is `-`, the standard input is used in its place.

*file1* and *file2* must be sorted in increasing collating sequence as determined by `LC_COLLATE` on the fields on which they are to be joined, normally the first in each line (see [sort\(1\)](#)).

**用法**

See [largefile\(5\)](#) for the description of the behavior of `join` when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

**示例**

**示例 1** Joining the password File and Group File

The following command line joins the password file and the group file, matching on the numeric group ID, and outputting the login name, the group name and the login directory. It is assumed that the files have been sorted in ASCII collating sequence on the group ID fields.

```
example% join -j1 4-j2 3 -o 1.1 2.1 1.6 -t:/etc/passwd /etc/group
```

**示例 2** Using the `-o` Option

The `-o 0` field essentially selects the union of the join fields. For example, given file phone:

```
!Name Phone Number
Don +1 123-456-7890
Hal +1 234-567-8901
Yasushi +2 345-678-9012
```

and file fax:

```
!Name Fax Number
Don +1 123-456-7899
Keith +1 456-789-0122
Yasushi +2 345-678-9011
```

## 示例 2 Using the -o Option (续)

where the large expanses of white space are meant to each represent a single tab character), the command:

```
example% join -t"tab" -a 1 -a 2 -e '(unknown)' -o 0,1.2,2.2 phone fax
```

would produce

| !Name   | Phone Number    | Fax Number      |
|---------|-----------------|-----------------|
| Don     | +1 123-456-7890 | +1 123-456-7899 |
| Hal     | +1 234-567-8901 | (unknown        |
| Keith   | (unknown)       | +1 456-789-012  |
| Yasushi | +2 345-678-9012 | +2 345-678-9011 |

### 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of join: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, LC\_COLLATE, and NLS\_PATH.

### 退出状态

The following exit values are returned:

- 0 All input files were output successfully.
- >0 An error occurred.

### 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

### 另请参见

[awk\(1\)](#), [comm\(1\)](#), [sort\(1\)](#), [uniq\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

### 附注

With default field separation, the collating sequence is that of `sort -b`; with `-t`, the sequence is that of a plain sort.

The conventions of the `join`, `sort`, `comm`, `uniq`, and `awk` commands are wildly incongruous.



|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | kbd – 操控键盘的状态、显示键盘的类型或更改缺省的键盘异常中止序列影响                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 用法概要 | <pre> kbd [-r] [-t ] [-l] [-a enable   disable   alternate]       [-c on   off] [-d keyboard device]       [-D autorepeat delay] [-R autorepeat rate]  kbd [-i] [-d keyboard device]  kbd -s [language]  kbd -b [keyboard   console] frequency </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 描述   | <p>kbd 实用程序操控键盘的状态、显示键盘类型或允许更改缺省的键盘异常中止序列影响。异常中止序列也适用于串口控制台设备。kbd 实用程序将 <code>/dev/kbd</code> 设置为缺省键盘设备。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 扩展描述 | <p><code>-i</code> 选项从键盘配置服务 <code>svc:/system/keymap:default</code> 读取并处理 <code>keyclick</code> 和 <code>keyboard</code> 异常中止设置的缺省值。只有支持按键发音器的键盘可以响应 <code>-c</code> 选项。要缺省打开按键音，请在 <code>keymap</code> 服务中将 <code>keymap/keyclick</code> 属性的值添加或更改为：</p> <pre> \$ svccfg -s keymap:default setprop keymap/keyclick=true \$ svcadm refresh keymap </pre> <p>接下来，运行命令 <code>kbd -i</code> 来更改设置。<code>keymap/keyclick</code> 属性的有效设置为 <code>true</code> 或 <code>false</code>。所有其他值均会被忽略。如果在 <code>keymap</code> 服务中未指定 <code>keymap/keyclick</code> 属性，该设置没有变化。</p> <p>只有超级用户可以使用 <code>-a</code> 选项更改键盘异常中止序列影响。通常，该序列在 SPARC 系统的键盘上为 <code>Stop-A</code> 或 <code>L1-A</code> 和 <code>Shift-Pause</code>，在 x86 系统上为 <code>F1-A</code> 和 <code>Shift-Pause</code>，在大多数系统的串口控制台输入设备上为 <code>BREAK</code>。</p> <p>无法将由错误的电子信号导致的 <code>BREAK</code> 情况与由远程 DCE 故意发送的该指令进行区分。作为修正措施，可以将 <code>-a</code> 选项与“替代中断”结合使用来切换中断解释。由于存在序列解释错误的风险，在施行“替代中断”序列时，不应通过串口控制台端口运行二进制协议（如 SLIP 和其他协议）。</p> <p>尽管 PPP 是二进制协议，但它具有避免使用会干扰串行操作的字符的能力。缺省的“替代中断”序列为 <code>CTRL-m ~ CTRL-b</code> 或十六进制的 <code>0D 7E 02</code>。在 PPP 中，这可以通过在 ACCM 中设置 <code>0x00000004</code> 或 <code>0x00002000</code> 来避免。这分别强制实施 <code>CTRL-b</code> 和 <code>CTRL-m</code> 字符的转义。</p> <p>要在 Solaris PPP 4.0 中执行此操作，应将以下内容：</p> <pre> asynmap 0x00002000 </pre> <p>添加到 <code>/etc/ppp/options</code> 文件或用于连接的任何其他配置文件。请参见 <a href="#">pppd(1M)</a>。</p> <p>SLIP 没有类似功能，使用“替代中断”序列时，不得使用 SLIP。</p> <p>“替代中断”序列对键盘异常中止没有影响。有关“替代中断”序列的更多信息，请参见 <a href="#">zs(7D)</a>、<a href="#">se(7D)</a> 和 <a href="#">asy(7D)</a>。</p> |

在许多系统上，键盘异常中止序列的缺省影响是暂停操作系统并进入调试器或监视器。某些系统具有带有一个安全位置的键开关。在这些系统上，将键开关设置到该安全位置可以覆盖该命令的任何软件缺省设置。

要永久更改键盘异常中止序列的软件缺省影响，请首先在 `keymap` 服务中将 `keymap/keyboard_abort` 属性添加或更改为：

```
$ svccfg -s keymap:default setprop keymap/keyboard_abort=disable
$ svcadm refresh keymap
```

接下来，运行命令 `kbd -i` 来更改设置。有效设置为 `enable`、`disable` 和 `alternate`；所有其他值均被忽略。如果在 `keymap` 服务中未指定该变量，则该设置没有变化。

要将异常中止序列设置为硬件 `BREAK`，应在 `keymap` 服务中将 `keymap/keyboard_abort` 的值设置为：

```
$ svccfg -s keymap:default setprop keymap/keyboard_abort=enable
$ svcadm refresh keymap
```

要更改当前设置，请运行命令 `kbd -i`。要将异常中止序列设置为“替代中断”字符序列，请首先在 `keymap` 服务中将 `keyboard_abort` 属性的当前值设置为：

```
$ svccfg -s keymap:default setprop keymap/keyboard_abort=alternate
$ svcadm refresh keymap
```

接下来，运行命令 `kbd -i` 来更改设置。当施行“替代中断”序列时，只有串口控制台设备会受影响。

要设置缺省的自动重复延迟，应在 `keymap` 服务中将 `repeat_delay` 属性设置为所需值，以毫秒 (ms) 为单位。为避免因印刷错误而导致键盘无法使用，将拒绝低于 `KIOCRPTDELAY_MIN`（在 `/usr/include/sys/kbio.h` 中定义的）的延迟值并返回 `EINVAL`：

```
$ svccfg -s keymap:default setprop keymap/repeat_delay=500
$ svcadm refresh keymap
```

要设置缺省的自动重复频率，请在 `keymap` 服务中将 `repeat_rate` 属性设置为所需值，以毫秒为单位。负值和零值重复频率将被拒绝并返回 `EINVAL`。

```
$ svccfg -s keymap:default setprop keymap/repeat_rate=40
$ svcadm refresh keymap
```

要更改 `delay` 和 `rate` 的当前设置，请运行命令 `kbd -i`。在施行自动重复延迟和/或自动重复频率时，只有命令行模式会受影响。

要设置缺省语言，请在 `keymap` 服务中将 `keymap/layout` 属性设置为所需语言。可以通过运行 `kbd -s` 查明内核中支持的这些语言。其他值均会被忽略。例如，以下示例将键盘布局设置为西班牙语：

```
$ svccfg -s keymap:default setprop keymap/layout=Spanish
$ svcadm refresh keymap
```

接下来，运行 `kbd -i` 以更改该设置。当 Solaris 重新引导时，会将西班牙语按键表装入到内核中。这些布局对 `usb` 和 `ps/2` 键盘有效。

要设置缺省的键盘蜂鸣器频率，请在 `keymap` 服务中将 `keymap/kbd_beeper_freq` 属性设置为所需值，以 HZ 为单位。该值应该介于 0 和 32767 之间（包含两者）。否则，它将被拒绝并返回 `EINVAL`：

```
$ svccfg -s keymap:default setprop keymap/kbd_beeper_freq=2000
$ svcadm refresh keymap
```

要设置缺省的控制台蜂鸣器频率，请在 `keymap` 服务中将 `keymap/console_beeper_freq` 属性设置为所需值，以 HZ 为单位。该值应该介于 0 和 32767 之间（包含两者）。否则，它将被拒绝并返回 `EINVAL`：

```
$ svccfg -s keymap:default setprop keymap/console_beeper_freq=900
$ svcadm refresh keymap
```

要更改键盘蜂鸣器频率和控制台蜂鸣器频率的当前设置，请运行 `kbd -i`。

## 选项

支持以下选项：

- |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-a enable   disable   alternate</code> | <p>启用、禁用或替代键盘异常中止序列影响。缺省情况下，在大多数系统上，键盘异常中止序列会暂停操作系统。通常，该序列在 SPARC 系统的键盘上为 <code>Stop-A</code> 或 <code>L1-A</code> 和 <code>Shift-Pause</code>，在 x86 系统上为 <code>F1-A</code> 和 <code>Shift-Pause</code>，在串口控制台设备上为 <code>BREAK</code>。</p> <p>可以使用该选项更改缺省的键盘行为。只有超级用户可以使用 <code>-a</code> 选项。</p> <p><code>enable</code>      启用键盘异常中止序列的缺省影响（暂停操作系统并进入调试器或监视器）。</p> <p><code>disable</code>      禁用缺省/替代影响并忽略键盘异常中止序列。</p> <p><code>alternate</code>    在控制台上接收到“替代中断”字符序列时，启用键盘异常中止序列的替代影响（暂停操作系统并进入调试器或监视器）。“替代中断”序列是由驱动程序 <code>zs(7D)</code>、<code>se(7D)</code>、<code>asy(7D)</code> 定义的。由于存在序列解释错误的风险，在使用该值时，二进制协议无法通过串口控制台端口运行。</p> |
| <code>-b keyboard   console</code>           | <p>设置键盘或控制台的蜂鸣器频率。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

|           |                         |                                                                                                                                                                                                                                                                           |
|-----------|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <b>keyboard</b>         | 将键盘蜂鸣器频率设置为操作数，以 HZ 为单位。请参见“操作数”部分。                                                                                                                                                                                                                                       |
|           | <b>console</b>          | 将控制台蜂鸣器频率设置为操作数，以 HZ 为单位。请参见“操作数”部分。                                                                                                                                                                                                                                      |
| <b>-c</b> | <b>on   off</b>         | 打开或关闭键盘按键音。<br><br><b>on</b> 启用按键音<br><b>off</b> 禁用按键音                                                                                                                                                                                                                    |
| <b>-d</b> | <i>keyboard device</i>  | 指定要进行设置的键盘设备。缺省设置为 <code>/dev/kbd</code> 。                                                                                                                                                                                                                                |
| <b>-D</b> | <i>autorepeat delay</i> | 设置自动重复延迟，以毫秒为单位。                                                                                                                                                                                                                                                          |
| <b>-i</b> |                         | 通过 <code>keymap</code> 服务设置键盘属性。除了 <code>-d keyboard device</code> 之外，该选项不能与任何其他选项一起使用。 <code>-i</code> 选项指示键盘命令从 <code>keymap</code> 服务中的键盘属性读取和处理 <code>keyclick</code> 和 <code>keyboard</code> 异常中止缺省值。 <code>-i</code> 选项只能由具有 "Device Security"（设备安全）权限配置文件的用户或角色使用。 |
| <b>-l</b> |                         | 返回正在使用的键盘布局代码，以及正在使用的自动重复延迟和自动重复频率。<br><br>如果与 <code>-R</code> 或 <code>-D</code> 选项一起使用，该选项将返回更改之前的值。                                                                                                                                                                     |
| <b>-r</b> |                         | 将键盘重置为刚打开电源时的设置。                                                                                                                                                                                                                                                          |
| <b>-R</b> | <i>autorepeat rate</i>  | 设置自动重复速率，以毫秒为单位。                                                                                                                                                                                                                                                          |
| <b>-s</b> | [ <i>language</i> ]     | 将键盘布局设置到内核。<br><br>如果指定了 <i>language</i> ，则会将布局设置为 <i>language</i> 。如果未指定 <i>language</i> ，则会显示可用的布局列表，用以提示用户指定 <i>language</i> 。请参见“操作数”部分。                                                                                                                              |
| <b>-t</b> |                         | 返回正在使用的键盘的类型。                                                                                                                                                                                                                                                             |

**操作数**

支持下列操作数：

|                  |                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------|
| <b>frequency</b> | 所指定的要在内核中设置的频率值。该值的接收方是由 <code>-b</code> 选项指定的。该值应该介于 0 和 32767 之间，否则将会被拒绝并返回 <code>EINVAL</code> 。 |
| <b>language</b>  | 所指定的要在内核中设置的语言。如果未发现语言，则会列出受支持的语言以供选择。它仅适用于 <code>-s</code> 选项。                                     |

## 示例

### 示例1 显示键盘类型

以下示例显示了键盘类型：

```
example% kbd -t
Type 4 Sun keyboard
example%
```

### 示例2 设置键盘缺省值

以下示例将键盘缺省值设置为 `keymap` 服务中指定的值：

```
example# kbd -i
example#
```

### 示例3 显示信息

以下示例显示了键盘类型和布局代码。它还显示了自动重复延迟和自动重复频率设置。

```
example% kbd -l
type=4
layout=43 (0x2b)
delay(ms)=500
rate(ms)=33
example%
```

### 示例4 设置键盘自动重复延迟

以下示例设置键盘自动重复延迟：

```
example% kbd -D 300
example%
```

### 示例5 设置键盘自动重复频率

以下示例设置键盘自动重复频率：

```
example% kbd -R 50
example%
```

### 示例6 选择和设置键盘语言

以下示例从指定的语言列表中选择并设置键盘语言：

```
example% kbd -s
1. Albanian
2. Belarusian
3. Belgian
4. Bulgarian
5. Croatian
6. Danish
16. Malta_UK
17. Malta_US
18. Norwegian
19. Portuguese
20. Russian
21. Serbia-And-Montenegro
```

**示例 6** 选择和设置键盘语言 (续)

```
7. Dutch
.....
22. Slove
```

To select the keyboard layout, enter a number [default n]:

```
example%
```

以下示例设置指定的键盘语言：

```
example% kbd -s Dutch
example%
```

**示例 7** 设置键盘蜂鸣器频率

以下示例设置键盘蜂鸣器频率：

```
example% kbd -b keyboard 1000
example%
```

**文件** /dev/kbd 键盘设备文件

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

**另请参见** [kmbd\(1\)](#)、[loadkeys\(1\)](#)、[svcs\(1\)](#)、[inetd\(1M\)](#)、[inetadm\(1M\)](#)、[svcadm\(1M\)](#)、[pppd\(1M\)](#)、[keytables\(4\)](#)

**附注** 某些服务器系统具有带有一个**安全**键位置的键开关，系统软件可以读取该键位置。该键位置覆盖了键盘异常中止序列影响的正常缺省值，并更改缺省值以禁用该影响。在这些系统上，当键开关位于**安全**位置时，键盘异常中止序列影响不能被可使用 `kbd` 实用程序设置的软件缺省值覆盖。

目前，没有方法可用来确定键盘单击设置的状态。

`kdb` 服务由服务管理工具 [smf\(5\)](#) 管理，其服务标识符为：

```
svc:/system/keymap:default
```

可以使用 [svcadm\(1M\)](#) 来对此服务执行管理操作（如启用、禁用或请求重新启动）。启动和重新启动该服务的职责已委托给 [inetd\(1M\)](#)。使用 [inetadm\(1M\)](#) 可以为该服务进行配置更改以及查看该服务的配置信息。可以使用 [svcs\(1\)](#) 命令来查询服务的状态。

|      |                                                                                                                                                                                                                                                                                                                                                                            |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | kdestroy – 销毁 Kerberos 票证                                                                                                                                                                                                                                                                                                                                                  |
| 用法概要 | <code>/usr/bin/kdestroy [-q] [-c cache_name]</code>                                                                                                                                                                                                                                                                                                                        |
| 描述   | <p><code>kdestroy</code> 实用程序通过向包含用户的活动 Kerberos 授权票证的指定凭证中写入零来销毁这些票证。如果未指定凭证高速缓存，则会销毁缺省凭证高速缓存。如果凭证高速缓存不存在，<code>kdestroy</code> 会显示有关此影响的一条消息。</p> <p>覆盖高速缓存后，<code>kdestroy</code> 会从系统中删除该高速缓存。该实用程序会显示一条指示操作成功或失败的消息。如果 <code>kdestroy</code> 无法销毁高速缓存，它将通过在您的终端发出嘟嘟声来警告您。</p> <p>如果需要，您可以将 <code>kdestroy</code> 命令放在您的 <code>.logout</code> 文件中，以便在您注销时自动销毁您的票证。</p> |
| 选项   | <p>支持以下选项：</p> <p><code>-c cache_name</code>      使用 <code>cache_name</code> 作为凭证（票证）高速缓存名称和位置。如果未使用此选项，将使用缺省高速缓存名称和位置。</p> <p><code>-q</code>                    静默运行。如果 <code>kdestroy</code> 无法销毁票证，您的终端将不会发出嘟嘟声。</p>                                                                                                                                                   |
| 环境变量 | <p><code>kdestroy</code> 使用以下环境变量：</p> <p><code>KRB5CCNAME</code>      凭证（票证）高速缓存的位置。有关语法和详细信息，请参见 <a href="#">krb5envvar(5)</a>。</p>                                                                                                                                                                                                                                      |
| 文件   | <code>/tmp/krb5cc_uid</code> 缺省凭证高速缓存（ <code>uid</code> 是用户的十进制 UID）。                                                                                                                                                                                                                                                                                                      |
| 属性   | 有关下列属性的说明，请参见 <a href="#">attributes(5)</a> ：                                                                                                                                                                                                                                                                                                                              |

| 属性类型  | 属性值                         |
|-------|-----------------------------|
| 可用性   | service/security/kerberos-5 |
| 接口稳定性 | Committed（已确定）              |
| 命令参数  | Committed（已确定）              |
| 命令输出  | Uncommitted（未确定）            |

另请参见 [kinit\(1\)](#)、[klist\(1\)](#)、[attributes\(5\)](#)、[kerberos\(5\)](#)、[krb5envvar\(5\)](#)

**已知问题** 只有指定凭证高速缓存中的票证会被销毁。用于保存根实例票证和口令更改票证的票证高速缓存是不同的。这些文件也可能应被销毁，或者用户的所有票证应保留在一个凭证高速缓存中。

**引用名** keylogin – 使用 keysevr 解密和存储密钥

**用法概要** /usr/bin/keylogin [-r]

**描述** keylogin 命令会提示输入口令，并使用它解密用户的密钥。密钥可在 /etc/publickey 文件（请参见 [publickey\(4\)](#)）或用户的主域中的 NIS 映射 "publickey.byname" 中找到。源及其查找顺序是在 /etc/nsswitch.conf 文件中指定的。请参见 [nsswitch.conf\(4\)](#)。解密后，用户的密钥将由本地密钥服务器进程 [keysevr\(1M\)](#) 进行存储。向任何安全的 RPC 服务（如 NFS）发送请求时，都会使用所存储的该密钥。程序 [keylogout\(1\)](#) 可用于删除由 [keysevr](#) 存储的密钥。

如果 keylogin 无法获取调用者的密钥，或给定的口令不正确，则它将失败。对于新用户或主机，可以使用 [newkey\(1M\)](#) 来添加新密钥。

如果为系统配置了多个验证机制，则所配置的每个机制的密钥将由 [keysevr\(1M\)](#) 进行解密和存储。

**选项** 支持以下选项：

-r 更新 /etc/.rootkey 文件。此文件中包含超级用户的未加密密钥。只有超级用户可以使用此选项。使用此选项时，以超级用户身份运行的进程可以发出已验证的请求，从而不要求管理员在系统启动时以超级用户身份显式运行 keylogin。请参见 [keysevr\(1M\)](#)。-r 选项应在以下情况下由管理员使用：公钥数据库中的主机条目发生更改，涉及到公钥数据库中存储的实际密钥对的 /etc/.rootkey 文件已过期。/etc/.rootkey 文件的权限允许系统中的超级用户读取或写入该文件，但不允许其他用户。

如果为系统配置了多个验证机制，则所配置的每个机制的密钥都将存储在 /etc/.rootkey 文件中。

**文件** /etc/.rootkey 超级用户的密钥

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

**另请参见** [chkey\(1\)](#)、[keylogout\(1\)](#)、[login\(1\)](#)、[keysevr\(1M\)](#)、[newkey\(1M\)](#)、[nsswitch.conf\(4\)](#)、[publickey\(4\)](#)、



**引用名** keylogout – 删除随 keyserv 存储的密钥

**用法概要** /usr/bin/keylogout [-f]

**描述** keylogout 删除由密钥服务器进程 [keyserv\(1M\)](#) 存储的密钥。尽管对密钥的进一步访问已撤销，但是，当前会话密钥在过期或刷新之前仍有效。

删除由 keyserv 存储的密钥会导致需要安全 RPC 服务的所有后台作业或已调度的 [at\(1\)](#) 作业失败。因为在计算机中仅保存了一份密钥副本，所以，最好不要在 .logout 文件中调用此命令，因为这会影响同一计算机上的其他会话。

**选项** 支持以下选项：

-f 强制 keylogout 删除超级用户的密钥。缺省情况下，不允许超级用户执行 keylogout，因为它会中断超级用户启动的所有 RPC 服务（如 NFS）。

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

**另请参见** [at\(1\)](#)、[chkey\(1\)](#)、[login\(1\)](#)、[keylogin\(1\)](#)、[keyserv\(1M\)](#)、[newkey\(1M\)](#)、[publickey\(4\)](#)、[attribu](#)

**引用名** kill – terminate or signal processes

**用法概要**

```
/usr/bin/kill -s signal_name pid...
/usr/bin/kill -l [exit_status]
/usr/bin/kill [-signal_name] pid...
/usr/bin/kill [-signal_number] pid...
```

**描述** The `kill` utility sends a signal to the process or processes specified by each *pid* operand.

For each *pid* operand, the `kill` utility performs actions equivalent to the `kill(2)` function called with the following arguments:

1. The value of the *pid* operand is used as the *pid* argument.
2. The *sig* argument is the value specified by the `-s` option, the `-signal_name` option, or the `-signal_number` option, or, if none of these options is specified, by SIGTERM.

The signaled process must belong to the current user unless the user is the super-user.

See NOTES for descriptions of the shell built-in versions of `kill`.

**选项** The following options are supported:

- `-l` (The letter ell.) Writes all values of *signal\_name* supported by the implementation, if no operand is specified. If an *exit\_status* operand is specified and it is a value of the `?` shell special parameter and `wait` corresponding to a process that was terminated by a signal, the *signal\_name* corresponding to the signal that terminated the process is written. If an *exit\_status* operand is specified and it is the unsigned decimal integer value of a signal number, the *signal\_name* corresponding to that signal is written. Otherwise, the results are unspecified.
- `-s signal_name` Specifies the signal to send, using one of the symbolic names defined in the `<signal.h>` description. Values of *signal\_name* is recognized in a case-independent fashion, without the SIG prefix. In addition, the symbolic name `0` is recognized, representing the signal value zero. The corresponding signal is sent instead of SIGTERM.
- `-signal_name` Equivalent to `-s signal_name`.
- `-signal_number` Specifies a non-negative decimal integer, *signal\_number*, representing the signal to be used instead of SIGTERM, as the *sig* argument in the effective call to `kill(2)`.

**操作数** The following operands are supported:

*pid* One of the following:

1. A decimal integer specifying a process or process group to be signaled. The process or processes selected by positive, negative and zero values of the *pid* operand is as described for the kill function. If process number 0 is specified, all processes in the process group are signaled. If the first *pid* operand is negative, it should be preceded by `—` to keep it from being interpreted as an option.
2. A job control job ID that identifies a background process group to be signaled. The job control job ID notation is applicable only for invocations of `kill` in the current shell execution environment.

The job control job ID type of *pid* is available only on systems supporting the job control option.

*exit\_status* A decimal integer specifying a signal number or the exit status of a process terminated by a signal.

## 用法

Process numbers can be found by using `ps(1)`.

The job control job ID notation is not required to work as expected when `kill` is operating in its own utility execution environment. In either of the following examples:

```
example% nohup kill %1 &
example% system("kill %1");
```

`kill` operates in a different environment and does not share the shell's understanding of job numbers.

## Output

When the `-l` option is not specified, the standard output is not be used.

When the `-l` option is specified, the symbolic name of each signal is written in the following format:

```
"%s%c", <signal_name>, <separator>
```

where the *<signal\_name>* is in upper-case, without the SIG prefix, and the *<separator>* is either a newline character or a space character. For the last signal written, *<separator>* is a newline character.

When both the `-l` option and *exit\_status* operand are specified, the symbolic name of the corresponding signal is written in the following format:

```
"%s\n", <signal_name>
```

## 示例

示例 1 Sending the kill signal

Any of the commands:

### 示例 1 Sending the kill signal (续)

```
example% kill -9 100 -165
example% kill -s kill 100 -165
example% kill -s KILL 100 -165
```

sends the SIGKILL signal to the process whose process ID is 100 and to all processes whose process group ID is 165, assuming the sending process has permission to send that signal to the specified processes, and that they exist.

### 示例 2 Avoiding ambiguity with an initial negative number

To avoid an ambiguity of an initial negative number argument specifying either a signal number or a process group, the former is always be the case. Therefore, to send the default signal to a process group (for example, 123), an application should use a command similar to one of the following:

```
example% kill -TERM -123
example% kill -- -123
```

### 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `kill`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

### 退出状态

The following exit values are returned:

- 0 At least one matching process was found for each *pid* operand, and the specified signal was successfully processed for at least one matching process.
- >0 An error occurred.

### 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/kill, csh,  
ksh88, sh

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

ksh

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | system/core-os  |
| CSI                 | Enabled         |
| Interface Stability | Uncommitted     |

另请参见 [csh\(1\)](#), [getconf\(1\)](#), [jobs\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [ps\(1\)](#), [sh\(1\)](#), [shell\\_builtins\(1\)](#), [wait\(1\)](#), [kill\(2\)](#), [signal\(3C\)](#), [signal.h\(3HEAD\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

## 附注

`/usr/bin/kill` The number of realtime signals supported is defined by the [getconf\(1\)](#) value `_POSIX_RTSIG_MAX`.

`sh` The Bourne shell, `sh`, has a built-in version of `kill` to provide the functionality of the `kill` command for processes identified with a *jobid*. The `sh` syntax is:

```
kill [-sig] [pid] [%job] ...
kill -l
```

`csh` The C-shell, `csh`, also has a built-in `kill` command, whose syntax is:

```
kill [-sig][pid][%job] ...
kill -l
```

The `csh kill` built-in sends the `TERM` (terminate) signal, by default, or the signal specified, to the specified process ID, the *job* indicated, or the current *job*. Signals are either specified by number or by name. There is no default. Typing `kill` does not send a signal to the current job. If the signal being sent is `TERM` (terminate) or `HUP` (hangup), then the job or process is sent a `CONT` (continue) signal as well.

`-l` Lists the signal names that can be sent.

`ksh88` The syntax of the `ksh88 kill` is:

```
kill [-sig][pid][%job] ...
kill -l
```

The `ksh88 kill` sends either the `TERM` (terminate) signal or the specified signal to the specified jobs or processes. Signals are either specified by number or by names (as specified in [signal.h\(3HEAD\)](#) stripped of the `SIG` prefix). If the signal being sent is `TERM` (terminate) or `HUP` (hangup), then the job or process is sent a `CONT` (continue) signal if it is stopped. The argument *job* can be the process id of a process that is not a member of one of the active jobs. In the second form, `kill -l`, the signal numbers and names are listed.

`ksh` The syntax of the `ksh kill` is:

```
kill [-n signum] [-s signame] job ...
kill [-n signum] [-s signame] -l [arg ...]
```

With the first form in which `-l` is not specified, `kill` sends a signal to one or more processes specified by *job*. This normally terminates the processes unless the signal is being caught or ignored.

Specify *job* as one of the following:

*number* The process id of *job*.

- *number*    The process group id of *job*.
- %*number*    The job number.
- %*string*    The job whose name begins with *string*.
- %?*string*    The job whose name contains *string*.
- %+            The current job.
- %%            The current job.
- %-            The previous job.

If the signal is not specified with either the -n or the -s option, the SIGTERM signal is used.

If -l is specified, and no *arg* is specified, then `kill` writes the list of signals to standard output. Otherwise, *arg* can be either a signal name, or a number representing either a signal number or exit status for a process that was terminated due to a signal. If a name is specified the corresponding signal number is written to standard output. If a number is specified the corresponding signal name is written to standard output.

- l            List signal names or signal numbers rather than sending signals as described above. The -n and -s options cannot be specified.
- n *signum*    Specify a signal number to send. Signal numbers are not portable across platforms, except for the following:
  - 0    No signal.
  - 1    HUP
  - 2    INT
  - 3    QUIT
  - 6    ABRT
  - 9    KILL
  - 14  ALRM
  - 15  TERM
- s *signame*    Specify a signal name to send. The signal names are derived from their names in `<signal.h>` without the SIG prefix and are case insensitive. `kill -l` generates the list of signals on the current platform.

`kill` in `ksh` exits with one of the following values:

- 0    At least one matching process was found for each job operand, and the specified signal was successfully sent to at least one matching process.
- >0    An error occurred.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | kinit – 获取和缓存 Kerberos 票证授予票证 (ticket-granting ticket)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 用法概要 | <pre> /usr/bin/kinit [-ARvV] [-p   -P] [-f   -F] [-a] [-c cache_name]                [-C] [-E] [-k [-t keytab_file]] [-l lifetime]                [-r renewable_life] [-s start_time] [-n] [-S service_name]                [-X attribute[=value]] [-T armor_ccache] [principal] </pre>                                                                                                                                                                                                                                                                                                                                                    |
| 描述   | <p>kinit 命令用于获取和缓存 <i>principal</i> 的初始票证授予票证（凭证）。此票证用于 Kerberos 系统进行验证。只有拥有 Kerberos 主体的用户才可以使用 Kerberos 系统。有关 Kerberos 主体的信息，请参见 <a href="#">kerberos(5)</a>。</p> <p>当使用 kinit 而未指定选项时，实用程序将提示您输入 <i>principal</i> 和 Kerberos 口令，并尝试使用本地 Kerberos 服务器验证您的登录。如果需要，可以在命令行上指定 <i>principal</i>。</p> <p>如果登录尝试通过了 Kerberos 的验证，kinit 将检索您的初始票证授予票证并将其放到票证高速缓存中。缺省情况下，票证存储在 <code>/tmp/krb5cc_uid</code> 文件中，其中 <i>uid</i> 表示用户标识号。票证将在指定的生命周期后过期，之后必须再次运行 kinit。高速缓存中的任何现有内容都将被 kinit 销毁。</p> <p>对于 <i>lifetime</i> 和 <i>renewable_life</i>，在命令行中指定的值将覆盖在 Kerberos 配置文件中指定的值。</p> <p><a href="#">kdestroy(1)</a> 命令可用于在结束登录会话之前销毁任何活动票证。</p> |
| 选项   | <p>支持以下选项：</p> <ul style="list-style-type: none"> <li>-a                    请求具有本地地址的票证。</li> <li>-A                    请求无地址的票证。</li> <li>-c <i>cache_name</i>      使用 <i>cache_name</i> 作为凭证（票证）高速缓存名称和位置。如果未使用此选项，将使用缺省高速缓存名称和位置。</li> <li>-C                    请求标准化主体名称。</li> <li>-E                    将主体名称视为企业名称。</li> <li>-f                    请求可转发票证。</li> <li>-F                    不可转发。请求不可转发票证。</li> </ul> <p>在一个主机上获取的票证通常不可在其他主机上使用。客户机可以请求将票证标记为可转发。在票证上设置 <code>TKT_FLG_FORWARDABLE</code> 标志后，用户可以使用此票证来请求具有其他 IP 地址的新票证。这样，用户可以使用当前凭证来获取在其他计算机上有效的凭证。此选项允许用户显式获取不可转发票证。</p>                                         |

- `-k [-t keytab_file]` 请求从本地主机的 *keytab* 文件中的密钥获取主机票证。可使用 `-t keytab_file` 选项来指定 *keytab* 文件的名称和位置。否则，将使用缺省名称和位置。
- `-l lifetime` 请求生命周期为 *lifetime* 的票证。如果不指定 `-l` 选项，将使用缺省票证生命周期（由各个站点配置）。如果指定的生命周期大于最大票证生命周期（由各个站点配置），则会将票证的生命周期设置为最大生命周期。有关可以为 *lifetime* 指定的有效持续时间格式，请参见“[时间格式](#)”部分。有关 `getprinc` 命令如何验证服务器主体的生命周期值，请参见 [kdc.conf\(4\)](#) 和 [kadadmin\(1M\)](#)。
- 返回的票证生命周期是下列值中的最小值：
- 在命令行上指定的值。
  - 在 KDC 配置文件中指定的值。
  - 在 Kerberos 数据库中为服务器主体指定的值。对 `kinit` 而言，该值是 `krbtgt/realm name`。
  - 在 Kerberos 数据库中为用户主体指定的值。
- `-n` 请求匿名处理。
- 支持两种匿名主体类型。对于完全匿名 Kerberos，需要在 KDC 上配置 `pkinit` 并在客户机的 `krb5.conf` 中配置 `pkinit_anchors`。然后使用 `-n` 选项并以 `@REALM` 形式（空主体名称后跟有 `@` 符号和领域名称）指定主体。如果 KDC 允许，将返回匿名票证。
- 此外还支持另一种形式的匿名票证。这类公开领域的票证隐藏客户机身份但不隐藏客户机领域。对于此模式，请使用 `kinit -n` 并以常规形式指定主体名称。如果 KDC 支持，将用匿名主体替换主体（但不替换领域）。自发行版 1.8 起，MIT Kerberos KDC 仅支持完全匿名操作。
- `-p` 请求可代理票证。
- `-P` 不可代理。请求不可代理票证。
- 可代理票证是这样一种票证：它允许您为服务获取一个 IP 地址与票证授权票证的 IP 地址不同的票证。此选项允许用户显式获取不可代理票证。
- `-r renewable_life` 请求总生命周期为 *renewable\_life* 的可更新票证。有关可以为 *renewable\_life* 指定的有效持续时间格式，请参见“[时间格式](#)”部分。有关 `getprinc` 命令如何验证服务器主体的生命周期值，请参见 [kdc.conf\(4\)](#) 和 [kadadmin\(1M\)](#)。



为票证返回的可更新生命周期是下列值中的最小值：

- 在命令行上指定的值。
- 在 KDC 配置文件中指定的值。
- 在 Kerberos 数据库中为服务器主体指定的值。对 `kinit` 而言，该值是 `krbtgt/realm name`。
- 在 Kerberos 数据库中为用户主体指定的值。

- `-R` 请求更新票证授予票证。请注意，过期票证即使仍在可更新生命周期内也无法更新。
- `-s start_time` 请求以后生效的票证，该票证从 `start_time` 开始生效。以后生效的票证是在设有 `invalid` 标志的情况下发出的，并且在使用前需要馈送回 KDC。有关可以为 `start_time` 设置的有效绝对时间或持续时间格式，请参见“**时间格式**”部分。`kinit` 首先尝试匹配绝对时间，然后再尝试匹配持续时间。
- `-S service_name` 指定在获取初始票证时使用的备用服务名。
- `-T armor_ccache` 如果 KDC 支持，指定已包含票证的凭证高速缓存 (`ccache`) 的名称。此 `ccache` 用于封装请求，这样攻击者必须同时拥有封装票证的密钥和用于验证的主体的密钥才能攻击请求。
- 封装还可以确保来自 KDC 的响应在传输过程中不会被修改。
- `-v` 请求将高速缓存中的票证授权票证（设置了 `invalid` 标志）传递到 KDC 进行验证。如果票证处于其请求的时间范围内，将使用通过验证的票证来替换高速缓存中的内容。
- `-V` 详细输出。向用户显示更多信息，例如确认验证和版本。
- `-X attribute[=value]` 指定要传递给预验证插件的预验证属性和值。可接受的 `attribute` 和 `value` 值因预验证插件而异。可以多次指定此选项以指定多个属性。如果未指定值，则假定为 `yes`。

OpenSSL `pkinit` 预验证机制可识别下列属性：

`X509_user_identity=URI`

指定到何处查找用户的 X509 身份信息。

有效的 URI 类型为 `FILE`、`DIR`、`PKCS11`、`PKCS12` 和 `ENV`。有关详细信息，请参见“**PKINIT URI 类型**”部分。

`X509_anchors=URI`

指定到何处查找可信 X509 锚信息。

有效的 URI 类型为 `FILE` 和 `DIR`。有关详细信息，请参见“**PKINIT URI 类型**”部分。

`flag_RSA_PROTOCOL[=yes]`  
指定使用 RSA 而不是缺省的 Diffie-Hellman 协议。

## PKINIT URI 类型

`FILE:file-name[,key-file-name]`

此选项的行为特定于上下文。

`X509_user_identity` *file-name* 指定包含用户证书的 PEM 格式文件的名称。如果未指定 *key-file-name*，则认为用户的私钥也在 *file-name* 中。否则，将使用 *key-file-name* 作为包含私钥的文件的名称。

`X509_anchors` *file-name* 被认为是 OpenSSL 式 ca-bundle 文件的名称。ca-bundle 文件应采用 base-64 编码。

`DIR:directory-name`

此选项的行为特定于上下文。

`X509_user_identity` *directory-name* 指定包含名为 \*.crt 和 \*.key 的文件的目录，其中文件名的第一部分是相同的，以便成对匹配证书和私钥文件。如果找到文件名以 .crt 结尾的文件，则假定以 .key 结尾的匹配文件中包含私钥。如果找不到这类文件，则不使用 .crt 文件中的证书。

`X509_anchors` *directory-name* 被认为是 OpenSSL 式散列 CA 目录，其中每个 CA 证书都存储在一个名为 hash-of-ca-cert.# 的文件中。建议使用此基础结构，但这样会检查目录中的所有文件，如果其中包含证书（PEM 格式），则使用相应的文件。

`PKCS12:pkcs12-file-name`

*pkcs12-file-name* 是包含用户证书和私钥的 PKCS #12 格式文件的名称。

`PKCS11:[slotid=slot-id][:token=token-label][:certid=cert-id][:certlabel=cert-label]`

所有关键字和值都是可选的。PKCS11 模块（例如 `opensc-pkcs11.so`）必须安装为 `libpkcs11(3LIB)` 下的加密提供程序。可以指定 `slotid=` 和/或 `token=` 以强制使用特定智能卡读取器或令牌（如果有多个可用）。可以指定 `certid=` 和/或 `certlabel=` 以强制选择设备上的特定证书。有关选择特定证书供 `pkinit` 使用的更多方法，请参见 `pkinit_cert_match` 配置选项。

`ENV:environment-variable-name`

*environment-variable-name* 指定环境变量的名称，其值已设置为与以前的某个值相符合。例如 `ENV:X509_PROXY`，其中环境变量 `X509_PROXY` 已设置为

`FILE:/tmp/my_proxy.pem`。

## 时间格式

下列绝对时间格式可用于 `-s start_time` 选项。这些示例中的日期和时间均为 1999 年 7 月 2 日 1:35:30 p.m。

| 绝对时间格式                      | 示例           |
|-----------------------------|--------------|
| <code>yymmddhhmm[ss]</code> | 990702133530 |

| 绝对时间格式                                    | 示例                    |
|-------------------------------------------|-----------------------|
| <i>hhmm</i> [ <i>ss</i> ]                 | 133530                |
| <i>yy.mm.dd.hh.mm.ss</i>                  | 99:07:02:13:35:30     |
| <i>hh:mm</i> [ <i>:ss</i> ]               | 13:35:30              |
| <i>ldate:ltime</i>                        | 07-07-99:13:35:30     |
| <i>dd-month-yyyy:hh:mm</i> [ <i>:ss</i> ] | 02-july-1999:13:35:30 |

## Variable

| 变量           | 说明                                                 |
|--------------|----------------------------------------------------|
| <i>dd</i>    | 日                                                  |
| <i>hh</i>    | 时（24 小时制）                                          |
| <i>mm</i>    | 分                                                  |
| <i>ss</i>    | 秒                                                  |
| <i>yy</i>    | 一个世纪内的年份（0-68 表示 2000 至 2068；69-99 表示 1969 至 1999） |
| <i>yyyy</i>  | 年份（包括世纪）                                           |
| <i>month</i> | 语言环境的月份名称的全称或缩写。                                   |
| <i>ldate</i> | 语言环境的相应日期表示形式                                      |
| <i>ltime</i> | 语言环境的相应时间表示形式                                      |

下列持续时间格式可用于 `-l lifetime`、`-r renewable_life` 和 `-s start_time` 选项。这些示例中的持续时间为 14 天 7 小时 5 分 30 秒。

| 持续时间格式               | 示例          |
|----------------------|-------------|
| <i>#d</i>            | 14d         |
| <i>#h</i>            | 7h          |
| <i>#m</i>            | 5m          |
| <i>#s</i>            | 30s         |
| <i>#d#h#m#s</i>      | 14d7h5m30s  |
| <i>#h#m[#s]</i>      | 7h5m30s     |
| <i>days-hh:mm:ss</i> | 14-07:05:30 |

|                      |         |
|----------------------|---------|
| 持续时间格式               | 示例      |
| <i>hours:mm[:ss]</i> | 7:05:30 |

|     |     |
|-----|-----|
| 分隔符 | 说明  |
| d   | 天数  |
| h   | 小时数 |
| m   | 分钟数 |
| s   | 秒数  |

|              |           |
|--------------|-----------|
| 变量           | 说明        |
| #            | 数字        |
| <i>days</i>  | 天数        |
| <i>hours</i> | 小时数       |
| <i>hh</i>    | 时（24 小时制） |
| <i>mm</i>    | 分         |
| <i>ss</i>    | 秒         |

## 环境变量

kinit 使用以下环境变量：

**KRB5CCNAME** 凭证（票证）高速缓存的位置。有关语法和详细信息，请参见 [krb5envvar\(5\)](#)。

## 文件

**/tmp/krb5cc\_***uid* 缺省凭证高速缓存（*uid* 是用户的十进制 UID）。

**/etc/krb5/krb5.keytab** 本地主机的 keytab 文件的缺省位置。

**/etc/krb5/krb5.conf** 本地主机的配置文件的缺省位置。请参见 [krb5.conf\(4\)](#)。

## 属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值                         |
|-------|-----------------------------|
| 可用性   | service/security/kerberos-5 |
| 接口稳定性 | 请参见下文。                      |

命令参数是 "Committed"（已确定）。命令输出是 "Uncommitted"（未确定）。

**另请参见**

[kdestroy\(1\)](#)、[klist\(1\)](#)、[kadmin\(1M\)](#)、[kttkt\\_warnd\(1M\)](#)、[libpkcs11\(3LIB\)](#)、[kdc.conf\(4\)](#)、[krb5](#)

**附注**

如果成功，`kinit` 将在初始凭证（票证授予票证）即将过期时通知 [kttkt\\_warnd\(1M\)](#) 去警告用户。

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | klist – 列出当前保留的 Kerberos 票证                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 用法概要 | <pre>/usr/bin/klist [-e]                 [ [-c] [-f] [-s] [-a [-n]] [cache_name]]                 [-k [-t] [-K] [keytab_file]]</pre>                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 描述   | <p>klist 实用程序输出凭证高速缓存的名称、票证适用于的主体（在票证文件中列出）的标识、用户当前持有的所有 Kerberos 票证的主体名称，以及每个认证者的问题和到期时间。主体名称以 <i>name/instance@realm</i> 形式列出，如果不包括 instance 则忽略 "/"，如果不包括 realm 则忽略 "@"。</p> <p>如果未指定 <i>cache_file</i> 或 <i>keytab_name</i>，klist 将根据情况显示缺省凭证高速缓存或密钥表文件中的凭证。缺省情况下，您的票证存储在文件 <i>/tmp/krb5cc_uid</i> 中，其中 <i>uid</i> 是用户的当前用户 ID。</p>                                                                                                                                                                                                                   |
| 选项   | <p>支持以下选项：</p> <ul style="list-style-type: none"> <li>-a 显示凭证中地址的列表。使用所配置的名称服务，将数字网络地址转换为关联的主机名（如果可能）。</li> <li>-c [<i>cache_name</i>] 列出凭证高速缓存中保留的票证。如果 -c 或 -k 均未指定，则此为缺省值。</li> <li>-e 显示凭证高速缓存中每个凭证的会话密钥和票证的加密类型，或密钥表文件中的每个密钥的加密类型。</li> <li>-f 显示凭证中存在的标识，使用下列缩写： <ul style="list-style-type: none"> <li>a 匿名</li> <li>A 已预验证</li> <li>d 以后生效的</li> <li>D 可以以后生效的</li> <li>f 已转发</li> <li>F 可转发</li> <li>H 已通过硬件验证</li> <li>i 无效</li> <li>I 初始</li> <li>O 可以委托</li> <li>p 代理</li> <li>P 可代理</li> <li>R 可续租</li> <li>T 已检查传输策略</li> </ul> </li> </ul> |

- k [*keytab\_file*] 列出 *keytab* 文件中保留的密钥。
- K 显示密钥表文件中每个密钥表条目中加密密钥的值。
- n 显示数字 IP 地址而不是反向解析地址。仅在与 -a 选项一起使用时有效。
- s 使 *klist* 无提示地运行（不生成任何输出），但仍然根据是否找到凭证高速缓存来设置退出状态。如果 *klist* 找到了某个凭证高速缓存，则退出状态为 0；如果未找到凭证高速缓存或本地领域 TGT 已过期，则退出状态为 1。
- t 显示密钥表文件中每个密钥表条目的时间条目时间戳。

**环境变量**

*klist* 使用以下环境变量：

**KRB5CCNAME** 凭证（票证）高速缓存的位置。有关语法和详细信息，请参见 [krb5envvar\(5\)](#)。

**文件**

*/tmp/krb5cc\_uid* 缺省凭证高速缓存（*uid* 是用户的十进制 UID）。

*/etc/krb5/krb5.keytab* 本地主机的 *keytab* 文件的缺省位置。

*/etc/krb5/krb5.conf* 本地主机的配置文件的缺省位置。请参见 [krb5.conf\(4\)](#)。

**属性**

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值                         |
|-------|-----------------------------|
| 可用性   | service/security/kerberos-5 |
| 接口稳定性 | 请参见下文。                      |

命令参数是 "Committed"（已确定）。命令输出是 "Uncommitted"（未确定）。

**另请参见**

[kdestroy\(1\)](#)、[kinit\(1\)](#)、[krb5.conf\(4\)](#)、[attributes\(5\)](#)、[krb5envvar\(5\)](#)、[kerberos\(5\)](#)

**已知问题**

读取某个文件作为服务密钥文件时，几乎不执行错误检查。

**引用名** kldb – 现场内核调试器中

## 用法概要

### 引导时装入

SPARC

```
ok boot [device-specifier] -k [-d] [boot-flags]
```

```
ok boot [device-specifier] kldb [-d] [boot-flags]
```

x86

```
kernel$ /platform/i86pc/kernel/$ISADIR/unix -k [-d] [boot-flags]
```

### 运行时装入

mdb -K

## 描述

kldb 是一个交互式内核调试器，可在即时内核上下文中实现 **mdb(1)** 的用户界面和功能。kldb 提供允许控制内核执行以及检查与修改即时内核状态的功能。kldb 可在引导会话开始时装入或者在引导系统之后装入。

本手册页介绍 kldb 独有的特性和功能、或者与 **mdb(1)** 相比 kldb 中不同的特性和功能。有关 **mdb(1)** 的更多信息，或者有关 kldb 所实现特性和功能的更多详细信息，请参见 **mdb(1)** 手册页和《Oracle Solaris Modular Debugger Guide》。

## 装入和卸载

### 引导时装入

被请求时，内核运行时链接程序 (krtld) 会在将控制权传送给内核之前先装入 kldb。如果使用 **-d** 标志，调试器会在执行 **unix** 目标文件中的初始函数之前获取对系统的控制权。如果未使用 **-d**，会装入 kldb，但要等到显式进入它时才会获取控制权。请参见下文的“调试器输入”一节。有关导致在引导时装入 kldb 的引导命令列表，请参见上面的“用法概要”部分。有关在 SPARC 计算机上使用该命令指定在引导时始终装入 kldb 的示例，请参见 **eeeprom(1M)**。

只能以系统重新引导的方式卸载在引导时装入的 kldb。

kldb 的一些功能依赖于内核服务的存在，并且不会立即可供引导时装入的 kldb 使用。特别是，直到初始化模块子系统后，**dmod** 的装入和卸载才可用。在得到处理前，请求会一直排队。同样，初始化 VM 系统之后，才可将虚拟地址转换为物理地址。尝试的转换将失败，直到转换工具可用。

### 运行时装入

对 **mdb(1)** 使用 **-K** 标志，还可在引导系统后装入 kldb。以这种方式装入时，它会立即获取对系统的控制权。可对 **mdb(1)** 使用 **-U** 标志卸载运行时装入的 kldb，或者对 **::quit dcmd** 使用 **-u** 标志将其从调试器内卸载。

### 终端类型

装入时，kldb 尝试确定系统控制台上使用中的正确终端类型。如果正在调试的系统连接了同时用于系统控制台的键盘和本地显示器，kldb 会使用适用于计算机的终端类型：对于 SPARC 使用 'sun'；对于 x86 使用 'sun-color'。串行控制台处于使用中时，引导时装入的 kldb 缺省为终端类型 '\t100'。运行时装入的 kldb 缺省为由 **mdb(1)** 请求的终端类型。**mdb(1)** 请求由 **TERM** 环境变量的值指定的终端类型，除非



被 `-T` 标志覆盖。`::term` 可用于查看当前的终端类型。

## 调试器输入

可以显式或隐式请求调试器输入。“**执行控制**”一节中将讨论在使用断点或其他执行控制功能时遇到的隐式输入。

显式调试器输入的主要方式是对于具有本地控制台的系统使用键盘中止序列，对于具有串行控制台的系统使用 `BREAK` 字符。具有本地控制台的 SPARC 系统的中止序列是 `STOP-A` 或 `Shift-Pause`，具有本地控制台的 x86 系统的中止序列是 `F1A` 或 `Shift-Pause`。有关中止序列的讨论以及有关禁用中止序列的说明，请参见 `kbd(1)`。

请求输入到调试器中的第二种方式是使用 `mdb(1)` 命令。在装入调试器后使用 `-k` 标志调用 `mdb(1)` 会触发调试器输入。

如果内核出现紧急情况并且装入了 `kmdb`，缺省情况下，`panic` 例程会进入 `kmdb` 以进行即时调试。如果指定了转储设备，并且您输入了 `::cont`，则调试器会退出并且会执行故障转储。要在遇到紧急情况时阻止内核进入 `kmdb`，可将 `nopanicdebug` 变量设置为 1。使用 `kmdb` 或在 `/etc/system` 中包含以下一行，将 `nopanicdebug` 变量设置为 1：

```
set nopanicdebug = 1
```

如果您希望确保保持装入 `kmdb` 但又始终希望紧急情况在不进入调试器的情况下触发故障转储，则这样做会很有帮助。

## 执行控制

在大多数情况下，`kmdb` 为内核提供的执行控制功能会镜像由 `mdb(1)` 进程目标提供的那些执行控制功能。可使用断点 (`::bp`)、监视点 (`::wp`)、`::continue` 以及各种类型的 `::step`。

与内核提供的无限制用户进程监视点相反，`kmdb` 仅限于一组 CPU 监视点，用于对允许的监视点数量、大小和类型进行限制。如果某个监视点与硬件支持的监视点不兼容，`::wp` 命令不允许创建该监视点。

## 调试程序模块 (dmods)

与 `mdb(1)` 一样，`kmdb` 随许多特定于子系统的调试器模块或 `dmod` 一起安装。`dmod` 会随着它们支持的子系统的装入和卸载而自动装入和卸载。还可以使用 `::load` 和 `::unload` 显式装入和卸载 `dmod`。

`kmdb` 使用内核功能装入和卸载 `dmod`，并且必须恢复系统执行才能执行每个请求的操作。`dmod` 装入或卸载完成时，系统会停止，而且会自动重新进入调试器。对于 `dmod` 装入，会在请求的 `dmod` 装入成功或失败时完成处理。在任一情况中都会提供状态消息。

## 特定于处理器的功能

某些功能是个别处理器类型所特有的。此类功能的一个示例就是各种 x86 处理器提供的分支跟踪。随特定于处理器的 `dcmd` 一起提供对这些处理器特定功能的访问权限，这些 `dcmd` 只存在于支持它们的系统上。`::status dcmd` 的输出中指明了特定于处理器的支持的可用性。调试器依赖于内核来确定处理器类型。即使调试器可能为某个给定处理器类型提供支持，但此支持要到内核前进到处理器标识完成时才会公开。

内核宏 调试器提供对一组预编译到调试器中的宏的访问权限。只有预编译的宏可用。与 `mdb(1)` 不同，`$<dcmd` 可能无法用于从任意位置装入宏。使用 `$M` 命令列出可用的宏。

内置 dcmd 本节列出对于 `kmdb` 唯一的 `dcmd`，或与 `mdb(1)` 相比在 `kmdb` 中具有不同行为的 `dcmd`。

`[address]::bp [+/-dDestT] [-c cmd] [-n count] sym ...`

`address :b [cmd ...]`

在指定位置设置断点。`::bp dcmd` 在每个指定的地址或符号处设置断点，包括 `dcmd` 前某个显式表达式指定的可选位置，以及 `dcmd` 之后的每个字符串或即时值。参数可能是符号名称或表示关注的特定虚拟地址的即时值。

如果指定了符号名称，此名称可能是指某个还无法被计算的符号。它可能包含某个尚未打开的装入目标文件中的目标文件名称和函数名称。在这种情况下，断点会被延迟，而且直到装入了与给定名称匹配的目标文件时断点才会在目标中处于活动状态。打开装入目标文件时会自动启用断点。

`-d`、`-D`、`-e`、`-s`、`-t`、`-T`、`-c` 和 `-n` 选项具有与用于 `::evset dcmd` 时的相同意义。有关 `::evset` 的说明，请参见 `mdb(1)`。如果使用 `:b` 形式的 `dcmd`，则仅会在由 `dcmd` 之前的表达式指定的虚拟地址处设置断点。`:b dcmd` 之后的参数会串联在一起形成回调字符串。如果此字符串包含元字符，必须引用该字符串。

`::branches [-v]`

(仅适用于 x86)

显示 CPU 采取的最后分支。此 `dcmd` 只在 x86 系统上受支持，而且仅当检测到并启用了特定于处理器的支持时才可用。显示的分支数量和类型取决于 CPU 提供的分支跟踪工具的功能。使用 `-v` 选项时，会显示给定分支之前的指令。

`[function]::call [arg [arg ...]]`

使用指定参数调用指定函数。被调用的函数必须列出为装入模块的符号表中的函数。字符串参数是通过引用传递的。调用完成时，会显示该函数的返回值。

使用此 `dcmd` 时务必格外谨慎。进行调用时将不会恢复内核。被调用的函数可能无法就任何内核服务的可用性做出任何假设，而且不能执行可能阻塞的操作或调用。用户还必须知晓被调用函数导致的任何负面影响，如内核稳定性可能受到影响。

`[addr]::cpuregs [-c cpuid]`

以 `::regs` 使用的格式显示为指定 CPU 设置的当前通用暂存器。

`[addr]::cpustack [-c cpuid]`

为指定 CPU 输出 C 栈回溯。显示的回溯针对进入指定 CPU 或该 CPU 被调试器停止时的点。

`addr[,len]::in [-L len]`

(仅适用于 x86)

从 `addr` 指定的 I/O 端口读取 `len` 字节。`-L` 选项的值（如果提供）优先于重复计数的值。读取长度必须为 1、2 或 4 字节，端口地址的对齐方式必须与长度相同。

`addr[,len]::out [-L len] value`

(仅适用于 x86)

将值写入 `addr` 指定的 len-byte I/O 端口。-L 选项的值（如果提供）优先于重复计数的值。写入长度必须为 1、2 或 4 字节，端口地址的对齐方式必须与长度相同。

`::quit [-u]`

`$q`

导致调试器退出。使用 -u 选项时，会恢复系统，并且会卸载调试器。如果调试器是在引导时装入的，-u 选项可能无法使用。不使用 -u 选项时，SPARC 系统将退出以引导 PROM ok 提示。go 命令可用于重新进入调试器。在 x86 系统上，会显示提示来请求重新引导计算机的权限。

`::step [over|out|branch]`

将目标步进一个指令。可选的 `over` 参数用于步过子例程调用。指定了可选 `out` 参数时，目标程序会继续，直到从当前函数返回控制权。

仅当检测到并启用了特定于处理器的支持时，才能在 x86 系统上使用可选的 `branch` 参数。指定了 `::step branch` 时，目标程序会继续，直到遇到下个分支指令。

在 SPARC 系统上，`::step dcmd` 可能无法用于步进 'ta' 指令。同样，它可能无法用于在 x86 系统上步进 'int' 指令。如果步进导致调试器无法解析的陷阱，会输出有关该影响的消息，然后步进将失败。

`cpuid::switch`

`cpuid:x`

将指定 CPU 用作代表。栈跟踪、通用暂存器转储和类似功能会将新的代表 CPU 用作数据源。完全执行控制功能可用于新的代表 CPU 上。

`::term`

显示当前终端类型。

`addr[,len]::wp [+/-dDestT] [-rwx] [-pi] [-n count] [-c cmd]`

`addr[,len]:a [cmd ...]`

`addr[,len]:p [cmd ...]`

`addr[,len]:w [cmd ...]`

在指定地址设置监视点，缺省情况下被解释为虚拟地址。如果使用 -p 选项，该地址被解释为物理地址。在 x86 平台上，可使用 -i 选项在 I/O 端口上设置监视点。使用 -i 选项时，该地址被解释为 I/O 端口的地址。

可通过在 `dcmd` 之前指定可选的重复计数来设置被监视区域的长度（以字节为单位）。如果没有显式设置长度，缺省值为一个字节。`::wp dcmd` 允许将监视点配置为在存在任何读取（-r 选项）、写入（-w 选项）或执行（-x 选项）访问的组合时触发。

-d、-D、-e、-s、-t、-T、-c 和 -n 选项具有与用于 `::evset dcmd` 时的相同意义。有关 `::evset` 的说明，请参见 [mdb\(1\)](#)。`:a dcmd` 可在指定地址处设置读取访问监视点。`:p dcmd` 可在指定地址处设置执行访问监视点。`:w dcmd` 可在指定地址处设置写入访问监视点。`:a`、`:p` 和 `:w dcmd` 之后的参数会串联在一起形成回调字符

串。如果此字符串包含元字符，必须引用该字符串。

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值                 |
|-------|---------------------|
| 可用性   | system/kernel       |
|       | developer/debug/mdb |
| 接口稳定性 | Committed（已确定）      |

**另请参见** [mdb\(1\)](#)、[boot\(1M\)](#)、[dumpadm\(1M\)](#)、[eeprom\(1M\)](#)、[kernel\(1M\)](#)、[system\(4\)](#)、[attributes\(5\)](#)  
 《Oracle Solaris Modular Debugger Guide》

**仅 SPARC** [kbd\(1\)](#)

## 附注

### 对可供调试器使用的内存的限制

装入调试器时，会分配可供调试器使用的内存区域，并在此时固定该区域。如果 `dcmd` 尝试分配超过可用量的内存，将终止这些 `dcmd`（如果可能）。调试器将尝试从内存不足的情况中正常恢复，但可能无法或者被强制终止系统。此约束在 32 位的 x86 系统上尤为严格。

### 性能影响

装入 `kmdb` 会对系统性能造成负面影响，因为调试器会消耗内核内存以及其他有限的系统资源。

### 引导到 `kmdb` 中以捕获 `panic()` 栈

要对 SPARC 计算机上的 `panic()` 进行故障排除，使用 [eeprom\(1M\)](#) 指定系统始终在引导时装入 `kmdb` 可能会很有用。紧急情况后，系统会开始重新引导，从而清除控制台中的 `panic` 栈。通过引导到 `kmdb` 中，可捕获和解释 `panic` 栈。有关指定在引导时装入 `kmdb` 的示例，请参见 [eeprom\(1M\)](#)。

|      |                                                                                                                                                                                                                                                                                                                           |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | kmfcfg – 密钥管理策略和插件配置实用程序                                                                                                                                                                                                                                                                                                  |
| 用法概要 | <code>kmfcfg subcommand [option ...]</code>                                                                                                                                                                                                                                                                               |
| 描述   | <p><code>kmfcfg</code> 命令允许用户配置密钥管理框架 (Key Management Framework, KMF) 策略数据库。KMF 策略数据库 (database, DB) 限制使用通过 KMF 框架管理的密钥和证书。</p> <p><code>kmfcfg</code> 提供在系统缺省数据库文件 <code>/etc/security/kmfpolicy.xml</code> 或用户定义数据库文件中列出、创建、修改、删除、导入和导出策略定义的能力。</p> <p>对于插件配置, <code>kmfcfg</code> 允许用户显示插件信息、安装或卸载 KMF 插件以及修改插件选项。</p> |

子命令 支持以下子命令：

#### create

将新策略添加到策略数据库文件中。

`create` 子命令的格式如下：

```
create [dbfile=dbfile] policy=policyname
 [ignore-date=true|false]
 [ignore-unknown-eku=true|false]
 [ignore-trust-anchor=true|false]
 [validity-adjusttime=adjusttime]
 [ta-name=trust anchor subject DN]
 [ta-name=trust anchor subject DN | search]
 [ta-serial=trust anchor serial number]
 [ocsp-responder=URL]
 [ocsp-proxy=URL]
 [ocsp-use-cert-responder=true|false]
 [ocsp-response-lifetime=timelimit]
 [ocsp-ignore-response-sign=true|false]
 [ocsp-responder-cert-name=Issuer DN]
 [ocsp-responder-cert-serial=serial number]
 [crl-basefilename=basefilename]
 [crl-directory=directory]
 [crl-get-crl-uri=true|false]
 [crl-proxy=URL]
 [crl-ignore-crl-sign=true|false]
 [crl-ignore-crl-date=true|false]
 [keyusage=digitalSignature|nonRepudiation
 |keyEncipherment | dataEncipherment |
 keyAgreement |keyCertSign |
 cRLSign | encipherOnly | decipherOnly],[...]
 [ekunames=serverAuth | clientAuth |
 codeSigning | emailProtection |
 ipsecEndSystem | ipsecTunnel |
 ipsecUser | timeStamping |
 OCSPSigning],[...]
```

```
[ekuoids=OID,OID,OID...]
[mapper-name=name of the mapper]
[mapper-dir=dir where mapper library resides]
[mapper-path=full pathname of mapper library]
[mapper-options=mapper options]
```

create 子命令支持以下选项：

**crl-basefilename=filename**

**crl-directory=directory**

这两个属性用于指定 CRL 文件的位置。crl-basefilename 属性表示 CRL 文件的基文件名。crl-directory 属性表示 CRL 文件的目录，缺省为当前目录。

如果 crl-get-crl-uri 属性设置为 true 并且没有指定 crl-basefilename，则高速缓存的 CRL 文件的 basefilename 是用于提取 CRL 文件的 URI 的基名。

如果 crl-get-crl-uri 属性设置为 false，需要指定 crl-basefilename 来指示输入 CRL 文件。缺省情况下，crl-get-crl-uri 的设置为 false。

这两个属性只适用于基于文件的 CRL 插件。当前基于文件的 CRL 插件为 file 和 pkcs11 Keystore。对于 nss Keystore，CRL 位置始终为 NSS 内部数据库。

**crl-get-crl-uri=true | false**

配置是否将动态提取和高速缓存 CRL 文件作为证书验证的一部分，使用来自证书的分发点扩展的 URI 信息。

此属性的缺省值为 false。

**crl-ignore-crl-date=true | false**

如果 crl-ignore-crl-date 设置为 true，不会检查 CRL 的有效时间段。

此属性的缺省值为 false。

**crl-ignore-crl-sign=true | false**

如果 crl-ignore-crl-sign 设置为 true，不会检查 CRL 的签名。

此属性的缺省值为 false。

**crl-proxy=URL**

crl-get-crl-uri 设置为 true 时，设置代理服务器名称和端口以便动态检索 CRL 文件。

端口号为可选。如果未指定端口号，缺省值为 8080。crl-proxy 设置的一个示例为：crl-proxy=webcache.sfbay:8080。

**dbfile=dbfile**

添加新策略的 DB 文件。如果未指定，缺省值将是系统 KMF 策略数据库文件 /etc/security/kmfpolicy.xml。

**ekuoids=EKUOIDS**

正在定义的策略所需的以逗号分隔的扩展密钥使用 OID 列表。OID 以点记法表示，例如 1.2.3.4。ekuoids 设置的一个示例为：ekuoids=1.2.3.4,9.8.7.6.5。

**ekunames=EKUNAMES**

正在定义的策略所需的以逗号分隔的扩展密钥使用名称列表。允许用于 *EKUNAMES* 的值列表

为: `serverAuth`、`clientAuth`、`codeSigning`、`emailProtection`、`ipsecEndSystem`、`ipsecTunnel`、`ipsecUser`、`timeStamping` 和 `OCSPSigning`

缺省情况下, OCSP、CRL、密钥使用和扩展密钥使用检查处于关闭状态。要启用其中任何一个, 请为特定检查指定一个或多个属性。例如, 如果设置了 `ocsp-responder` 属性, 则会启用 OCSP 检查。如果设置了 `ekuname` 属性或 `ekuoids` 属性, 则会启用扩展密钥使用检查。

**ignore-date=true | false**

为此策略设置 "Ignore Date" (忽略日期) 选项。缺省情况下此值为 `false`。如果指定了 `true`, 策略会在评估证书有效性时忽略证书中定义的有效期。

**ignore-unknown-eku=true | false**

为策略设置 "Ignore Unknown EKU" (忽略未知 EKU) 选项。缺省情况下此值为 `false`。如果为 `true`, 策略会忽略扩展密钥使用扩展中任何无法识别的 EKU 值。

**ignore-trust-anchor=true | false**

为此策略设置 "Ignore Trust Anchor" (忽略信任锚) 选项。缺省情况下此值为 `false`。如果指定了 `true`, 策略不会在验证时使用信任锚证书验证主题证书的签名。

**keyusage=KUVALUES**

正在定义的策略所需的以逗号分隔的扩展密钥使用列表。允许的值列表

为: `digitalSignature`、`nonRepudiation`、`keyEncipherment`、`dataEncipherment`、`keyAgreement`、`cRLSign`、`encipherOnly`、`decipherOnly`

**ocsp-ignore-response-sign=true | false**

如果该属性设置为 `true`, 则不会验证 OCSP 响应的签名。此属性值缺省为 `false`。

**ocsp-proxy=URL**

为 OCSP 设置代理服务器名称和端口。端口号为可选。如果未指定端口号, 则缺省值为 8080。ocsp-proxy 设置的一个示例

为: `ocsp-proxy="webcache.sfbay:8080"`

**ocsp-response-lifetime=timelimit**

设置响应必须处于的有效时间段。timelimit 可由数字-day、数字-hour、数字-minute 或数字-second 指定。ocsp-response-lifetime 设置的一个示例为: `ocsp-response-lifetime=6-hour`。

**ocsp-responder-cert-name=IssuerDN****ocsp-responder-cert-serial=serialNumber**

这两个属性表示 OCSP 响应者证书。ocsp-responder-cert-name 用于指定证书的签发者名称。有关示例, 请参见 ta-name 选项。ocsp-responder-cert-serial 表示序



列号，必须指定为十六进制值，例如 `0x0102030405060708090a0b0c0d0e0f`。如果 OCSP 响应者与证书的签发者不同，并且如果需要验证 OCSP 响应，应该提供 OCSP 响应者证书信息。

`ocsp-responder=URL`

设置 OCSP 响应者 URL 以便与 OCSP 验证方式一起使用。例如

`ocsp-responder=http://ocsp.verisign.com/ocsp/status`

`ocsp-use-cert-responder=true | false`

将此策略配置为始终使用证书自身中定义的响应者（如果可能）。

`policy=policyname`

要创建的策略记录。*policyname* 是必需的。

`ta-name=trust anchor subject DN | search`

`ta-name` 标识用于验证证书的信任锚。KMF 策略引擎不会执行完全 PKIX 路径验证，而只会将信任锚视为要验证的证书的父证书。

如果指定了显式“主题 DN”，它必须与 `ta-serial` 值结合，以便唯一地标识要使用的证书。此外，标识的证书必须可用于选择的 keystore 中。

如果使用了值 `search` 而不是显式主题和序列号，KMF 策略引擎会尝试查找与待验证证书的签发者名称匹配的证书并使用其进行验证。

如果使用 `search`，会忽略 `ta-serial` 值。

`ta-serial=trust anchor serial number`

如果将 `ta-name` 指定为显式主题名称，则此证书的序列号必须由 `ta-serial` 值指示。此序列号必须以十六进制格式表示，例如 `ta-serial=0x01020a0b`。

`validity-adjusttime=adjusttime`

为证书有效期的两个端点设置调整时间。时间可由**数字-day**、**数字-hour**、**数字-minute** 或**数字-second** 指定。`validity-adjusttime` 设置的一个示例

为：`validity-adjusttime=6-hour`。`ta-name="Subject DN"`

`ta-serial=serialNumber`

这两个属性表示信任锚证书，用于查找 keystore 中的信任锚证书。*ta-name* 用于指定信任锚证书主题名称的标识名。例如，`ta-name="O=Sun Microsystems Inc., \ OU=Solaris Security Technologies Group, \ L=Ashburn, ST=VA, C=US, CN=John Smith"` TA 证书的序列号。它和“签发者 DN”一起用于查找 keystore 中的 TA 证书。必须将序列号指定为十六进制值，例如

`0x0102030405060708090a0b0c0d0e` 如果 `ignore-trust-anchor` 属性的值为 `false`，需要设置信任锚属性。

`mapper-name=name`

`mapper-dir=directory`

`mapper-path=path`

`mapper-options=options`



这四个选项支持证书到名称映射。`mapper-name` 提供映射器的名称。例如，`cn` 名称表示映射器目标文件 `kmf_mapper_cn.so.1`。`mapper-dir` 覆盖缺省映射器目录 `/lib/crypto`。`mapper-path` 指定映射器目标文件的全路径。`mapper-options` 是最长为 255 字节的仅 ASCII 字符串。它的格式是映射器特定的，但映射器会接受以逗号分隔的选项列表，例如 `casesensitive,ignoredomain`。`mapper-path` 和 `mapper-name` 互斥。只有设置了 `mapper-name` 才能设置 `mapper-dir`。只有设置了 `mapper-name` 或 `mapper-path` 才能设置 `mapper-options`。尝试使用上述任何一个不正确设置都会导致错误，且无法修改策略数据库。

#### delete

删除与指定策略名称匹配的任何策略。无法删除系统缺省策略 (`default`)。

`delete` 子命令的格式如下：

```
delete [dbfile=dbfile] policy=policyname
```

`delete` 子命令支持以下选项：

`dbfile=dbfile` 从指定文件中读取策略定义。如果未指定 `dbfile`，缺省值将是系统 KMF 策略数据库文件：`/etc/security/kmfpolicy.xml`。

`policy=policyname` 要删除的策略的名称。如果使用系统数据库，则需要 `policyname`。

#### export

将策略从一个策略数据库文件导出到另一个策略数据库文件。

`export` 子命令的格式如下：

```
kmfcfg export policy=policyname outfile=newdbfile [dbfile=dbfile]
```

`export` 子命令支持以下选项：

`dbfile=dbfile` 从中读取导出策略的 DB 文件。如果未指定 `dbfile`，缺省值将是系统 KMF 策略数据库文件：`/etc/security/kmfpolicy.xml`。

`outfile=outputdbfile` 在其中存储导出策略的 DB 文件。

`policy=policyname` 要导出的策略记录。

#### help

显示关于 `kmfcfg` 命令的帮助。

`help` 子命令的格式如下：

```
help
```

#### import

将策略从一个策略数据库文件导入到另一个策略数据库文件。

`import` 子命令的格式如下：

```
kmfcfg import policy=policyname infile=inputdbfile [dbfile=dbfile]
```

import 子命令支持以下选项：

*policy=policyname* 要导入的策略记录。

*infile=inputdbfile* 要从中读取策略的 DB 文件。

*dbfile=outdbfile* 添加新策略的 DB 文件。如果未指定，缺省值将是系统 KMF 策略数据库文件 `/etc/security/kmfpolicy.xml`。

#### list

如果不指定参数，将列出缺省系统数据库中的所有策略定义。

list 子命令的格式如下：

```
list [dbfile=dbfile] [policy=policyname]
```

list 子命令支持以下选项：

*dbfile=dbfile* 从指定文件读取策略定义。如果未指定，缺省值将是系统 KMF 策略数据库文件 `/etc/security/kmfpolicy.xml`。

*policy=policyname* 只显示指定策略的策略定义。

#### modify

修改与指定名称匹配的任何策略。无法修改系统缺省策略 (default)。

modify 子命令的格式如下：

```
modify [dbfile=dbfile] policy=policyname
 [ignore-date=true|false]
 [ignore-unknown-eku=true|false]
 [ignore-trust-anchor=true|false]
 [validity-adjusttime=adjusttime]
 [ta-name=trust anchor subject DN]
 [ta-serial=trust anchor serial number]
 [ocsp-responder=URL]
 [ocsp-proxy=URL]
 [ocsp-use-cert-responder=true|false]
 [ocsp-response-lifetime=timelimit]
 [ocsp-ignore-response-sign=true|false]
 [ocsp-responder-cert-name=Issuer DN]
 [ocsp-responder-cert-serial=serial number]
 [ocsp-none=true|false]
 [crl-basefilename=basefilename]
 [crl-directory=directory]
 [crl-get-crl-uri=true|false]
 [crl-proxy=URL]
 [crl-ignore-crl-sign=true|false]
 [crl-ignore-crl-date=true|false]
 [crl-none=true|false]
```

```

[keyusage=digitalSignature| nonRepudiation
 |keyEncipherment | dataEncipherment |
 keyAgreement |keyCertSign |
 cRLSign | encipherOnly | decipherOnly],[...]
[keyusage-none=true|false]
[ekunames=serverAuth | clientAuth |
 codeSigning | emailProtection |
 ipsecEndSystem | ipsecTunnel |
 ipsecUser | timeStamping |
 OCSPSigning],[...]
[ekuoids=OID,OID,OID]
[eku-none=true|false]
[mapper-name=name of the mapper]
[mapper-dir=dir where mapper library resides]
[mapper-path=full pathname of mapper library]
[mapper-options=mapper options]

```

`modify` 子命令支持与 `create` 子命令相同的许多选项。有关共享选项的说明，请参见 `create` 子命令。

`modify` 子命令支持以下唯一选项：

|                                          |                                                                                                                                                                          |
|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>crl-none=true   false</code>       | 如果 <code>crl-none</code> 设置为 <code>true</code> ，将禁用 CRL 检查。如果此属性设置为 <code>true</code> ，则无法设置其他 CRL 属性。                                                                   |
| <code>dfile=[<i>dbfile</i> ]</code>      | 要修改策略的数据库文件。如果未指定，缺省值将是系统 KMF 策略数据库文件 <code>/etc/security/kmfpolicy.xml</code> 。                                                                                         |
| <code>eku-none=true   false</code>       | 如果 <code>eku-none</code> 设置为 <code>true</code> ，将禁用扩展密钥使用检查。如果 <code>eku-none</code> 设置为 <code>true</code> ，无法同时设置扩展密钥使用属性 <code>ekuname</code> 和 <code>ekuoids</code> 。 |
| <code>keyusage-none=true   false</code>  | 如果 <code>keyusage-none</code> 设置为 <code>true</code> ，将禁用密钥使用检查。<br><br>如果此属性设置为 <code>true</code> ，无法同时设置 <code>keyusage</code> 属性。                                      |
| <code>ocsp-none=true   false</code>      | 如果 <code>ocsp-none</code> 设置为 <code>true</code> ，将禁用 OCSP 检查。如果此属性设置为 <code>true</code> ，不会同时设置任何其他 OCSP 属性。                                                             |
| <code>policy=<i>policyname</i></code>    | 要修改的策略的名称。 <i>policyname</i> 是必需的。无法修改系统 KMF 策略数据库中的 <i>default</i> 策略。                                                                                                  |
| <code>mapper-name=<i>name</i></code>     |                                                                                                                                                                          |
| <code>mapper-dir=<i>directory</i></code> |                                                                                                                                                                          |
| <code>mapper-path=<i>path</i></code>     |                                                                                                                                                                          |

`mapper-options=options` 有关更多信息，请参见 `create` 子命令。

#### 插件子命令

`install keystore=keystore_name modulepath=pathname\ [option=option_str]`  
 将插件安装到系统中。`modulepath` 字段指定 KMF 插件共享库目标文件的路径名。如果未将 `pathname` 指定为绝对路径名，共享库目标文件会假定为与 `/lib/security/$ISA/` 相对。`ISA` 标记会由实施定义的目录名称替换，该名称可定义相对于调用程序指令集体系结构的路径名。

`list plugin`

显示 KMF 插件信息。

不使用 `plugin` 关键字，`kmfcfg list` 会如“子命令”部分所述显示策略信息。

`modify plugin keystore=keystore_name option=option_str`

修改 `plugin` 选项。`plugin` 选项是由插件定义的，并且由此插件专门解释，因此此命令接受任何选项字符串。

不使用 `plugin` 关键字，`kmfcfg modify` 会如“子命令”部分所述更新策略配置。

`uninstall keystore=keystore_name`

卸载具有 `keystore_name` 的插件。

#### 示例

示例1 创建新策略

以下示例会在系统数据库中创建一个名为 IPSEC 的新策略：

```
$ kmfcfg create IPSEC \
ignore-trust-anchor=true \
ocsp-use-cert-responder=true \
keyusage=keyAgreement,keyEncipherment,dataEncipherment \
ekuname=ipsecTunnel,ipsecUser
```

#### 退出状态

将返回以下退出值：

0 成功完成。

>0 出现错误。

#### 文件

`/etc/security/kmfpolicy.xml` 缺省系统策略数据库

#### 属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值              |
|-------|------------------|
| 可用性   | system/core-os   |
| 接口稳定性 | Uncommitted（未确定） |

另请参见

[attributes\(5\)](#)

**引用名** kpasswd – 更改用户的 Kerberos 口令

**用法概要** /usr/bin/kpasswd [*principal*]

**描述** kpasswd 命令用于更改 Kerberos 主体的口令。kpasswd 会提示您输入当前 Kerberos 口令，以便从用户 Kerberos 领域的 KDC 获取 changepw 票证。如果 kpasswd 成功获取了 changepw 票证，将两次提示用户输入新口令，然后口令将被更改。

如果主体是由某个策略控制的，该策略指定了新口令的长度和/或字符类数量，则新口令必须符合该策略。（五种字符类为小写、大写、数字、标点符号和所有其他字符。）

**操作数** 支持下列操作数：

*principal* 更改 Kerberos 主体 *principal* 的口令。否则，主体将从调用 kpasswd 命令的用户的标识派生。

**文件** /tmp/ovsec\_admin.xxxxxx 在口令更改操作的生命周期内使用的临时凭证高速缓存。(xxxxxx 是一个随机字符串。)

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值                         |
|------|-----------------------------|
| 可用性  | service/security/kerberos-5 |
| CSI  | Enabled (已启用)               |

**另请参见** [kerberos\(5\)](#)

**已知问题** 如果暂停 kpasswd，changepw 票证可能不会被销毁。

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | krb5-config – 针对已安装的 Kerberos 库进行链接                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 用法概要 | <pre>krb5-config   [--all   --cflags   --exec-prefix   --help   --libs <i>library</i>     --prefix   --vendor   --version]</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 描述   | krb5-config 标识和显示针对已安装的 Kerberos 库编译和链接程序时所需的特殊标志。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 选项   | <p>支持以下选项：</p> <ul style="list-style-type: none"> <li>--all            显示版本、供应商、前缀和 exec-prefix。</li> <li>--cflags        显示构建 Kerberos 时使用的编译器标志。</li> <li>--exec-prefix   显示构建 Kerberos 时使用的 exec-prefix。</li> <li>--help          显示用法消息。</li> <li>                 这是缺省值。</li> <li>--libs <i>library</i> 显示与 <i>library</i> 进行链接时所需的编译器选项。</li> <li>                 支持以下 <i>library</i> 值： <ul style="list-style-type: none"> <li>krb5     Kerberos 5 应用程序</li> </ul> </li> <li>--prefix        显示构建 Kerberos 时使用的前缀。</li> <li>--vendor        显示已安装的 Kerberos 实现的供应商。</li> <li>--version       显示已安装的 Kerberos 实现的版本。</li> </ul> |
| 示例   | <p>示例1 使用 --cflags 选项</p> <p>以下示例显示了使用 <code>libkrb5(3LIB)</code> 时所需的 C 编译器标志：</p> <pre>% krb5-config --cflags -I/usr/include/kerberosv5</pre> <p>示例2 使用 --libs 选项</p> <p>以下示例显示了针对 <code>libkrb5(3LIB)</code> 进行链接时所需的 C 编译器选项：</p> <pre>% krb5-config --libs -L/usr/lib -R/usr/lib -lkrb5</pre>                                                                                                                                                                                                                                                                                                                                     |
| 退出状态 | <p>将返回以下退出值：</p> <ul style="list-style-type: none"> <li>0     成功完成。</li> <li>&gt;0   出现错误。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

**属性**            有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值                         |
|-------|-----------------------------|
| 可用性   | service/security/kerberos-5 |
| 接口稳定性 | Volatile (可变)               |

**另请参见**        [libgss\(3LIB\)](#)、[libkrb5\(3LIB\)](#)、[attributes\(5\)](#)



|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | ksh, ksh93, rksh – Korn Shell, 一种标准和受限的命令与编程语言                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 用法概要 | <pre>ksh [<math>\pm</math>abcefhikmnoqrstuvxBCD] [-R <i>file</i>] [<math>\pm</math>o <i>option</i>] ...   [-] [<i>arg</i> ...]  rksh [<math>\pm</math>abcefhikmnoqrstuvxBCD] [-R <i>file</i>] [<math>\pm</math>o <i>option</i>] ...   [-] [<i>arg</i> ...]</pre>                                                                                                                                                                                                                                                                                                                                                                       |
| 描述   | <p>ksh 是一种执行从终端或文件读取的命令的命令与编程语言。rksh 是命令解释程序 ksh 的受限版本。rksh 用于设置登录名和执行环境，其功能比标准 shell 的功能更受约束。</p> <p>有关此 shell 的参数含义，请参见 <a href="#">调用</a>。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 定义   | <p><b>元字符</b>定义为下列字符之一：</p> <pre>; &amp; ( )   &lt; &gt; NEWLINE SPACE TAB</pre> <p>空白为制表符或空格。</p> <p><b>标识符</b>是以字母或下划线开头的字母、数字或下划线的序列。标识符用作<b>变量名称</b>的组成部分。</p> <p><i>vname</i> 是由句点(.)分隔的一个或多个标识符的序列，并且可选择性地在此序列前面添加一个句点(.)。vname 用作函数和变量名称。</p> <p><b>单词</b>是当前语言环境定义的字符集中的<b>字符</b>的序列（不包括不带引号的<b>元字符</b>）。</p> <p><b>命令</b>是 shell 语言的语法中的字符序列。shell 读取各个命令，并直接执行或通过调用单独的实用程序执行所需操作。内置命令是由 shell 自身执行且未创建单独的进程的命令。某些命令用作内置命令的原因仅仅是出于方便，这些命令不在本手册页的论述范围内。本手册页记录了对 shell 环境造成负面影响的内置命令以及在执行路径搜索前发现的内置命令（请参见 <a href="#">执行</a>）。出于历史原因，其中的某些内置命令的行为不同于其他内置命令的行为，这些命令称为特殊内置命令。</p>                                                 |
| 命令   | <p><b>简单命令</b>是一个变量赋值的列表（请参见 <a href="#">变量赋值</a>），或是一个可在其前面添加变量赋值列表的以空白分隔的单词序列。请参见本手册页的 <a href="#">环境</a> 部分。</p> <p>第一个单词指定要执行的命令的名称。除非在此节中指定，否则其余单词将作为参数传递到调用的命令。命令名称作为参数 0 传递。请参见 <a href="#">exec(2)</a>。简单命令的<b>值</b>即为其退出状态。如果简单命令正常终止，其值为 0-255。如果简单命令异常终止，其值为 256+<i>signum</i>。与退出状态对应的信号的名称可通过 kill 内置实用程序的 -l 选项获取。</p> <p><b>管道</b>是由   分隔的一个或多个命令的序列。每个命令（最后一个命令除外）的标准输出都通过 <a href="#">pipe(2)</a> 连接到下一命令的标准输入。每个命令（可能除最后一个命令以外）都作为单独的进程运行。shell 等待最后一个命令终止。除非启用了 <a href="#">pipefail</a> 选项，否则管道的退出状态即为最后一个命令的退出状态。每个管道前面都可以添加保留字 !。如果最后一个命令的退出状态为<b>非零</b>，此操作会导致管道的退出状态变为 0；如果最后一个命令的退出状态为 0，则会导致管道的退出状态变为 1。</p> |

列表是由 ;、&、|&、&& 或 | 分隔的一个或多个管道的序列，并且该序列可以选择性地以 ;、& 或 |& 结束。在这五个符号中，;、& 和 |& 的优先级相同，这些符号的优先级低于 && 和 || 的优先级。同时，符号 && 和 || 也具有相同的优先级。

分号 (;) 导致按顺序执行前面的管道。和符号 (&) 导致异步执行前面的管道，也即，shell 不会等待该管道完成。符号 |& 导致异步执行前面的管道，并建立一个到父 shell 的双向管道。通过向命令应用重定向运算符 <& 和 >& 以及 arg p，并通过使用内置命令 read 和 print 的 -p 选项，父 shell 可以写入和读取派生的管道的标准输入和输出。符号 && (||) 导致仅当前面的管道返回零（非零）值时才会执行该符号后面的列表。列表中可显示一个或多个换行符（而非分号）来分隔命令。如果列表的第一个管道的第一个项是一个并非以重定向开头且未出现在 while、until 或 if 列表中的简单命令，则可以在此项前面添加一个分号。除非使用 set 内置命令按照说明启用 showme 选项，否则将忽略此分号。

命令要么是一个简单命令，要么是以下列表中的命令之一。除非另行说明，否则命令返回的值即为该命令中执行的最后一个简单命令的值。

```
for vname [in word ...];do list ;done
```

每次执行 for 命令时，vname 都会设置为从 in word 列表获取的下一个 word。如果省略了 in word...，for 命令针对每个位置参数执行一次 do list，这些位置参数从 1 开始设置。当此列表中没有任何其他单词时，执行随即结束。请参见参数扩展。

```
(([expr1] ; [expr2] ; [expr3])) ;do list ;done
```

首先计算算术表达式 expr1。反复计算算术表达式 expr2，直到其计算结果为零为止；如果计算结果为非零，则会执行 list，并计算算术表达式 expr3。如果省略了任何表达式，该表达式按照计算结果为 1 的方式运行。请参见算术计算。

```
select vname [in word ...];do list ;done
```

select 命令在标准错误（文件描述符 2）中输出一组 word，每个 word 前面均带有一个数字。如果省略了 in word...，则改用从 1 开始的位置参数。请参见参数扩展。输出 PS3 提示，并从标准输入中读取一个行。如果此行包含列出的某个 word 的数字，那么变量 vname 的值设置为与该数字对应的 word。如果此行为空，则会重新输出选择列表。否则，变量 vname 的值设置为 null。从标准输入读取的行的内容保存在变量 REPLY 中。在遇到中断符或 EOF 之前，系统会针对每个选择执行 list。如果通过执行 list 将 REPLY 变量设置为 null，则会输出选择列表，然后为下一选择显示 PS3 提示。

```
case word in [(| pattern [| pattern] ...) list ;;] ... esac
```

case 命令执行与 word 匹配的 first pattern 的关联 list。模式的格式与用于生成文件名的格式相同。请参见文件名生成。

; ; 运算符导致终止执行 case。如果使用 ;& 取代 ; ;，则会执行下一个后续列表（如果有）。

```
if list ;then list [;elif list ;then list] ... [;else list] ;fi
```

执行 if 后面的 list，并且如果返回零退出状态，则执行第一个 then 后面的 list。否则执行 elif 后面的 list，如果其值为零，则执行下一个 then 后面的 list。如果执行

每个连续的 `elif list` 失败，则会执行 `else list`。如果 `if list` 具有**非零**退出状态，并且没有 `else list`，`if` 命令会返回**零**退出状态。

`while list ; do list ; done`

`until list ; do list ; done`

`while` 命令反复执行 `while list`，并且如果此列表中的最后一个命令的退出状态为零，则会执行 `do list`，否则循环将终止。如果未执行 `do list` 中的命令，`while` 命令将返回**零**退出状态，并且可使用 `until` 替换 `while`，以便对循环终止测试求反。

**((表达式))**

使用本手册页中介绍的算术计算规则计算 `expression`。如果算术表达式的值为**非零**，退出状态为 `0`。否则，退出状态为 `1`。

**(list;)**

在单独的环境中执行 `list`。如果需要两个相邻的左括号以便进行嵌套，必须插入一个空格以免作为算术命令计算（如本部分所属）。

完全执行 `list`。与元字符不同的是，`(` 和 `)` 以及 `{` 和 `}` 是**保留字**，它们必须出现在行开头或 `;` 之后才能识别。

**[[ 表达式 ]]**

计算 `expression` 并在 `expression` 为 `True` 时返回**零**退出状态。有关 `expression` 的说明，请参见**条件表达式**。

`function varname { list ; }`

`varname () { list ; }`

定义 `varname` 引用的函数。其 `varname` 包含 `.` 的函数称为**规程函数**，并且最后一个 `.` 前面的 `varname` 部分必须引用现有变量。

函数主体是 `{` 和 `}` 之间的命令的 `list`。此外，使用函数 `varname` 语法定义的函数还可用作 `.` 特殊内置命令的参数，以便获取使用 `varname()` 语法定义它的等效行为。请参见**函数**。

`time [ pipeline ]`

如果省略了 `pipeline`，则会在标准错误中输出当前 `shell` 和已完成的子进程的用户和系统时间。否则执行 `pipeline`，并在标准错误中输出已用时间以及用户和系统时间。`TIMEFORMAT` 变量可设置为格式字符串，用于指定时间信息的显示方式。有关 `TIMEFORMAT` 变量的说明，请参见 `Shell 变量`。

下列保留字仅作为命令的第一个字且未带引号时才能识别为保留字：

```
case
do
done
else
elif
esac
for
```

```

fi
function
if
select
then
time
until
while
{ }
[[]]
!
```

**变量赋值**

一个或多个变量赋值可启动简单命令，也可用作 `typeset`、`export` 或 `readonly` 特殊内置命令的参数。**赋值**语法格式如下所示：

*varname*=*word*

*varname*[*word*]=*word*

禁止在 *varname* 和 = 之间或者 = 和 *word* 之间使用空格。

*varname*=(*assignlist*)

禁止在 *varname* 和 = 之间使用空格。*assignlist* 可以为下列值之一：

*word* ...

索引数组赋值。

[*word*]=*word* ...

关联数组赋值。如果带有 `typeset -a` 前缀，则改为创建索引数组。

**赋值** ...

复合变量赋值。这会创建一个带有子变量的复合变量 *varname*，子变量的格式为 *varname.name*，其中 *name* 是赋值的名称部分。*varname* 的值包含所有赋值元素。同时，对 *varname* 的子变量进行的其他赋值也会显示为 *varname* 的值的一部分。如果未指定 *assignment*，*varname* 则为允许定义后续子元素的复合变量。

`typeset [options] assignment ...`

嵌套变量赋值。通过用 ; 分隔每个赋值，可以指定多个赋值。前一个值将在赋值之前取消设置。

此外，可以使用 += 替换 = 以表示添加或附加到前一个值。如果将 += 应用到算术类型，*word* 将作为算术表达式计算，并添加到当前值。如果应用到字符串变量，则会将 *word* 定义的值附加到当前值。对于复合赋值，不会取消设置前一个值，并且只要类型兼容，新值便会附加到当前值。

**注释**

以 # 开头的单词将导致注释掉或忽略该单词以及换行符之前的所有后续字符。

**别名设置**

如果为各个命令的第一个单词定义了别名，则会使用别名文本替换此单词。别名由任意数目的字符组成，其中不包括元字符、引号字符、文件扩展字符、参数扩展字符、命令替换字符以及 =。替换字符串可包含任何有效的 shell 脚本，其中包括在 **命令**

部分中列出的元字符。除正在替换的所有文本以外，将对被替换文本中的各个命令的第一个单词的别名进行测试。如果别名值的最后一个字符为空白，还会检查此别名后的单词以便进行别名替换。

别名可用于重新定义内置命令，但不能用于重新定义在**命令**部分中列出的保留字。使用 `alias` 命令可以创建并列出别名，使用 `unalias` 命令可以删除别名。

别名设置在读取脚本时执行，而不是在执行脚本时执行。为使别名生效，必须在读取引用该别名的命令之前执行 `alias` 定义命令。下列别名已编译到 `shell` 中，但是可以取消设置或重新定义这些别名：

```
autoload='typeset -fu'
command='command '
fc=hist
float='typeset -lE'
functions='typeset -f'
hash='alias -t --'
history='hist -l'
integer='typeset -li'
nameref='typeset -n'
nohup='nohup '
r='hist -s'
redirect='command exec'
source='command .'
stop='kill -s STOP'
suspend='kill -s STOP $$'
times='{ { time;} 2>&1;}'
type='whence -v'
```

#### 波浪号替换

执行别名替换之后，系统将检查每个单词，查看此单词是否以不带引号的波浪号 (~) 开头。要进行波浪号替换，`word` 还会引用参数扩展的 `word` 部分。请参见**参数扩展**。

如果满足此条件，则会检查 / 前的单词，确定它是否与口令数据库中的用户名相匹配。如果找到匹配项，~ 及匹配的登录名将替换为匹配用户的登录目录。如果未找到匹配项，原始文本将保留不变。单独的 ~ 或位于 / 之前的 ~ 替换为 `$HOME`。后跟有 + 或 - 的 ~ 分别替换为 `$PWD` 和 `$OLDPWD` 的值。

此外，当赋值的值以 ~ 开头以及当 ~ 显示在冒号 (:) 之后时，扩展**变量赋值**时会尝试替换波浪号。同时，~ 登录名还会以 : 结束。

#### 命令替换

前面带有美元符号 (\$) 或一对重音符 ("") 的括号所括起的命令的标准输出可用作单词的一部分或整个单词。尾随的换行符将被删除。在第二种格式 (已过时) 中，在执行命令之前将处理引号中的字符串，以便替换为特殊引号字符。请参见**引用**。

命令替换 `$(cat file)` 可替换为速度更快的等效 `${<file}`。命令替换 `$(n<#)` 扩展至文件描述符 `n` 的当前字节偏移。

**算术替换** 前面带有美元符号的双括号所括起的算术表达式( $\$( (arithmetic\_expression) )$ ) 替换为双括号中的算术表达式的值。

**进程替换** 仅在支持使用 `/dev/fd` 目录命名打开文件的 UNIX 操作系统版本上提供进程替换。

`<(list)` 或 `>(list)` 格式的每个命令参数都会运行异步连接到 `/dev/fd` 中的某个文件的进程 `list`。此文件的名称即为该命令的参数。如果选择采用 `>` 的格式，在此文件中写入会为 `list` 提供输入。如果采用 `<`，作为参数传递的文件则包含 `list` 进程的输出。

例如，

```
paste <(cut -f1 file1) <(cut -f3 file2) | tee \
 >(process1) >(process2)
```

分别对文件 `file1` 和 `file2` 中的字段 1 和 3 执行 `cut` 操作，将结果 `paste` 在一起，并将其发送到进程 `process1` 和 `process2`。该命令还会向标准输出显示此结果。作为参数传递给命令的文件为 UNIX `pipe(2)`。需要对此文件执行 `lseek(2)` 的程序无法运行。

**参数扩展** 参数是一个变量、一个或多个数字或 `*`、`@`、`#`、`?`、`-`、`$` 和 `!` 中的任意字符。变量通过 `vname` 表示。要创建其 `vname` 包含 `.` 的变量，必须已存在一个如下变量：其 `vname` 在最后一个 `.` 之前包括所有内容。变量具有一个值以及零个或多个属性。通过使用 `typeset` 特殊内置命令，可以为变量分配值和属性。后文介绍了 shell 支持的属性以及 `typeset` 特殊内置命令。导出的变量向环境传递值和属性。

此 shell 支持索引数组和关联数组。数组变量的元素由下标引用。索引数组的下标通过 `[` 和 `]` 之间的算术表达式表示（请参见**算术计算**）。使用 `set -A vname value ...` 为索引数组分配值。所有下标的值必须介于范围 `0` 至 `1,048,575` 之间。索引数组不需要声明。对带有有效下标的变量的任何引用都是合法的，并且会根据需要创建数组。

关联数组使用 `typeset` 的 `-A` 选项创建。关联数组的下标通过括在 `[` 和 `]` 内的字符串表示。

引用不带下标的任何数组与引用带有下标 `0` 的数组等效。

变量的值可通过以下命令分配：

```
vname=value [vname=value] ...
```

或

```
vname[subscript]=value [vname[subscript]=value] ...
```

禁止在 `=` 前后使用空格。`nameref` 是一个引用其他变量的变量。`nameref` 使用 `typeset` 的 `-n` 属性创建。执行 `typeset` 命令时的变量值成为使用 `nameref` 变量时引用的变量。`nameref` 的名称不能包含点 (`.`)。如果变量或函数名称包含 `.`，并且第一个 `.` 前的名称部分与 `nameref` 的名称匹配，通过将 `nameref` 部分替换为 `nameref` 引用的变量名称可以获取引用的变量。如果将 `nameref` 用作 `for` 循环的索引，则会为列表中的每一项建立一个名称引用。`nameref` 提供了一种在函数中引用变量的方便方法，该变量的名称作为参数传递给函数。例如，如果变量名称作为第一个参数传递给函数，该函数中的命令



```
typeset -n var=$1
```

导致将 *var* 的引用和赋值作为其名称已传递给函数的变量的引用和赋值。如果为 *vname* 设置了浮点值属性 *-E* 或 *-F* 或者整数属性 *-i*，该 *value* 则受算术计算的约束，如本手册页所述。位置参数是通过数字表示的参数，可使用 *set* 特殊内置命令向这些参数分配值。当调用 shell 时，参数 *\$0* 从参数零开始设置。字符 *\$* 用于引入可替换的参数。

```
${parameter}
```

此 shell 读取从 *\${* 到匹配的 *}* 之间的所有字符，并将其作为同一个单词的一部分，即使此部分包含花括号或元字符。该参数的值（如果有）将被替换。当 *parameter* 后跟有不会作为其名称的一部分解释的字母、数字或下划线时，当变量名称包含 *.* 时或者当变量带有下标时，都需要使用花括号。如果 *parameter* 是一个或多个数字，该参数则为位置参数。包含多个数字的位置参数必须括在花括号中。如果 *parameter* 是 *\** 或 *@*，则会替换从 *\$1* 开始的所有位置参数，并使用字段分隔符分隔这些参数。如果使用带有下标 *\** 或 *@* 的数组 *vname*，则会替换每个元素的值，并使用 *IFS* 的值的第一个字符分隔这些值。

```
${#parameter}
```

如果 *parameter* 为 *\** 或 *@*，则会替换位置参数的数字。否则替换 *parameter* 的值的长度。

```
${#vname[*]}
```

```
${#vname[@]}
```

替换数组 *vname* 中的元素数目。

```
${!vname}
```

扩展至 *vname* 引用的变量的名称。除非 *vname* 为名称引用，否则为 *vname*。

```
${!vname[subscript]}
```

除非 *subscript* 为 *\** 或 *@*，否则扩展至下标的名称。当 *subscript* 为 *\** 时，生成 *vname* 的数组下标的列表。对于不是数组的变量，如果设置了该变量，则值为 *0*。否则为 *null*。当 *subscript* 为 *@* 时，该命令与 *\${vname[\*]}* 相同，除非用于双引号中，此时，每个数组下标将生成一个单独的参数。

```
${!prefix*}
```

扩展至其名称以 *prefix* 开头的变量的名称。

```
${parameter:-word}
```

如果 *parameter* 已设置且不为 *Null*，则替换其值。否则替换 *word*。

除非将 *word* 用作被替换的字符串，否则不会计算它的值。

在下面的示例中，仅当 *d* 未设置或者为 *NULL* 时，才会执行 *pwd*：

```
print ${d:-$(pwd)}
```

如果表达式中省略了冒号 (:), 此 shell 仅检查是否设置了 *parameter*。

```

${parameter:offset:length}
${parameter:offset}

```

扩展至 *parameter* 的值的以下部分：从作为算术表达式扩展偏移量所确定的字符开始（从 0 开始算起），并且该部分包含的字符数由 *length* 定义的算术表达式确定。

在第二种格式中，使用相应值的剩余部分。负偏移量从 *parameter* 末尾开始往后计数。

减号前需要一个或多个空白，以免此 shell 将此运算符解释为 :-。如果 *parameter* 为 \* 或 @，或者为 \* 或 @ 索引的数组名称，*offset* 和 *length* 分别表示数组索引和元素数目。相对于索引数组的最高下标而言，采用的负 *offset* 比此最高下标大 1。未指定关联数组的顺序。

```

${parameter#pattern}
${parameter##pattern}

```

如果 shell *pattern* 与 *parameter* 的值的开头匹配，此扩展的值即为删除了匹配部分的 *parameter* 的值。否则会替换此 *parameter* 的值。在第一种格式中，删除了最小匹配 *pattern*，在第二种格式中，删除了最大匹配 *pattern*。如果 *parameter* 为 @、\*，或者为带有下标 @ 或 \* 的数组变量，则会依次对每个元素应用求子串操作。

```

${parameter%pattern}
${parameter%%pattern}

```

如果 shell *pattern* 与 *parameter* 的值的末尾匹配，此扩展的值即为删除了匹配部分的 *parameter* 的值。否则会替换 *parameter* 值。在第一种格式中，删除了最小匹配 *pattern*，在第二种格式中，删除了最大匹配 *pattern*。如果 *parameter* 为 @、\*，或者为带有下标 @ 或 \* 的数组变量，则会依次对每个元素应用求子串操作。

```

${parameter/pattern/string}
${parameter//pattern/string}
${parameter/#pattern/string}
${parameter/%pattern/string}

```

扩展 *parameter*，并使用指定 *string* 替换 *pattern* 的最长匹配项。*string* 中的 \n 的每个匹配项都替换为与第 *n* 个子模式匹配的 *parameter* 部分。

如果 *string* 为 null，则会删除 *pattern*，并且可省略 *string* 前面的 /。如果 *parameter* 为 @、\*，或者为带有



下标@或\*的数组变量，则会依次对每个元素应用替换操作。此种情况下，将针对每个元素重新计算 *word* 的 *string* 部分。

在第一种格式中，仅替换 *pattern* 的第一个匹配项。

在第二种格式中，*pattern* 的每个匹配项都将替换为指定 *string*。

第三种格式将模式匹配限制为 *string* 的开头。

第四种格式将模式匹配限制为 *string* 的末尾。

shell 自动设置下列参数：

|             |                                                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------------------|
| #           | 位置参数的数目（以十进制表示）。                                                                                                      |
| -           | 在调用时或通过 <code>set</code> 命令向 shell 提供的选项。                                                                             |
| ?           | 执行的最后一个命令所返回的十进制值。                                                                                                    |
| \$          | 此 shell 的进程数。                                                                                                         |
| _           | 一开始时，_ 的值是在环境中传递时所执行的 shell 或脚本的绝对路径名。然后会为该值分配上一命令的最后一个参数。                                                            |
|             | 对于异步命令，未设置此参数。此参数还可用于保存查看邮件时的匹配 MAIL 文件的名称。                                                                           |
| !           | 调用的最后一个后台命令的进程数或使用 <code>bg</code> 内置命令放置到后台的最新作业。                                                                    |
| .sh.command | 当处理 DEBUG 陷阱时，此变量包含即将运行的最新命令行。                                                                                        |
| .sh.edchar  | 此变量包含处理 KEYBD 陷阱时已输入的键盘字符（如果第一个字符为 ESC、ASCII 033，则为字符序列）的值。如果在陷阱操作中更改了此值，则会使用新值替换导致此陷阱的键（或键序）。请参见本手册页的 <b>键绑定</b> 部分。 |
| .sh.edcol   | 执行最新的 KEYBD 陷阱时光标所在的字符位置。                                                                                             |
| .sh.edmode  | 当在 vi 插入模式下处理 KEYBD 陷阱时，该值设置为 ESC。否则，当处理 KEYBD 陷阱时，.sh.edmode 为 null。请参见本手册页的 <b>vi 编辑模式</b> 部分。                      |
| .sh.edtext  | 执行最新的 KEYBD 陷阱时输入缓冲区中的字符。如果未在处理 KEYBD 陷阱，该值则为 null。                                                                   |
| .sh.file    | 包含当前命令的文件的名称。                                                                                                         |
| .sh.fun     | 正在执行的当前函数的名称。                                                                                                         |

|                            |                                                                                                                                                                                                                         |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>.sh.match</code>     | 将最新匹配项及子模式匹配项存储在匹配的条件模式匹配项以及使用运算符 <code>#</code> 、 <code>%</code> 或 <code>/</code> 的变量扩展之后的索引数组。第 <code>0</code> 个元素存储完整匹配项，第 <code>i</code> 个元素存储第 <code>i</code> 个子匹配项。当为扩展的变量分配了新值时， <code>.sh.match</code> 变量将取消设置。 |
| <code>.sh.name</code>      | 设置为调用规程函数时的变量的名称。                                                                                                                                                                                                       |
| <code>.sh.subscript</code> | 设置为调用规程函数时的变量的名称下标。                                                                                                                                                                                                     |
| <code>.sh.subshell</code>  | 子 shell 和命令替换的当前深度。                                                                                                                                                                                                     |
| <code>.sh.value</code>     | 设置为调用 <code>set</code> 或 <code>append</code> 规程函数时的变量的值。                                                                                                                                                                |
| <code>.sh.version</code>   | 设置为标识此 shell 的版本的值。                                                                                                                                                                                                     |
| <code>LINENO</code>        | 正在执行的脚本或函数中的当前行号。                                                                                                                                                                                                       |
| <code>OLDPWD</code>        | <code>cd</code> 命令设置的早期工作目录。                                                                                                                                                                                            |
| <code>OPTARG</code>        | <code>getopts</code> 内置命令处理的最后一个选项参数的值。                                                                                                                                                                                 |
| <code>OPTIND</code>        | <code>getopts</code> 内置命令处理的最后一个选项参数的索引。                                                                                                                                                                                |
| <code>PPID</code>          | shell 的父级的进程数。                                                                                                                                                                                                          |
| <code>PWD</code>           | <code>cd</code> 命令设置的当前工作目录。                                                                                                                                                                                            |
| <code>RANDOM</code>        | 每次引用此变量时，将生成一个在 <code>0</code> 和 <code>32767</code> 之间统一分布的随机整数。通过为 <code>RANDOM</code> 分配一个数值，可以初始化随机数字的序列。                                                                                                            |
| <code>REPLY</code>         | 如果未提供参数，此变量则由 <code>select</code> 语句和 <code>read</code> 内置命令设置。                                                                                                                                                         |
| <code>SECONDS</code>       | 每次引用此变量时，将返回自调用 shell 以来所经过的秒数。如果为此变量分配了一个值，引用时返回的值即为此分配的值以及自分配以来所经过的秒数。                                                                                                                                                |

shell 使用下列变量：

**CDPATH**

定义 `cd` 命令的搜索路径。

**COLUMNS**

定义 shell 编辑模式的编辑窗口以及用于输出选择列表的编辑窗口的宽度。

**EDITOR**

如果未设置 `VISUAL` 变量，则会按照 `VISUAL` 的说明针对模式检查此变量的值，并启用相应的编辑选项。

请参见本手册页**特殊命令**部分中的 `set` 命令。

**ENV**

对相应值执行参数扩展、命令替换和算术替换，以便生成调用 shell 时所执行的脚本的路径名。此文件通常用于别名和函数定义。缺省值为 `$HOME/.kshrc`。

请参见本手册页的**调用**部分。

shell 不会设置 ENV。

**FCEDIT**

hist 命令的缺省编辑器名称的已过时名称。如果设置了 HISTEDIT，则不会使用 FCEDIT。

shell 指定 FCEDIT 的缺省值。

**FIGNORE**

定义执行文件名匹配时忽略的文件名集的模式。

**FPATH**

函数定义的搜索路径。当引用具有 -u 属性的函数时，以及当未找到命令时，系统会在此路径中的目录内搜索其名称与此函数或命令相同的文件。如果找到名称与此命令对应的可执行文件，则会在当前环境中读取和执行此文件。与 PATH 不同的是，必须明确使用句点 (.)（而非相邻冒号 (: ) 字符，或者起始或结束冒号 (: )）表示当前目录。

**HISTCMD**

历史文件中的当前命令的数目。

**HISTEDIT**

hist 命令的缺省编辑器的名称。

**HISTFILE**

如果在调用 shell 时设置了此变量，此变量的值即为存储命令历史所使用的文件的路径名。请参见本手册页的**重新输入命令**部分。

**HISTSIZ**

如果在调用 shell 时设置了此变量，此 shell 可访问的以前输入的命令的数目则大于或等于此数目。缺省值为 512。

**HOME**

cd 命令的缺省参数（起始目录）。

shell 不会设置 HOME。HOME 由 `login(1)` 设置。

**IFS**

内部字段分隔符，通常为空格、制表符和换行符，这些分隔符用于分隔命令替换或参数扩展的结果，还可用于通过内置命令 read 分隔字段。IFS 变量的第一个字符用于分隔参数以便进行 "\$\*" 替换。请参见本手册页的**引用**部分。

要分隔的字符串中出现的、未包含在 isspace 字符类中的每个 IFS 字符以及包含在 isspace 字符类中的 IFS 中的任何相邻字符均可用于分隔字段。IFS 中包含在

`issspace` 字符类中的一个或多个字符可用于分隔字段。此外，如果 `IFS` 中连续显示同一 `issspace` 字符，则会按照此字符不包含在 `issspace` 类中的方式来处理此字符，因此，如果 `IFS` 包含两个制表符，两个相邻制表符可分隔一个 `null` 字段。

`shell` 指定 `IFS` 的缺省值。

#### LANG

对于并非使用以 `LC_` 或 `LANG` 开头的变量专门选择的任何类别，此变量确定这些类别的语言环境类别。

#### LC\_ALL

此变量覆盖 `LANG` 变量以及任何其他 `LC_` 变量的值。

#### LC\_COLLATE

此变量确定字符整理信息的语言环境类别。

#### LC\_CTYPE

此变量确定字符处理函数的语言环境类别。它确定用于模式匹配的字符类。请参见本手册页的**文件名生成**部分。

#### LC\_NUMERIC

此变量确定小数点字符的语言环境类别。

#### LINES

如果设置了此变量，则使用此变量的值确定列长度以便输出选择列表。选择列表纵向输出，直到填充了大约 2/3 的 `LINES` 行为止。

#### MAIL

如果此变量设置为邮件文件的名称，并且未设置 `MAILPATH` 变量，`shell` 则会在指定文件中通知用户已收到邮件。

`shell` 不会设置 `MAIL`。在某些系统中，`MAIL` 由 `login(1)` 设置。

#### MAILCHECK

指定 `shell` 检查 `MAILPATH` 或 `MAIL` 变量指定的所有文件的修改时间更改的频率（秒）。缺省值为 600 秒。经过此时间后，`shell` 会在发出下一提示前进行检查。

`shell` 指定 `MAILCHECK` 的缺省值。

#### MAILPATH

以冒号 (:) 分隔的文件名列表。如果设置了此变量，`shell` 会通知用户在最后 `MAILCHECK` 秒内对指定文件所做的任何修改。每个文件名后面都可以带有 ? 以及输出的消息。此消息会进行参数扩展、命令替换和算术替换，并将变量 `$_` 定义为已发生更改的文件的名称。缺省消息为在 `$_` 中您有一个邮件。

#### PATH

命令的搜索路径。除非在 `.profile` 中，否则用户无法在 `rksh` 下执行时更改 `PATH`。请参见本手册页的**执行**部分。

`shell` 指定 `PATH` 的缺省值。

**PS1**

扩展此变量的值，以便进行参数扩展、命令替换和算术替换，从而定义主提示字符串，缺省情况下，主提示字符串为 `$`。主提示字符串中的字符 `!` 替换为命令编号。当输出提示字符串时，出现的两个连续 `!` 可生成一个 `!`。请参见本手册页的**重新输入命令**部分。

`shell` 指定 `PS1` 的缺省值。

**PS2**

辅助提示字符串，缺省情况下为 `>`。

`shell` 指定 `PS2` 的缺省值。

**PS3**

在选择循环中使用的选择提示字符串，缺省情况下为 `#?`。

`shell` 指定 `PS3` 的缺省值。

**PS4**

扩展此变量的值，以便进行参数计算、命令替换和算术替换，并且此变量的值会在每一行前面放置一个执行跟踪。缺省情况下，`PS4` 为 `+`。如果取消设置 `PS4`，执行跟踪提示同样为 `+`。

`shell` 指定 `PS4` 的缺省值。

**SHELL**

在环境中保存 `shell` 的路径名。如果此变量的基名在调用时为 `rsh`、`rksh`、`rksh` 或 `krsh`，此 `shell` 即受限制。

`shell` 不会设置 `SHELL`。在某些系统中，`SHELL` 由 `login(1)` 设置。

**TIMEFORMAT**

此参数的值用作格式字符串，指定带有 `time` 保留字前缀的管道的时间信息的显示方式。`%` 字符引入扩展为时间值或其他信息的格式序列。

格式序列及其含义如下所示。

`%%`  
字符 `%`。

`%[p][\]R`  
已用时间（秒）。

`%[p][\]U`  
在用户模式下所用的 CPU 秒数。

`%[p][\]S`  
在系统模式下所用的 CPU 秒数。

`%P`  
CPU 百分比，计算公式为  $(U + S)/R$ 。

花括号表示可选部分。可选 *p* 是一个用于指定**精度**（即小数点后面的小数位数）的数字。值为 0 导致不会输出小数点或小数。最多可显示小数点后面的三个数位。大于 3 的 *p* 值视为 3。如果未指定 *p*，则使用值 3。

可选 *l* 指定较长的格式，其中包括小时（如果大于零）、分钟和秒钟，并且格式为 *HHhMMmSS.FFs*。*p* 的值确定是否包含小数。

所有其他字符在输出时都不会发生任何更改，并且会添加一个尾随的换行符。如果未设置，则使用缺省值 `$_\nreal\t%2LR\nuser\t%2LU\nsys\t%2LS'`。如果值为 null，则不会显示时间信息。

#### TMOUT

如果设置为大于零的值，TMOUT 即为 read 内置命令的缺省超时值。如果输入来自于终端，select 复合命令则会在 TMOUT 秒后终止。否则，当从终端读取时，如果未在规定的秒数内输入某个行，shell 将终止。可以为 shell 编译此值的最大限制，不能超过此最大限制。

shell 指定 TMOUT 的缺省值。

#### VISUAL

如果此变量的值与模式 `*[Vv][Ii]*` 匹配，则会启用 vi 选项。请参见**特殊命令**。如果此值与模式 `*gmacs*` 匹配，则会启用 gmacs 选项。如果此值与模式 `*macs*` 匹配，则会启用 emacs 选项。VISUAL 的值会覆盖 EDITOR 的值。

#### 字段分隔

执行参数扩展和命令替换之后，系统将扫描替换结果中的字段分隔符（即位于 IFS 中的字段分隔符），并在找到此类字符的位置将结果分隔为不同字段。系统将保留显式 null 字段（`"` 或 `'`），并删除隐式 null 字段，这些字段是由没有任何值的参数或没有任何输出的命令替换所生成的。

如果设置了 braceexpand (-B) 选项，则会检查由 IFS 生成的每个字段，以查看它们是否包含一个或多个花括号模式。有效花括号模式包括：`{*,*}`、`{l1..l2}`、`{n1..n2}`、`{n1..n2%fmt}`、`{n1..n2..n3}` 或 `{n1..n2..n3%fmt}`，其中 \* 表示任意字符，*l1*、*l2* 表示字母，*n1*、*n2*、*n3* 表示带符号的数字，*fmt* 表示 printf 使用的指定格式。在每种情况下，通过将 { 前的字符放置在 { 和 } 之间的字符所生成的每个字符串之前，并将 } 后的字符附加到此类字符串之后来创建字段。系统将检查生成的字段，查看这些字段是否包含任何花括号模式。

在第一种格式中，将为 { 和 , 、 , 和 } 之间的每个字符串创建一个字段。\* 表示的字符串可包含不带引号的匹配的嵌入式 { 和 }。否则，必须引用带有 \* 的每个 { 和 }。

在第二种格式中，*l1* 和 *l2* 在 C 语言环境中必须同时为大写字母或小写字母字符。在此种情况下，将为从 *l1* 到 *l2* 的每个字符创建一个字段。

在其余的格式中，将为从 *n1* 开始的每个数字创建一个字段。此操作将持续进行，直达到 *n2* 并使 *n1* 递增 *n3* 为止。如果 *n1* ≤ *n2*，在未指定 *n3* 的情况下的行为就像 *n3* 为 1 一样，否则则像 *n3* 为 -1 一样。在指定了 *%fmt* 的格式中，可指定所有格式标志、宽

度和精度，并且 *fmt* 可以任何说明符 *cdiouxX* 结尾。例如，`{a,z}{1..5..3%02d}{b..c}x` 扩展为 8 个字段，即 `a01bx`、`a01cx`、`a04bx`、`a04cx`、`z01bx`、`z01cx`、`z04bx` 和 `z4cx`。

## 文件名生成

分隔之后，除非设置了 `-f` 选项，否则将扫描每个字段，以查看是否包含字符 `*`、`?`、`(` 和 `[`。如果显示其中的某个字符，则会将单词视为模式。

包含任何模式字符的每个文件名组件都会替换为与该目录中的模式相匹配的按字典顺序排序的名称集合。如果未找到与此模式匹配的文件名，该文件名组件将保持不变，除非此模式带有 `~(N)` 前缀，此种情况下，将删除此文件名组件。如果设置了 `FIGNORE`，当生成匹配的文件名时，将忽略与 `FIGNORE` 的值定义的模式相匹配的每个文件名组件。此外，还会忽略名称 `.` 和 `..`。如果未设置 `FIGNORE`，则会忽略每个文件名组件开头的字符 `.`，除非与此组件对应的模式的第一个字符为字符 `.` 自身。对于模式匹配的其他使用情况，不会对 `/` 和 `.` 进行特殊处理。

**\*** 匹配任何字符串，包括 `null` 字符串。当用于文件名扩展时，如果启用了 `globstar` 选项，单独的两个相邻 `*` 匹配所有文件和零个或多个目录及子目录。如果两个相邻 `*` 后跟有一个 `/`，则仅匹配目录和子目录。

**?** 匹配任何单个字符。

**[...]** 匹配包括的任何一个字符。使用 `-` 分隔的一对字符将在词汇上匹配这对字符之间的任何字符，包括这对字符。如果左 `[` 后面的第一个字符为 `!`，则匹配任何不包括的字符。通过将 `-` 作为第一个或最后一个字符，可以将其包含在字符集合中。在 `[` 和 `]` 中，可使用语法 `[:class:]` 指定字符类，其中 *class* 表示按 ANSI-C 标准定义的下列类之一：

```
alnum alpha blank cntrl digit graph
lower print punct space upper
word xdigit
```

*word* 等效于 *alnum* 和字符 `_`。在 `[` 和 `]` 中，可使用语法 `[=c=]` 指定等效类，这会匹配主整理权重（根据当前语言环境定义）与字符 *c* 相同的所有字符。在 `[` 和 `]` 中，`[.symbol.]` 匹配整理符号 *symbol*。

*pattern-list* 是使用 `&` 或 `|` 彼此分隔的一个或多个模式的列表。`&` 表示必须匹配所有模式，而 `|` 要求仅匹配一个模式。复合模式可以由下列一个或多个子模式组成：

`?(pattern-list)` 选择性地匹配任何指定模式之一。

`*(pattern-list)` 匹配出现的零个或多个指定模式。

`+(pattern-list)` 匹配出现的一个或多个指定模式。

`{n}(pattern-list)` 匹配指定模式的 *n* 个实例。

`{m,n}(pattern-list)` 匹配出现的 *m* 到 *n* 个指定模式。如果省略了 *m*，则使用 `0`。如果省略了 *n*，则至少匹配出现的 *m* 个指定模式。

`@(pattern-list)` 与某个指定模式完全匹配。



!(*pattern-list*) 与除某个指定模式以外的所有内容匹配。

缺省情况下，每个模式或子模式匹配最长的字符串，此字符串可能与生成最长的完整匹配项相一致。如果可能有多个匹配项，则选择起始位置离字符串开头最近的匹配项。但是，对于每个复合模式，可以在 ( 前面插入一个 -，以便使用指定 *pattern-list* 的最短匹配项。

如果 *pattern-list* 包含在括号中，则会以特殊方式处理反斜杠字符 \，即使它位于字符类中也是如此。所有 ANSI-C 转义符均可识别，并且与指定字符匹配。此外，还可识别下列转义序列：

- \d 匹配数字类中的任意字符。
- \D 匹配数字类中不包括的任意字符。
- \s 匹配空格类中的任意字符。
- \S 匹配空格类中不包括的任意字符。
- \w 匹配单词类中的任意字符。
- \W 匹配单词类中不包括的任意字符。

格式为 *%(pattern-pairs)* 的模式是可用于匹配嵌套字符表达式的子模式。每个 *pattern-pair* 都是一个包含两个字符的序列，该序列不能包含 & 或 |。第一个 *pattern-pair* 指定匹配项的起始字符和结束字符。每个后续 *pattern-pair* 表示在计算起始字符和结束字符匹配项数目时跳过的嵌套组的起始字符和结束字符。如果 *pattern-pair* 的第一个字符是除下列字符以外的字母数字，则行为是未指定的：

- D 导致结束字符在未找到匹配项的情况下终止对此模式的搜索。
- E 导致将结束字符解释为转义符。
- L 导致将结束字符解释为引号字符，从而导致在查找匹配项时忽略所有字符。
- Q 导致将结束字符解释为引号字符，从而导致在查找匹配项时忽略除任何转义符以外的所有字符。

*%({}Q"E\)* 匹配从 { 开始的字符，直到找到匹配的 } 为止（位于双引号字符串中或前面带有转义符 \ 的任何 { 或 } 不会计算在内）。如果没有 {}，此模式则与任何 C 语言字符串匹配。

复合模式中的每个子模式都根据 ( 在此模式中的位置从 1 开始进行编号。序列 \*n*（其中 *n* 表示一个数字，并且 \*n* 位于第 *n* 个子模式之后）匹配与子模式自身相同的字符串。

模式可包含 *~(options:pattern-list)* 格式的子模式，其中可省略 *options* 或 *:pattern-list*。与其他复合模式不同的是，这些子模式不会计算在带有编号的子模式之内。如果存在 *options*，则可以包括下列一个或多个字符：



- + 启用下列选项。这是缺省值。
- 禁用下列选项。
- E 模式的提示使用扩展的正则表达式语法，例如，`egrep(1)` 命令。
- F 模式的提示使用 `fgrep(1)` 表达式语法。
- g 归档最长匹配项（最长一致）。  
这是缺省值。
- G 模式的提示使用基本正则表达式语法，例如，`grep(1)` 命令。
- i 将匹配项视为不区分大小写。
- K 模式的提示使用 shell 模式语法。  
这是缺省值。
- l 将模式固定在左侧。  
这是 k 样式模式的缺省值。
- N 忽略此字符。但是，如果该字符是第一个字母，用于生成文件名，并且没有匹配项，则文件模式将扩展为空字符串。
- r 将模式固定在右侧。  
这是 k 样式模式的缺省值。

如果同时指定了 *options* 和 *:pattern-list*，这些选项仅适用于 *pattern-list*。否则，这些选项将保持有效，直到由后续 *~(...)* 禁用或位于含有 *~(...)* 的子模式的末尾为止。

## 引用

定义中列出的每个元字符都对 shell 具有特殊意义。

- g 归档最长匹配项（最长一致）。这是缺省值。
- i 将匹配项视为不区分大小写。

如果同时指定了 *options* 和 *:pattern-list*，这些选项仅适用于 *pattern-list*。否则，这些选项将保持有效，直到由后续 *~(...)* 禁用或位于含有 *~(...)* 的子模式的末尾为止。

本手册页定义部分中列出的每个元字符都对 shell 具有特殊意义，并且会导致单词结束（除非这些元字符带有引号）。通过在字符前面添加一个反斜杠（\），可以引用此字符，即，使此字符代表其自身。\\ 换行符对将被删除。前面没有 \$ 的单引号对（' '）中包含的所有字符将被引起来。单引号无法显示在单引号中。如果单引号前面带有不带引号的 \$，单引号中的字符串将作为 ANSI-C 字符串处理，但下列字符串除外：

- \0 导致忽略字符串的提示。
- \cx 扩展至字符 CTRL-x。

`\C[.name.]` 扩展至整理元素 *name*。

`\e` 等效于转义符 (ASCII 033)。

`\E` 等效于转义符 (ASCII 033)。

参数和命令替换在双引号 (") 中执行，并且 \ 引用字符 \、'、" 和 \$。在 C 或 POSIX 语言环境中，将忽略带有双引号的字符串前面的 \$，在其他语言环境中，这可能导致将此字符串替换为语言环境特定的字符串。当未带有引号时，或者当用作变量赋值的值或文件名时，\$\* 的含义与 \$@ 相同。但是，当用作命令参数时，"\$\*" 等效于 "\$1\$d\$2d..."，其中 *d* 是 IFS 变量的第一个字符，而 "\$@" 等效于 "\$1" "\$2" ... 在重音符引号 (") 中，\ 将字符 \、' 和 \$ 引起来。如果重音符引号出现在双引号中，\ 还会引用字符 "。

通过引用保留字的任何字符，可以删除保留字或别名的特殊意义。函数名称或内置命令名称的识别方式无法通过引用它们来进行更改。

## 算术计算

此 shell 通过执行算术计算来进行算术替换，以便计算算术命令、索引数组下标以及内置命令 `shift` 和 `let` 的参数。此外，还会对内置命令 `printf` 的参数操作数执行算术计算，该命令与格式操作数中的数字格式说明符相对应。请参见 `printf(1)`。使用双精度浮点运算或长整型双精度浮点数对提供此数据类型的系统执行计算。浮点常量遵循 ANSI-C 编程语言浮点约定。整数常量遵循 ANSI-C 编程语言整数常量约定，但只能识别单字节字符常量，而不能识别字符强制类型转换。常量可以采用 `[base#]n` 格式，其中 *base* 是介于 2 到 64 之间的表示算术基的十进制数字，*n* 是该基中的数字。大于 9 的数字分别通过小写字母、大写字母、@ 和 \_ 表示。对于小于或等于 36 的基，可以交替使用大写和小写字符。

算术表达式使用与 C 语言相同的表达式语法、优先级和关联。所有适用于浮点值的 C 语言运算符均可使用。此外，运算符 `**` 可用于求幂。同乘法相比，它具有较高的优先级，并且会保留关联性。如果算术变量或子表达式的值可表示为长整数，则可以执行所有 C 语言整数算术运算。算术表达式中的名称可引用变量，而不必使用参数扩展语法。引用变量时，该变量的值将作为算术表达式计算。

C 数学库包含的下列任何数学库函数均可在算术表达式中使用：

```
abs acos acosh asin asinh atan atan2 atanh cbrt
copysign cos cosh erf erfc exp exp2 expm1 fabs
fdim finite floor fma fmax fmod hypot ilogb
int isinf isnan lgamma log log2 logb
nearbyint nextafter nexttoward pow remainder
rint round sin sinh sqrt tan tanh tgamma trunc
```

使用 `typeset` 特殊内置命令的 `-E [n]` 或 `-F [n]` 选项可以指定作为双精度浮点数的变量的内部表示。`-E` 选项导致在扩展值时使用科学记数法表示此值的扩展。可选选项参数 *n* 定义重要数字的数目。`-F` 选项导致在扩展时将扩展表示为十进制浮点数。此种情况下，可选选项参数 *n* 定义小数点后面的小数位数。

使用 `typeset` 特殊内置命令的 `-i [n]` 选项可以指定变量的内部整数表示。可选选项参数 `n` 指定扩展变量时使用的算术基。如果未指定算术基，则使用基 10。

使用 `-E`、`-F` 或 `-i` 选项对变量的每个赋值的值执行算术计算。向整数类型的变量分配浮点数会导致截断小数部分。

#### 提示

当交互使用时，`shell` 在扩展 `PS1` 以便进行参数扩展、命令替换和算术替换之后通过 `PS1` 的值进行提示，然后再读取命令。此外，提示中的每个 `!` 都会替换为命令编号。只有 `#!` 才能在提示中放置 `!`。如果在任何时间键入了换行符，并且需要进行更多输入才能完成命令，则会发出辅助提示，即 `PS2` 的值。

#### 条件表达式

**条件表达式** 与 `[]` 复合命令配合使用，以便测试文件属性和比较字符串。不会对 `[]` 和 `]]` 之间的单词执行字段分隔和文件名生成。

每个表达式均可由下面的一个或多个一元表达式或二元表达式构成：

|                         |                                                       |
|-------------------------|-------------------------------------------------------|
| <code>-a file</code>    | True (如果 <i>file</i> 存在)。                             |
|                         | 此选项与 <code>-e</code> 相同。此选项已过时。                       |
| <code>-b file</code>    | True (如果 <i>file</i> 存在且为块特殊文件)。                      |
| <code>-c file</code>    | True (如果 <i>file</i> 存在且为字符特殊文件)。                     |
| <code>-d file</code>    | True (如果 <i>file</i> 存在且为目录)。                         |
| <code>-e file</code>    | True (如果 <i>file</i> 存在)。                             |
| <code>-f file</code>    | True (如果 <i>file</i> 存在且为普通文件)。                       |
| <code>-g file</code>    | True (如果 <i>file</i> 存在且设置了其 <code>setgid</code> 位)。  |
| <code>-G file</code>    | True (如果 <i>file</i> 存在，并且其组与此进程的有效组 ID 匹配)。          |
| <code>-h file</code>    | True (如果 <i>file</i> 存在且为符号链接)。                       |
| <code>-k file</code>    | True (如果 <i>file</i> 存在且设置了其 <code>sticky</code> 位)。  |
| <code>-L file</code>    | True (如果 <i>file</i> 存在且为符号链接)。                       |
| <code>-n string</code>  | True (如果 <i>string</i> 的长度为非零)。                       |
| <code>-N file</code>    | True (如果 <i>file</i> 存在，并且修改时间大于上次访问时间)。              |
| <code>-o option</code>  | True (如果已启用名为 <i>option</i> 的选项)。                     |
| <code>-o ?option</code> | True (如果名为 <i>option</i> 的选项为有效选项名称)。                 |
| <code>-O file</code>    | True (如果 <i>file</i> 存在，并且归此进程的有效用户 ID 所有)。           |
| <code>-p file</code>    | True (如果 <i>file</i> 存在且为 <code>FIFO</code> 特殊文件或管道)。 |
| <code>-r file</code>    | True (如果 <i>file</i> 存在，并且可供当前进程读取)。                  |

|                                   |                                                                                                                                                   |
|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-s file</code>              | True (如果 <i>file</i> 存在且其大小大于零)。                                                                                                                  |
| <code>-S file</code>              | True (如果 <i>file</i> 存在且为套接字)。                                                                                                                    |
| <code>-t fildes</code>            | True (如果文件描述符编号 <i>fildes</i> 已打开且与终端设备相关联)。                                                                                                      |
| <code>-u file</code>              | True (如果 <i>file</i> 存在且设置了其 <code>setuid</code> 位)。                                                                                              |
| <code>-w file</code>              | True (如果 <i>file</i> 存在, 并且可供当前进程写入)。                                                                                                             |
| <code>-x file</code>              | True (如果 <i>file</i> 存在, 并且可供当前进程执行)。如果 <i>file</i> 存在且为目录, 并且如果当前进程有权在此目录中搜索, 则为 True。                                                           |
| <code>-z string</code>            | True (如果 <i>string</i> 的长度为零)。                                                                                                                    |
| <code>file1 -ef file2</code>      | True (如果 <i>file1</i> 和 <i>file2</i> 存在, 并且引用同一文件)。                                                                                               |
| <code>file1 -nt file2</code>      | True (如果 <i>file1</i> 存在且 <i>file2</i> 不存在, 或者 <i>file1</i> 比 <i>file2</i> 更新)。                                                                   |
| <code>file1 -ot file2</code>      | True (如果 <i>file2</i> 存在且 <i>file1</i> 不存在, 或者 <i>file1</i> 比 <i>file2</i> 更旧)。                                                                   |
| <code>string</code>               | True (如果 <i>string</i> 不为 null)。                                                                                                                  |
| <code>string == pattern</code>    | True (如果 <i>string</i> 与 <i>pattern</i> 匹配)。可引用 <i>pattern</i> 的任何部分, 以将其作为字符串匹配。成功匹配 <i>pattern</i> 后, <code>.sh.match</code> 数组变量将包含匹配项和子模式匹配项。 |
| <code>string = pattern</code>     | 与 <code>==</code> 相同, 但已过时。                                                                                                                       |
| <code>string != pattern</code>    | True (如果 <i>string</i> 与 <i>pattern</i> 不匹配)。如果 <i>string</i> 与 <i>pattern</i> 匹配, <code>.sh.match</code> 数组变量将包含匹配项和子模式匹配项。                      |
| <code>string =~ ere</code>        | True (如果 <i>string</i> 与模式 <code>~(E)ere</code> 匹配, 其中 <i>ere</i> 是扩展的正则表达式)。                                                                     |
| <code>string1 &lt; string2</code> | True (如果根据字符的 ASCII 值, <i>string1</i> 位于 <i>string2</i> 之前)。                                                                                      |
| <code>string1 &gt; string2</code> | True (如果根据字符的 ASCII 值, <i>string1</i> 位于 <i>string2</i> 之后)。                                                                                      |

在下列每个表达式中, 如果 *file* 格式为 `/dev/fd/n` (其中 *n* 为整数), 测试则应用于描述符编号为 *n* 的打开文件。支持下列已过时的算术比较:

|                            |                                           |
|----------------------------|-------------------------------------------|
| <code>exp1 -eq exp2</code> | True (如果 <i>exp1</i> 等于 <i>exp2</i> )。    |
| <code>exp1 -ge exp2</code> | True (如果 <i>exp1</i> 大于或等于 <i>exp2</i> )。 |
| <code>exp1 -gt exp2</code> | True (如果 <i>exp1</i> 大于 <i>exp2</i> )。    |
| <code>exp1 -le exp2</code> | True (如果 <i>exp1</i> 小于或等于 <i>exp2</i> )。 |
| <code>exp1 -lt exp2</code> | True (如果 <i>exp1</i> 小于 <i>exp2</i> )。    |
| <code>exp1 -ne exp2</code> | True (如果 <i>exp1</i> 不等于 <i>exp2</i> )。   |

通过使用下列任何基元，可以使用这些基元构成复合表达式（这些基元按优先级的降序顺序列出）：

|                                          |                                                           |
|------------------------------------------|-----------------------------------------------------------|
| (表达式)                                    | True（如果 <i>expression</i> 为 True）。用于对表达式进行分组。             |
| ! 表达式                                    | True（如果 <i>expression</i> 为 False）。                       |
| <i>expression1</i> && <i>expression2</i> | True（如果 <i>expression1</i> 和 <i>expression2</i> 均为 True）。 |
| <i>expression1</i>    <i>expression2</i> | True（如果 <i>expression1</i> 或 <i>expression2</i> 为 True）。  |

## 输入和输出

执行命令之前，使用 shell 解释的特殊表示法可以重定向该命令的输入和输出。以下内容可显示在简单命令中的任意位置，也可以位于命令之前或之后，并且不会传递给调用的命令。除非此部分另行说明，否则在使用 *word* 或 *digit* 之前执行命令替换、参数扩展和算术替换。仅当 shell 为交互式 shell 并且模式与某个文件匹配时，才会执行文件名生成。不会执行字段分隔。

在下列每个重定向中，如果 *file* 格式为 */dev/sctp/host/port*、*/dev/tcp/host/port* 或 */dev/udp/host/port*（其中 *host* 为主机名或主机地址，*port* 为名称指定的服务或整数端口号），重定向会尝试建立到相应套接字的 *tcp*、*sctp* 或 *udp* 连接。

禁止在重定向运算符的字符之间使用中间空格。

|                   |                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| < <i>word</i>     | 使用文件 <i>word</i> 作为标准输入（文件描述符 0）。                                                                                                                                                                                                                                                                                                                         |
| > <i>word</i>     | 使用文件 <i>word</i> 作为标准输出（文件描述符 1）。如果文件不存在，则会创建一个文件。如果文件存在，并且启用了 <i>noclobber</i> 选项，则会导致错误。否则会将其截断为零长度。                                                                                                                                                                                                                                                    |
| >  <i>word</i>    | 与 > 相同，但它会覆盖 <i>noclobber</i> 选项。                                                                                                                                                                                                                                                                                                                         |
| >> <i>word</i>    | 使用文件 <i>word</i> 作为标准输出。如果文件存在，则会将输出附加到此文件（通过首先查找到文件结尾）。否则会创建文件。                                                                                                                                                                                                                                                                                          |
| <> <i>word</i>    | 打开文件 <i>word</i> 以便作为标准输入读取和写入。                                                                                                                                                                                                                                                                                                                           |
| <<[-] <i>word</i> | 读取在删除任何引用后与 <i>word</i> 相同的行之前或文件结尾之前的 shell 输入。不会对 <i>word</i> 执行参数替换、命令替换、算术替换或文件名生成。生成的名为 <i>here-document</i> 的文档将成为标准输入。如果引用了 <i>word</i> 的任何字符，则不会对文档的字符进行任何解释。否则会执行参数扩展、命令替换和算术替换，忽略 \ 换行符，并且必须使用 \ 引用字符 \、\$、'。如果将 - 附加到 <<，则会从 <i>word</i> 和文档删除所有前导制表符。如果将 # 附加到 <<，则会从文档的第一行中删除前导空格和制表符，并从其余行和 <i>word</i> 中最多删除等效的缩排。制表停止位置应每 8 列出现一次，以便确定缩排。 |
| <<< <i>word</i>   | 生成 <i>here document</i> 的简捷形式，在此简捷形式中，在执行任何参数扩展、命令替换和算术替换之后， <i>word</i> 将成为 <i>here-document</i> 的内容。                                                                                                                                                                                                                                                    |

|                              |                                                                                                                                     |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;&amp;digit</code>  | 从文件描述符 <i>digit</i> 复制标准输入，类似地，对于标准输出，请使用 <code>&gt;&amp;digit</code> 。请参见 <a href="#">dup(2)</a> 。                                 |
| <code>&lt;&amp;digit-</code> | 将 <i>digit</i> 指定的文件描述符移至标准输入。类似地，对于标准输出，请使用 <code>&gt;&amp;digit-</code> 。                                                         |
| <code>&lt;&amp;-</code>      | 关闭标准输入。类似地，对于标准输出，请使用 <code>&gt;&amp;-</code> 。                                                                                     |
| <code>&lt;&amp;p</code>      | 将协同进程的输入移至标准输入。                                                                                                                     |
| <code>&gt;&amp;p</code>      | 将协同进程的输出移至标准输出。                                                                                                                     |
| <code>&lt;#((expr))</code>   | 计算算术表达式 <i>expr</i> ，并将文件描述符 0 放置在离文件开头的字节数为生成的值的相应位置。当计算 <i>expr</i> 时，变量 <code>CUR</code> 和 <code>EOF</code> 的计算结果分别为当前偏移和文件结尾偏移。 |
| <code>&gt;#((expr))</code>   | 与 <code>&lt;#</code> 相同，但应用于文件描述符 1。                                                                                                |
| <code>&lt;#pattern</code>    | 向前查找到包含模式的下一行的开头。                                                                                                                   |
| <code>&lt;##pattern</code>   | 与 <code>&lt;#</code> 相同，但将跳过的文件部分复制到标准输出。                                                                                           |

如果某个重定向运算符前面带有一个数字，并且没有任何中间空格，引用的文件描述符编号即为该数字（而非缺省值 0 或 1）指定的文件描述符编号。如果除 `>&-`、`>#` 和 `<#` 格式以外的其他某个重定向运算符前面带有 `{varname}`，且无中间空格，shell 则会选择 `> 10` 的文件描述符编号，并将其存储在变量 *varname* 中。如果 `>&-`、任何 `>#` 及 `<#` 格式前面带有 `{varname}`，*varname* 的值定义要关闭或定位的文件描述符。例如：

```
... 2>&1
```

表示打开文件描述符 2 以便作为文件描述符 1 的副本写入，

```
exec [n]<file
```

表示打开文件以便读取，并将文件描述符编号存储在变量 *n* 中。重定向的指定顺序至关重要。shell 根据评估时的（文件描述符、文件）关联来评估每个重定向。例如：

```
... 1>fname 2>&1
```

首先将文件描述符 1 与文件 *fname* 关联。然后，它将文件描述符 2 与文件描述符 1 的关联文件（即 *fname*）相关联。如果颠倒重定向顺序，文件描述符 2 将与终端关联（假定文件描述符 1 已与此终端关联），文件描述符 1 将与文件 *fname* 关联。如果命令后跟有 `&`，并且作业控制处于非活动状态，该命令的缺省标准输入则为空文件 `/dev/null`。否则，命令执行环境包含调用根据输入和输出规范修改的 shell 的文件描述符。

**环境** 是按照标准参数列表的相同传递方式传递给执行程序的名称-值对列表。请参见 [environ\(5\)](#)。

这些名称必须是**标识符**，值必须是字符串。shell 采用多种方式与环境交互。shell 在调用时扫描环境，为找到的每个名称创建一个变量，为变量指定相应值和属性，并将其

标记为 `export`。执行命令继承此环境。如果用户使用 `export` 或 `typeset -x` 命令修改了这些变量的值或创建了新的变量，这些变量将成为环境的一部分。因此，向任何执行命令显示的环境包括 `shell` 最初继承的所有名称-值对（当前 `shell` 可修改这些名称-值对的值）以及必须在 `export` 或 `typeset -x` 命令中说明的所有添加内容。通过在任何简单命令或函数前面添加一个或多个变量赋值作为前缀，可以扩展这些简单命令或函数的环境。变量赋值参数是**标识符=值**格式的单词。因此：

```
TERM=450 cmd args
```

和

```
(export TERM; TERM=450; cmd args)
```

等效（就执行 `cmd` 而言，**内置命令**部分中列出的特殊内置命令除外，这些命令的前面带有一个剑形符号）。如果设置了已过时的 `-k` 选项，所有变量赋值参数都会放置到环境中，即使这些变量赋值参数出现在命令名称之后。

下面的示例首先输出 `a=b c`，然后输出 `c`：

```
echo a=b c
set -k
echo a=b c
```

此功能旨在用于为早期 `shell` 版本编写的脚本，强烈建议不要将此功能用于新脚本中。

## 函数

出于历史原因，定义函数有两种方法，即 `name()` 语法和 `function name` 语法。本手册页的**命令**部分介绍了这两种方法。

`Shell` 函数在内部读取和存储。别名在读取函数时解析。函数的执行方式与将参数作为位置参数传递的命令相似。有关详细信息，请参见本手册页的**执行**部分。

根据 `function name` 语法定义并按照名称调用的函数与调用者在同一进程中执行，并与调用者共享所有文件和现有工作目录。调用者捕获的陷阱在函数中重置为其缺省操作。函数未捕获或忽略的陷阱条件导致该函数终止，并将此条件传递给调用者。函数完成之后，在 `EXIT` 时在函数内部设置的陷阱将在调用者的环境中执行。通常在调用程序和函数之间共享变量。但是，在函数中使用的 `typeset` 特殊内置命令定义其作用域包括当前函数的局部变量。这些局部变量可以在调用之前传递给变量赋值列表中的调用函数，或者作为以名称引用形式传递的参数传递。函数中的错误将控制权返回给调用者。

通过 `.` 特殊内置命令调用的使用 `name()` 语法定义的函数以及使用 `function name` 语法定义的函数在调用者的环境中执行，并且与调用者共享所有变量和陷阱。在执行这些函数的过程中出现的错误导致函数所在的脚本中止。

特殊内置命令 `return` 用于从函数调用返回。

使用 `typeset` 特殊内置命令的 `-f` 或 `+f` 选项可以列出函数名称。使用 `-f` 还可以列出函数文本（如果可用）。使用 `unset` 特殊内置命令的 `-f` 选项可以取消定义函数。



函数通常在 shell 执行 shell 脚本时取消设置。必须跨不同 shell 调用定义的函数应放置在目录中，并且 `FPATH` 变量应包含此目录的名称。也可以在 `ENV` 文件中指定这些函数。

## 规程函数

每个变量都可以关联有零个或多个规程函数。shell 最初了解规程名称 `get`、`set`、`append` 和 `unset`，但是在大多数系统中，通过 `builtin` 内置实用程序提供的 C 编程接口扩展可以在运行时添加其他规程名称。如果为变量定义了 `get` 规程，则会在引用指定变量时调用此规程。如果为规程函数中的变量 `.sh.value` 分配了一个值，引用的变量的计算结果将为此值。如果为变量定义了 `set` 规程，则会在为指定变量分配值时调用此规程。如果为变量定义了 `append` 规程，则会在将值附加到指定变量时调用此规程。在调用规程之前，为变量 `.sh.value` 指定该变量的值，在规程结束之后，为该变量分配 `.sh.value` 的值。如果在规程中取消设置 `.sh.value`，该值将保持不变。如果为变量定义了 `unset` 规程，则会在取消设置指定变量时调用此规程。除非从此规程函数中显式取消设置变量，否则无法取消设置此变量。

变量 `.sh.name` 包含调用了规程函数的变量的名称，`.sh.subscript` 是此变量的下标，`.sh.value` 包含在 `set` 规程函数中分配的值。对于 `set` 规程，更改 `.sh.value` 会更改分配的值。

## 作业

如果启用 `set` 命令的 `monitor` 选项，交互式 shell 会将作业与每个管道相关联。shell 保存当前作业的表，并使用 `jobs` 命令输出此表，然后为这些作业分配较小的整数。使用 `&` 异步启动作业时，shell 输出如下行：

```
[1] 1234
```

这表示异步启动的作业的作业编号为 1，并且该作业具有一个（顶级）进程，其进程 ID 为 1234。

如果您正在运行作业，并且希望停止此作业，按 `Ctrl-z` 可向当前作业发送一个 `STOP` 信号。shell 通常会显示一则该作业已停止的消息，并且会显示其他提示。然后，您可以处理此作业的状态，使用 `bg` 命令将其放置到后台，或者运行某些其他命令，最后使用前台命令 `fg` 将其放回到前台。`Ctrl-z` 立即生效，并在键入时放弃暂挂输出和未读输入，因此它与中断相似。

如果在后台运行的作业尝试从终端读取，则会停止该作业。通常允许后台作业生成输出，但是通过指定命令 `sttytostop` 可以禁用此功能。如果设置了此 `tty` 选项，后台作业将在尝试生成输出时停止，就像在尝试读取输入时停止一样。

在 shell 中引用作业有多种方法。根据作业的任何进程的进程 ID 或下列各项之一可以引用作业：

- `%number` 带有指定编号的作业。
- `%string` 命令行以字符串开头的所有作业。
- `%?string` 命令行包含字符串的所有作业。
- `%%` 当前作业。



%+           等效于%%。

%-           以前的作业。

当进程更改状态时，shell 可立即获悉。当作业已被阻塞以致无法执行其他进程时，shell 通常会向您发送通知，但仅在输出提示前发送此通知。其目的是为了避免打扰您的工作。set 命令的通知选项导致 shell 在作业发生更改时立即输出这些作业更改消息。

启用 monitor 选项后，完成的每个后台作业都会触发为 CHLD 设置的所有陷阱。

如果尝试在作业运行或停止时保留 shell，将警告您 You have stopped(running) jobs。您可以使用 jobs 命令来查看这些作业。如果您立即尝试再次退出，shell 不会向您发出第二次警告，而会终止停止的作业。登录 shell 收到 HUP 信号后，它会将 HUP 信号发送到未使用 disown 内置命令否认的每个作业。

**信号**           如果调用的命令后跟有 &，并且 monitor 选项处于非活动状态，则会忽略该命令的 INT 和 QUIT 信号。否则，信号具有 shell 从父级继承的值。请参见 trap 内置命令。

**执行**           每次读取命令时都会执行替换。如果命令名称与本手册页的**特殊内置命令**部分中的某个命令名称相匹配，则会在当前 shell 进程中执行此命令。接着，系统会检查此命令名称，查看其是否与某个用户定义函数匹配。如果匹配，则会保存位置参数，并将其重置为函数调用的参数。函数同样在当前 shell 进程中执行。函数完成或发出返回命令之后，系统将恢复位置参数列表。对于使用 function name 语法定义的函数，系统将执行在 EXIT 时在函数中设置的所有陷阱。函数的退出值即为执行的最后一个命令的值。如果命令名称不是特殊内置命令或用户定义函数，而是某个内置命令，则会在当前 shell 进程中执行此命令。

shell 变量 PATH 定义此命令所在的目录的搜索路径。备用路径名称使用冒号(:) 分隔。缺省路径为 /bin:/usr/bin:，并按此顺序指定 /bin、/usr/bin 和当前目录。可以使用两个或多个相邻冒号指定当前目录，也可以使用位于路径列表开头或末尾的一个冒号来指定当前目录。如果命令名称包含斜杠(/)，则不会使用搜索路径。否则，将在此路径的每个目录中搜索具有指定名称的可执行文件（不是目录）。如果找到此文件，并且如果 shell 确定存在与指定路径名对应的内置命令版本，则会在当前进程中调用此内置命令。如果找到此文件，并且此目录还包含在 FPATH 变量的值中，则会将此文件装载到当前 shell 环境，就像是 . 命令的参数一样（除非只扩展预设别名），并按照本手册页中的说明执行具有指定名称的函数。如果未找到此文件，但找到文件 .paths，并且此文件包含一个 FPATH=path 格式的行，其中 path 是现有目录，并且此目录包含具有指定名称的文件，那么，则会将此文件装载到当前 shell 环境，就像是 .特殊内置命令的参数一样，并执行具有指定名称的函数。否则，如果找到此文件，则会创建一个进程，并尝试使用 exec(2) 执行命令。

找到可执行文件后，系统将在此可执行文件所在的目录中搜索名为 .paths 的文件。如果找到此文件，并且该文件包含一个 BUILTIN\_LIB=value 格式的行，则会搜索根据 value 命名的库，就像是 builtin -f 的选项参数一样，并且如果此文件包含具有指定名称的内置命令，则会执行此内置命令，而不会执行具有此名称的命令。否

则，如果找到此文件，并且该文件的第一行或第二行中包含一个 *name=value* 格式的行，则会通过将 *value* 指定的目录添加到目录列表开头来修改环境变量 *name*。如果 *value* 不是绝对目录，则会指定一个相对于可执行文件所在目录的目录。如果环境变量 *name* 不存在，则会将其添加到指定命令的环境列表中。

如果此文件具有执行权限，但不是 *a.out* 文件，则假定此文件是含有 shell 命令的文件。系统会派生一个单独的 shell 来读取此文件。此种情况下，将删除所有非导出的变量。如果 shell 命令文件没有读取权限，并且/或者如果在此文件中设置了 *setuid* 和 *setgid* 位，此 shell 则会执行其作业为设置权限的代理，并执行此 shell，同时将此 shell 命令文件作为打开文件文件向下传递。带有括号的命令在子 shell 中执行，而不会删除非导出的变量。

#### 命令重新输入

从终端设备输入的最后一个 HISTSIZE（缺省值为 512）命令的文本保存在历史文件中。如果未设置 HISTFILE 变量或者如果此变量指定的文件无法写入，则会使用 *\$HOME/.sh\_history*。shell 可以访问使用同一名称 HISTFILE 的所有交互式 shell 的命令。内置命令 *hist* 用于列出或编辑此文件的某个部分。要编辑或列出的文件部分可以根据编号进行选择，或者通过指定此命令的第一个字符或前几个字符来进行选择。可以指定一个命令或一系列命令。如果未将编辑器程序指定为 *hist* 的参数，则会使用变量 HISTEDIT 的值。如果取消设置 HISTEDIT，则使用已过时的变量 FCEDIT。如果未定义 FCEDIT，则使用 */bin/ed*。在退出编辑器时，除非您在未写入的情况下退出，否则将重新输出和执行已编辑的命令。*-s* 选项（以及已过时的版本中的编辑器名称 *-*）用于跳过编辑短语并重新执行此命令。此种情况下，在执行之前，可使用 *old=new* 格式的替换参数来修改命令。例如，使用 '*hist -s*' 的预设别名 *r*，键入 '*r bad=good c*' 将重新执行以字母 *c* 开头的最新命令，并将出现的第一个字符串 *bad* 替换为字符串 *good*。

#### 内嵌编辑选项

通常，从终端设备输入的每个命令行在键入时仅后跟有一个换行符（回车或换行符）。如果已激活 *emacs*、*gmacs* 或 *vi* 选项，用户则可以编辑命令行。要进入这些编辑模式，请设置相应选项。每次向 VISUAL 或 EDITOR 变量分配一个以上述选项名称结尾的值时，都会自动选择编辑选项。

编辑功能要求用户终端将回车作为无换行符的回车接受，并且空格必须覆盖屏幕上的当前字符。

除非启用 *multiline* 选项，否则编辑模式会实现在当前行查看整个窗口的内容的概念。窗口宽度为 COLUMNS 的值（如果定义），否则为 *80*。如果窗口宽度太小以致无法显示提示，并且至少保留了 *8* 列来输入相关输入，则会从左侧截断提示。如果此行长于窗口宽度减去 *2* 的值，则会在窗口末端显示一个标记以便通知用户。当光标移动并达到窗口边界时，该窗口会将光标放置在中间位置。如果该行向窗口右侧、左侧或两侧延伸，标记分别为 *>*、*<*、*\**。

每种编辑模式下的搜索命令都可以提供对历史文件的访问。仅匹配字符串（而非模式），但字符串中的前导 *^* 将匹配范围限制为从行中的第一个字符开始。

每种编辑模式都具有列出与部分输入的单词匹配的文件或命令的操作。如果对行中的第一个单词或者对 ;、|、& 或 ( 后的第一个单词应用此功能，并且此单词不以 ~ 开头或不包含 /，则会显示 PATH 变量定义的可与部分单词匹配的别名、函数和可执行命令的列表。否则显示与指定单词匹配的文件列表。如果部分输入的单词不包含任何文件扩展字符，则会在生成这些列表之前附加一个 \*。显示生成的列表之后，将重新绘制输入行。这些操作分别称为命令行列出和文件名列出操作。此外，还存在一些称为命令名称完成和文件名完成的其他操作，这些操作计算匹配命令或文件的列表，但不会输出此列表，而是用完整匹配项或部分匹配项替换当前单词。对于文件名完成操作，如果匹配项是唯一的，则会附加一个 /（如果该文件为目录）和空格（如果该文件不为目录）。否则会使用所有匹配文件的最长共同前缀替换此单词。对于命令名称完成，仅使用最后一个 / 后面的文件名部分来查找最长的命令前缀。如果只有一个名称与此前缀匹配，则会将此单词替换为此命令名称，并在后面添加一个空格。当对未生成唯一匹配项的完成操作使用制表符时，后续制表符将提供带有编号的匹配项的列表。通过输入后跟有制表符的选择编号，可以选择特定项。

## 键绑定

KEYBD 陷阱可用于在键入键时拦截键，并更改 shell 实际看到的字符。当从终端读取时，将在输入每个字符（如果第一个字符为 ESC，则为字符序列）后执行此陷阱。

变量 `.sh.edchar` 包含生成此陷阱的字符或字符序列。在陷阱操作中更改 `.sh.edchar` 的值将导致 shell 的行为就像新值（而非原始值）是从键盘输入的一样。变量 `.sh.edcol` 设置为输入时光标所在的输入列编号。当处于 vi 插入模式时，变量 `.sh.edmode` 设置为 ESC，在其他情况下则设置为空。在分配给 `.sh.edchar` 的值前面添加 `${.sh.editmode}` 将导致 shell 更改为此控制模式（如果未处于此模式下）。

对于作为编辑指令的参数输入的字符，或者当读取字符搜索的输入时，不会调用此陷阱。

## emacs 编辑模式

通过启用 `emacs` 或 `gmacs` 选项，可以进入此模式。这两种模式之间的唯一区别在于它们对 ^T 的处理方式。要进行编辑，用户应将光标移至需要校正的位置，然后根据需要插入或删除字符或单词。所有编辑命令均为控制字符或转义序列。控制字符的表示法为后跟有字符的插入记号 (^)。

例如，^F 是 Ctrl/F 的表示法。按住 Ctrl（控制）键并按下 f 可以输入 ^F。不按 Shift 键。（表示法 ^? 表示 Del（删除）键。）

转义序列的表示法为后跟有字符的 M-。例如，通过先后按下 Esc (ASCII 033) 和 f 可以输入 M-f（发音为 Meta f）。M-F 是后跟有 F 的 ESC 的表示法。

所有编辑命令都可从行中的任意位置运行，而不仅限于行开头。除非另行说明，否则不会在编辑命令之后输入回车或换行符键。

^F           将光标向前（向右）移动一个字符。

M-[C        将光标向前（向右）移动一个字符。

M-f         将光标向前移动一个单词。对于 emacs 编辑器而言，单词是仅由字母、数字和下划线组成的字符串。

---

|                       |                                                                                                       |
|-----------------------|-------------------------------------------------------------------------------------------------------|
| <code>^B</code>       | 将光标向后（向左）移动一个字符。                                                                                      |
| <code>M-[D</code>     | 将光标向后（向左）移动一个字符。                                                                                      |
| <code>M-b</code>      | 将光标向后移动一个单词。                                                                                          |
| <code>^A</code>       | 将光标移至行开头。                                                                                             |
| <code>M-[H</code>     | 将光标移至行开头。                                                                                             |
| <code>^M</code>       | 将光标移至行尾。                                                                                              |
| <code>M-[Y</code>     | 将光标移至行尾。                                                                                              |
| <code>^]char</code>   | 将光标向前移动到当前行中的字符 <i>char</i> 。                                                                         |
| <code>M-^]char</code> | 将光标向后移动到当前行中的字符 <i>char</i> 。                                                                         |
| <code>^X^X</code>     | 交换光标和标记。                                                                                              |
| <i>erase</i>          | 删除上一字符。用户定义的清除字符使用 <code>stty(1)</code> 命令定义，该字符通常为 <code>^H</code> 或 <code>#</code> 。                |
| <i>lnext</i>          | 删除下一字符的编辑功能。用户定义的下一文本字符使用 <code>stty(1)</code> 命令定义，如果未定义此字符，则为 <code>^V</code> 。                     |
| <code>^D</code>       | 删除当前字符。                                                                                               |
| <code>M-d</code>      | 删除当前单词。                                                                                               |
| <code>M-^H</code>     | 元 Back Space 键。删除上一单词。                                                                                |
| <code>M-h</code>      | 删除上一单词。                                                                                               |
| <code>M-^?</code>     | 元 Del 键。删除上一单词。如果您使用的中断字符为 <code>^?(DEL</code> ，缺省值)，则此命令不会起作用。                                       |
| <code>^T</code>       | 在 <code>emacs</code> 模式下将当前字符与上一字符换位并向前移动光标。在 <code>gmacs</code> 模式下将前两个字符换位。                         |
| <code>^C</code>       | 大写当前字符。                                                                                               |
| <code>M-c</code>      | 大写当前单词。                                                                                               |
| <code>M-l</code>      | 将当前单词更改为小写字母。                                                                                         |
| <code>^K</code>       | 删除光标至行尾之间的内容。如果此命令前面带有一个其值小于当前光标位置的数字参数，则删除指定位置到光标之间的内容。如果此命令前面带有一个其值大于当前光标位置的数字参数，则删除光标到指定光标位置之间的内容。 |
| <code>^由 w</code>     | 删除光标到标记之间的内容。                                                                                         |
| <code>M-p</code>      | 将光标到标记之间的区域推送到栈中。                                                                                     |

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>kill</i>            | 删除整个当前行。用户定义的删除字符使用 <code>stty(1)</code> 命令定义，该字符通常为 <code>^G</code> 或 <code>@</code> 。如果连续输入了两个删除字符，从此时起输入的所有删除字符都会导致一个换行符。当使用纸张终端时，此功能非常有用。                                                                                                                                                                                                                                                                                                                                     |
| <code>^Y</code>        | 恢复从行中删除的上的一项。在行中重新输出此项。                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <code>^L</code>        | 换行并输出当前行。                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>M-^L</code>      | 清除屏幕。                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>^@</code>        | 空字符。设置标记。                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>M-space</code>   | 元空格。设置标记。                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>^J</code>        | 新行。执行当前行。                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>^M</code>        | 返回。执行当前行。                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>EOF</code>       | 仅当当前行为空时，文件结尾字符（通常为 <code>^D</code> ）才会作为文件结尾处理。                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <code>^P</code>        | 获取上一命令。每次输入 <code>^P</code> 时，都会访问前面的上一命令。如果该命令不在多行命令的第一行中，则往回移动一行。                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>M-[A</code>      | 等效于 <code>^P</code> 。                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>M-&lt;</code>    | 获取最早（最旧）的历史行。                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>M-&gt;</code>    | 获取最近（最新）的历史行。                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <code>^N</code>        | 获取下一命令行。每次输入 <code>^N</code> 时，都会访问后面的下一命令。                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>M-[B</code>      | 等效于 <code>^N</code> 。                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>^Rstring</code>  | 颠倒含有 <i>string</i> 的上一命令行的搜索历史。如果指定参数 0，搜索则为正向搜索。 <i>string</i> 以回车或换行符结束。如果 <i>string</i> 前面带有 <code>^</code> ，匹配行必须以 <i>string</i> 开头。如果省略了 <i>string</i> ，则会访问包含最新的 <i>string</i> 的下一命令行。此种情况下，参数 0 将颠倒搜索方向。                                                                                                                                                                                                                                                                   |
| <code>^O</code>        | 运行。执行当前行，获取历史文件中相对于当前行的下一行。                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>M-digits</code>  | 转义。定义数字参数。这些数字作为下一命令的参数。接受参数的命令包括： <code>^F</code> 、 <code>^B</code> 、 <code>ERASE</code> 、 <code>^C</code> 、 <code>^D</code> 、 <code>^K</code> 、 <code>^R</code> 、 <code>^P</code> 、 <code>^N</code> 、 <code>^]</code> 、 <code>M-.</code> 、 <code>M-、</code> 、 <code>M-^</code> 、 <code>M-]</code> 、 <code>M-]</code> 、 <code>M-#</code> 、 <code>M-b</code> 、 <code>M-c</code> 、 <code>M-d</code> 、 <code>M-f</code> 、 <code>M-h</code> 、 <code>M-l</code> 和 <code>M-^H</code> 。 |
| <code>M-letter</code>  | 软键。根据名称 <i>letter</i> 在别名列表中搜索别名。如果定义了 <i>letter</i> 的名称，则在输入队列中插入其值。 <i>letter</i> 不能是此部分中的元函数之一。                                                                                                                                                                                                                                                                                                                                                                                |
| <code>M-[letter</code> | 软键。根据名称 <i>letter</i> 在别名列表中搜索别名。如果定义了此名称的别名，则在输入队列中插入其值。此命令可用于在许多终端上对功能键进行编程。                                                                                                                                                                                                                                                                                                                                                                                                    |

|       |                                                                                                      |
|-------|------------------------------------------------------------------------------------------------------|
| M-.   | 在行中插入上一命令的最后一个单词。如果此命令前面带有一个数字参数，此参数的值确定要插入的单词，而不是最后一个单词。                                            |
| M_    | 与 M-. 相同。                                                                                            |
| M-*   | 尝试根据当前单词生成文件名。如果单词不与任何文件匹配或不包含任何特殊模式字符，则会附加一个星号。                                                     |
| M-ESC | 按照本手册页中的说明执行命令或文件名完成操作。                                                                              |
| ^ITAB | 尝试按照本手册页中的说明执行命令或文件名完成操作。如果发生部分完成，重复此操作的行为就像输入了 M-= 一样。如果未找到匹配项或者在空格后输入，则会插入一个制表符。                   |
| M-=   | 如果此命令前面没有数字参数，则会按照本手册页中的说明生成匹配的命令或文件名的列表。否则，光标下的单词将替换为与最近生成的命令或文件列表中的数字参数的值相对应的项。如果光标不在单词上方，则会插入此单词。 |
| ^U    | 将下一命令的参数乘以 4。                                                                                        |
| \     | 对下一字符进行转义。如果前面带有 \，则可以在命令行或搜索字符串中输入编辑字符、用户的清除、删除和中断（通常为 ^?）字符。\<br>删除下一字符的编辑功能（如果有）。                 |
| M-^V  | 显示 shell 版本。                                                                                         |
| M-#   | 如果行不以 # 开头，则会在此行开头以及每个换行符后面插入一个 #，并输入此行。这会导致在历史文件中插入注释。如果行以 # 开头，则会删除 #，并删除每个换行符后面的一个 #。             |

## vi 编辑模式

有两种键入模式。最初，当您输入命令时，您处于输入模式下。要进行编辑，用户应通过键入 ESC (033) 进入控制模式，将光标移至需要校正的位置，然后根据需要插入或删除字符或单词。大多数控制命令都接受命令前面的可选重复 *count*。

当在大多数系统上处于 vi 模式时，如果速度为 1200 波特或更大值，并且命令包含任何控制字符，或者自输出提示以来所经过的时间不到 1 秒，则会在一开始时启用标准化处理，并重新回显命令。ESC 字符终止命令提示的标准化处理，然后用户可以修改命令行。此方案具有标准化处理以及对原始模式进行输入提示回显的优势。

如果还设置了选项 `viraw`，终端始终会禁用标准化处理。对于不支持两种备用的行尾分隔符的系统，此模式是固有的，并且对某些终端可能会很有用。

### 输入编辑命令

缺省情况下，编辑器处于输入模式。

支持下列输入编辑命令：

**ERASE**     使用 `stty` 命令定义的用户定义清除字符，通常为 ^H 或 #。删除上一字符。



- ^由 w** 删除由空格分隔的上一单词。在某些系统上，可能需要使用 `viraw` 选项，此命令才能正常运行。
- EOF** 除非设置了 `ignoreeof` 选项，否则行的第一个字符将导致 `shell` 终止。否则，将忽略此字符。
- lnext*** 使用 `stty(1)` 定义的下一用户定义文本字符，如果未定义此字符，则为 `^V`。删除下一字符的编辑功能（如果有）。在某些系统上，可能需要使用 `viraw` 选项，此命令才能正常运行。
- \** 将下一 ERASE 或 KILL 字符转义。
- ^I TAB** 尝试按照本手册页中的说明执行命令或文件名完成操作，并返回到输入模式。如果发生部分完成，重复此操作的行为就像从控制模式输入了 `=` 一样。如果未找到匹配项或者在空格后输入，则会插入一个制表符。

### 运动编辑命令

运动编辑命令可移动光标。

支持下列运动编辑命令：

- [count]l** 将光标向前（向右）移动一个字符。
- [count][C** 将光标向前（向右）移动一个字符。
- [count]w** 将光标向前移动一个字母数字单词。
- [count]W** 将光标移至空格后的下一个单词的开头。
- [count]e** 将光标移至单词末尾。
- [count]E** 将光标移至以当前空格分隔的单词的末尾。
- [count]h** 将光标向后（向左）移动一个字符。
- [count][D** 将光标向后（向左）移动一个字符。
- [count]b** 将光标向后移动一个单词。
- [count]B** 将光标移至以空格分隔的上一单词。
- [count]|** 将光标移至列 *count*。
- [count]fc** 在当前行中查找下一字符 *c*。
- [count]Fc** 在当前行中查找上一个字符 *c*。
- [count]tC** 等效于 `f` 后跟 `h`。
- [count]Tc** 等效于 `F` 后跟 `l`。
- [count];** 重复最后一个单字符查找命令 *count* 次：`f`、`F`、`t` 或 `T`。
- [count],** 反向最后一个单字符查找命令 *count* 次。

- 0 将光标移到行首。
- ^ 将光标移到行首。
- [H 将光标移到行中的第一个非空白字符。
- \$ 将光标移至行尾。
- [Y 将光标移至行尾。
- % 移动到对称的 (、)、{、}、[ 或 ]。如果光标不在本部分所述的任何一个字符上，将先在该行的剩余部分中搜索第一次出现的这些字符之一。

### 搜索编辑命令

搜索编辑命令可以访问您的命令历史记录。

支持下列搜索编辑命令：

- [count]k 获取上一命令。每次输入 k 时，都会访问之前的上一条命令。
- [count]- 获取上一命令。每次输入 k 时，都会访问之前的上一条命令。  
等效于 k。
- [count][A 获取上一命令。每次输入 k 时，都会访问之前的上一条命令。  
等效于 k。
- [count]j 获取下一条命令。每次输入 j 时，都会访问以后的下一条命令。
- [count]+ 获取下一条命令。每次输入 j 时，都会访问以后的下一条命令。  
等效于 j。
- [count][B 获取下一条命令。每次输入 j 时，都会访问以后的下一条命令。  
等效于 j。
- [count]G 获取命令编号 *count*。缺省命令为最早的历史命令。
- /字符串 在历史记录中向后搜索包含 *string* 的上一条命令。*string* 以回车或换行符结束。如果 *string* 前面带有 ^，匹配行必须以 *string* 开头。如果 *string* 为空，则使用上一个字符串。
- ?字符串 在历史记录中向前搜索包含 *string* 的上一条命令。*string* 以回车或换行符结束。如果 *string* 前面带有 ^，匹配行必须以 *string* 开头。如果 *string* 为空，则使用上一个字符串。  
  
除了搜索方向是向前外，其余均与 / 相同。
- n 向后搜索 / 或 ? 命令的最后一个模式的下一个匹配项。
- N 向前搜索 / 或 ? 的最后一个模式的下一个匹配项。



## 文本修改编辑命令

以下命令可修改行：

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| a               | 进入输入模式并在当前字符后输入文本。                                                          |
| A               | 将文本附加到行尾。等效于 \$a。                                                           |
| [count]cmotion  | 从当前字符删除到 <i>motion</i> 将光标移动到的字符，并进入输入模式。如果 <i>motion</i> 为 c，将删除整行并进入输入模式。 |
| c[count]motion  | 从当前字符删除到 <i>motion</i> 将移动到的字符。如果 <i>motion</i> 为 d，将删除整行。                  |
| C               | 从当前字符删除到行尾并进入输入模式。等效于 c\$。                                                  |
| S               | 等效于 cc。                                                                     |
| [count]s        | 在输入模式中替换光标下方的字符。                                                            |
| D[count]dmotion | 从当前字符删除到行尾。等效于 d\$。                                                         |
| d[count]motion  | 从当前字符删除到 <i>motion</i> 将移动到的字符。如果 <i>motion</i> 为 d，将删除整行。                  |
| i               | 进入输入模式并在当前字符前插入文本。                                                          |
| I               | 在行首之前插入文本。等效于 0i。                                                           |
| [count]P        | 在光标之前放置以前修改的文本。                                                             |
| [count]p        | 在光标之后放置以前修改的文本。                                                             |
| R               | 进入输入模式并将屏幕上的字符替换为您以覆盖方式键入的字符。                                               |
| [count]rc       | 将从当前光标位置开始的 <i>count</i> 个字符替换为 <i>c</i> ，并向前移动光标。                          |
| [count]x        | 删除当前字符。                                                                     |
| [count]X        | 删除前面的字符。                                                                    |
| [count].        | 重复上一条文本修改命令。                                                                |
| [count]~        | 转换从当前光标位置开始的 <i>count</i> 个字符的大小写，并向前移动光标。                                  |
| [count]_        | 促使附加上一条命令中的 <i>count</i> 单词并进入输入模式。如果省略 <i>count</i> ，则使用最后一个单词。            |
| *               | 促使 * 附加到当前的单词并尝试生成文件名。如果未找到匹配项，则会发出铃声。否则，将该单词替换为匹配模式并进入输入模式。                |
| \               | 按照本手册页中的说明执行命令或文件名完成操作。                                                     |

## 其他编辑命令

支持下列其他编辑命令：

|                             |                                                                                                                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>[count]ymotion</code> | 将当前字符到 <i>motion</i> 将光标移动到的字符全部移出。将移出的字符放在删除缓冲区中。文本和光标位置保持不变。                                                                                                                       |
| <code>y[count]motion</code> | 移出当前行。                                                                                                                                                                               |
| <code>yy</code>             | 移出从当前的光标位置直到行尾的当前行。等效于 <code>y\$</code> 。                                                                                                                                            |
| <code>Y</code>              | 撤消最后一个文本修改命令。                                                                                                                                                                        |
| <code>u</code>              | 撤消对当前行执行的所有文本修改命令。                                                                                                                                                                   |
| <code>U</code>              | 返回以下命令：                                                                                                                                                                              |
| <code>[count]V</code>       | <pre>hist -e \${VISUAL:-\${EDITOR:-vi}} count</pre> <p>该命令位于输入缓冲区中。如果省略 <i>count</i>，则使用当前行。</p>                                                                                     |
| <code>^L</code>             | 换行并输出当前行。该命令仅在控制模式下使用。                                                                                                                                                               |
| <code>^J</code>             | 新行。无论处于什么模式下，都执行当前行。                                                                                                                                                                 |
| <code>^M</code>             | 返回。无论处于什么模式下，都执行当前行。                                                                                                                                                                 |
| <code>#</code>              | <p>如果该命令的第一个字符是 <code>#</code>，则删除该 <code>#</code> 及换行符之后的每个 <code>#</code>。</p> <p>否则，在该命令的每行前面插入 <code>#</code> 之后发送该行。</p> <p>该命令非常有用，可以将当前行作为注释插入历史记录中，以及取消注释历史文件中以前带有注释的命令。</p> |
| <code>[count]=</code>       | <p>如果未指定 <i>count</i>，则生成本手册页中所述的匹配命令或文件名的列表。</p> <p>否则，将当前光标位置的单词替换为最近生成的命令或文件列表中的 <i>count</i> 项。如果光标未停留在单词上，则会在当前光标位置之后插入该项。</p>                                                  |
| <code>@letter</code>        | 在别名列表中按名称 <i>letter</i> 搜索别名。如果该名称的别名已经定义，则在输入队列中插入其值以进行处理。                                                                                                                          |
| <code>^A</code>             | 显示 shell 的版本。                                                                                                                                                                        |

## 内置命令

以下的简单命令将在 shell 进程中执行。允许输入和输出重定向。除非另有说明，否则输出将写入在文件描述符 1 中，并且退出状态（如果没有语法错误）为 0。除了 `:、true、false、echo、newgrp` 和 `login` 外，其他所有的内置命令都接受 `--` 以指示选项结束。它们还可以将选项 `--man` 解释为在标准错误中显示手册页的请求，以及将 `-?` 解释为在标准错误中输出用法消息的帮助请求。

前面添加了一个或两个 ++ 符号的命令是特殊的内置命令，将通过以下方式进行特殊处理：

1. 命令完成时，该命令前面的变量赋值列表仍然有效。
2. 在变量赋值之后处理 I/O 重定向。
3. 脚本中包含的错误会导致其中止。
4. 它们不是有效的函数名称。
5. 对于跟在前面添加了 ++ 的命令之后且格式为变量赋值格式的单词，将通过与变量赋值相同的规则进行扩展。这意味着将在 = 符号之后执行波浪号替换，并且不执行字段分割和文件名生成。

**+** : [arg ...]  
该命令仅扩展参数。

**+ . name** [arg ...]  
如果 *name* 是一个通过 `function name` 保留字语法定义的函数，则会在当前环境中执行该函数（就好像它是通过 `name()` 语法定义的一样）。否则，如果 *name* 表示一个文件，则会读取整个文件并在当前的 shell 环境中执行命令。通过 `PATH` 指定的搜索路径用于查找包含该文件的目录。如果指定了任何参数 *arg*，则在处理 `.命令` 时这些参数将变为位置参数，并在完成时恢复原始的位置参数。否则，位置参数保持不变。退出状态是最后执行的命令的退出状态。

**++ alias** [-ptx] [*name* [=value]] ...  
不包含参数的 `alias` 可在标准输出中以 `name=value` 的格式输出别名列表。可通过 `-p` 选项在每个名称之前插入单词 `alias`。如果指定一个或多个参数，将为已指定其 *value* 的每个 *name* 定义 `alias`。可通过 *value* 中的结尾空格选中下一个单词进行别名替换。过时的 `-t` 选项用于设置和列出被跟踪的别名。被跟踪别名的值是与指定的 *name* 对应的完整路径名。如果 `PATH` 的值重置但别名仍被跟踪，被跟踪别名的值将变成未定义的值。如果没有 `-t` 选项，则对于参数列表中没有为其指定 *value* 的每个 *name*，都将输出别名的名称和值。过时的 `-x` 选项没有作用。如果指定了 *name*，但没有提供值，也没有为 *name* 定义别名，则退出状态为**非零**。

**bg** [ *job* ... ]  
该命令仅适用于支持作业控制的系统。将指定的每个 *job* 放入后台。如果未指定 *job*，则将当前作业放入后台。请参见本手册页的 `Jobs` 部分了解 *job* 格式的说明。

**+ break** [*n*]  
从封闭的 `for`、`while`、`until` 或 `select` 循环（如果有）中退出。如果指定了 *n*，则分成 *n* 个级别。

**builtin** [-ds] [-f *file*] [*name* ...]  
如果未指定 *name* 和 `-f` 选项，将在标准输出中输出内置名称。`-s` 选项仅输出特殊的内置名称。否则，每个 *name* 都代表其基名为内置名称的路径名。入口点函数名称通过在内置名称前面加上 *b* 确定。内置命令 `mycommand` 的 ISO C/C++ 样例为 `bmycommand(int argc, char *argv[], void *context)`，其中 *argv* 是 *argc* 元素的数组，*context* 是指向如 `<ast/shell.h>` 中所述的 `Shell_t` 结构的可选指针。不能将特殊内置名称绑定到路径名或删除。`-d` 选项可删除所指定的每个内置名称。在支持

动态装载的系统上，`-f` 选项可命名包含内置名称代码的共享库。可以省略共享库的前缀和/或后缀，具体取决于系统。装载某个库之后，其符号将可用于 `builtin` 的后续调用。可通过单独调用 `builtin` 命令指定多个库。按指定库时的反向顺序搜索库。装载某个库之后，该库将在名称为 `lib_init()` 的库中查找函数，并将此函数与参数 `0` 一起调用。

```
cd [-LP] [arg]
```

```
cd [-LP] old new
```

该命令具有两种形式。

在第一种形式中，它将当前目录更改为 `arg`。如果 `arg` 为 `-`，目录将被更改为以前的目录。`shell` 变量 `HOME` 是缺省的 `arg`。变量 `PWD` 被设置为当前目录。`shell` 变量 `CDPATH` 定义包含 `arg` 的目录的搜索路径。备用路径名称使用冒号 (`:`) 分隔。缺省路径是 `NULL`（指定当前目录）。当前目录是通过空路径名指定的，可以直接显示在等号之后或显示在路径列表中任何其他位置的冒号分隔符之间。如果 `arg` 以 `/` 开头，则不使用搜索路径。否则，将搜索路径中每个目录的 `arg`。

`cd` 的第二种形式在当前目录名称 `PWD` 中用字符串 `new` 替换字符串 `old`，并尝试更改为此新目录。缺省情况下，在查找目录名称时会按字面意思处理符号链接名称。这等效于 `-L` 选项。可通过 `-P` 选项在确定目录时解析符号链接。命令行中的 `-L` 或 `-P` 的最后一个实例可确定使用的方法。不能通过 `rksh` 执行 `cd` 命令。

```
command [-pvVx] name [arg...]
```

如果没有 `-v` 或 `-V` 选项，则通过由 `arg` 指定的参数执行 `name`。

可通过 `-p` 选项搜索缺省路径，而不是 `PATH` 的值指定的路径。查找 `name` 时不会搜索函数。此外，如果 `name` 表示特殊内置名称，将忽略与前导剑形符号关联的任何特殊属性。例如，预定义别名 `redirect='command exec'` 可在指定的重定向无效时阻止脚本终止。

在带有 `-x` 选项的情况下，如果由于参数过多会导致命令执行失败（`errno E2BIG`），`shell` 将多次调用命令 `name`，且每次调用时包含部分参数。每次调用时，将会传递在扩展到多个参数的第一个单词之前以及在扩展到多个参数的最后一个单词之后出现的参数。退出状态为最大调用退出状态。

在带有 `-v` 选项的情况下，`command` 等效于本部分中所述的内置 `whence` 命令。`-V` 选项使 `command` 的作用相当于 `whence -v`。

```
+continue [n]
```

继续执行封闭的 `for`、`while`、`until` 或 `select` 循环的下一次重复。如果指定了 `n`，则从第 `n` 次封闭循环继续执行。

```
disown [job...]
```

促使该 `shell` 在登录 `shell` 终止时不将 `HUP` 信号发送至指定的每个 `job`，或不发送至所有活动作业（如果省略 `job`）。

**echo** [*arg...*]

如果第一个 *arg* 不以 - 开头，并且所有的参数均不包含反斜杠 (\)，则会输出其所有参数，并用空格隔开以及换行符终止。否则，**echo** 的行为与系统相关，并且应使用本部分中所述的 **print** 或 **printf**。请参见 **echo(1)** 了解用法和说明。

**+eval** [*arg...*]

将这些参数作为输入读取到 shell 并执行生成的命令。

**+exec** [-c] [-a *name...*] [*arg...*]

如果指定了 *arg*，将执行这些参数指定的命令而不是此 shell，同时不创建新进程。可通过 -c 选项在应用与 **exec** 调用关联的变量赋值之前清除环境。可通过 -a 选项将 *name*（而不是第一个 *arg*）变成新进程的 `argv[0]`。输入参数与输出参数可以显示并影响当前进程。如果未指定 *arg*，则该命令的作用是修改输入/输出重定向列表所指定的文件描述符。在这种情况下，任何通过该机制打开的 2 以上的文件描述符编号将在调用其他程序时关闭。

**+exit** [*n*]

促使 shell 退出，退出状态由 *n* 指定。该值是指定状态的最低有效 8 位。如果省略 *n*，则退出状态为最后执行的命令的退出状态。此外，也可通过文件结尾使 shell 退出，但启用了 **ignoreeof** 选项的 shell 除外。请参见 **set**。

**++export** [-p] [*name[=value]*] ...

如果未指定 *name*，将输出具有导出属性的每个变量的名称和值，并且以能够重新输入的方式将值括起来。可通过 -p 选项在每个名称之前插入单词 **export**。否则，将对指定的 *name* 添加标记以自动导出到后续执行的命令的环境中。

**false**

不执行任何操作，退出 1。与 **until** 结合使用以实现死循环。

**fg** [*job...*]

该命令仅适用于支持作业控制的系统。将指定的每个 *job* 放入前台并按指定顺序等待。否则，将当前作业放入前台。请参见 **Jobs** 了解 *job* 格式的说明。

**getconf** [*name* [*pathname*]]

输出 *name* 指定的配置参数的当前值。配置参数是通过 IEEE POSIX 1003.1 和 IEEE POSIX 1003.2 标准定义的。请参见 **pathconf(2)** 和 **sysconf(3C)**。

其值取决于文件系统中的位置的参数需要 *pathname* 参数。如果未指定参数，**getconf** 将输出当前配置参数的名称和值。对需要 *pathname* 的每个参数使用路径名 */*。

**getopts** [-a *name*] *optstring* *vname* [*arg...*]

检查 *arg* 是否具有合法选项。如果省略了 *arg*，则会使用位置参数。选项参数以 + 或 - 开始。不是以 + 或 - 开始的选项或者参数 -- 会终止选项。当 *optstring* 以 + 开始时，仅识别以 + 开始的选项。*optstring* 包含 **getopts** 可识别的字母。如果字母后面跟有一个 :，则该选项需要有一个参数。选项与参数之间可以用空格隔开。选项 -? 促使 **getopts** 在标准错误输出中生成用法消息。-a 选项可用于指定用法消息使用的名称，缺省情况下为 \$0。每次调用时，**getopts** 都将它找到的下一个选项字母放在变量 *vname* 内。如果 *arg* 以 + 开始，会在选项字母前面加上 +。下一个 *arg* 的索引存

储在 OPTIND 中。选项参数（如果有）存储在 OPTARG 中。*optstring* 中的前导：促使 *getopts* 将无效选项的字母存储在 OPTARG 中，并在缺少所需的选项参数时针对未知选项和 to: 将 *vname* 设置为 ?。其他情况下，*getopts* 将显示一条错误消息。如果没有更多选项，则退出状态是**非零的**。无法指定以下任何选项：:、+、-、?、[ 和 ]。只能将 # 选项指定为第一个选项。

```
hist [-e ename][-nlr] [first[last]]
```

```
hist -s [old=new] [command]
```

在第一种形式中，从 *first* 到 *last* 的命令范围是从在终端键入的最后的 HISTSIZE 命令中选择的。参数 *first* 和 *last* 可以指定为数字或字符串。字符串用于从指定字符串开始查找最近的命令。负数用作到当前命令数的偏移。如果选择 -l 选项，命令将列在标准输出中。否则，在包含这些键盘命令的文件中调用编辑器程序 *ename*。如果未提供 *ename*，将使用变量 HISTEDIT 的值。如果未设置 HISTEDIT，将 FCEDIT（缺省为 /bin/ed）用作编辑器。编辑完成后，如果更改已经保存，将执行编辑后的命令。如果未指定 *last*，将它设置为 *first*。如果未指定 *first*，针对编辑和列举的缺省值分别为上一条命令和 -16。选项 -r 颠倒命令顺序，选项 -n 在列举时隐藏命令数。在第二种形式中，将 *command* 解释为本部分中所述的 *first*，缺省为最后执行的命令。在执行可选替代 *old=new* 之后执行生成的命令。

```
jobs -lnp [job...]
```

列出有关每个指定作业或（如果省略 *job*）所有活动作业的信息。除普通信息外，-l 还列出进程 ID。-n 选项仅显示自上次通知起已停止或退出的作业。可通过 -p 选项只列出进程组。请参见 Jobs 了解 *job* 格式的说明。

```
kill [-s signame] job...
```

```
kill [-n signal] job...
```

```
kill -l [sig...]
```

向指定作业或进程发送 TERM（终端）信号或指定信号。通过具有 -n 选项的数字或具有 -s 选项的名称指定信号（如 <signal.h> 中指定，除去前缀 'SIG，除非将 SIGCLD 命名为 CHLD）。为实现向后兼容，可以省略 n 和 s，将数字或名称紧跟 - 之后。如果已发送的信号是 TERM（终端）或 HUP（挂断），在作业或进程停止时将其作为 CONT（继续）信号发送。参数 *job* 可以是不属于某个活动作业的进程的进程 ID。请参见 Jobs 了解 *job* 格式的说明。在第三种形式 kill -l 中，如果未指定 *sig*，将列出信号名称。否则，对于作为名称的每个 *sig*，将列出相应的信号编号。对于作为编号的每个 *sig*，将列出与 *sig* 的最低有效 8 位对应的信号名称。

```
let [arg...]
```

每个 *arg* 都是一个可求值的独立算术表达式。请参见本手册页的 Arithmetic Evaluation 部分，了解算术表达式求值的说明。如果最后一个表达式的值是**非零**，则退出状态为 0，否则为 1。

```
+newgrp [arg...]
```

等效于 exec /bin/newgrp *arg*...

```
print [-Renprs] [-u unit] [-f format] [arg...]
```

如果没有选项或者具有选项 - 或 --，将在标准输出中输出每个 *arg*。可通过 -f 选项按 printf 所述输出参数。在这种情况下，忽略任何 e、n、r 或 R 选项。除非指定 -R 或 -r，否则应用以下转义约定：



\a 警报字符 (ASCII 07)  
 \b 退格字符 (ASCII 010)  
 \c 使输出终止，而不处理更多参数，且不添加换行符  
 \f 换页字符 (ASCII 014)  
 \n 换行符 (ASCII 012)  
 \r RETURN 字符 (ASCII 015)  
 \t TAB 字符 (ASCII 011)  
 \v 垂直 TAB 字符 (ASCII 013)  
 \E 转义符 (ASCII 033)  
 \\ 反斜杠字符 \  
 \0x 通过 *x* 指定的 1、2 或 3 位八进制字符串定义的字符

-R 选项输出除 -n 之外的所有后续参数和选项。可通过 -e 应用转义约定，这是缺省行为。它颠倒原先 -r 的效果。可通过 -p 选项将参数写入到通过 |& 产生的进程管道中，而不是标准输出中。可通过 -s 选项将参数写入到历史文件中而不是标准输出中。-u 选项可用于指定输出所在的一位文件描述符单元编号 *unit*。缺省值为 1。如果使用 -n 选项，则不会将换行符添加到输出中。

#### printf *format*[*arg*...]

根据与格式字符串 *format* 关联的 ANSI-C 格式化规则，在标准输出中输出参数 *arg*。如果参数的数目超过格式指定的数目，将重用该格式字符串格式化剩余的参数。还可以使用以下表达式：可以用 %b 格式代替 %s 以按照 print 中的说明扩展对应 *arg* 中的转义序列。可通过 %B 选项将每个参数作为变量名称处理并输出变量的二进制值。这对于带有属性 b 的变量非常有用。可以使用 %H 格式代替 %s，以便将 *arg* 中存在的对于 HTML 和 XML 而言为特殊字符的字符输出为其实体名称。可以用 %P 格式代替 %s，以便将 *arg* 解释为扩展的正则表达式并作为 shell 模式输出。可以用 %R 格式代替 %s，以便将 *arg* 解释为 shell 模式并作为扩展的正则表达式输出。可以用 %q 格式代替 %，以便通过可重新输入到 shell 的方式引用生成的字符串。%(*date-format*)T 格式可用于将参数作为日期/时间字符串来处理并根据为 `date(1)` 命令定义的 *date-format* 格式化日期/时间。%Z 格式可输出其值为 0 的字节。%d 格式的精度字段可以后跟 a. 和输出基数。在这种情况下，可通过 # 标志字符使 base# 前置。# 标志与 d 说明符一起使用但没有输出基数时，使输出以千分位显示，后缀 kMGTPe 中的其中一个表示单位。# 标志与 i 说明符一起使用时，使输出以 1024 显示，后缀 Ki Mi Gi Ti Pi Ei 中的其中一个表示单位。已添加 = 标志以使输出在指定的字段宽度内居中。

#### pwd [-LP]

输出当前工作目录的值。-L 选项为缺省选项，用于输出当前目录的逻辑名称。如果指定了 -P 选项，将基于该名称解析所有的符号链接。命令行中的 -L 或 -P 的最后一个实例可确定使用的方法。

```
read [-Aprs] [-d delim] [-n n] [[-N n] [[-t timeout] [-u unit] [vname?prompt] [vname...]
```

shell 输入机制。读取一行并使用 IFS 中的字符作为分隔符将该行分为多个字段。转义符 \ 用于去除下一个字符和续行符的任何特殊含义。可通过 -d 选项将读取延续到 *delim*（而不是换行符）的第一个字符。可通过 -n 选项最多只读取 *n* 字节而不是整行，但在从速度较慢的设备中读取时，只要已读取任何字符即将其返回。可通过 -N 选项恰好读取 *n* 字节，除非到达文件结尾或因设置了 -t 选项而导致读取超时。在原始模式下 (-r)，不特殊处理 \ 字符。第一个字段分配给第一个 *vname*，第二个字段分配给第二个 *vname*，以此类推，剩余的字段分配给最后一个 *vname*。如果 *vname* 具有二进制属性并且已指定 -n 或 -N，读取的字节将直接存储在变量中。如果已指定 -v，将第一个 *vname* 的值用作从终端设备读取时的缺省值。可通过 -A 选项取消设置变量 *vname*，并使读取的每个字段存储在索引数组 *vname* 的连续元素中。通过 -p 选项，使用 |& 从 shell 产生的进程输入管道中获取输入行。如果存在 -s 选项，将输入另存为历史文件中的命令。选项 -u 可用于指定从中读取的一位文件描述符单元 *unit*。可以通过特殊的内置命令 `exec` 打开文件描述符。单元 *n* 的缺省值是 0。选项 -t 用于指定从终端或管道读取时以秒为单位的超时。如果省略 *vname*，则 REPLY 将用作缺省的 *vname*。具有 -p 选项的文件结尾可清除该进程，这样可以生成其他进程。如果第一个参数包含 ?，则当 shell 为交互式 shell 时，该单词的剩余部分将用作标准错误输出中的提示。除非到达文件结尾或读取超时，否则退出状态为 0。

```
++readonly [-p] [vname[=value]] ...
```

如果未指定 *vname*，将输出具有只读属性的每个变量的名称和值，并且以能够重新输入的方式将值括起来。通过 -p 选项将单词 `readonly` 插入在每个变量之前。否则，将对指定的 *vname* 标记 `readonly`，且不能通过后续赋值更改这些名称。

```
+return [n]
```

使 shell 函数或脚本返回到具有 *n* 指定的退出状态的调用脚本。该值是指定状态的最低有效 8 位。如果省略 *n*，则返回状态为最后执行的命令的返回状态。如果调用的返回不在函数或脚本中，则与退出的行为相同。

```
+set [±BCGabefhkmnoprstuvx] [±o [option]] ... [±A vname] [arg...]
```

set 命令支持以下选项：

-a

自动导出已定义的所有后续变量。

-A

数组赋值。取消设置变量 *vname*，并按顺序从 *arg* 列表中赋值。如果使用 +A，则首先不取消设置变量 *vname*。

-b

一旦后台作业更改状态，即输出作业完成消息，而不是等待下一提示。

-B

启用大括号模式字段生成。这是缺省行为。



- C  
阻止重定向 (>) 截断现有文件。通过 `O_EXCL` 模式打开已创建的文件。要求 `>|` 在启用时截断文件。
- e  
如果命令具有**非零**退出状态，则执行 `ERR` 捕获（如果已设置）并退出。该模式在读取配置文件时被禁用。
- f  
禁用文件名生成。
- G  
在用于文件名生成时，使模式 `**` 本身与文件及零个或多个目录和子目录匹配。如果后跟 `/`，则仅匹配目录和子目录。
- h  
每个命令在第一次遇到时将成为被跟踪别名。
- k  
已过时。将所有的变量赋值参数放在命令的环境中，而不仅仅是命令名称前面的参数。
- m  
后台作业在单独的进程组中运行，并在完成时输出一行。在完成消息中报告后台作业的退出状态。在具有作业控制的系统上，为交互式 `shell` 自动启用该选项。
- n  
读取命令并检查其中是否存在语法错误，但不执行这些命令。对于交互式 `shell`，忽略此命令。
- o  
如果未提供选项名称，则会将选项及其当前设置的列表写入到标准输出。当与 `+` 一起调用此命令时，这些选项将以可重新输入到 `shell` 的格式写入，以便恢复相应设置。重复此选项可以启用或禁用多个选项。  
  
下列参数可以是下列选项名称之一：
  - `allexport`  
与 `-a` 相同。
  - `bgnice`  
以较低的优先级运行所有后台作业。这是缺省模式。
  - `braceexpand`  
与 `-B` 相同。
  - `emacs`  
使您进入 `emacs` 样式的内嵌编辑器以便输入命令。

`errexit`

与 `-e` 相同。

`globstar`

与 `-G` 相同。

`gmacs`

使您进入 `gmacs` 样式的内嵌编辑器以便输入命令。

`ignoreeof`

此 shell 不会在文件结尾退出。必须使用 `exit` 命令。

`keyword`

与 `-k` 相同。

`markdirs`

文件名生成操作生成的所有目录名称都附加有尾随 `/`。

`monitor`

与 `-m` 相同。

`multiline`

对于长于屏幕宽度的行，内置编辑器在屏幕上使用多个行。这可能不适用于所有终端。

`noclobber`

与 `-c` 相同。

`noexec`

与 `-n` 相同。

`noglob`

与 `-f` 相同。

`nolog`

不会在历史文件中保存函数定义。

`notify`

与 `-b` 相同。

`nounset`

与 `-u` 相同。

`pipefail`

管道将在所有管道组件完成后完成，并且返回值为失败的最后一个**非零**命令的值，如果没有任何命令失败，则为零。

`privileged`

与 `-p` 相同。

`showme`

启用此命令后，将显示前面带有分号 (`;`) 的简单命令或管道，就像启用了 `xtrace` 选项但未执行此选项一样。否则，将忽略前导 `;`。

`trackall`

与 `-h` 相同。

`verbose`

与 `-v` 相同。

`vi`

使您进入 `vi` 样式的内嵌编辑器的插入模式，直到您键入转义符 `033` 为止。这会让您进入控制模式。回车可发送此行。

`viraw`

各个字符在 `vi` 模式下键入时处理。

`xtrace`

与 `-x` 相同。

如果未提供选项名称，则输出当前选项设置。

`-p`

禁用对 `$HOME/.profile` 文件的处理，并使用文件 `/etc/suid_profile`（而非 `ENV` 文件）。当有效 `uid (gid)` 不等于实际 `uid (gid)` 时，将启用此模式。禁用此模式会导致将有效 `uid` 和 `gid` 设置为实际 `uid` 和 `gid`。

`-r`

启用受限 `shell`。此选项一旦设置便不能取消设置。

`-s`

按字典顺序对位置参数进行排序。

`-t`

已过时。读取和执行一个命令后退出。

`-u`

替换时将 `unset` 参数视为错误。

`-v`

在读取 `shell` 输入行时输出这些行。

`-x`

在执行命令及其参数时输出这些命令及其参数。

`--`

不更改任何选项。在将 `$1` 设置为以 `-` 开头的值时，此命令非常有用。如果此选项后没有任何参数，则取消设置位置参数。

作为一项已过时的功能，如果第一个 `arg` 为 `-`，则会禁用 `-x` 和 `-v` 选项，并将下一个 `arg` 视为第一个参数。使用 `+`（而非 `-`）将导致禁用这些选项。此外，还可以在调用 `shell` 时使用这些选项。当前选项集可在 `$-` 中找到。除非指定了 `-A`，否则剩余的参数都是位置参数，并按照顺序分配为 `$1 $2 ...`。如果未指定参数，则会在标准输出中输出所有变量的名称和值。

**+shift [n]**

$\$n+1 \dots$  中的位置参数重命名为  $\$1 \dots$ ，缺省的  $n$  为 1。参数  $n$  可以为任何算术表达式，此算术表达式的计算结果为小于或等于  $\$#$  的非负数字。

**+trap -p [action] [sig] ...**

**-p** 选项导致输出与参数指定的每个陷阱相关联的陷阱操作，并进行相应引用。否则，当 shell 接收信号 *sig* 时，则会处理 *action*，就像它是 `eval` 的参数一样。可以以编号或信号名称的形式指定每个 *sig*。陷阱命令按照信号编号的顺序执行。尝试对输入到当前 shell 时忽略的信号设置陷阱将不起作用。如果省略了 *action*，并且第一个 *sig* 为编号，或者如果 *action* 为 `-`，则会将每个 *sig* 的陷阱重置为其原始值。如果 *action* 为空字符串，shell 及其调用的命令将忽略此信号。如果 *sig* 为 `ERR`，当命令的退出状态为**非零**时，将执行 *action*。如果 *sig* 为 `DEBUG`，则会在每个命令之前执行 *action*。当运行 *action* 时，变量 `.sh.command` 包含当前命令行的内容。如果 *sig* 为 `0` 或 `EXIT`，并且在使用 `function name` 语法定义的函数主体内执行陷阱语句，则会在此函数完成后执行命令 *action*。对于在任何函数外部设置的陷阱，如果 *sig* 为 `0` 或 `EXIT`，则会在退出时从 shell 执行命令 *action*。如果 *sig* 为 `KEYBD`，则会在 `emacs`、`gmacs` 或 `vi` 模式下读取键时执行 *action*。不带参数的 `trap` 命令输出与各信号编号关联的命令的列表。

**true**

不执行任何操作，退出 `0`。与 `while` 一起用于无限循环。

**++typeset [±AHflabnprtux ] [ ±EFLRZi[n] ] [ vname[=value ] ]**

设置 shell 变量和函数的属性及值。当在使用 `function name` 语法定义的函数内部调用时，将创建一个新的 *vname* 变量实例，并在此函数完成时恢复变量的值和类型。

使用 `+`（而非 `-`）将导致禁用这些选项。如果未指定 *vname* 参数，则输出变量的 *vname*（以及可选 *value*）的列表。使用 `+`（而非 `-`）可防止输出值。**-p** 选项导致在每个名称（而非选项名称）之前输出后跟有选项字母的 `typeset`。如果指定了除 **-p** 以外的任何选项，则仅输出具有所有指定选项的变量。否则输出具有属性的所有变量的 *vname* 和 *attributes*。

可以指定下列属性列表：

**-a** 将 *vname* 声明为索引数组。这是可选属性（除非用于复合变量赋值）。

**-A** 将 *vname* 声明为关联数组。下标是字符串，而不是算术表达式。

**-b** 此变量可保存任意字节数目的数据。数据可以是文本或二进制数据。此变量的值通过数据的 `base64` 编码表示。如果还指定了 **-z**，缓冲区中的数据大小（以字节为单位）则由与 **-z** 关联的大小确定。如果分配的 `base64` 字符串导致生成更多数据，则会截断此数据。否则，则会使用值为零的字节填充此数据。`printf` 格式 `%B` 可用于输出此缓冲区中的实际数据，而非数据的 `base64` 编码。

**-E** 将 *vname* 声明为双精度浮点数。如果  $n$  为**非零**，则定义扩展 *vname* 时使用的有效数字的数目。否则使用 10 个有效数字。

- f 这些名称引用函数名称，而非变量名称。不可赋值，且其他有效选项只有 `-t`、`-u` 和 `-x`。`-t` 选项开启对此函数的执行跟踪。`-u` 选项导致将此函数标记为未定义。搜索 `FPATH` 变量，以便在引用函数时查找函数定义。如果指定了除 `-f` 以外的选项，则会在标准输出中显示函数定义。如果指定了 `+f`，则会显示一个行，此行包含后跟有一条 `shell` 注释的函数名称，此注释包含行号以及定义此函数所在的文件的路径名（如果有）。
  - i 属性无法与 `-f` 一起指定。
- F 将 `vname` 声明为双精度浮点数。如果 `n` 为**非零**，则定义扩展 `vname` 时使用的小数点后面的小数位数。否则使用小数点后面的 10 个小数位数。
- H 此选项提供非 UNIX 计算机上的 UNIX 到主机名文件的映射。
- i 将要在内部表示的 `vname` 声明为整数。当指定为整数时，赋值右侧将作为算术表达式计算。如果 `n` 为**非零**，则会定义输出算术基数，否则输出基数为 10。
  - i 属性不能与 `-R`、`-L`、`-Z` 或 `-f` 一起指定。
- l 所有大写字符都将转换为小写字符。禁用大写字母选项 `-u`。
- L 向左调整 `value`，并删除其中的前导空格。如果 `n` 为**非零**，则定义字段宽度，否则根据第一个赋值的值的宽度确定字段宽度。如果为此变量赋值，则会根据需要使用空格填充此值的右侧或者截断此值，使其适合此字段。禁用 `-R` 选项。
  - i 属性无法与 `-L` 一起指定。
- n 将 `vname` 声明为对某个变量的引用，此变量的名称根据变量 `vname` 的值定义。这通常用于引用函数中其名称作为参数传递的变量。
- R 向右调整，并使用前导空格填充。如果 `n` 为**非零**，则定义字段宽度，否则根据第一个赋值的值的宽度确定字段宽度。如果重新为此变量赋值，则会使用空格填充此字段的左侧，或者截断此字段的末尾。禁用 `-L` 选项。
  - i 属性无法与 `-R` 一起指定。
- r 指定 `vname` 标记为只读，后续赋值不能更改这些名称。
- t 标记变量。用户可以定义标记，对 `shell` 而言，这些标记没有任何特殊意义。
- u 所有小写字符都将转换为大写字符。禁用小写字母选项 `-l`。
- x 标记指定 `vname`，以便自动导出到随后执行的命令的环境。无法导出其名称包含 `.` 的变量。
- Z 如果第一个非空字符为数字，并且未设置 `-L` 选项，则会向右调整，并使用前导零进行填充。如果同时设置了 `-L` 选项，则删除前导零。如果 `n` 为**非零**，则定义字段宽度，否则根据第一个赋值的值的宽度确定字段宽度。

-i 属性无法与 -z 一起指定。

**ulimit [-HSacdfmnpstv] [ *limit*]**

设置或显示资源限制。许多系统都不支持一个或多个这些限制。如果指定了 *limit*，则会设置指定资源的限制。*limit* 的值可以是一个数字（采用随各资源指定的单位为单位），也可以为值 **unlimited**。如果指定了多种资源，则会在值之前输出限制名称及单位。

如果未指定任何选项，则采用 -f。

下面列出了一些可用的资源限制：

- a 列出当前的所有资源限制。
- c 核心转储大小中的 512 字节块的数目。
- d 数据区域大小中的千字节数目。
- f 当前进程或子进程可写入的文件中的 512 字节块的数目（可以读取任意大小的文件）。
- H 指定指定资源的硬限制。

硬限制一旦设置便不能增加。

如果未指定 -H 和 -S 选项，此限制同时适用于两个选项。如果省略了 *limit*，则会输出当前资源限制。此种情况下，除非指定 -H，否则将输出软限制。

- m 物理内存大小中的千字节数目。
- n 文件描述符数目加 1。
- p 用于管道缓冲的 512 字节块的数目。
- s 栈区域大小中的千字节数目。
- S 指定指定资源的软限制。

软限制最多可增加至硬限制的值。

如果未指定 -H 和 -S 选项，此限制同时适用于两个选项。如果省略了 *limit*，则会输出当前资源限制。此种情况下，除非指定 -H，否则将输出软限制。

- t 各进程使用的 CPU 秒数。
- v 用于虚拟内存的千字节数目。

**umask [-S] [*mask*]**

用户文件创建掩码设置为 *mask*。*mask* 可以是八进制数字或符号值，如 **chmod(1)** 所述。

如果指定了符号值，新 `umask` 值即为对以前的 `umask` 值的补数应用 `mask` 的结果的补数。如果省略了 `mask`，则会输出掩码的当前值。`-s` 选项导致将模式输出为符号值。否则输出八进制掩码。

请参见 [umask\(2\)](#)

`+unalias [-a] name`

从别名列表中删除 `name` 列表指定的别名。`-a` 选项导致取消设置所有别名。

`+unset [-fnv] vname`

未分配 `vname` 列表指定的变量，即，已清除这些变量的值和属性。无法取消设置只读变量。如果设置了 `-f` 选项，名称则引用函数名称。如果设置了 `-v` 选项，名称则引用变量名称。`-f` 选项可覆盖 `-v`。如果设置了 `-n`，并且 `name` 为名称引用，则会取消设置 `name`（而非其引用的变量）。缺省值等效于 `-v`。取消设置 `LINENO`、`MAILCHECK`、`OPTARG`、`OPTIND`、`RANDOM`、`SECONDS`、`TMOUT` 和 `_` 将删除其特殊含义，即使在随后对这些变量赋值也是如此。

`wait [job]`

等待指定作业，并报告其终止状态。如果未指定 `job`，则等待当前处于活动状态的所有子进程。如果指定了 `job`，此命令的退出状态即为等待的最后一个进程的退出状态；否则为零。请参见 [Jobs](#) 了解 `job` 格式的说明。

`whence [-afpv] name ...`

对于每个 `name`，此命令指示将其用作命令名称时的解释方式。`-v` 选项生成更详细的报告。`-f` 选项跳过函数搜索。即使名称为别名、函数或保留字，`-p` 选项也会针对 `name` 进行路径搜索。`-a` 选项与 `-v` 选项相似，但前者导致报告指定名称的所有解释。

## 调用

如果使用 [exec\(2\)](#) 调用 shell，并且参数的第一个字符 `0` (`$0`) 为 `.`，则假定此 shell 为登录 shell，并从 `/etc/profile` 读取命令，然后从当前目录的 `.profile` 或 `$HOME/.profile` 中读取命令（如果存在任何一个文件）。接着，对于交互式 shell，首先从 `/etc/ksh.kshrc` 读取命令，然后从通过对环境变量 `ENV` 的值执行参数扩展、命令替换和算术替换进行命名的文件中读取命令（如果存在此文件）。如果 `-s` 选项不存在，并且存在 `arg` 以及名为 `arg` 的文件，则会读取和执行此脚本。否则，如果第一个 `arg` 不包含 `/`，则会对第一个 `arg` 执行路径搜索，以便确定要执行的脚本的名称。脚本 `arg` 必须具有执行权限，并且忽略任何 `setuid` 和 `setgid` 设置。如果未在此路径中找到脚本，则会处理 `arg`，就像指定了内置命令或函数一样。

然后按照说明读取命令，并且 shell 会在调用时解释下列选项：

- `-c`            如果存在 `-c` 选项，则从第一个 `arg` 读取命令。其余所有参数都将成为从 `0` 开始的位置参数。
- `-D`            标准输出中将输出前面带有 `$` 且包含在双引号中的所有字符串的列表，并且 shell 退出。当语言环境不是 C 或 POSIX 时，此组字符串则会转换为相应语言。不会执行任何命令。
- `-i`            如果存在 `-i` 选项，或者如果将 shell 输入和输出附加到终端（根据 [tcgetattr\(3C\)](#) 的指示），此 shell 则为交互式 shell。此种情况下，将忽

略 TERM（因此，kill 0 不会中止交互式 shell），并捕获和忽略 INTR（因此，可以中断等待）。在所有情况下，shell 将忽略 QUIT。

- R *filename*     -R *filename* 选项用于生成交叉引用数据库，此数据库可供单独的实用程序用于查找变量和命令的定义及引用。
- r                如果存在 -r 选项，shell 则为受限 shell。
- s                如果存在 -s 选项，或者如果未保留任何参数，则从标准输入读取命令。Shell 输出（列出的**特殊命令**的输出除外）写入到文件描述符 2。

其余选项和参数在 set 命令下说明。忽略作为第一个参数的可选 -。

仅适用于 rksh

rksh 用于设置登录名和执行环境，其功能比标准 shell 的功能更受约束。

rksh 的操作与 ksh 的操作相同，但前者不允许以下操作：

- 取消设置受限选项
- 更改目录。请参见 [cd\(1\)](#)。
- 设置或取消设置 SHELL、ENV、FPATH 或 PATH 的值或属性
- 指定包含 / 的路径或命令名称。
- 重定向输出（>、>|、<> 和 >>）。
- 添加或删除内置命令。
- 使用 command -p 调用命令。

在解释 .profile 和 ENV 文件后强制这些限制。

如果发现要执行的命令为 shell 过程，rksh 则会调用 ksh 来执行此命令。因此，可以向最终用户提供有权访问标准 shell 的完整功能的 shell 过程，并强制有限的命令菜单。此方案假定最终用户没有同一目录的写入和执行权限。这些规则的实际结果是，通过执行保证的设置操作，并将用户保留在适当的目录（可能不是登录目录）中，.profile 的写入者具有用户操作的完整控制权。系统管理员通常设置可由 rksh 安全调用的命令目录（例如，/usr/rbin）。

用法

有关在遇到大小大于或等于 2 GB（2<sup>31</sup> 字节）的文件时 ksh 和 rksh 的行为的说明，请参见 [largefile\(5\)](#)。

退出状态

将返回以下退出值：

#### 非零值

当 shell 检测到错误（如语法错误）时，返回**非零**。

如果以非交互方式使用 shell，除非子 shell 中出现错误（此种情况下将放弃子 shell），否则会放弃执行此 shell 文件。

#### 执行的最后一个命令的退出状态

返回执行的最后一个命令的退出状态。



通过输出命令或函数名称以及错误状态，报告 shell 检测到的运行时错误。如果出现错误的行号大于 1，则还会在命令或函数名称后面的方括号 ([]) 中输出此行号。

有关其他详细信息，请参见 `ksh exit` 命令。

## 文件

`/etc/profile`

系统初始化文件，针对 `login shell` 执行。

`/etc/ksh.kshrc`

系统级启动文件，针对交互式 shell 执行。

`$HOME/.profile`

专用初始化文件，在 `/etc/profile` 后针对登录 shell 执行。

`$HOME/.kshrc`

缺省专用初始化文件，当未设置 `ENV` 时，在 `/etc/ksh.kshrc` 后针对交互式 shell 执行。

`/etc/suid-profile`

备用初始化文件，在实际和有效用户或组 ID 不匹配时代替专用初始化文件执行。

`/dev/null`

空设备。

## 作者

David Korn, `dgk@research.att.com`

## 属性

有关下列属性的说明，请参见 `attributes(5)`：

| 属性类型  | 属性值       |
|-------|-----------|
| 可用性   | shell/ksh |
| 接口稳定性 | 请参见下文。    |

脚本接口是 "Uncommitted"（未确定）。环境变量、`.paths` 功能以及编辑模式是 `Volatile`（可变）接口。

## 另请参见

`cat(1)`、`cd(1)`、`chmod(1)`、`cut(1)`、`date(1)`、`egrep(1)`、`echo(1)`、`egrep(1)`、`env(1)`、`fgrep(1)`、

由 Bolsky, Morris I. 和 Korn, David G. 合著的《The New KornShell Command and Programming Language》，Prentice Hall 出版，1995。

《POSIX-Part 2: Shell and Utilities, IEEE Std 1003.2-1992, ISO/IEC 9945-2》，IEEE 出版，1993 年。

## 附注

`ksh` 脚本应选择 ISO C99、C++ 和 JAVA 语言的保留字所使用的名称空间之外的 shell 函数名称，以免与 `ksh` 的将来改进冲突。

如果执行命令，并在搜索路径中原始命令所在目录之前的目录中安装同名命令，shell 会继续执行原始命令。使用 `alias` 命令的 `-t` 选项可以更正此情况。

某些很早的 shell 脚本包含插入记号 (^)，并将其用作管道字符 (|) 的同义词。

在复合命令中使用 `hist` 内置命令会导致 `whole` 命令从历史文件中消失。

内置命令 `.file` 在执行任何命令之前读取整个文件。文件中的 `alias` 和 `unalias` 命令不适用于该文件中定义的所有命令。

当作业正在等待前台进程时，不会处理陷阱。因此，在前台作业终止之前，不会执行 `CHLD` 中的陷阱。

最好在算术表达式中的逗号运算符后保留一个空格，以免在特定语言环境中将逗号解释为小数点字符。

创建 `.paths` 文件可能存在一些限制，该文件可移植到其他操作系统中。

如果系统支持 64 位指令集，`/bin/ksh` 将执行 ksh 的 64 位版本。

**引用名** ksh88, rksh88 – KornShell, a standard/restricted command and programming language

**用法概要**

```

/usr/sunos/bin/ksh [± abCefhikmnoprstuvx] [± o option]...
 [arg]...

/usr/sunos/bin/ksh -c [± abCefhikmnoprstuvx]
 [± o option]... command_string
 [command_name [arg...]]

/usr/xpg4/bin/sh [± abCefhikmnoprstuvx]
 [± o option]... [arg]...

/usr/xpg4/bin/sh -c [± abCefhikmnoprstuvx]
 [± o option]... command_string
 [command_name [arg...]]

/usr/sunos/bin/rksh [± abCefhikmnoprstuvx] [± o option]...
 [arg]...

/usr/sunos/bin/rksh -c [± abCefhikmnoprstuvx]
 [± o option]... command_string
 [command_name [arg...]]

```

**描述**

The `/usr/xpg4/bin/sh` utility is a standards compliant shell. This utility provides all the functionality of `/usr/sunos/bin/ksh`, except in cases where differences in behavior exist. See Arithmetic Expansions section for details.

`/usr/sunos/bin/ksh` is a command and programming language that executes commands read from a terminal or a file. `rksh` is a restricted version of the command interpreter `ksh`; it is used to set up login names and execution environments whose capabilities are more controlled than those of the standard shell. See the Invocation section for the meaning of arguments to the shell.

**Definitions**

A *metacharacter* is one of the following characters:

```
; & () | < > NEWLINE SPACE TAB
```

A *blank* is a TAB or a SPACE. An *identifier* is a sequence of letters, digits, or underscores starting with a letter or underscore. Identifiers are used as names for *functions* and *variables*. A *word* is a sequence of *characters* separated by one or more non-quoted *metacharacters*.

A *command* is a sequence of characters in the syntax of the shell language. The shell reads each command and carries out the desired action either directly or by invoking separate utilities. A *special-command* is a command that is carried out by the shell without creating a separate process. Except for documented side effects, most special commands can be implemented as separate utilities.

**Commands**

A *simple-command* is a sequence of blank-separated words which can be preceded by a variable assignment list. See Environment. The first word specifies the name of the command to be executed. Except as specified, the remaining words are passed as arguments to the

invoked command. The command name is passed as argument 0 (see `exec(2)`). The *value* of a simple-command is its exit status if it terminates normally. If it terminates abnormally due to receipt of a signal, the *value* is the signal number plus 128. See `signal.h(3HEAD)` for a list of signal values. Obviously, normal exit status values 129 to 255 cannot be distinguished from abnormal exit caused by receiving signal numbers 1 to 127.

A *pipeline* is a sequence of one or more *commands* separated by `|`. The standard output of each command but the last is connected by a `pipe(2)` to the standard input of the next command. Each command is run as a separate process; the shell waits for the last command to terminate. The exit status of a pipeline is the exit status of the last command.

A *list* is a sequence of one or more *pipelines* separated by `;`, `&`, `&&`, or `|`, and optionally terminated by `;`, `&`, or `|&`. Of these five symbols, `;`, `&`, and `|&` have equal precedence, which is lower than that of `&&` and `|`. The symbols `&&` and `|` also have equal precedence. A semicolon (`;`) causes sequential execution of the preceding pipeline; an ampersand (`&`) causes asynchronous execution of the preceding pipeline (that is, the shell does *not* wait for that pipeline to finish). The symbol `|&` causes asynchronous execution of the preceding command or pipeline with a two-way pipe established to the parent shell.

The standard input and output of the spawned command can be written to and read from by the parent shell using the `-p` option of the special commands `read` and `print` described in Special Commands. The symbol `&&( | )` causes the *list* following it to be executed only if the preceding pipeline returns 0 (or a non-zero) value. An arbitrary number of new-lines can appear in a *list*, instead of a semicolon, to delimit a command.

A *command* is either a *simple-command* or one of the following. Unless otherwise stated, the value returned by a command is that of the last simple-command executed in the command.

`for identifier [ in word ... ] ; do list ; done`

Each time a `for` command is executed, *identifier* is set to the next *word* taken from the *in word list*. If *in word ...* is omitted, then the `for` command executes the *do list* once for each positional parameter that is set. See Parameter Substitution. Execution ends when there are no more words in the list.

`select identifier [ in word ... ] ; do list ; done`

A `select` command prints to standard error (file descriptor 2), the set of *words*, each preceded by a number. If *in word ...* is omitted, then the positional parameters are used instead. See Parameter Substitution. The PS3 prompt is printed and a line is read from the standard input. If this line consists of the number of one of the listed *words*, then the value of the variable *identifier* is set to the *word* corresponding to this number. If this line is empty the selection list is printed again. Otherwise the value of the variable *identifier* is set to NULL. (See Blank Interpretation about NULL). The contents of the line read from standard input is saved in the shell variable `REPLY`. The *list* is executed for each selection until a break or EOF is encountered. If the `REPLY` variable is set to NULL by the execution of *list*, then the selection list is printed before displaying the PS3 prompt for the next selection.

`case word in [ pattern [ | pattern ] ) list ; ; ] ... esac`

A `case` command executes the *list* associated with the first *pattern* that matches *word*. The form of the patterns is the same as that used for file-name generation. See File Name Generation.

`if list ; then list ; [ elif list ; then list ; ... ] [ else list ; ] fi`

The *list* following `if` is executed and, if it returns an exit status of 0, the *list* following the first `then` is executed. Otherwise, the *list* following `elif` is executed and, if its value is 0, the *list* following the next `then` is executed. Failing that, the `else list` is executed. If no `else list` or `then list` is executed, then the `if` command returns 0 exit status.

`while list ; do list ; done`

`until list ; do list ; done`

A `while` command repeatedly executes the `while list` and, if the exit status of the last command in the `do list` is 0, executes the `do list`; otherwise the loop terminates. If no commands in the `do list` are executed, then the `while` command returns 0 exit status. `until` can be used in place of `while` to negate the loop termination test.

`(list)`

Execute *list* in a separate environment. If two adjacent open parentheses are needed for nesting, a space must be inserted to avoid arithmetic evaluation.

`{list}`

*list* is simply executed. Unlike the metacharacters ( and ), { and } are *reserved words* and must occur at the beginning of a line or after a ; in order to be recognized.

`[[expression]]`

Evaluates *expression* and returns 0 exit status when *expression* is true. See Conditional Expressions for a description of *expression*.

`function identifier { list ; }`

`identifier( ) { list ; }`

Define a function which is referenced by *identifier*. The body of the function is the *list* of commands between { and }. See Functions.

`time pipeline`

The *pipeline* is executed and the elapsed time as well as the user and system time are printed to standard error.

The following reserved words are only recognized as the first word of a command and when not quoted:

|          |        |       |       |      |      |      |
|----------|--------|-------|-------|------|------|------|
| !        | if     | then  | else  | elif | fi   | case |
| esac     | for    | while | until | do   | done | { }  |
| function | select | time  | [[ ]] |      |      |      |

Comments

A word beginning with # causes that word and all the following characters up to a new-line to be ignored.

## Aliasing

The first word of each command is replaced by the text of an alias if an alias for this word has been defined. An alias name consists of any number of characters excluding metacharacters, quoting characters, file expansion characters, parameter and command substitution characters, and =. The replacement string can contain any valid shell script including the metacharacters listed above. The first word of each command in the replaced text, other than any that are in the process of being replaced, is tested for aliases. If the last character of the alias value is a *blank* then the word following the alias is also be checked for alias substitution. Aliases can be used to redefine special builtin commands but cannot be used to redefine the reserved words listed above. Aliases can be created, listed, and exported with the `alias` command and can be removed with the `unalias` command. Exported aliases remain in effect for scripts invoked by name, but must be reinitialized for separate invocations of the shell. See *Invocation*. To prevent infinite loops in recursive aliasing, if the shell is not currently processing an alias of the same name, the word is replaced by the value of the alias; otherwise, it is not be replaced.

Aliasing is performed when scripts are read, not while they are executed. Therefore, for an alias to take effect, the `alias` definition command has to be executed before the command which references the alias is read.

Aliases are frequently used as a short hand for full path names. An option to the aliasing facility allows the value of the alias to be automatically set to the full pathname of the corresponding command. These aliases are called *tracked* aliases. The value of a *tracked* alias is defined the first time the corresponding command is looked up and becomes undefined each time the `PATH` variable is reset. These aliases remain *tracked* so that the next subsequent reference redefines the value. Several tracked aliases are compiled into the shell. The `-h` option of the `set` command makes each referenced command name into a tracked alias.

The following *exported aliases* are compiled into (and built-in to) the shell but can be unset or redefined:

```
autoload='typeset -fu'
functions='typeset -f'
history='fc -l'
integer='typeset -i'
nohup='nohup '
r='fc -e -'
```

An example concerning trailing blank characters and reserved words follows. If the user types:

```
$ alias foo="/bin/ls "
$ alias while=""
```

the effect of executing:

```
$ while true
> do
> echo "Hello, World"
> done
```

is a never-ending sequence of `Hello, World` strings to the screen. However, if the user types:

```
$ foo while
```

the result is an `ls` listing of `/`. Since the alias substitution for `foo` ends in a space character, the next word is checked for alias substitution. The next word, `while`, has also been aliased, so it is substituted as well. Since it is not in the proper position as a command word, it is not recognized as a reserved word.

If the user types:

```
$ foo; while
```

`while` retains its normal reserved-word properties.

#### Tilde Substitution

After alias substitution is performed, each word is checked to see if it begins with an unquoted `~`. If it does, then the word up to a `/` is checked to see if it matches a user name. If a match is found, the `~` and the matched login name are replaced by the login directory of the matched user. This is called a *tilde* substitution. If no match is found, the original text is left unchanged. A `~` by itself, or in front of a `/`, is replaced by `$HOME`. A `~` followed by a `+` or `-` is replaced by `$PWD` and `$OLDPWD`, respectively.

In addition, *tilde* substitution is attempted when the value of a *variable assignment* begins with a `~`.

#### Tilde Expansion

A *tilde-prefix* consists of an unquoted tilde character at the beginning of a word, followed by all of the characters preceding the first unquoted slash in the word, or all the characters in the word if there is no slash. In an assignment, multiple tilde-prefixes can be used: at the beginning of the word (that is, following the equal sign of the assignment), following any unquoted colon or both. A tilde-prefix in an assignment is terminated by the first unquoted colon or slash. If none of the characters in the tilde-prefix are quoted, the characters in the tilde-prefix following the tilde are treated as a possible login name from the user database.

A portable login name cannot contain characters outside the set given in the description of the `LOGNAME` environment variable. If the login name is null (that is, the tilde-prefix contains only the tilde), the tilde-prefix is replaced by the value of the variable `HOME`. If `HOME` is unset, the results are unspecified. Otherwise, the tilde-prefix is replaced by a pathname of the home directory associated with the login name obtained using the `getpwnam` function. If the system does not recognize the login name, the results are undefined.

Tilde expansion generally occurs only at the beginning of words, but an exception based on historical practice has been included:

```
PATH=/posix/bin:~dgc/bin
```

is eligible for tilde expansion because tilde follows a colon and none of the relevant characters is quoted. Consideration was given to prohibiting this behavior because any of the following are reasonable substitutes:

```
PATH=$(printf %s ~karels/bin : ~bostic/bin)
for Dir in ~maat/bin ~srb/bin .
do
 PATH=${PATH:+$PATH:}$Dir
done
```

With the first command, explicit colons are used for each directory. In all cases, the shell performs tilde expansion on each directory because all are separate words to the shell.

Expressions in operands such as:

```
make -k mumble LIBDIR=~/chet/lib
```

do not qualify as shell variable assignments and tilde expansion is not performed (unless the command does so itself, which `make` does not).

The special sequence `$~` has been designated for future implementations to evaluate as a means of forcing tilde expansion in any word.

Because of the requirement that the word not be quoted, the following are not equivalent; only the last causes tilde expansion:

```
~h1j/ ~h\1j/ ~"h1j"/ ~h1j\ / ~h1j/
```

The results of giving tilde with an unknown login name are undefined because the KornShell `~+` and `~--` constructs make use of this condition, but, in general it is an error to give an incorrect login name with tilde. The results of having `HOME` unset are unspecified because some historical shells treat this as an error.

**Command Substitution** The standard output from a *command* enclosed in parenthesis preceded by a dollar sign (that is, `$(command)`) or a pair of grave accents (“”) can be used as part or all of a word. Trailing new-lines are removed. In the second (archaic) form, the string between the quotes is processed for special quoting characters before the command is executed. See **Quoting**. The command substitution `$(cat file)` can be replaced by the equivalent but faster `$(<file)`. Command substitution of most special commands that do not perform input/output redirection are carried out without creating a separate process.

Command substitution allows the output of a command to be substituted in place of the command name itself. Command substitution occurs when the command is enclosed as follows:

```
$(command)
```

or (backquoted version):

```
`command`
```

The shell expands the command substitution by executing *command* in a subshell environment and replacing the command substitution (the text of *command* plus the enclosing `$()` or backquotes) with the standard output of the command, removing sequences



of one or more newline characters at the end of the substitution. Embedded newline characters before the end of the output is not be removed; however, they can be treated as field delimiters and eliminated during field splitting, depending on the value of IFS and quoting that is in effect.

Within the backquoted style of command substitution, backslash shall retain its literal meaning, except when followed by:

\$ ' \

(dollar-sign, backquote, backslash). The search for the matching backquote is satisfied by the first backquote found without a preceding backslash. During this search, if a non-escaped backquote is encountered within a shell comment, a here-document, an embedded command substitution of the  $\$(command)$  form, or a quoted string, undefined results occur. A single- or double-quoted string that begins, but does not end, within the `'...'` sequence produces undefined results.

With the  $\$(command)$  form, all characters following the open parenthesis to the matching closing parenthesis constitute the *command*. Any valid shell script can be used for *command*, except:

- A script consisting solely of redirections produces unspecified results.
- See the restriction on single subshells.

The results of command substitution are not field splitting and pathname expansion processed for further tilde expansion, parameter expansion, command substitution or arithmetic expansion. If a command substitution occurs inside double-quotes, it is not be performed on the results of the substitution.

Command substitution can be nested. To specify nesting within the backquoted version, the application must precede the inner backquotes with backslashes; for example:

```
'\`command\`'
```

The  $\$( )$  form of command substitution solves a problem of inconsistent behavior when using backquotes. For example:

| Command              | Output |
|----------------------|--------|
| echo '\\$x'          | \\$x   |
| echo `echo '\\$x`    | \$x    |
| echo \$(echo '\\$x') | \\$x   |

Additionally, the backquoted syntax has historical restrictions on the contents of the embedded command. While the new  $\$( )$  form can process any kind of valid embedded script,

the backquoted form cannot handle some valid scripts that include backquotes. For example, these otherwise valid embedded scripts do not work in the left column, but do work on the right:

|                             |                             |
|-----------------------------|-----------------------------|
| echo `                      | echo \$(                    |
| cat <<eof                   | cat <<eof                   |
| a here-doc with `           | a here-doc with )           |
| eof                         | eof                         |
| `                           | )                           |
| echo `                      | echo \$(                    |
| echo abc # a comment with ` | echo abc # a comment with ) |
| `                           | )                           |
| echo `                      | echo \$(                    |
| echo ""                     | echo `)`                    |
| `                           | )                           |

Because of these inconsistent behaviors, the backquoted variety of command substitution is not recommended for new applications that nest command substitutions or attempt to embed complex scripts.

If the command substitution consists of a single subshell, such as:

```
$(command)
```

a portable application must separate the \$( and ( into two tokens (that is, separate them with white space). This is required to avoid any ambiguities with arithmetic expansion.

#### Arithmetic Expansion

An arithmetic expression enclosed in double parentheses preceded by a dollar sign (`$(arithmetic-expression)`) is replaced by the value of the arithmetic expression within the double parenthesis. Arithmetic expansion provides a mechanism for evaluating an arithmetic expression and substituting its value. The format for arithmetic expansion is as follows:

```
$((expression))
```

The expression is treated as if it were in double-quotes, except that a double-quote inside the expression is not treated specially. The shell expands all tokens in the expression for parameter expansion, command substitution and quote removal.

Next, the shell treats this as an arithmetic expression and substitute the value of the expression. The arithmetic expression is processed according to the rules of the ISO C with the following exceptions:

- Only integer arithmetic is required.
- The `sizeof()` operator and the prefix and postfix `++` and `--` operators are not required.
- Selection, iteration, and jump statements are not supported.
- `/usr/sunos/bin/ksh` and `/usr/sunos/bin/rksh` treat prefix 0 through 9 as decimal constants. See the following examples:

| Command                           | Result in <code>/bin/ksh</code> | Result in <code>/usr/xpg4/bin/sh</code> |
|-----------------------------------|---------------------------------|-----------------------------------------|
| <code>echo \$((010+10))</code>    | 20                              | 18                                      |
| <code>echo \$((019+10))</code>    | 29                              | error                                   |
| <code>[ 10 -le \$((011)) ]</code> | true                            | false                                   |

As an extension, the shell can recognize arithmetic expressions beyond those listed. If the expression is invalid, the expansion fails and the shell writes a message to standard error indicating the failure.

A simple example using arithmetic expansion:

```
repeat a command 100 times
x=100
while [$x -gt 0]
do
 command
 x=$((x-1))
done
```

#### Process Substitution

This feature is available in SunOS and only on versions of the UNIX operating system that support the `/dev/fd` directory for naming open files. Each command argument of the form `<(list)` or `>(list)` runs process `list` asynchronously connected to some file in `/dev/fd`. The name of this file becomes the argument to the command. If the form with `>` is selected, then writing on this file provides input for `list`. If `<` is used, then the file passed as an argument contains the output of the `list` process. For example:

```
paste <(cut -f1 file1) <(cut -f3 file2) | tee >(process1) >(process2)
```

`cut`s fields 1 and 3 from the files `file1` and `file2`, respectively, pastes the results together, and sends it to the processes `process1` and `process2`, as well as putting it onto the standard output. The file, which is passed as an argument to the command, is a UNIX [pipe\(2\)](#) so programs that expect to [lseek\(2\)](#) on the file does not work.

#### Parameter Substitution

A *parameter* is an *identifier*, one or more digits, or any of the characters `*`, `@`, `#`, `?`, `-`, `$`, and `!`. A *variable* (a *parameter* denoted by an *identifier*) has a *value* and zero or more *attributes*. *Variables* can be assigned *values* and *attributes* by using the `typeset` special command. The attributes supported by the shell are described later with the `typeset` special command. Exported variables pass values and attributes to the environment.

The shell supports a one-dimensional array facility. An element of an array variable is referenced by a *subscript*. A *subscript* is denoted by a [, followed by an *arithmetic expression*, followed by a ]. See Arithmetic Evaluation. To assign values to an array, use `set -A name value . . .`. The *value* of all subscripts must be in the range of 0 through 4095. Arrays need not be declared. Any reference to a variable with a valid subscript is legal and an array is created if necessary. Referencing an array without a subscript is equivalent to referencing the element 0. If an array *identifier* with subscript \* or @ is used, then the value for each of the elements is substituted (separated by a field separator character).

The *value* of a *variable* can be assigned by writing:

```
name=value [name=value] . . .
```

If the integer attribute, -i, is set for *name*, the *value* is subject to arithmetic evaluation.

Positional parameters, parameters denoted by a number, can be assigned values with the set special command. Parameter \$0 is set from argument zero when the shell is invoked. If *parameter* is one or more digits then it is a positional parameter. A positional parameter of more than one digit must be enclosed in braces.

Parameter Expansion    The format for parameter expansion is as follows:

```
${expression}
```

where *expression* consists of all characters until the matching }. Any } escaped by a backslash or within a quoted string, and characters in embedded arithmetic expansions, command substitutions and variable expansions, are not examined in determining the matching }.

The simplest form for parameter expansion is:

```
${parameter}
```

The value, if any, of *parameter* is substituted.

The parameter name or symbol can be enclosed in braces, which are optional except for positional parameters with more than one digit or when *parameter* is followed by a character that could be interpreted as part of the name. The matching closing brace are determined by counting brace levels, skipping over enclosed quoted strings and command substitutions.

If the parameter name or symbol is not enclosed in braces, the expansion uses the longest valid name whether or not the symbol represented by that name exists. When the shell is scanning its input to determine the boundaries of a name, it is not bound by its knowledge of what names are already defined. For example, if F is a defined shell variable, the command:

```
echo $Fred
```

does not echo the value of \$F followed by red; it selects the longest possible valid name, Fred, which in this case might be unset.

If a parameter expansion occurs inside double-quotes:

- Pathname expansion is not performed on the results of the expansion.
- Field splitting is not performed on the results of the expansion, with the exception of @.

In addition, a parameter expansion can be modified by using one of the following formats. In each case that a value of *word* is needed (based on the state of *parameter*), *word* is subjected to tilde expansion, parameter expansion, command substitution and arithmetic expansion. If *word* is not needed, it is not expanded. The } character that delimits the following parameter expansion modifications is determined as described previously in this section and in dquote. (For example, `_${foo-bar}xyz` would result in the expansion of `foo` followed by the string `xyz` if `foo` is set, else the string `barxyz`).

|                                   |                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>_\${parameter:-word}</code> | Use Default Values . If <i>parameter</i> is unset or null, the expansion of <i>word</i> is substituted. Otherwise, the value of <i>parameter</i> is substituted.                                                                                                                                                                                |
| <code>_\${parameter:=word}</code> | Assign Default Values . If <i>parameter</i> is unset or null, the expansion of <i>word</i> is assigned to <i>parameter</i> . In all cases, the final value of <i>parameter</i> is substituted. Only variables, not positional parameters or special parameters, can be assigned in this way.                                                    |
| <code>_\${parameter:?word}</code> | Indicate Error if Null or Unset . If <i>parameter</i> is unset or null, the expansion of <i>word</i> (or a message indicating it is unset if <i>word</i> is omitted) is written to standard error and the shell exits with a non-zero exit status. Otherwise, the value of <i>parameter</i> is substituted. An interactive shell need not exit. |
| <code>_\${parameter:+word}</code> | Use Alternative Value . If <i>parameter</i> is unset or null, null is substituted. Otherwise, the expansion of <i>word</i> is substituted.                                                                                                                                                                                                      |

In the parameter expansions shown previously, use of the colon in the format results in a test for a parameter that is unset or null. Omission of the colon results in a test for a parameter that is only unset. The following two tables summarize the effect of the colon:

|                                   | parameter set and not null  | parameter set and null      |
|-----------------------------------|-----------------------------|-----------------------------|
| <code>_\${parameter:-word}</code> | substitute <i>parameter</i> | substitute <i>word</i>      |
| <code>_\${parameter-word}</code>  | substitute <i>parameter</i> | substitute null             |
| <code>_\${parameter:=word}</code> | substitute <i>parameter</i> | assign <i>word</i>          |
| <code>_\${parameter=word}</code>  | substitute <i>parameter</i> | substitute <i>parameter</i> |
| <code>_\${parameter:?word}</code> | substitute <i>parameter</i> | error, exit                 |
| <code>_\${parameter?word}</code>  | substitute <i>parameter</i> | substitute null             |
| <code>_\${parameter:+word}</code> | substitute <i>word</i>      | substitute null             |

|                                  | parameter set and not null | parameter set and null |
|----------------------------------|----------------------------|------------------------|
| <code>\${parameter+word}</code>  | substitute <i>word</i>     | substitute <i>word</i> |
|                                  |                            |                        |
|                                  |                            | parameter unset        |
| <code>\${parameter:-word}</code> | substitute <i>word</i>     |                        |
| <code>\${parameter-word}</code>  | substitute <i>word</i>     |                        |
| <code>\${parameter:=word}</code> | assign <i>word</i>         |                        |
| <code>\${parameter=word}</code>  | assign null                |                        |
| <code>\${parameter:?word}</code> | error, exit                |                        |
| <code>\${parameter?word}</code>  | error, exit                |                        |
| <code>\${parameter:+word}</code> | substitute null            |                        |
| <code>\${parameter+word}</code>  | substitute null            |                        |

In all cases shown with “substitute”, the expression is replaced with the value shown. In all cases shown with “assign”, *parameter* is assigned that value, which also replaces the expression.

`${#parameter}`     **String Length.** The length in characters of the value of *parameter*. If *parameter* is \* or @, then all the positional parameters, starting with \$1, are substituted (separated by a field separator character).

The following four varieties of parameter expansion provide for substring processing. In each case, pattern matching notation (see `patmat`), rather than regular expression notation, is used to evaluate the patterns. If *parameter* is \* or @, then all the positional parameters, starting with \$1, are substituted (separated by a field separator character). Enclosing the full parameter expansion string in double-quotes does not cause the following four varieties of pattern characters to be quoted, whereas quoting characters within the braces has this effect.

`${parameter%word}`     **Remove Smallest Suffix Pattern.** The *word* is expanded to produce a pattern. The parameter expansion then results in *parameter*, with the smallest portion of the suffix matched by the *pattern* deleted.

`${parameter%%word}`     **Remove Largest Suffix Pattern.** The *word* is expanded to produce a pattern. The parameter expansion then results in *parameter*, with the largest portion of the suffix matched by the *pattern* deleted.

`${parameter#word}` Remove Smallest Prefix Pattern. The *word* is expanded to produce a pattern. The parameter expansion then results in *parameter*, with the smallest portion of the prefix matched by the *pattern* deleted.

`${parameter##word}` Remove Largest Prefix Pattern. The *word* is expanded to produce a pattern. The parameter expansion then results in *parameter*, with the largest portion of the prefix matched by the *pattern* deleted.

Examples:

`${parameter:-word}`

In this example, `ls` is executed only if `x` is `null` or `unset`. (The `$(ls)` command substitution notation is explained in [Command Substitution](#) above.)

```
 ${x:-$(ls)}
```

`${parameter:=word}`

```
unset X
echo ${X:=abc}
abc
```

`${parameter:?word}`

```
unset posix
echo ${posix:?}
sh: posix: parameter null or not set
```

`${parameter:+word}`

```
set a b c
echo ${3:+posix}
posix
```

`${#parameter}`

```
HOME=/usr/posix
echo ${#HOME}
10
```

`${parameter%word}`

```
x=file.c
echo ${x%.c}.o
file.o
```

`${parameter%%word}`

```
x=posix/src/std
echo ${x%%/*}
posix
```

```
${parameter#word}
```

```
x=$HOME/src/cmd
echo ${x#$HOME}
/src/cmd
```

```
${parameter##word}
```

```
x=/one/two/three
echo ${x##*/}
three
```

Parameters Set by Shell The following parameters are automatically set by the shell:

|        |                                                                                                                                                                                                                                                                                                                                                                                 |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| #      | The number of positional parameters in decimal.                                                                                                                                                                                                                                                                                                                                 |
| —      | Flags supplied to the shell on invocation or by the set command.                                                                                                                                                                                                                                                                                                                |
| ?      | The decimal value returned by the last executed command.                                                                                                                                                                                                                                                                                                                        |
| \$     | The process number of this shell.                                                                                                                                                                                                                                                                                                                                               |
| _      | Initially, the value of <code>_</code> is an absolute pathname of the shell or script being executed as passed in the <i>environment</i> . Subsequently it is assigned the last argument of the previous command. This parameter is not set for commands which are asynchronous. This parameter is also used to hold the name of the matching MAIL file when checking for mail. |
| !      | The process number of the last background command invoked.                                                                                                                                                                                                                                                                                                                      |
| ERRNO  | The value of <code>errno</code> as set by the most recently failed system call. This value is system dependent and is intended for debugging purposes.                                                                                                                                                                                                                          |
| LINENO | The line number of the current line within the script or function being executed.                                                                                                                                                                                                                                                                                               |
| OLDPWD | The previous working directory set by the <code>cd</code> command.                                                                                                                                                                                                                                                                                                              |
| OPTARG | The value of the last option argument processed by the <code>getopts</code> special command.                                                                                                                                                                                                                                                                                    |
| OPTIND | The index of the last option argument processed by the <code>getopts</code> special command.                                                                                                                                                                                                                                                                                    |
| PPID   | The process number of the parent of the shell.                                                                                                                                                                                                                                                                                                                                  |
| PWD    | The present working directory set by the <code>cd</code> command.                                                                                                                                                                                                                                                                                                               |



|         |                                                                                                                                                                                                                                                               |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RANDOM  | Each time this variable is referenced, a random integer, uniformly distributed between 0 and 32767, is generated. The sequence of random numbers can be initialized by assigning a numeric value to RANDOM.                                                   |
| REPLY   | This variable is set by the <code>select</code> statement and by the <code>read</code> special command when no arguments are supplied.                                                                                                                        |
| SECONDS | Each time this variable is referenced, the number of seconds since shell invocation is returned. If this variable is assigned a value, then the value returned upon reference is the value that was assigned plus the number of seconds since the assignment. |

Variables Used by Shell The following variables are used by the shell:

|         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CDPATH  | The search path for the <code>cd</code> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| COLUMNS | If this variable is set, the value is used to define the width of the edit window for the shell edit modes and for printing <code>select</code> lists.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| EDITOR  | If the value of this variable ends in <code>emacs</code> , <code>gmacs</code> , or <code>vi</code> and the <code>VISUAL</code> variable is not set, then the corresponding option is turned on. See the <code>set</code> special command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| ENV     | <p>This variable, when and only when an interactive shell is invoked, is subjected to parameter expansion by the shell and the resulting value is used as a pathname of a file containing shell commands to execute in the current environment. The file need not be executable. If the expanded value of <code>ENV</code> is not an absolute pathname, the results are unspecified. <code>ENV</code> is ignored if the user's real and effective user IDs or real and effective group IDs are different.</p> <p>This variable can be used to set aliases and other items local to the invocation of a shell. The file referred to by <code>ENV</code> differs from <code>\$HOME/.profile</code> in that <code>.profile</code> is typically executed at session startup, whereas the <code>ENV</code> file is executed at the beginning of each shell invocation. The <code>ENV</code> value is interpreted in a manner similar to a dot script, in that the commands are executed in the current environment and the file needs to be readable, but not executable. However, unlike dot scripts, no <code>PATH</code> searching is performed. This is used as a guard against Trojan Horse security breaches.</p> |
| FCEDIT  | The default editor name for the <code>fc</code> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| FPATH   | The search path for function definitions. By default, the <code>FPATH</code> directories are searched after the <code>PATH</code> variable. If an executable file is found, then it is read and executed in the current environment. <code>FPATH</code> is searched before <code>PATH</code> when a function with the <code>-u</code> attribute is referenced. The preset alias <code>autoload</code> causes a function with the <code>-u</code> attribute to be created.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HISTFILE    | If this variable is set when the shell is invoked, then the value is the pathname of the file that is used to store the command history. See <code>Command re-entry</code> .                                                                                                                                                                                                                                                                                                                                                                    |
| HISTSIZE    | If this variable is set when the shell is invoked, then the number of previously entered commands that are accessible by this shell is greater than or equal to this number. The default is 128.                                                                                                                                                                                                                                                                                                                                                |
| HOME        | The default argument (home directory) for the <code>cd</code> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| IFS         | Internal field separators, normally space, tab, and <code>new-line</code> that are used to separate command words which result from command or parameter substitution and for separating words with the special command <code>read</code> . The first character of the <code>IFS</code> variable is used to separate arguments for the <code>\$*</code> substitution. See <code>Quoting</code> .                                                                                                                                                |
| LANG        | Provide a default value for the internationalization variables that are unset or null. If any of the internationalization variables contains an invalid setting, the utility behaves as if none of the variables had been defined.                                                                                                                                                                                                                                                                                                              |
| LC_ALL      | This variable provides a default value for the <code>LC_*</code> variables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| LC_COLLATE  | This variable determines the behavior of range expressions, equivalence classes and multi-byte character collating elements within pattern matching.                                                                                                                                                                                                                                                                                                                                                                                            |
| LC_CTYPE    | Determines how the shell handles characters. When <code>LC_CTYPE</code> is set to a valid value, the shell can display and handle text and filenames containing valid characters for that locale. If <code>LC_CTYPE</code> (see <code>environ(5)</code> ) is not set in the environment, the operational behavior of the shell is determined by the value of the <code>LANG</code> environment variable. If <code>LC_ALL</code> is set, its contents are used to override both the <code>LANG</code> and the other <code>LC_*</code> variables. |
| LC_MESSAGES | This variable determines the language in which messages should be written.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| LINENO      | This variable is set by the shell to a decimal number representing the current sequential line number (numbered starting with 1) within a script or function before it executes each command. If the user unsets or resets <code>LINENO</code> , the variable can lose its special meaning for the life of the shell. If the shell is not currently executing a script or function, the value of <code>LINENO</code> is unspecified.                                                                                                            |
| LINES       | If this variable is set, the value is used to determine the column length for printing <code>select</code> lists. <code>Select</code> lists print vertically until about two-thirds of <code>LINES</code> lines are filled.                                                                                                                                                                                                                                                                                                                     |
| MAIL        | If this variable is set to the name of a mail file <i>and</i> the <code>MAILPATH</code> variable is not set, then the shell informs the user of arrival of mail in the specified file.                                                                                                                                                                                                                                                                                                                                                          |

---

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAILCHECK | This variable specifies how often (in seconds) the shell checks for changes in the modification time of any of the files specified by the MAILPATH or MAIL variables. The default value is 600 seconds. When the time has elapsed the shell checks before issuing the next prompt.                                                                                                                                                          |
| MAILPATH  | A colon (:) separated list of file names. If this variable is set, then the shell informs the user of any modifications to the specified files that have occurred within the last MAILCHECK seconds. Each file name can be followed by a ? and a message that is printed. The message undergoes parameter substitution with the variable \$_ defined as the name of the file that has changed. The default message is you have mail in \$_. |
| NLSPATH   | Determine the location of message catalogues for the processing of LC_MESSAGES.                                                                                                                                                                                                                                                                                                                                                             |
| PATH      | The search path for commands. See Execution. The user cannot change PATH if executing under rksh (except in .profile).                                                                                                                                                                                                                                                                                                                      |
| PPID      | This variable is set by the shell to the decimal process ID of the process that invoked the shell. In a subshell, PPID is set to the same value as that of the parent of the current shell. For example, echo \$PPID and (echo \$PPID) would produce the same value.                                                                                                                                                                        |
| PS1       | The value of this variable is expanded for parameter substitution to define the primary prompt string which by default is "\$ ". The character ! in the primary prompt string is replaced by the <i>command</i> number. See Command Re-entry. Two successive occurrences of ! produces a single ! when the prompt string is printed.                                                                                                        |
| PS2       | Secondary prompt string, by default "> ".                                                                                                                                                                                                                                                                                                                                                                                                   |
| PS3       | Selection prompt string used within a select loop, by default "#? ".                                                                                                                                                                                                                                                                                                                                                                        |
| PS4       | The value of this variable is expanded for parameter substitution and precedes each line of an execution trace. If omitted, the execution trace prompt is "+ ".                                                                                                                                                                                                                                                                             |
| PWD       | Set by the shell to be an absolute pathname of the current working directory, containing no components of type symbolic link, no components that are dot, and no components that are dot-dot when the shell is initialized. If an application sets or unsets the value of PWD, the behaviors of the cd and pwd utilities are unspecified                                                                                                    |
| SHELL     | The pathname of the <i>shell</i> is kept in the environment. At invocation, if the basename of this variable is rsh, rksh, or krsh, then the shell becomes restricted.                                                                                                                                                                                                                                                                      |

|        |                                                                                                                                                                                                                                                     |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TMOUT  | If set to a value greater than zero, the shell terminates if a command is not entered within the prescribed number of seconds after issuing the PS1 prompt. The shell can be compiled with a maximum bound for this value which cannot be exceeded. |
| VISUAL | If the value of this variable ends in emacs, gmacs, or vi, then the corresponding option is turned on. See Special Command set.                                                                                                                     |

The shell gives default values to PATH, PS1, PS2, PS3, PS4, MAILCHECK, FCEDIT, TMOUT, and IFS, while HOME, SHELL, ENV, and MAIL are not set at all by the shell (although HOME *is* set by [login\(1\)](#)). On some systems MAIL and SHELL are also set by [login](#).

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Blank Interpretation | After parameter and command substitution, the results of substitutions are scanned for the field separator characters (those found in IFS) and split into distinct arguments where such characters are found. Explicit null arguments ( "" ) or ( ' ' ) are retained. Implicit null arguments (those resulting from <i>parameters</i> that have no values) are removed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| File Name Generation | <p>Following substitution, each command <i>word</i> is scanned for the characters *, ?, and [ unless the -f option has been set. If one of these characters appears, the word is regarded as a <i>pattern</i>. The word is replaced with lexicographically sorted file names that match the pattern. If no file name is found that matches the pattern, the word is left unchanged. When a <i>pattern</i> is used for file name generation, the character period ( . ) at the start of a file name or immediately following a /, as well as the character / itself, must be matched explicitly. A file name beginning with a period is not matched with a pattern with the period inside parentheses. That is, <code>ls .@(r*)</code> would locate a file named <code>.restore</code>, but <code>ls @( .r*)</code> would not. In other instances of pattern matching, the / and . are not treated specially.</p> <ul style="list-style-type: none"> <li>* Matches any string, including the null string.</li> <li>? Matches any single character.</li> <li>[...] Matches any one of the enclosed characters. A pair of characters separated by – matches any character lexically between the pair, inclusive. If the first character following the opening “[” is a “!”, then any character not enclosed is matched. A – can be included in the character set by putting it as the first or last character.</li> </ul> |

A *pattern-list* is a list of one or more patterns separated from each other with a |. Composite patterns can be formed with one or more of the following:

|                          |                                                         |
|--------------------------|---------------------------------------------------------|
| ?( <i>pattern-list</i> ) | Optionally matches any one of the given patterns.       |
| *( <i>pattern-list</i> ) | Matches zero or more occurrences of the given patterns. |
| +( <i>pattern-list</i> ) | Matches one or more occurrences of the given patterns.  |
| @( <i>pattern-list</i> ) | Matches exactly one of the given patterns.              |
| !( <i>pattern-list</i> ) | Matches anything, except one of the given patterns.     |

- Quoting Each of the *metacharacters* listed above (see Definitions) has a special meaning to the shell and causes termination of a word unless quoted. A character can be *quoted* (that is, made to stand for itself) by preceding it with a `\`. The pair `\ NEWLINE` is removed. All characters enclosed between a pair of single quote marks (`' '`) are quoted. A single quote cannot appear within single quotes. Inside double quote marks (`"`), parameter and command substitution occur and `\` quotes the characters `\`, `'`, `"`, and `$`. The meaning of `$*` and `@` is identical when not quoted or when used as a parameter assignment value or as a file name. However, when used as a command argument, `$*` is equivalent to `"$1d $2d. . ."`, where *d* is the first character of the IFS variable, whereas `@` is equivalent to `$1 $2 . . .`. Inside grave quote marks (```), `\` quotes the characters `\`, `'`, and `$`. If the grave quotes occur within double quotes, then `\` also quotes the character `"`.
- The special meaning of reserved words or aliases can be removed by quoting any character of the reserved word. The recognition of function names or special command names listed cannot be altered by quoting them.
- Arithmetic Evaluation An ability to perform integer arithmetic is provided with the special command `let`. Evaluations are performed using *long* arithmetic. Constants are of the form `[ base# ] n` where *base* is a decimal number between two and thirty-six representing the arithmetic base and *n* is a number in that base. If *base* is omitted then base 10 is used.
- An arithmetic expression uses the same syntax, precedence, and associativity of expression as the C language. All the integral operators, other than `++`, `--`, `?:`, and `,` are supported. Variables can be referenced by name within an arithmetic expression without using the parameter substitution syntax. When a variable is referenced, its value is evaluated as an arithmetic expression.
- An internal integer representation of a *variable* can be specified with the `-i` option of the `typeset` special command. Arithmetic evaluation is performed on the value of each assignment to a variable with the `-i` attribute. If you do not specify an arithmetic base, the first assignment to the variable determines the arithmetic base. This base is used when parameter substitution occurs.
- Since many of the arithmetic operators require quoting, an alternative form of the `let` command is provided. For any command which begins with a `(`, all the characters until a matching `)` are treated as a quoted expression. More precisely, `(( . . . ))` is equivalent to `let " . . . "`.
- Prompting When used interactively, the shell prompts with the parameter expanded value of `PS1` before reading a command. If at any time a new-line is typed and further input is needed to complete a command, then the secondary prompt (that is, the value of `PS2`) is issued.

## Conditional Expressions

A *conditional expression* is used with the `[[` compound command to test attributes of files and to compare strings. Word splitting and file name generation are not performed on the words between `[[` and `]]`. Each expression can be constructed from one or more of the following unary or binary expressions:

- a *file* True, if *file* exists.
- b *file* True, if *file* exists and is a block special file.
- c *file* True, if *file* exists and is a character special file.
- d *file* True, if *file* exists and is a directory.
- e *file* True, if *file* exists.
- f *file* True, if *file* exists and is an ordinary file.
- g *file* True, if *file* exists and has its setgid bit set.
- h *file* True, if *file* exists and is a symbolic link.
- k *file* True, if *file* exists and has its sticky bit set.
- n *string* True, if length of *string* is non-zero.
- o *option* True, if option named *option* is on.
- p *file* True, if *file* exists and is a fifo special file or a pipe.
- r *file* True, if *file* exists and is readable by current process.
- s *file* True, if *file* exists and has size greater than zero.
- t *fildev* True, if file descriptor number *fildev* is open and associated with a terminal device.
- u *file* True, if *file* exists and has its setuid bit set.
- w *file* True, if *file* exists and is writable by current process.
- x *file* True, if *file* exists and is executable by current process. If *file* exists and is a directory, then the current process has permission to search in the directory.
- z *string* True, if length of *string* is zero.
- L *file* True, if *file* exists and is a symbolic link.
- O *file* True, if *file* exists and is owned by the effective user id of this process.
- G *file* True, if *file* exists and its group matches the effective group id of this process.
- S *file* True, if *file* exists and is a socket.

|                                 |                                                                                                                                                |
|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>file1</i> -nt <i>file2</i>   | True, if <i>file1</i> exists and is newer than <i>file2</i> .                                                                                  |
| <i>file1</i> -ot <i>file2</i>   | True, if <i>file1</i> exists and is older than <i>file2</i> .                                                                                  |
| <i>file1</i> -ef <i>file2</i>   | True, if <i>file1</i> and <i>file2</i> exist and refer to the same file.                                                                       |
| <i>string</i>                   | True if the string <i>string</i> is not the null string.                                                                                       |
| <i>string</i> == <i>pattern</i> | True, if <i>string</i> matches <i>pattern</i> .                                                                                                |
| <i>string</i> = <i>pattern</i>  | Same as ==, but is obsolete.                                                                                                                   |
| <i>string</i> != <i>pattern</i> | True, if <i>string</i> does not match <i>pattern</i> .                                                                                         |
| <i>string1</i> < <i>string2</i> | True, if <i>string1</i> comes before <i>string2</i> based on strings interpreted as appropriate to the locale setting for category LC_COLLATE. |
| <i>string1</i> > <i>string2</i> | True, if <i>string1</i> comes after <i>string2</i> based on strings interpreted as appropriate to the locale setting for category LC_COLLATE.  |
| <i>exp1</i> -eq <i>exp2</i>     | True, if <i>exp1</i> is equal to <i>exp2</i> .                                                                                                 |
| <i>exp1</i> -ne <i>exp2</i>     | True, if <i>exp1</i> is not equal to <i>exp2</i> .                                                                                             |
| <i>exp1</i> -lt <i>exp2</i>     | True, if <i>exp1</i> is less than <i>exp2</i> .                                                                                                |
| <i>exp1</i> -gt <i>exp2</i>     | True, if <i>exp1</i> is greater than <i>exp2</i> .                                                                                             |
| <i>exp1</i> -le <i>exp2</i>     | True, if <i>exp1</i> is less than or equal to <i>exp2</i> .                                                                                    |
| <i>exp1</i> -ge <i>exp2</i>     | True, if <i>exp1</i> is greater than or equal to <i>exp2</i> .                                                                                 |

In each of the above expressions, if *file* is of the form */dev/fd/n*, where *n* is an integer, then the test is applied to the open file whose descriptor number is *n*.

A compound expression can be constructed from these primitives by using any of the following, listed in decreasing order of precedence.

|                                          |                                                                   |
|------------------------------------------|-------------------------------------------------------------------|
| ( <i>expression</i> )                    | True, if <i>expression</i> is true. Used to group expressions.    |
| ! <i>expression</i>                      | True if <i>expression</i> is false.                               |
| <i>expression1</i> && <i>expression2</i> | True, if <i>expression1</i> and <i>expression2</i> are both true. |
| <i>expression1</i>    <i>expression2</i> | True, if either <i>expression1</i> or <i>expression2</i> is true. |

#### Input/Output

Before a command is executed, its input and output can be redirected using a special notation interpreted by the shell. The following can appear anywhere in a simple-command or can precede or follow a *command* and are *not* passed on to the invoked command. Command and parameter substitution occur before *word* or *digit* is used except as noted. File name generation occurs only if the pattern matches a single file, and blank interpretation is not performed.

<*word*            Use file *word* as standard input (file descriptor 0).

|                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&gt;word</code>          | Use file <i>word</i> as standard output (file descriptor 1). If the file does not exist then it is created. If the file exists, and the <code>-noclobber</code> option is on, this causes an error; otherwise, it is truncated to zero length.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>&gt; word</code>         | Same as <code>&gt;</code> , except that it overrides the <code>-noclobber</code> option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>&gt;&gt;word</code>      | Use file <i>word</i> as standard output. If the file exists, output is appended to it (by first seeking to the EOF). Otherwise, the file is created.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>&lt;&gt;word</code>      | Open file <i>word</i> for reading and writing as standard input.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>&lt;&lt; [- ]word</code> | The shell input is read up to a line that is the same as <i>word</i> , or to an EOF. No parameter substitution, command substitution, or file name generation is performed on <i>word</i> . The resulting document, called a <i>here-document</i> , becomes the standard input. If any character of <i>word</i> is quoted, no interpretation is placed upon the characters of the document. Otherwise, parameter and command substitution occur, <code>\NEWLINE</code> is ignored, and <code>\</code> must be used to quote the characters <code>\</code> , <code>\$</code> , <code>'</code> , and the first character of <i>word</i> . If <code>-</code> is appended to <code>&lt;&lt;</code> , then all leading tabs are stripped from <i>word</i> and from the document. |
| <code>&lt;&amp;digit</code>    | The standard input is duplicated from file descriptor <i>digit</i> (see <code>dup(2)</code> ). Similarly for the standard output using <code>&gt;&amp;digit</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>&lt;&amp;-</code>        | The standard input is closed. Similarly for the standard output using <code>&gt;&amp;-</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <code>&lt;&amp;p</code>        | The input from the co-process is moved to standard input.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <code>&gt;&amp;p</code>        | The output to the co-process is moved to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

If one of the above is preceded by a digit, then the file descriptor number referred to is that specified by the digit (instead of the default 0 or 1). For example:

```
... 2>&1
```

means file descriptor 2 is to be opened for writing as a duplicate of file descriptor 1.

The order in which redirections are specified is significant. The shell evaluates each redirection in terms of the (*file descriptor, file*) association at the time of evaluation. For example:

```
... 1>fname 2>&1
```

first associates file descriptor 1 with file *fname*. It then associates file descriptor 2 with the file associated with file descriptor 1 (that is, *fname*). If the order of redirections were reversed, file descriptor 2 would be associated with the terminal (assuming file descriptor 1 had been) and then file descriptor 1 would be associated with file *fname*.



If a command is followed by `&` and job control is not active, then the default standard input for the command is the empty file `/dev/null`. Otherwise, the environment for the execution of a command contains the file descriptors of the invoking shell as modified by input/output specifications.

#### Environment

The *environment* (see [environ\(5\)](#)) is a list of name-value pairs that is passed to an executed program in the same way as a normal argument list. The names must be *identifiers* and the values are character strings. The shell interacts with the environment in several ways. On invocation, the shell scans the environment and creates a variable for each name found, giving it the corresponding value and marking it *export*. Executed commands inherit the environment. If the user modifies the values of these variables or creates new ones, using the `export` or `typeset -x` commands, they become part of the environment. The environment seen by any executed command is thus composed of any name-value pairs originally inherited by the shell, whose values can be modified by the current shell, plus any additions which must be noted in `export` or `typeset -x` commands.

The environment for any *simple-command* or *function* can be augmented by prefixing it with one or more variable assignments. A variable assignment argument is a word of the form *identifier=value*. Thus:

```
TERM=450 cmd args
```

and

```
(export TERM; TERM=450; cmd args)
```

are equivalent (as far as the above execution of *cmd* is concerned, except for special commands listed that are preceded with an asterisk).

If the `-k` flag is set, *all* variable assignment arguments are placed in the environment, even if they occur after the command name. The following first prints `a=b c` and then `c`:

```
echo a=b c
set -k echo
a=b c
```

This feature is intended for use with scripts written for early versions of the shell and its use in new scripts is strongly discouraged. It is likely to disappear someday.

#### Functions

The `function` reserved word, described in the [Commands](#) section above, is used to define shell functions. Shell functions are read in and stored internally. Alias names are resolved when the function is read. Functions are executed like commands with the arguments passed as positional parameters. See [Execution](#).

Functions execute in the same process as the caller and share all files and present working directory with the caller. Traps caught by the caller are reset to their default action inside the

function. A trap condition that is not caught or ignored by the function causes the function to terminate and the condition to be passed on to the caller.

A trap on EXIT set inside a function is executed after the function completes in the environment of the caller. This is true only for non-POSIX-style functions, that is, functions declared as

```
function func
```

as opposed to POSIX-style functions, declared as

```
func()
```

Ordinarily, variables are shared between the calling program and the function. However, the `typeset` special command used within a function defines local variables whose scope includes the current function and all functions it calls.

The special command `return` is used to return from function calls. Errors within functions return control to the caller.

The names of all functions can be listed with `typeset -f`. `typeset -f` lists all function names as well as the text of all functions. `typeset -f function-names` lists the text of the named functions only. Functions can be undefined with the `-f` option of the `unset` special command.

Ordinarily, functions are unset when the shell executes a shell script. The `-xf` option of the `typeset` command allows a function to be exported to scripts that are executed without a separate invocation of the shell. Functions that need to be defined across separate invocations of the shell should be specified in the ENV file with the `-xf` option of `typeset`.

#### Function Definition Command

A function is a user-defined name that is used as a simple command to call a compound command with new positional parameters. A function is defined with a *function definition command*.

The format of a function definition command is as follows:

```
fname() compound-command[io-redirect ...]
```

The function is named `fname`; it must be a name. An implementation can allow other characters in a function name as an extension. The implementation maintains separate name spaces for functions and variables.

The `()` in the function definition command consists of two operators. Therefore, intermixing blank characters with the `fname`, `(`, and `)` is allowed, but unnecessary.

The argument *compound-command* represents a compound command.

When the function is declared, none of the expansions in `wordexp` is performed on the text in *compound-command* or *io-redirect*; all expansions is performed as normal each time the

function is called. Similarly, the optional *io-redirect* redirections and any variable assignments within *compound-command* is performed during the execution of the function itself, not the function definition.

When a function is executed, it has the syntax-error and variable-assignment properties described for the special built-in utilities.

The *compound-command* is executed whenever the function name is specified as the name of a simple command. The operands to the command temporarily becomes the positional parameters during the execution of the *compound-command*; the special parameter # is also changed to reflect the number of operands. The special parameter 0 is unchanged. When the function completes, the values of the positional parameters and the special parameter # is restored to the values they had before the function was executed. If the special built-in return is executed in the *compound-command*, the function completes and execution resumes with the next command after the function call.

An example of how a function definition can be used wherever a simple command is allowed:

```
If variable i is equal to "yes",
define function foo to be ls -l
#
["$i" = yes] && foo() {
 ls -l
}
```

The exit status of a function definition is 0 if the function was declared successfully; otherwise, it is greater than zero. The exit status of a function invocation is the exit status of the last command executed by the function.

## Jobs

If the *monitor* option of the *set* command is turned on, an interactive shell associates a job with each pipeline. It keeps a table of current jobs, printed by the *jobs* command, and assigns them small integer numbers. When a job is started asynchronously with *&*, the shell prints a line which looks like:

```
[1] 1234
```

indicating that the job, which was started asynchronously, was job number 1 and had one (top-level) process, whose process id was 1234.

If you are running a job and wish to do something else you can press the key *^Z* (Control-Z) which sends a *STOP* signal to the current job. The shell normally indicates that the job has been 'Stopped', and print another prompt. You can then manipulate the state of this job, putting it in the background with the *bg* command, or run some other commands and then eventually bring the job back into the foreground with the foreground command *fg*. A *^Z* takes effect immediately and is like an interrupt in that pending output and unread input are discarded when it is typed.

A job being run in the background stops if it tries to read from the terminal. Background jobs are normally allowed to produce output, but this can be disabled by giving the command “`stty tostop`”. If you set this tty option, then background jobs stop when they try to produce output as they do when they try to read input.

There are several ways to refer to jobs in the shell. A job can be referred to by the process id of any process of the job or by one of the following:

- `%number` The job with the given number.
- `%string` Any job whose command line begins with *string*.
- `%?string` Any job whose command line contains *string*.
- `%%` Current job.
- `%+` Equivalent to `%%`.
- `%-` Previous job.

The shell learns immediately whenever a process changes state. It normally informs you whenever a job becomes blocked so that no further progress is possible, but only just before it prints a prompt. This is done so that it does not otherwise disturb your work.

When the monitor mode is on, each background job that completes triggers any trap set for CHLD.

When you try to leave the shell while jobs are running or stopped, you are warned with the message, ‘You have stopped (running) jobs.’ You can use the jobs command to see what they are. If you do this or immediately try to exit again, the shell does not warn you a second time, and the stopped jobs is terminated. If you have jobs running for which the nohup command was invoked and attempt to logout, you are warned with the message:

You have jobs running.

You need to logout a second time to actually logout. However, your background jobs continue to run.

Signals The INT and QUIT signals for an invoked command are ignored if the command is followed by `&` and the `-monitor` option is not active. Otherwise, signals have the values inherited by the shell from its parent. See the trap special command section.

Execution Each time a command is executed, the above substitutions are carried out. If the command name matches one of the Special Commands listed, it is executed within the current shell process. Next, the command name is checked to see if it matches one of the user defined functions. If it does, the positional parameters are saved and then reset to the arguments of the function call. When the function completes or issues a return, the positional parameter list is restored and any trap set on EXIT within the function is executed. The value of a function is

the value of the last command executed. A function is also executed in the current shell process. If a command name is not a special command or a user defined function, a process is created and an attempt is made to execute the command using `exec(2)`.

The shell variable `PATH` defines the search path for the directory containing the command. Alternative directory names are separated by a colon (:). The default path is `/usr/bin:` (specifying `/usr/bin` and the current directory in that order). The current directory can be specified by two or more adjacent colons, or by a colon at the beginning or end of the path list. If the command name contains a / then the search path is not used. Otherwise, each directory in the path is searched for an executable file. If the file has execute permission but is not a directory or an `a.out` file, it is assumed to be a file containing shell commands. A sub-shell is spawned to read it. All non-exported aliases, functions, and variables are removed in this case. A parenthesized command is executed in a sub-shell without removing non-exported quantities.

**Command Re-entry** The text of the last `HISTSIZE` (default 128) commands entered from a terminal device is saved in a `history` file. The file `$HOME/.sh_history` is used if the `HISTFILE` variable is not set or if the file it names is not writable. A shell can access the commands of all *interactive* shells which use the same named `HISTFILE`. The special command `fc` is used to list or edit a portion of this file. The portion of the file to be edited or listed can be selected by number or by giving the first character or characters of the command. A single command or range of commands can be specified. If you do not specify an editor program as an argument to `fc` then the value of the variable `FCEDIT` is used. If `FCEDIT` is not defined, then `/bin/ed` is used. The edited command(s) is printed and re-executed upon leaving the editor. The editor name `-` is used to skip the editing phase and to re-execute the command. In this case a substitution parameter of the form `old=new` can be used to modify the command before execution. For example, if `r` is aliased to `'fc -e -'` then typing `'r bad=good c'` re-executes the most recent command which starts with the letter `c`, replacing the first occurrence of the string `bad` with the string `good`.

**In-line Editing Option** Normally, each command line entered from a terminal device is simply typed followed by a new-line (RETURN or LINEFEED). If either the `emacs`, `gmacs`, or `vi` option is active, the user can edit the command line. To be in either of these edit modes set the corresponding option. An editing option is automatically selected each time the `VISUAL` or `EDITOR` variable is assigned a value ending in either of these option names.

The editing features require that the user's terminal accept RETURN as carriage return without line feed and that a space must overwrite the current character on the screen.

The editing modes implement a concept where the user is looking through a window at the current line. The window width is the value of `COLUMNS` if it is defined, otherwise 80. If the window width is too small to display the prompt and leave at least 8 columns to enter input, the prompt is truncated from the left. If the line is longer than the window width minus two, a mark is displayed at the end of the window to notify the user. As the cursor moves and reaches

the window boundaries the window are centered about the cursor. The mark is a > if the line extends on the right side of the window, < if the line extends on the left, and \* if the line extends on both sides of the window.

The search commands in each edit mode provide access to the history file. Only strings are matched, not patterns, although a leading caret (^) in the string restricts the match to begin at the first character in the line.

#### emacs Editing Mode

This mode is entered by enabling either the emacs or gmacs option. The only difference between these two modes is the way they handle ^T. To edit, move the cursor to the point needing correction and then insert or delete characters or words as needed. All the editing commands are control characters or escape sequences. The notation for control characters is caret (^) followed by the character. For example, ^F is the notation for control F. This is entered by depressing 'f' while holding down the CTRL (control) key. The SHIFT key is *not* depressed. (The notation ^? indicates the DEL (delete) key.)

The notation for escape sequences is M- followed by a character. For example, M- f (pronounced Meta f) is entered by depressing ESC (ascii 033) followed by 'f'. (M- F would be the notation for ESC followed by SHIFT (capital) 'F'.)

All edit commands operate from any place on the line (not just at the beginning). Neither the RETURN nor the LINEFEED key is entered after edit commands except when noted.

|              |                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| ^F           | Move cursor forward (right) one character.                                                                                                      |
| M- f         | Move cursor forward one word. (The emacs editor's idea of a word is a string of characters consisting of only letters, digits and underscores.) |
| ^B           | Move cursor backward (left) one character.                                                                                                      |
| M- b         | Move cursor backward one word.                                                                                                                  |
| ^A           | Move cursor to start of line.                                                                                                                   |
| ^E           | Move cursor to end of line.                                                                                                                     |
| ^]char       | Move cursor forward to character <i>char</i> on current line.                                                                                   |
| M- ^]char    | Move cursor backward to character <i>char</i> on current line.                                                                                  |
| ^X^X         | Interchange the cursor and mark.                                                                                                                |
| <i>erase</i> | (User defined erase character as defined by the <code>stty(1)</code> command, usually ^H or #.) Delete previous character.                      |
| ^D           | Delete current character.                                                                                                                       |
| M- d         | Delete current word.                                                                                                                            |
| M- ^H        | (Meta-backspace) Delete previous word.                                                                                                          |

---

|             |                                                                                                                                                                                                                                                                                                                                                |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| M-h         | Delete previous word.                                                                                                                                                                                                                                                                                                                          |
| M-^?        | (Meta-DEL) Delete previous word (if your interrupt character is ^? (DEL, the default) then this command does not work).                                                                                                                                                                                                                        |
| ^T          | Transpose current character with next character in emacs mode. Transpose two previous characters in gmacs mode.                                                                                                                                                                                                                                |
| ^C          | Capitalize current character.                                                                                                                                                                                                                                                                                                                  |
| M-c         | Capitalize current word.                                                                                                                                                                                                                                                                                                                       |
| M-l         | Change the current word to lower case.                                                                                                                                                                                                                                                                                                         |
| ^K          | Delete from the cursor to the end of the line. If preceded by a numerical parameter whose value is less than the current cursor position, then delete from given position up to the cursor. If preceded by a numerical parameter whose value is greater than the current cursor position, then delete from cursor up to given cursor position. |
| ^W          | Kill from the cursor to the mark.                                                                                                                                                                                                                                                                                                              |
| M-p         | Push the region from the cursor to the mark on the stack.                                                                                                                                                                                                                                                                                      |
| <i>kill</i> | (User defined kill character as defined by the <a href="#">stty(1)</a> command, usually ^G or @.) Kill the entire current line. If two <i>kill</i> characters are entered in succession, all kill characters from then on cause a line feed (useful when using paper terminals).                                                               |
| ^Y          | Restore last item removed from line. (Yank item back to the line.)                                                                                                                                                                                                                                                                             |
| ^L          | Line feed and print current line.                                                                                                                                                                                                                                                                                                              |
| ^@          | (null character) Set mark.                                                                                                                                                                                                                                                                                                                     |
| M-space     | (Meta space) Set mark.                                                                                                                                                                                                                                                                                                                         |
| J           | (New line) Execute the current line.                                                                                                                                                                                                                                                                                                           |
| M           | (Return) Execute the current line.                                                                                                                                                                                                                                                                                                             |
| <i>eof</i>  | End-of-file character, normally ^D, is processed as an End-of-file only if the current line is null.                                                                                                                                                                                                                                           |
| ^P          | Fetch previous command. Each time ^P is entered the previous command back in time is accessed. Moves back one line when not on the first line of a multi-line command.                                                                                                                                                                         |
| M-<         | Fetch the least recent (oldest) history line.                                                                                                                                                                                                                                                                                                  |
| M->         | Fetch the most recent (youngest) history line.                                                                                                                                                                                                                                                                                                 |
| ^N          | Fetch next command line. Each time ^N is entered the next command line forward in time is accessed.                                                                                                                                                                                                                                            |

---

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>^Rstring</code>    | Reverse search history for a previous command line containing <i>string</i> . If a parameter of zero is given, the search is forward. <i>string</i> is terminated by a RETURN or NEW LINE. If <i>string</i> is preceded by a ^, the matched line must begin with <i>string</i> . If <i>string</i> is omitted, then the next command line containing the most recent <i>string</i> is accessed. In this case a parameter of zero reverses the direction of the search. |
| <code>^O</code>          | Operate. Execute the current line and fetch the next line relative to current line from the history file.                                                                                                                                                                                                                                                                                                                                                             |
| <code>M-digits</code>    | (Escape) Define numeric parameter, the digits are taken as a parameter to the next command. The commands that accept a parameter are ^F, ^B, <i>erase</i> , ^C, ^D, ^K, ^R, ^P, ^N, ^], M-., M-^], M-_, M-b, M-c, M-d, M-f, M-h, M-l and M-^H.                                                                                                                                                                                                                        |
| <code>M-letter</code>    | Soft-key. Your alias list is searched for an alias by the name <code>_letter</code> and if an alias of this name is defined, its value is inserted on the input queue. The <i>letter</i> must not be one of the above meta-functions.                                                                                                                                                                                                                                 |
| <code>M-[letter</code>   | Soft-key. Your alias list is searched for an alias by the name <code>__letter</code> and if an alias of this name is defined, its value is inserted on the input queue. The can be used to program functions keys on many terminals.                                                                                                                                                                                                                                  |
| <code>M-.</code>         | The last word of the previous command is inserted on the line. If preceded by a numeric parameter, the value of this parameter determines which word to insert rather than the last word.                                                                                                                                                                                                                                                                             |
| <code>M-<u>  </u></code> | Same as <code>M-.</code>                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>M-*</code>         | An asterisk is appended to the end of the word and a file name expansion is attempted.                                                                                                                                                                                                                                                                                                                                                                                |
| <code>M-ESC</code>       | File name completion. Replaces the current word with the longest common prefix of all filenames matching the current word with an asterisk appended. If the match is unique, a / is appended if the file is a directory and a space is appended if the file is not a directory.                                                                                                                                                                                       |
| <code>M=<u>  </u></code> | List files matching current word pattern if an asterisk were appended.                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>^U</code>          | Multiply parameter of next command by 4.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>\</code>           | Escape next character. Editing characters, the user's erase, kill and interrupt (normally ^?) characters can be entered in a command line or in a search string if preceded by a \ . The \ removes the next character's editing features (if any).                                                                                                                                                                                                                    |
| <code>^V</code>          | Display version of the shell.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>M-#</code>         | Insert a # at the beginning of the line and execute it. This causes a comment to be inserted in the history file.                                                                                                                                                                                                                                                                                                                                                     |



|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| vi Editing Mode      | <p>There are two typing modes. Initially, when you enter a command you are in the <i>input</i> mode. To edit, enter <i>control</i> mode by typing ESC (033) and move the cursor to the point needing correction and then insert or delete characters or words as needed. Most control commands accept an optional repeat <i>count</i> prior to the command.</p> <p>When in vi mode on most systems, canonical processing is initially enabled and the command is echoed again if the speed is 1200 baud or greater and it contains any control characters or less than one second has elapsed since the prompt was printed. The ESC character terminates canonical processing for the remainder of the command and the user can then modify the command line. This scheme has the advantages of canonical processing with the type-ahead echoing of raw mode.</p> <p>If the option <code>vi raw</code> is also set, the terminal always have canonical processing disabled. This mode is implicit for systems that do not support two alternate end of line delimiters, and can be helpful for certain terminals.</p> |
| Input Edit Commands  | <p>By default the editor is in input mode.</p> <p><i>erase</i> (User defined erase character as defined by the <code>stty(1)</code> command, usually ^H or #.)<br/>Delete previous character.</p> <p>^W Delete the previous blank separated word.</p> <p>^D Terminate the shell.</p> <p>^V Escape next character. Editing characters and the user's erase or kill characters can be entered in a command line or in a search string if preceded by a ^V. The ^V removes the next character's editing features (if any).</p> <p>\ Escape the next <i>erase</i> or <i>kill</i> character.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Motion Edit Commands | <p>The following commands move the cursor:</p> <p>[<i>count</i>]l Cursor forward (right) one character.</p> <p>[<i>count</i>]w Cursor forward one alpha-numeric word.</p> <p>[<i>count</i>]W Cursor to the beginning of the next word that follows a blank.</p> <p>[<i>count</i>]e Cursor to end of word.</p> <p>[<i>count</i>]E Cursor to end of the current blank delimited word.</p> <p>[<i>count</i>]h Cursor backward (left) one character.</p> <p>[<i>count</i>]b Cursor backward one word.</p> <p>[<i>count</i>]B Cursor to preceding blank separated word.</p> <p>[<i>count</i>]  Cursor to column <i>count</i>.</p> <p>[<i>count</i>]fc Find the next character <i>c</i> in the current line.</p>                                                                                                                                                                                                                                                                                                                                                                                                            |

|                    |                                                                                                                                                                                                     |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [ <i>count</i> ]Fc | Find the previous character <i>c</i> in the current line.                                                                                                                                           |
| [ <i>count</i> ]tc | Equivalent to <i>f</i> followed by <i>h</i> .                                                                                                                                                       |
| [ <i>count</i> ]Tc | Equivalent to <i>F</i> followed by <i>l</i> .                                                                                                                                                       |
| [ <i>count</i> ];  | Repeats <i>count</i> times, the last single character find command, <i>f</i> , <i>F</i> , <i>t</i> , or <i>T</i> .                                                                                  |
| [ <i>count</i> ],  | Reverses the last single character find command <i>count</i> times.                                                                                                                                 |
| 0                  | Cursor to start of line.                                                                                                                                                                            |
| ^                  | Cursor to first non-blank character in line.                                                                                                                                                        |
| \$                 | Cursor to end of line.                                                                                                                                                                              |
| %                  | Moves to balancing ( , ) , { , } , [ , or ] . If cursor is not on one of the above characters, the remainder of the line is searched for the first occurrence of one of the above characters first. |

Search Edit Commands These commands access your command history.

|                   |                                                                                                                                                                                                                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [ <i>count</i> ]k | Fetch previous command. Each time <i>k</i> is entered the previous command back in time is accessed.                                                                                                                                                                               |
| [ <i>count</i> ]– | Equivalent to <i>k</i> .                                                                                                                                                                                                                                                           |
| [ <i>count</i> ]j | Fetch next command. Each time <i>j</i> is entered, the next command forward in time is accessed.                                                                                                                                                                                   |
| [ <i>count</i> ]+ | Equivalent to <i>j</i> .                                                                                                                                                                                                                                                           |
| [ <i>count</i> ]G | The command number <i>count</i> is fetched. The default is the least recent history command.                                                                                                                                                                                       |
| / <i>string</i>   | Search backward through history for a previous command containing <i>string</i> . <i>string</i> is terminated by a RETURN or NEWLINE. If <i>string</i> is preceded by a ^, the matched line must begin with <i>string</i> . If <i>string</i> is NULL, the previous string is used. |
| ? <i>string</i>   | Same as / except that search is in the forward direction.                                                                                                                                                                                                                          |
| n                 | Search for next match of the last pattern to / or ? commands.                                                                                                                                                                                                                      |
| N                 | Search for next match of the last pattern to / or ?, but in reverse direction. Search history for the <i>string</i> entered by the previous / command.                                                                                                                             |

Text Modification Edit Commands These commands modifies the line.

|   |                                                              |
|---|--------------------------------------------------------------|
| a | Enter input mode and enter text after the current character. |
| A | Append text to the end of the line. Equivalent to \$a.       |

---

|                             |                                                                                                                                                                                                                                                                                             |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>[count]cmotion</code> |                                                                                                                                                                                                                                                                                             |
| <code>c[count]motion</code> | Delete current character through the character that <i>motion</i> would move the cursor to and enter input mode. If <i>motion</i> is <code>c</code> , the entire line is deleted and input mode entered.                                                                                    |
| <code>C</code>              | Delete the current character through the end of line and enter input mode. Equivalent to <code>c\$</code> .                                                                                                                                                                                 |
| <code>[count]s</code>       | Delete <i>count</i> characters and enter input mode.                                                                                                                                                                                                                                        |
| <code>S</code>              | Equivalent to <code>cc</code> .                                                                                                                                                                                                                                                             |
| <code>D</code>              | Delete the current character through the end of line. Equivalent to <code>d\$</code> .                                                                                                                                                                                                      |
| <code>[count]dmotion</code> |                                                                                                                                                                                                                                                                                             |
| <code>d[count]motion</code> | Delete current character through the character that <i>motion</i> would move to. If <i>motion</i> is <code>d</code> , the entire line is deleted.                                                                                                                                           |
| <code>i</code>              | Enter input mode and insert text before the current character.                                                                                                                                                                                                                              |
| <code>I</code>              | Insert text before the beginning of the line. Equivalent to <code>0i</code> .                                                                                                                                                                                                               |
| <code>[count]P</code>       | Place the previous text modification before the cursor.                                                                                                                                                                                                                                     |
| <code>[count]p</code>       | Place the previous text modification after the cursor.                                                                                                                                                                                                                                      |
| <code>R</code>              | Enter input mode and replace characters on the screen with characters you type overlay fashion.                                                                                                                                                                                             |
| <code>[count]rc</code>      | Replace the <i>count</i> character(s) starting at the current cursor position with <i>c</i> , and advance the cursor.                                                                                                                                                                       |
| <code>[count]x</code>       | Delete current character.                                                                                                                                                                                                                                                                   |
| <code>[count]X</code>       | Delete preceding character.                                                                                                                                                                                                                                                                 |
| <code>[count].</code>       | Repeat the previous text modification command.                                                                                                                                                                                                                                              |
| <code>[count]~</code>       | Invert the case of the <i>count</i> character(s) starting at the current cursor position and advance the cursor.                                                                                                                                                                            |
| <code>[count]_</code>       | Causes the <i>count</i> word of the previous command to be appended and input mode entered. The last word is used if <i>count</i> is omitted.                                                                                                                                               |
| <code>*</code>              | Causes an <code>*</code> to be appended to the current word and file name generation attempted. If no match is found, it rings the bell. Otherwise, the word is replaced by the matching pattern and input mode is entered.                                                                 |
| <code>\</code>              | Filename completion. Replaces the current word with the longest common prefix of all filenames matching the current word with an asterisk appended. If the match is unique, a <code>/</code> is appended if the file is a directory and a space is appended if the file is not a directory. |

Other Edit Commands    Miscellaneous commands.

[*count*]ymotion

y[*count*]motion    Yank current character through character that *motion* would move the cursor to and puts them into the delete buffer. The text and cursor are unchanged.

Y    Yanks from current position to end of line. Equivalent to y\$.

u    Undo the last text modifying command.

U    Undo all the text modifying commands performed on the line.

[*count*]v    Returns the command `fc -e ${VISUAL:-${EDITOR:-vi}}` *count* in the input buffer. If *count* is omitted, then the current line is used.

^L    Line feed and print current line. Has effect only in control mode.

J    (New line) Execute the current line, regardless of mode.

M    (Return) Execute the current line, regardless of mode.

#    If the first character of the command is a #, then this command deletes this # and each # that follows a newline. Otherwise, sends the line after inserting a # in front of each line in the command. Useful for causing the current line to be inserted in the history as a comment and removing comments from previous comment commands in the history file.

=    List the file names that match the current word if an asterisk were appended it.

@*letter*    Your alias list is searched for an alias by the name *\_letter* and if an alias of this name is defined, its value is inserted on the input queue for processing.

Special Commands    The following *simple-commands* are executed in the shell process. Input/Output redirection is permitted. Unless otherwise indicated, the output is written on file descriptor 1 and the exit status, when there is no syntax error, is 0. Commands that are preceded by one or two \* (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by \*\* that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.

- 
- \* : [ *arg* ... ]  
The command only expands parameters.
- \* . *file* [ *arg* ... ]  
Read the complete *file* then execute the commands. The commands are executed in the current shell environment. The search path specified by PATH is used to find the directory containing *file*. If any arguments *arg* are given, they become the positional parameters. Otherwise the positional parameters are unchanged. The exit status is the exit status of the last command executed.
- \*\* alias [ -tx ] [ *name*[=*value*] ] ...  
alias with no arguments prints the list of aliases in the form *name=value* on standard output. An *alias* is defined for each name whose *value* is given. A trailing space in *value* causes the next word to be checked for alias substitution. The -t flag is used to set and list tracked aliases. The value of a tracked alias is the full pathname corresponding to the given *name*. The value becomes undefined when the value of PATH is reset but the aliases remained tracked. Without the -t flag, for each *name* in the argument list for which no *value* is given, the name and value of the alias is printed. The -x flag is used to set or print *exported aliases*. An *exported alias* is defined for scripts invoked by name. The exit status is non-zero if a *name* is given, but no value, and no alias has been defined for the *name*.
- bg [ %*job*... ]  
This command is only on systems that support job control. Puts each specified *job* into the background. The current job is put in the background if *job* is not specified. See Jobs section above for a description of the format of *job*.
- \* break [ *n* ]  
Exit from the enclosed for, while, until, or select loop, if any. If *n* is specified then break *n* levels. If *n* is greater than the number of enclosing loops, the outermost enclosing loop shall be exited.
- \* continue [ *n* ]  
Resume the next iteration of the enclosed for, while, until, or select loop. If *n* is specified then resume at the *n*-th enclosed loop. If *n* is greater than the number of enclosing loops, the outermost enclosing loop shall be used.
- cd [ -L ] [ -P ] [ *arg* ]  
cd *old new*  
This command can be in either of two forms. In the first form it changes the current directory to *arg*. If *arg* is – the directory is changed to the previous directory. The shell variable HOME is the default *arg*. The environment variable PWD is set to the current directory. If the PWD is changed, the OLDPWD environment variable shall also be changed to the value of the old working directory, that is, the current working directory immediately prior to the call to change directory (cd). The shell variable CDPATH defines the search path for the directory containing *arg*. Alternative directory names are separated by a colon (:). The default path is null (specifying the current directory). The current directory is specified by a null path name, which can appear immediately after the equal sign or between the

colon delimiters anywhere else in the path list. If *arg* begins with a / then the search path is not used. Otherwise, each directory in the path is searched for *arg*. If unsuccessful, *cd* attempts to change directories to the pathname formed by the concatenation of the value of *PWD*, a slash character, and *arg*.

- L Handles the operation dot-dot ( . . ) logically. Symbolic link components are *not* resolved before dot-dot components are processed.
- P Handles the operand dot-dot physically. Symbolic link components *are* resolved before dot-dot components are processed.

If both -L and -P options are specified, the last option to be invoked is used and the other is ignored. If neither -L nor -P is specified, the operand is handled dot-dot logically.

The second form of *cd* substitutes the string *new* for the string *old* in the current directory name, *PWD*, and tries to change to this new directory. The *cd* command cannot be executed by *rksh*.

*command* [-p] [*command\_name*] [argument ...]

*command* [-v | -V] *command\_name*

The *command* utility causes the shell to treat the arguments as a simple command, suppressing the shell function lookup. The -p flag performs the command search using a default value for *PATH* that is guaranteed to find all of the standard utilities. The -v flag writes a string to standard output that indicates the pathname or command that is used by the shell, in the current shell execution environment, to invoke *command\_name*. The -V flag writes a string to standard output that indicates how the name given in the *command\_name* operand is interpreted by the shell, in the current shell execution environment.

*echo* [*arg* ...]

See [echo\(1\)](#) for usage and description.

\* *eval* [*arg* ...]

The arguments are read as input to the shell and the resulting command(s) executed.

\* *exec* [*arg* ...]

If *arg* is given, the command specified by the arguments is executed in place of this shell without creating a new process. Input/output arguments can appear and affect the current process. If no arguments are given the effect of this command is to modify file descriptors as prescribed by the input/output redirection list. In this case, any file descriptor numbers greater than 2 that are opened with this mechanism are closed when invoking another program.

\* *exit* [*n*]

Causes the calling shell or shell script to exit with the exit status specified by *n*. The value is the least significant 8 bits of the specified status. If *n* is omitted then the exit status is that of the last command executed. When *exit* occurs when executing a trap, the last command

refers to the command that executed before the trap was invoked. An EOF also causes the shell to exit except for a shell which has the `ignoreeof` option turned on. See `set`.

```
** export [name[=value]] ...
```

```
** export -p
```

The given *names* are marked for automatic export to the environment of subsequently-executed commands.

When `-p` is specified, `export` writes to the standard output the names and values of all exported variables in the following format:

```
"export %s=%s\n", name, value
```

if *name* is set, and:

```
"export %s\n", name
```

if *name* is unset.

The shell formats the output, including the proper use of quoting, so that it is suitable for reinput to the shell as commands that achieve the same exporting results, except for the following:

1. Read-only variables with values cannot be reset.
2. Variables that were unset at the time they were output are not reset to the unset state if a value is assigned to the variable between the time the state was saved and the time at which the saved output is reinput to the shell.

```
fc [-e ename] [-n l r] [first [last]]
```

```
fc -e - [old=new] [command]
```

```
fc -s [old=new] [command]
```

In the first form, a range of commands from *first* to *last* is selected from the last HISTSIZE commands that were typed at the terminal. The arguments *first* and *last* can be specified as a number or as a string. A string is used to locate the most recent command starting with the given string. A negative number is used as an offset to the current command number. If the `-l` flag is selected, the commands are listed on standard output. Otherwise, the editor program *ename* is invoked on a file containing these keyboard commands. If *ename* is not supplied, then the value of the variable FCEDIT (default `/bin/ed`) is used as the editor. When editing is complete, the edited command(s) is executed. If *last* is not specified then it is set to *first*. If *first* is not specified the default is the previous command for editing and `-16` for listing. The flag `-r` reverses the order of the commands and the flag `-n` suppresses command numbers when listing. In the second form the *command* is re-executed after the substitution *old=new* is performed. If there is not a *command* argument, the most recent command typed at this terminal is executed.

```
fg [%job...]
```

This command is only on systems that support job control. Each *job* specified is brought to the foreground. Otherwise, the current job is brought into the foreground. See “Jobs” section above for a description of the format of *job*.

`getopts optstring name [ arg ... ]`

Checks *arg* for legal options. If *arg* is omitted, the positional parameters are used. An option argument begins with a + or a -. An option not beginning with + or - or the argument - ends the options. *optstring* contains the letters that `getopts` recognizes. If a letter is followed by a :, that option is expected to have an argument. The options can be separated from the argument by blanks.

`getopts` places the next option letter it finds inside variable *name* each time it is invoked with a + prepended when *arg* begins with a +. The index of the next *arg* is stored in `OPTIND`. The option argument, if any, gets stored in `OPTARG`.

A leading : in *optstring* causes `getopts` to store the letter of an invalid option in `OPTARG`, and to set *name* to ? for an unknown option and to : when a required option is missing. Otherwise, `getopts` prints an error message. The exit status is non-zero when there are no more options. See `getoptcvt(1)` for usage and description.

`getopts` supports both traditional single-character short options and long options defined by Sun's Command Line Interface Paradigm (CLIP).

Each long option is an alias for a short option and is specified in parentheses following its equivalent short option. For example, you can specify the long option `file` as an alias for the short option `f` using the following script line:

```
getopts "f(file)" opt
```

Precede long options on the command line with `--` or `++`. In the example above, `--file` on the command line would be the equivalent of `-f`, and `++file` on the command line would be the equivalent of `+f`.

Each short option can have multiple long option equivalents, although this is in violation of the CLIP specification and should be used with caution. You must enclose each long option equivalent parentheses, as follows:

```
getopts "f:(file)(input-file)o:(output-file)"
```

In the above example, both `--file` and `--input-file` are the equivalent of `-f`, and `--output-file` is the equivalent of `-o`.

The variable name is always set to a short option. When a long option is specified on the command line, *name* is set to the short-option equivalent.

`hash [ name ... ]`

`hash [ -r ]`

For each *name*, the location in the search path of the command specified by *name* is determined and remembered by the shell. The `-r` option causes the shell to forget all remembered locations. If no arguments are given, information about remembered commands is presented. *Hits* is the number of times a command has been invoked by the shell process. *Cost* is a measure of the work required to locate a command in the search path. If a command is found in a relative directory in the search path, after changing to that



directory, the stored location of that command is recalculated. Commands for which this is done are indicated by an asterisk (\*) adjacent to the *hits* information. *Cost* is incremented when the recalculation is done.

`jobs [ -lnp ] [ %job ... ]`

Lists information about each given job; or all active jobs if *job* is omitted. The `-l` flag lists process ids in addition to the normal information. The `-n` flag displays only jobs that have stopped or exited since last notified. The `-p` flag causes only the process group to be listed. See "Jobs" section above and `jobs(1)` for a description of the format of *job*.

`kill [ -sig ] %job ...`

`kill [ -sig ] pid ...`

`kill -l`

Sends either the TERM (terminate) signal or the specified signal to the specified jobs or processes. Signals are either given by number or by names (as given in `signal.h(3HEAD)` stripped of the prefix "SIG" with the exception that SIGCHD is named CHLD). If the signal being sent is TERM (terminate) or HUP (hangup), then the job or process is sent a CONT (continue) signal if it is stopped. The argument *job* can be the process id of a process that is not a member of one of the active jobs. See Jobs for a description of the format of *job*. In the second form, `kill -l`, the signal numbers and names are listed.

`let arg...`

Each *arg* is a separate *arithmetic expression* to be evaluated. See the Arithmetic Evaluation section above, for a description of arithmetic expression evaluation.

The exit status is 0 if the value of the last expression is non-zero, and 1 otherwise.

`login argument . . .`

Equivalent to 'exec login *argument*....' See `login(1)` for usage and description.

`* newgrp [ arg ... ]`

Equivalent to `exec /bin/newgrp arg ....`

`print [ -Rnprsu[n ] ] [ arg ... ]`

The shell output mechanism. With no flags or with flag `-o` or `-`, the arguments are printed on standard output as described by `echo(1)`. The exit status is 0, unless the output file is not open for writing.

- `-n` Suppresses NEWLINE from being added to the output.
- `-R | -r` Raw mode. Ignores the escape conventions of `echo`. The `-R` option prints all subsequent arguments and options other than `-n`.
- `-p` Writes the arguments to the pipe of the process spawned with `|&` instead of standard output.
- `-s` Writes the arguments to the history file instead of standard output.
- `-u [ n ]` Specifies a one digit file descriptor unit number *n* on which the output is placed. The default is 1.

`pwd [ -L | -P ]`

Writes to the standard output an absolute pathname of the current working directory, which does not contain the filenames dot (.) or dot-dot (..).

- L If the PWD environment variable contains an absolute pathname of the current directory that does not contain the filenames dot or dot-dot, `pwd` writes this pathname to standard output. Otherwise, the `-L` option behaves like the `-P` option.
- P The absolute pathname written shall not contain filenames that, in the context of the pathname, refer to files of type symbolic link.

If both `-L` and `-P` are specified, the last one applies. If neither `-L` nor `-P` is specified, `pwd` behaves as if `-L` had been specified.

`read [ -prsu[ n ] ] [ name?prompt ] [ name ... ]`

The shell input mechanism. One line is read and is broken up into fields using the characters in IFS as separators. The escape character, (\), is used to remove any special meaning for the next character and for line continuation. In raw mode, `-r`, the \ character is not treated specially. The first field is assigned to the first *name*, the second field to the second *name*, etc., with leftover fields assigned to the last *name*. The `-p` option causes the input line to be taken from the input pipe of a process spawned by the shell using |&. If the `-s` flag is present, the input is saved as a command in the history file. The flag `-u` can be used to specify a one digit file descriptor unit *n* to read from. The file descriptor can be opened with the `exec` special command. The default value of *n* is 0. If *name* is omitted then REPLY is used as the default *name*. The exit status is 0 unless the input file is not open for reading or an EOF is encountered. An EOF with the `-p` option causes cleanup for this process so that another can be spawned. If the first argument contains a ?, the remainder of this word is used as a *prompt* on standard error when the shell is interactive. The exit status is 0 unless an EOF is encountered.

\*\* `readonly [ name[=value] ] ...`

\*\* `readonly -p`

The given *names* are marked `readonly` and these names cannot be changed by subsequent assignment.

When `-p` is specified, `readonly` writes to the standard output the names and values of all read-only variables, in the following format:

```
"readonly %s=%s\n", name, value
```

if *name* is set, and:

```
"readonly $s\n", name
```

if *name* is unset.

The shell formats the output, including the proper use of quoting, so that it is suitable for reinput to the shell as commands that achieve the same value and `readonly` attribute-setting results in a shell execution environment in which:

1. Variables with values set at the time they were output do not have the readonly attribute set.
2. Variables that were unset at the time they were output do not have a value at the time at which the saved output is reinput to the shell.

\* return [ *n* ]

Causes a shell function or ' . ' script to return to the invoking script with the return status specified by *n*. The value is the least significant 8 bits of the specified status. If *n* is omitted then the return status is that of the last command executed. If return is invoked while not in a function or a ' . ' script, then it is the same as an exit.

set [ ±abCefhkmnopstuvx ] [ ±o *option* ] . . . [ ±A *name* ] [ *arg* ... ]

The flags for this command have meaning as follows:

- A Array assignment. Unsets the variable *name* and assigns values sequentially from the list *arg*. If +A is used, the variable *name* is not unset first.
- a All subsequent variables that are defined are automatically exported.
- b Causes the shell to notify the user asynchronously of background job completions. The following message is written to standard error:

```
"[%d]%c %s%s\n", <job-number>, <current>, <status>, \
 whe<job-name>
```

where the fields are as follows:

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <current>    | The character + identifies the job that would be used as a default for the fg or bg utilities. This job can also be specified using the <i>job_id</i> %+ or %%. The character - identifies the job that would become the default if the current default job were to exit; this job can also be specified using the <i>job_id</i> %-. For other jobs, this field is a space character. At most one job can be identified with + and at most one job can be identified with -. If there is any suspended job, then the current job is a suspended job. If there are at least two suspended jobs, then the previous job is also a suspended job. |
| <job-number> | A number that can be used to identify the process group to the wait, fg, bg, and kill utilities. Using these utilities, the job can be identified by prefixing the job number with %.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <status>     | Unspecified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <job-name>   | Unspecified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

When the shell notifies the user a job has been completed, it can remove the job's process ID from the list of those known in the current shell execution environment. Asynchronous notification is not enabled by default.

- C Prevents existing files from being overwritten by the shell's > redirection operator. The >| redirection operator overrides this -noclobber option for an individual file.
- e If a command has a non-zero exit status, executes the ERR trap, if set, and exit. This mode is disabled while reading profiles.
- f Disables file name generation.
- h Each command becomes a tracked alias when first encountered.
- k All variable assignment arguments are placed in the environment for a command, not just those that precede the command name.
- m Background jobs runs in a separate process group and a line prints upon completion. The exit status of background jobs is reported in a completion message. On systems with job control, this flag is turned on automatically for interactive shells.
- n Reads commands and check them for syntax errors, but do not execute them. Ignored for interactive shells.
- o Writes the current option settings to standard output in a format that is suitable for reinput to the shell as commands that achieve the same option settings.
- o The following argument can be one of the following option names:
  - allexport Same as -a.
  - errexit Same as -e.
  - bgnice All background jobs are run at a lower priority. This is the default mode.
  - emacs Puts you in an emacs style in-line editor for command entry.
  - gmacs Puts you in a gmacs style in-line editor for command entry.
  - ignoreeof The shell does not exit onEOF. The command exit must be used.
  - keyword Same as -k.
  - markdirs All directory names resulting from file name generation have a trailing / appended.
  - monitor Same as -m.
  - noclobber Prevents redirection > from truncating existing files. Require >| to truncate a file when turned on. Equivalent to -C.
  - noexec Same as -n.
  - noglob Same as -f.
  - nolog Do not save function definitions in history file.

|                         |                                                                                                                                                                             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>notify</code>     | Equivalent to <code>-b</code> .                                                                                                                                             |
| <code>nounset</code>    | Same as <code>-u</code> .                                                                                                                                                   |
| <code>privileged</code> | Same as <code>-p</code> .                                                                                                                                                   |
| <code>verbose</code>    | Same as <code>-v</code> .                                                                                                                                                   |
| <code>trackall</code>   | Same as <code>-h</code> .                                                                                                                                                   |
| <code>vi</code>         | Puts you in insert mode of a <code>vi</code> style in-line editor until you hit escape character <code>033</code> . This puts you in control mode. A return sends the line. |
| <code>viraw</code>      | Each character is processed as it is typed in <code>vi</code> mode.                                                                                                         |
| <code>xtrace</code>     | Same as <code>-x</code> .                                                                                                                                                   |

If no option name is supplied, the current option settings are printed.

- `-p` Disables processing of the `$HOME/.profile` file and uses the file `/etc/suid_profile` instead of the `ENV` file. This mode is on whenever the effective uid is not equal to the real uid, or when the effective gid is not equal to the real gid. Turning this off causes the effective uid and gid to be set to the real uid and gid.
- `-s` Sorts the positional parameters lexicographically.
- `-t` Exits after reading and executing one command.
- `-u` Treats unset parameters as an error when substituting.
- `-v` Prints shell input lines as they are read.
- `-x` Prints commands and their arguments as they are executed.
- `-` Turns off `-x` and `-v` flags and stops examining arguments for flags.
- `—` Does not change any of the flags. Useful in setting `$1` to a value beginning with `-`. If no arguments follow this flag then the positional parameters are unset.

Using `+` rather than `-` causes these flags to be turned off. These flags can also be used upon invocation of the shell. The current set of flags can be found in `$-`. Unless `-A` is specified, the remaining arguments are positional parameters and are assigned, in order, to `$1 $2 ...`. If no arguments are given, the names and values of all variables are printed on the standard output.

`* shift [ n ]`

The positional parameters from `$n+1 $n+1 . . .` are renamed `$1 . . .`, default `n` is 1.

The parameter `n` can be any arithmetic expression that evaluates to a non-negative number less than or equal to `$#`.

`stop%jobid ...`

`stop pid ...`

`stop` stops the execution of a background job(s) by using its *jobid*, or of any process by using its *pid*. See [ps\(1\)](#).

`suspend`

Stops the execution of the current shell (but not if it is the login shell).

`test expression`

Evaluates conditional expressions. See Conditional Expressions section above and [test\(1\)](#) for usage and description.

`* times`

Prints the accumulated user and system times for the shell and for processes run from the shell.

`* trap [ arg sig ... ]`

*arg* is a command to be read and executed when the shell receives signal(s) *sig*. *arg* is scanned once when the trap is set and once when the trap is taken. *sig* can be specified as a signal number or signal name. `trap` commands are executed in order of signal number. Any attempt to set a trap on a signal number that was ignored on entry to the current shell is ineffective.

If *arg* is `-`, the shell resets each *sig* to the default value. If *arg* is null (`' '`), the shell ignores each specified *sig* if it arises. Otherwise, *arg* is read and executed by the shell when one of the corresponding *sigs* arises. The action of the trap overrides a previous action (either default action or one explicitly set). The value of `$?` after the trap action completes is the value it had before the trap was invoked.

*sig* can be `EXIT`, `0` (equivalent to `EXIT`) or a signal specified using a symbolic name, without the `SIG` prefix, for example, `HUP`, `INT`, `QUIT`, `TERM`. If *sig* is `0` or `EXIT` and the `trap` statement is executed inside the body of a function, then the command *arg* is executed after the function completes. If *sig* is `0` or `EXIT` for a `trap` set outside any function, the command *arg* is executed on exit from the shell. If *sig* is `ERR`, *arg* is executed whenever a command has a non-zero exit status. If *sig* is `DEBUG`, *arg* is executed after each command.

The environment in which the shell executes a trap on `EXIT` is identical to the environment immediately after the last command executed before the trap on `EXIT` was taken.

Each time the trap is invoked, *arg* is processed in a manner equivalent to `eval "$arg"`.

Signals that were ignored on entry to a non-interactive shell cannot be trapped or reset, although no error need be reported when attempting to do so. An interactive shell can reset or catch signals ignored on entry. Traps remain in place for a given shell until explicitly changed with another `trap` command.

When a subshell is entered, traps are set to the default args. This does not imply that the `trap` command cannot be used within the subshell to set new traps.

The `trap` command with no arguments writes to standard output a list of commands associated with each *sig*. The format is:

trap — %s %s ... <arg>, <sig> ...

The shell formats the output, including the proper use of quoting, so that it is suitable for reinput to the shell as commands that achieve the same trapping results. For example:

```
save_traps=$(trap)
. . .
eval "$save_traps"
```

If the trap name or number is invalid, a non-zero exit status is returned. Otherwise, 0 is returned. For both interactive and non-interactive shells, invalid signal names or numbers are not considered a syntax error and do not cause the shell to abort.

Traps are not processed while a job is waiting for a foreground process. Thus, a trap on CHLD won't be executed until the foreground job terminates.

type *name* ...

For each *name*, indicates how it would be interpreted if used as a command name.

\*\* typeset [ ±HLRZfilrtux[*n*] ] [ *name*[=*value*] ] ...

Sets attributes and values for shell variables and functions. When typeset is invoked inside a function, a new instance of the variables *name* is created. The variables *value* and type are restored when the function completes. The following list of attributes can be specified:

- H This flag provides UNIX to host-name file mapping on non-UNIX machines.
- L Left justifies and removes leading blanks from *value*. If *n* is non-zero it defines the width of the field. Otherwise, it is determined by the width of the value of first assignment. When the variable is assigned to, it is filled on the right with blanks or truncated, if necessary, to fit into the field. Leading zeros are removed if the -Z flag is also set. The -R flag is turned off.
- R Right justifies and fills with leading blanks. If *n* is non-zero it defines the width of the field, otherwise it is determined by the width of the value of first assignment. The field is left filled with blanks or truncated from the end if the variable is reassigned. The -L flag is turned off.
- Z Right justifies and fills with leading zeros if the first non-blank character is a digit and the -L flag has not been set. If *n* is non-zero it defines the width of the field. Otherwise, it is determined by the width of the value of first assignment.
- f The names refer to function names rather than variable names. No assignments can be made and the only other valid flags are -t, -u, and -x. The flag -t turns on execution tracing for this function. The flag -u causes this function to be marked undefined. The FPATH variable is searched to find the function definition when the function is referenced. The flag -x allows the function definition to remain in effect across shell procedures invoked by name.
- i Parameter is an integer. This makes arithmetic faster. If *n* is non-zero it defines the output arithmetic base; otherwise, the first assignment determines the output base.

- l All upper-case characters are converted to lower-case. The upper-case flag, -u is turned off.
- r The given *names* are marked readonly and these names cannot be changed by subsequent assignment.
- t Tags the variables. Tags are user definable and have no special meaning to the shell.
- u All lower-case characters are converted to upper-case characters. The lower-case flag, -l is turned off.
- x The given *names* are marked for automatic export to the environment of subsequently-executed commands.

The -i attribute cannot be specified along with -R, -L, -Z, or -f.

Using + rather than - causes these flags to be turned off. If no *name* arguments are given but flags are specified, a list of *names* (and optionally the *values*) of the *variables* which have these flags set is printed. (Using + rather than - keeps the values from being printed.) If no *names* and flags are given, the *names* and *attributes* of all *variables* are printed.

`ulimit [ -HSacdfnstv ] [ limit ]`

Sets or displays a resource limit. The available resources limits are listed in the following section. Many systems do not contain one or more of these limits. The limit for a specified resource is set when *limit* is specified. The value of *limit* can be a number in the unit specified with each resource, or the value `unlimited`. The string `unlimited` requests that the current limit, if any, be removed. The -H and -S flags specify whether the hard limit or the soft limit for the given resource is set. A hard limit cannot be increased once it is set. A soft limit can be increased up to the value of the hard limit. If neither the -H or -S options is specified, the limit applies to both. The current resource limit is printed when *limit* is omitted. In this case, the soft limit is printed unless -H is specified. When more than one resource is specified, the limit name and unit is printed before the value.

- a Lists all of the current resource limits.
- c The number of 512-byte blocks on the size of core dumps.
- d The number of K-bytes on the size of the data area.
- f The number of 512-byte blocks on files written by child processes (files of any size can be read).
- n The number of file descriptors plus 1.
- s The number of K-bytes on the size of the stack area.
- t The number of seconds to be used by each process.
- v The number of K-bytes for virtual memory.

If no option is given, -f is assumed.



`umask [-S] [ mask ]`

The user file-creation mask is set to *mask* (see [umask\(2\)](#)). *mask* can either be an octal number or a symbolic value as described in [chmod\(1\)](#). If a symbolic value is given, the new `umask` value is the complement of the result of applying *mask* to the complement of the previous `umask` value. If *mask* is omitted, the current value of the mask is printed. The `-S` flag produces symbolic output.

`unalias name ...`

`unalias -a`

The aliases given by the list of *names* are removed from the alias list. The `-a` option removes all alias definitions from the current execution environment.

`unset [-f] name ...`

The variables given by the list of *names* are unassigned, that is, their values and attributes are erased. readonly variables cannot be unset. If the `-f`, flag is set, then the names refer to *function* names. Unsetting `ERRNO`, `LINENO`, `MAILCHECK`, `OPTARG`, `OPTIND`, `RANDOM`, `SECONDS`, `TMOUT`, and `_` removes their special meaning even if they are subsequently assigned to.

`*wait [job]`

Waits for the specified *job* and report its termination status. If *job* is not given then all currently active child processes are waited for. The exit status from this command is that of the process waited for. See [Jobs](#) for a description of the format of *job*.

`whence [-pv] name ...`

For each *name*, indicates how it would be interpreted if used as a command name.

The `-v` flag produces a more verbose report.

The `-p` flag does a path search for *name* even if name is an alias, a function, or a reserved word.

## Invocation

If the shell is invoked by [exec\(2\)](#), and the first character of argument zero (`$0`) is `-`, then the shell is assumed to be a login shell and commands are read from `/etc/profile` and then from either `.profile` in the current directory or `$HOME/.profile`, if either file exists. Next, commands are read from the file named by performing parameter substitution on the value of the environment variable `ENV` if the file exists. If the `-s` flag is not present and *arg* is, then a path search is performed on the first *arg* to determine the name of the script to execute. The script *arg* must have read permission and any `setuid` and `setgid` settings are ignored. If the script is not found on the path, *arg* is processed as if it named a builtin command or function. Commands are then read as described as follows. The following flags are interpreted by the shell when it is invoked:

- `-c` Reads commands from the *command\_string* operand. Sets the value of special parameter `0` from the value of the *command\_name* operand and the positional parameters (`$1`, `$2`, and so on) in sequence from the remaining *arg* operands. No commands are read from the standard input.

- s If the `-s` flag is present or if no arguments remain, commands are read from the standard input. Shell output, except for the output of the Special Commands listed above, is written to file descriptor 2.
- i If the `-i` flag is present or if the shell input and output are attached to a terminal (as told by [ioctl\(2\)](#)), then this shell is *interactive*. In this case, `TERM` is ignored (so that `kill 0` does not kill an interactive shell) and `INTR` is caught and ignored (so that `wait` is interruptible). In all cases, `QUIT` is ignored by the shell.
- r If the `-r` flag is present the shell is a restricted shell.

The remaining flags and arguments are described under the `set` command above.

#### rksh Only

`rksh` is used to set up login names and execution environments whose capabilities are more controlled than those of the standard shell. The actions of `rksh` are identical to those of `ksh`, except that the following are disallowed:

- changing directory (see [cd\(1\)](#))
- setting the value of `SHELL`, `ENV`, or `PATH`
- specifying path or command names containing `/`
- redirecting output (`>`, `>|`, `<>`, and `>>`)
- changing group (see [newgrp\(1\)](#)).

The restrictions above are enforced after `.profile` and the `ENV` files are interpreted.

When a command to be executed is found to be a shell procedure, `rksh` invokes `ksh` to execute it. Thus, it is possible to provide to the end-user shell procedures that have access to the full power of the standard shell, while imposing a limited menu of commands; this scheme assumes that the end-user does not have write and execute permissions in the same directory.

The net effect of these rules is that the writer of the `.profile` has complete control over user actions, by performing guaranteed setup actions and leaving the user in an appropriate directory (probably *not* the login directory).

The system administrator often sets up a directory of commands (that is, `/usr/rbin`) that can be safely invoked by `rksh`.

#### 错误

Errors detected by the shell, such as syntax errors, cause the shell to return a non-zero exit status. Otherwise, the shell returns the exit status of the last command executed (see also the `exit` command above). If the shell is being used non-interactively then execution of the shell file is abandoned. Run time errors detected by the shell are reported by printing the command or function name and the error condition. If the line number that the error occurred on is greater than one, then the line number is also printed in square brackets (`[1]`) after the command or function name.

For a non-interactive shell, an error condition encountered by a special built-in or other type of utility causes the shell to write a diagnostic message to standard error and exit as shown in the following table:

| Error                                          | Special Built-in | Other Utilities |
|------------------------------------------------|------------------|-----------------|
| Shell language syntax error                    | exits            | exits           |
| Utility syntax error (option or operand error) | exits            | does not exit   |
| Redirection error                              | exits            | does not exit   |
| Variable assignment error                      | exits            | does not exit   |
| Expansion error                                | exits            | exits           |
| Command not found                              | n/a              | might exit      |
| Dot script not found                           | exits            | n/a             |

An expansion error is one that occurs when the shell expansions are carried out (for example, `!y`, because `!` is not a valid operator). An implementation can treat these as syntax errors if it is able to detect them during tokenization, rather than during expansion.

If any of the errors shown as “might exit” or “exits” occur in a subshell, the subshell exits or might exit with a non-zero status, but the script containing the subshell does not exit because of the error.

In all of the cases shown in the table, an interactive shell writes a diagnostic message to standard error without exiting.

## 用法

See [largefile\(5\)](#) for the description of the behavior of `ksh` and `rksh` when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

## 退出状态

Each command has an exit status that can influence the behavior of other shell commands. The exit status of commands that are not utilities is documented in this section. The exit status of the standard utilities is documented in their respective sections.

If a command is not found, the exit status is 127. If the command name is found, but it is not an executable utility, the exit status is 126. Applications that invoke utilities without using the shell should use these exit status values to report similar errors.

If a command fails during word expansion or redirection, its exit status is greater than zero.

When reporting the exit status with the special parameter `?`, the shell reports the full eight bits of exit status available. The exit status of a command that terminated because it received a signal reported as greater than 128.

文件

```

/etc/profile
/etc/suid_profile
$HOME/.profile
/tmp/sh*
/dev/null

```

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/sunos/bin/ksh,  
/usr/sunos/bin/rksh

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | shell/ksh88     |
| CSI            | Enabled         |

/usr/xpg4/bin/sh

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

另请参见

[cat\(1\)](#), [cd\(1\)](#), [chmod\(1\)](#), [cut\(1\)](#), [echo\(1\)](#), [env\(1\)](#), [getoptcv\(1\)](#), [jobs\(1\)](#), [login\(1\)](#), [newgrp\(1\)](#), [paste\(1\)](#), [pfksh\(1\)](#), [pfexec\(1\)](#), [ps\(1\)](#), [shell\\_builtins\(1\)](#), [stty\(1\)](#), [test\(1\)](#), [vi\(1\)](#), [dup\(2\)](#), [exec\(2\)](#), [fork\(2\)](#), [ioctl\(2\)](#), [lseek\(2\)](#), [pipe\(2\)](#), [ulimit\(2\)](#), [umask\(2\)](#), [rand\(3C\)](#), [signal\(3C\)](#), [signal.h\(3HEAD\)](#), [wait\(3C\)](#), [a.out\(4\)](#), [profile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

Morris I. Bolsky and David G. Korn, *The KornShell Command and Programming Language*, Prentice Hall, 1989.

警告

The use of setuid shell scripts is *strongly* discouraged.

附注

If a command which is a *tracked alias* is executed, and then a command with the same name is installed in a directory in the search path before the directory where the original command was found, the shell continues to exec the original command. Use the `-t` option of the `alias` command to correct this situation.

Some very old shell scripts contain a `^` as a synonym for the pipe character `|`.

Using the `fc` built-in command within a compound command causes the whole command to disappear from the history file.

The built-in command `.file` reads the whole file before any commands are executed. Therefore, `alias` and `unalias` commands in the file does not apply to any functions defined in the file.

When the shell executes a shell script that attempts to execute a non-existent command interpreter, the shell returns an erroneous diagnostic message that the shell script file does not exist.

**引用名** ktutil – Kerberos 密钥表维护实用程序

**用法概要** /usr/bin/ktutil

**描述** ktutil 命令是用于管理密钥表文件中的密钥列表的交互式命令行界面实用程序。您必须先读入密钥表的密钥列表，然后才能对其进行管理。此外，运行 ktutil 命令的用户必须对密钥表具有读取/写入权限。例如，如果密钥表由 root 拥有（通常如此），ktutil 必须作为 root 运行才能拥有适当权限。

**命令**

|                     |                                                            |
|---------------------|------------------------------------------------------------|
| clear_list          |                                                            |
| clear               | 清除当前密钥列表。                                                  |
| read_kt_file        |                                                            |
| rkt file            | 将密钥表读取到当前密钥列表。必须指定要读取的密钥表文件。                               |
| write_kt_file       |                                                            |
| wkt file            | 将当前密钥列表写入密钥表文件。必须制定要写入的密钥表文件。如果密钥表文件已存在，当前密钥列表会附加到现有密钥表文件。 |
| add_entry number    |                                                            |
| addent number       | 添加一项到当前密钥列表。使用密钥列表槽号指定项。                                   |
| delete_entry number |                                                            |
| delent number       | 从当前密钥列表删除项。使用密钥列表槽号指定项。                                    |
| list                |                                                            |
| l                   | 列出当前密钥列表。                                                  |
| list_request        |                                                            |
| lr                  | 列出可用请求（命令）。                                                |
| quit                |                                                            |
| exit                |                                                            |
| q                   | 退出实用程序。                                                    |

**示例** 示例1 从文件中删除主体

以下示例从 /etc/krb5/krb5.keytab 文件中删除 host/denver@ACME.com 主体。请注意，如果您要删除现有密钥表中的项，必须首先将密钥列表写入临时密钥表，然后使用该临时密钥表覆盖现有密钥表。这是因为 wkt 命令实际上会将当前密钥列表附加到现有密钥表，所以您不能使用它来覆盖密钥表。

```
example# /usr/krb5/bin/ktutil
ktutil: rkt /etc/krb5/krb5.keytab
ktutil: list
slot KVNO Principal

1 8 host/vail@ACME.COM
```

示例1 从文件中删除主体 (续)

```

2 5 host/denver@ACME.COM
 ktutil:delent 2
 ktutil:l
slot KVNO Principal

1 8 host/vail@ACME.COM
 ktutil:wkt /tmp/krb5.keytab
 ktutil:q
example# mv /tmp/krb5.keytab /etc/krb5/krb5.keytab

```

文件  
属性

/etc/krb5/krb5.keytab Kerberos 客户机的密钥表文件

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值                         |
|-------|-----------------------------|
| 可用性   | service/security/kerberos-5 |
| 接口稳定性 | 请参见下文。                      |

命令参数是 "Committed" (已确定)。命令输出是 "Uncommitted" (未确定)。

另请参见

[kadmin\(1M\)](#)、[k5srvutil\(1M\)](#)、[attributes\(5\)](#)、[kerberos\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | kvno – 输出 Kerberos 主体的主要版本号                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 用法概要 | kvno [-h] [-P] [-q] [-c <i>ccache</i> ] [-e <i>etype</i> ] [-S <i>sname</i> ]<br>[-U <i>for_user</i> ] <i>princ_host</i> [ <i>princ_host</i> ...]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 描述   | kvno 用于获取指定 Kerberos 主体的服务票证，并输出各个 Kerberos 主体的主要版本号。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 选项   | 支持以下选项： <ul style="list-style-type: none"> <li>-c <i>ccache</i>      指定要使用的非缺省凭证高速缓存的名称。</li> <li>-e <i>etype</i>      指定要为命令行上指定的所有服务的会话密钥请求的 <i>etype</i>。<br/>这在某些向后兼容性情形下非常有用。</li> <li>-h              输出用法语句并退出。</li> <li>-P              指定 <i>princ_host</i> 参数将被视为服务（应当使用约束委托获取其凭证）。<br/>此选项仅在与协议转换一起使用时才有效。</li> <li>-q              禁止输出。</li> <li>-S <i>sname</i>      指定使用 <code>krb5_sname_to_principal()</code> 来构建主体名称。如果指定此标志，<i>princ_host</i> 参数将被解释为主机名（而不是主体名称），<i>sname</i> 将被解释为服务名称。</li> <li>-U <i>for_user</i>    指定使用协议转换 (S4U2Self) 来代表 <i>for_user</i> 获取票证。<br/>如果未请求约束委托，则服务名称必须与凭证高速缓存客户端主体匹配。</li> </ul> |
| 操作数  | 支持下列操作数： <ul style="list-style-type: none"> <li><i>princ_host</i>    如果指定了 -s 选项，则指定要获取其主要版本号的目标主体的主体名称或主机名。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 环境变量 | 有关影响 kvno 执行的以下环境变量的描述，请参见 <a href="#">environ(5)</a> ：LANG、LC_ALL、LC_CTYPE、LC_MESSAGES 和 NLSPATH。<br>KRB5CCNAME    凭证（票证）高速缓存的位置。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 文件   | /tmp/krb5cc_ <i>uid</i> 凭证高速缓存的缺省位置。 <i>uid</i> 是用户的十进制 UID。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 属性   | 有关下列属性的说明，请参见 <a href="#">attributes(5)</a> ：                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

| 属性类型 | 属性值                         |
|------|-----------------------------|
| 可用性  | service/security/kerberos-5 |



---

| 属性类型  | 属性值             |
|-------|-----------------|
| 接口稳定性 | Committed (已确定) |

另请参见

[kdestroy\(1\)](#)、[kinit\(1\)](#)、[attributes\(5\)](#)、[krb5envvar\(5\)](#)

**引用名**           lari – 运行时接口的链接分析

**用法概要**        lari [-bCDsVv] [-a | -i | -o] *file* | *directory*...

lari [-CDosv] [-m [-d *mapdir*]] *file*

**描述**            lari 实用程序分析动态 ELF 目标文件的接口要求。有两种基本操作模式可用。第一种模式显示运行时接口信息。第二种模式生成接口定义。

动态目标文件提供符号定义，表示目标文件为外部使用者提供的接口。在运行时，会建立从一个目标文件的符号引用到另一个目标文件的符号定义的绑定。lari 分析指定目标文件的接口定义和运行时绑定。

显示运行时接口信息时，lari 可分析许多文件和/或目录。lari 分析在命令行上指定的每个 *file*。lari 以递归方式向下派生到在命令行上指定的每个 *directory* 中，处理找到的每个文件。

生成接口定义时，lari 只能处理在命令行上指定的单个 *file*。

如果不使用 -D 选项，lari 可通过使用 [ldd\(1\)](#) 将文件作为动态 ELF 目标文件进行处理。此处理使用以下选项：

```
-r and -e LD_DEBUG=files,bindings,detail
```

这些选项提供有关作为载入目标文件的一部分建立的所有绑定的信息。请注意，通过使用 [ldd](#)，将不会执行指定目标文件，因此不会发生用户控制的目标文件装入（例如，通过使用 [dlopen\(3C\)](#)）。要捕获来自执行进程的所有绑定信息，可直接将以下环境变量传递到运行时链接程序 [ld.so.1\(1\)](#)：

```
LD_DEBUG=files,bindings,detail LD_DEBUG_OUTPUT=lari.dbg \
LD_BIND_NOW=yes
```

lari 可使用 -D 选项处理所生成的调试输出 *lari.dbg.pid*。**注意：**lari 尝试分析已使用调试输出中定义的路径名进行处理的每个目标文件。因此，lari 必须可访问每个目标文件，才能提供完整和准确的分析。调试输出文件必须在 C 语言环境中生成。

显示接口信息时，lari 将分析已处理动态目标文件的接口，而且在缺省情况下，会显示所有需要关注的信息。请参见“扩展描述”部分的“需要关注的信息”。显示的信息还适合管道传输到其他工具。此功能可帮助开发者分析进程绑定或调试复杂绑定方案。

lari 的接口定义生产操作可用于完善所处理的动态目标文件的接口要求。创建动态目标文件时，您应定义一个显式、标明版本号接口。此定义控制外部用户可使用的符号定义。此外，此定义经常会减少目标文件的整体运行时执行成本。在链接编辑器创建目标文件时，可使用 -M 选项以及相关映射文件的指令将接口定义分配给此目标文件。有关将映射文件用于版本化目标文件的更多详细信息，请参见《[链接程序和库指南](#)》。lari 可创建这些映射文件的初始版本。

## 选项

支持以下选项。

- a 显示已分析目标文件的所有接口信息。**注意**：该选项的输出会非常多，但经常用于管道传输到其他分析工具。
- b 将接口信息限制为已显式绑定的那些符号。**注意**：可能已从定义目标文件内绑定到定义为受保护的符号。此绑定会在链接编辑时间得到满足，因此对运行时环境是不可见的。受保护符号会与此选项一起显示。
- c 取消改编 C++ 符号名。此选项可用于增加运行时接口信息。生成接口定义时，取消改编的名称会作为注释添加到**映射文件**。
- d *mapdir* 定义目录 *mapdir*，会在其中创建**映射文件**。缺省情况下会使用当前的工作目录。
- D 将任何输入文件解释为调试信息而不是动态目标文件。
- i 显示需要关注的接口绑定信息。如果没有提供其他输出控制选项，则此模式为缺省模式。请参见“扩展描述”部分的“需要关注的信息”。
- m 为处理的每个动态目标文件创建**映射文件**。这些**映射文件**会反映正在处理的输入文件所需的每个目标文件的接口要求。
- o 将接口信息限制为被视为系统开销的那些符号。创建**映射文件**时，会将任何系统开销符号列出为局部符号。请参见“扩展描述”部分的“系统开销信息”。
- s 保存从 `ldd(1)` 生成的绑定信息以便进一步分析。请参见“文件”部分。
- V 附加需要关注的符号的可见性。可使用此选项标识定义为**单件**或定义为**受保护的**符号。
- v 忽略任何已标明版本号的目标文件。标明版本号的目标文件已定义其接口，但会增加显示的接口信息。例如，标明版本号的共享目标文件可能显示某个特定进程的系统开销符号。共享目标文件通常旨在供多个进程使用，因此这些目标文件提供的接口可扩展超出任何一个进程的要求。因此，`-v` 选项可在显示接口信息时减少无用数据。

从 `lari` 生成的运行时接口信息具有以下格式：

```
[information]: symbol-name [demangled-name]: object-name
```

每一行描述接口符号（即 *symbol-name*）以及目标文件（即 *object-name*），在该目标文件中定义符号。如果符号代表一个函数，符号名称之后会带有 `()`。如果符号代表一个数据目标文件，符号名称之后会带有符号大小，括在 `[]` 中。如果使用 `-c` 选项，符号名称会附带符号取消改编的名称，即 *demangled-name*。信息字段提供下列一个或多个标记，描述符号的使用：

- cnt:bnd* 两个十进制值表示符号计数 *cnt* 以及到此目标文件的绑定数 *bnd*。符号计数是指在分析的目标文件中已发现此符号定义的出现次数。大于 1 的计数表示符号定义的多个实例。绑定数表示已由运行时链接程序绑定到此符号定义的目标文件数。
- E 已从外部目标文件绑定到此符号定义。
- S 已从相同目标文件绑定到此符号定义。
- D 已直接绑定到此符号定义。
- I 此符号定义为插入项而提供。将自身显式标识为插入项的目标文件会将所有全局符号定义为插入项。请参见 [ld\(1\)](#) 的 `-z interpose` 选项以及 [ld.so.1\(1\)](#) 的 `LD_PRELOAD` 变量。通过使用 `插入映射文件` 指令可将动态可执行文件中的单个符号定义为插入项。
- C 此符号定义是复制重定位的引用数据。
- F 此符号定义位于 `filtee` 中。
- P 此符号定义为受保护。此符号可能具有来自声明此符号所在目标文件的内部绑定。其他符号定义不能插入具有该属性的任何内部绑定。
- A 此符号定义是动态可执行文件中过程链接表项的地址。
- U 此符号查找是源自用户请求，例如 `dlsym(3C)`。
- R 此符号定义充当过滤器，并为重定向到过滤器而提供。
- r 符号搜索过程中的某个时刻拒绝了到此符号的绑定。当直接绑定请求发现某个符号已标记为阻止直接绑定时会发生拒绝。在此情况下，会使用缺省搜索模型重复符号搜索。绑定仍然可以解析为原始、被拒绝的符号。非缺省符号搜索发现某个符号被标识为 **单件** 时，也可能发生拒绝。同样，使用缺省搜索模型重复符号搜索。
- N 此符号定义将显式禁止直接绑定到定义。

有关这些符号分类的更多详细信息，请参见《[链接程序和库指南](#)》。

## 扩展描述

### 需要关注的信息

缺省情况下，或专门使用 `-i` 选项，`lari` 可过滤任何运行时接口信息以显示需要关注的事件。执行该过滤操作主要是为了减少大型应用程序可能生成的信息量。此外，此信息可作为调试复杂绑定情况的焦点，并且经常能突出显示存在问题的区域。但是，为任何特定应用程序分类哪些信息是需要关注的，是一门不精确的科学。您仍然可自由使用 `-a` 选项并为正调查的应用程序所独有的事件搜索绑定信息。

发现需要关注的符号定义时，会输出相同符号的所有其他定义。

需要关注的接口信息的焦点在于存在一个符号的多个定义。在这种情况下，一个符号通常会插入一个或多个其他符号定义。一个定义的绑定计数 *bnd* 不为零、而所有其他

定义的绑定计数都为零时，可以看到这一插入操作。从编译环境或链接环境产生的插入操作不会被描述为需要关注。这些插入事件的示例包括复制重定位 ([C]) 以及绑定到过程链接地址 ([A])。

插入通常是需要的。目的在于过载或替换来自共享目标文件的符号定义。使用 `ld(1)` 的 `-z interpose` 选项，可以显式标记 ([I]) 插入目标文件。这些目标文件可安全插入符号，无论这些目标文件在进程中是以什么顺序装入的。但是，使用非显式插入时请务必谨慎，因为这种插入与构成进程的目标文件的装入顺序有关。

用户创建的、多重定义的符号从 `lari` 输出为需要关注的符号。在此示例中，存在 `interpose1()` 的两个定义，但只会引用 `main` 中的定义。

```
[2:1E]: interpose1(): ./main
[2:0]: interpose1(): ./libA.so
```

插入也可能是由异常符号名称冲突导致的不需要的和意外的事件。这种插入的症状可能是虽然您知道存在对某个函数的引用，但是从来不会调用该函数。这种情况可标识为多重定义符号，如上一示例中所述。但是，当某个目标文件同时定义和引用特定符号时，通常会遇到更意外的情况。

这种情况的一个示例为两个动态目标文件定义和引用相同函数 `interpose2()`。对该符号的任何引用都会绑定到使用该进程装入的第一个动态目标文件。在这种情况下，目标文件 `libA.so` 中 `interpose2()` 的定义会插入目标文件 `libB.so` 中的 `interpose2()` 的定义并将其隐藏。`lari` 的输出可能为：

```
[2:2E5]: interpose2(): ./libA.so
[2:0]: interpose2(): ./libB.so
```

还可单独绑定到多重定义的符号。直接绑定生效 ([D]) 或由于符号具有受保护可见性 ([P]) 时可使用单独绑定。尽管可显式建立单独绑定，但仍可能存在异常或意外的实例。直接绑定符号和具有受保护可见性的符号会输出为需要关注的信息。

## 系统开销信息

使用 `-o` 选项时，`lari` 会显示可能会被视为系统开销的符号定义。

没有引用的全局符号会被视为系统开销。目标文件中提供的符号信息会不必要地增加目标文件的文本段大小。此外，符号信息会增加在运行时搜索目标文件内其他符号引用所需的处理操作。

只从相同目标文件引用的全局符号具有相同的特征。对符号引用进行运行时搜索会导致绑定到造成引用的相同目标文件，这是一项额外系统开销。

这两个符号定义都是通过定义目标文件接口降低到局部范围的候选。链接编辑器创建文件时，可使用 `-M` 选项以及相关联的**映射文件**指令将接口定义分配给此文件。有关**映射文件**的更多详细信息，请参见《[链接程序和库指南](#)》。将 `lari` 与 `-m` 选项一起使用可创建这些**映射文件**的初始版本。

如果 `lari` 用于生成映射文件，标明版本号的共享目标文件将创建映射文件，指示必须将它们系统开销符号降低为局部。此模型允许 `lari` 生成映射文件以便与现有接口定义相比较。创建映射文件时使用 `-v` 选项忽略标明版本号的共享目标文件。

复制重定位也被视为系统开销，而且一般是应该避免的。被复制数据的大小是其接口的定义。此定义会限制在定义数据的较新版本共享目标文件中更改数据大小的能力。可通过使用功能接口引用数据来避免这种限制以及处理复制重定位的成本。对于复制重定位，`lari` 的输出可能为：

```
[2:1EC]: __iob[0x140]: ./main
[2:0]: __iob[0x140]: ./libA.so.1
```

请注意，由于与系统库的历史编程交互，会存在多个小型复制重定位，例如上一示例中使用的 `__iob`。

系统开销信息的另一个示例为动态目标文件绑定到动态可执行文件的过程链接表项。如果动态可执行文件引用外部函数，会创建过程链接表项。此结构允许将引用绑定延迟到实际进行函数调用。如果动态目标文件采用同一被引用函数的地址，动态目标文件会绑定到动态可执行文件过程链接表项。这种类型事件的示例显示以下内容：

```
[2:1EA]: foo(): ./main
[2:1E]: foo(): ./libA.so
```

少量的这种类型绑定一般不需要担心。但是，大量这样的绑定（或许是由于转移表编程技术所导致）则会增加启动系统开销。这类地址重定位绑定要求在应用程序启动时进行重定位处理，而不是在直接调用函数时使用延迟重定位处理。使用此地址还需要在运行时进行间接操作。

## 示例

示例 1 分析多绑定的情况

以下示例显示对于一个存在多个符号定义的进程的分析。共享目标文件 `libX.so` 和 `libY.so` 都调用函数 `interpose()`。此函数同时存在于应用程序 `main` 和共享目标文件 `libA.so` 中。由于插入，两个引用都会绑定到 `main` 中 `interpose()` 的定义。

共享目标文件 `libX.so` 和 `libY.so` 都调用函数 `foo()`。此函数存在于应用程序 `main` 和共享目标文件 `libA.so`、`libX.so` 与 `libY.so` 中。由于 `libX.so` 和 `libY.so` 两者都是在启用直接绑定的情况下生成的，所以每个目标文件都会绑定到自己的定义。

```
example% lari ./main
[3:0]: foo(): ./libA.so
[3:15D]: foo(): ./libX.so
[3:15D]: foo(): ./libY.so
[2:0]: interpose(): ./libA.so
[2:2EP]: interpose(): ./main
```

要更彻底分析绑定信息，可保存绑定数据以便进一步检查。例如，上一个输出指示函数 `interpose()` 是从 `main` 外部的两个目标文件中调用的。对绑定输出的检查显示对此函数的引用源自何处。

### 示例1 分析多绑定的情况 (续)

```
example% lari -s ./main
lari: ./main: bindings information saved as: /usr/tmp/lari.dbg.main
.....
example% fgrep foo /usr/tmp/lari.dbg.main
binding file=./libX.so to file=./main: symbol 'interpose'
binding file=./libY.so to file=./main: symbol 'interpose'
```

**注意：**绑定输出一般比此处显示信息更多，因为输出还伴随有进程标识符、地址和其他绑定信息。

### 示例2 生成接口定义

以下示例会为应用程序及其相关性创建接口定义，同时忽略任何标明版本号 of 的系统库。应用程序 main 会对 foo.so 中的接口 one()、two() 和 three() 进行引用。

```
example% lari -omv ./main
example% cat mapfile-foo.so
#
Interface Definition mapfile for:
Dynamic Object: ./foo.so
Process: ./main
#
foo.so {
 global:
 one;
 three;
 two;
 local:
 _one;
 _three;
 _two;
 *;
};
```

文件  
属性

`$TMPDIR/lari.dbg.file` `ldd(1)` 生成的绑定输出。

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值                                |
|-------|------------------------------------|
| 可用性   | developer/base-developer-utilities |
| 接口稳定性 | 请参见下文。                             |

人类可读的输出是 "Uncommitted" (未确定)。选项为 "Committed" (已确定)。

另请参见

[ld\(1\)](#)、[ldd\(1\)](#)、[ld.so.1\(1\)](#)、[dlopen\(3C\)](#)、[dlsym\(3C\)](#)、[attributes\(5\)](#)

《链接程序和库指南》



**引用名** last – display login and logout information about users and terminals

**用法概要** last [-a] [-n *number* | -*number*] [-f *filename*]  
[*name* | *tty*]. . .

**描述** The `last` command looks in the `/var/adm/wtmpx` file, which records all logins and logouts, for information about a user, a terminal, or any group of users and terminals. Arguments specify names of users or terminals of interest. If multiple arguments are given, the information applicable to any of the arguments is printed. For example, `last root console` lists all of root's sessions, as well as all sessions on the console terminal. `last` displays the sessions of the specified users and terminals, most recent first, indicating the times at which the session began, the duration of the session, and the terminal on which the session took place. `last` also indicates whether the session is continuing or was cut short by a reboot.

The pseudo-user `reboot` logs in when the system is shutdown and when it reboots. Thus,

```
last reboot
```

gives an approximate record of when the operating system instance was shutdown and when it rebooted. This can be used to calculate the availability of the operating system over time.

`last` with no arguments displays a record of all logins and logouts, in reverse order.

If `last` is interrupted, it indicates how far the search has progressed in `/var/adm/wtmpx`. If interrupted with a quit signal (generated by a CTRL-`\`), `last` indicates how far the search has progressed, and then continues the search.

**选项** The following options are supported:

- a Displays the hostname in the last column.
- f *filename* Uses *filename* as the name of the accounting file instead of `/var/adm/wtmpx`.
- n *number* | -*number* Limits the number of entries displayed to that specified by *number*. These options are identical; the -*number* option is provided as a transition tool only and is removed in future releases.

**环境变量** Date and time format is based on locale specified by the `LC_ALL`, `LC_TIME`, or `LANG` environments, in that order of priority.

**文件** `/var/adm/wtmpx` accounting file

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

另请参见

[utmpx\(4\)](#), [attributes\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | lastcomm – display the last commands executed, in reverse order                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 用法概要 | lastcomm [-f <i>file</i> ] [-x] [ <i>command-name</i> ] ... [ <i>user-name</i> ] ...<br>[ <i>terminal-name</i> ] ...                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 描述   | <p>The lastcomm command gives information on previously executed commands. lastcomm with no arguments displays information about all the commands recorded during the current accounting file's lifetime. If called with arguments, lastcomm only displays accounting entries with a matching <i>command-name</i>, <i>user-name</i>, or <i>terminal-name</i>. If extended process accounting is active (see <a href="#">acctadm(1M)</a>) and is recording the appropriate data items, lastcomm attempts to take data from the current extended process accounting file. If standard process accounting is active, lastcomm takes data from the current standard accounting file (see <a href="#">acct(2)</a>).</p> <p>If <i>terminal-name</i> is '-', there was no controlling TTY for the process. The process was probably executed during boot time. If <i>terminal-name</i> is '??', the controlling TTY could not be decoded into a printable name.</p> <p>For each process entry, lastcomm displays the following items of information:</p> <ul style="list-style-type: none"> <li>▪ The command name under which the process was called.</li> <li>▪ One or more flags indicating special information about the process. The flags have the following meanings: <ul style="list-style-type: none"> <li>F The process performed a fork but not an exec.</li> <li>S The process ran as a set-user-id program.</li> </ul> </li> <li>▪ The name of the user who ran the process.</li> <li>▪ The terminal which the user was logged in on at the time (if applicable).</li> <li>▪ The amount of CPU time used by the process (in seconds).</li> <li>▪ The date and time the process exited.</li> </ul> |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-f <i>file</i> Uses <i>file</i> as the source of accounting data. <i>file</i> may be either an extended process accounting file or a standard process accounting file.</li> <li>-x Uses the currently active extended process accounting file. If extended processing accounting is inactive, no output will be produced.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 示例   | <p>示例 1 Listing executions of named commands</p> <p>The command</p> <pre>example% lastcomm a.out root term/01</pre> <p>produces a listing of all the executions of commands named a.out by user root while using the terminal term/01.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

示例 2 Listing all user commands

The command

example% **lastcomm root**

produces a listing of all the commands executed by user root.

文件

/var/adm/pacct            standard accounting file

/var/adm/exacct/proc    extended accounting file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

另请参见

[last\(1\)](#), [acctadm\(1M\)](#), [acct\(2\)](#), [acct.h\(3HEAD\)](#), [core\(4\)](#), [attributes\(5\)](#)

## 引用名

ld – 目标文件的链接编辑器

## 用法概要

```
ld [-32 | -64] [-Bdirect | nodirect] [-B dynamic | static]
[-B eliminate[=mode] | local[=mode]] [-B reduce] [-c name]
[-C] [-D [!]token1, [!]token2, ...] [-e epsym]
[-f name | -F name] [-G] [-h name] [-i] [-l x] [-L path]
[-m] [-M mapfile] [-N string] [-o outfile] [-p auditlib]
[-P auditlib] [-Q y | n] [-r] [-R path] [-s]
[-S supportlib] [-t] [-u symname] [-V] [-Y P,dirlist]
[-z allexttract | defaultextract | weakextract]
[-z altexec64] [-z aslr[=mode]] [-z ancillary[=outfile]]
[-z assert-deflib[=libname]] [-z deferred | nodeferred]
[-z defs | nodefs] [-z direct | nodirect]
[-z discard-unused=item1, item2, ...] [-z endfiltee]
[-z fatal-warnings | nofatal-warnings] [-z finiarray=function]
[-z globalaudit] [-z guidance[=item1, item2, ...]] [-z help]
[-z ignore | record] [-z initempty=function] [-z initfirst]
[-z interpose] [-z lazyload | nolazyload]
[-z ld32=arg1, arg2, ...] [-z ld64=arg1, arg2, ...]
[-z loadfltr] [-z mapfile-add=name] [-z muldefs]
[-z nocompstrtab] [-z nodefaultlib] [-z nodelete]
[-z nodlopen] [-z nodump] [-z noldynsym] [-z nopartial]
[-z origin] [-z parent=object] [-z preinitarray=function]
[-z relaxreloc] [-z rescanner] [-z rescanner-start ...-z rescanner-end]
[-z strip-class=[!]class1, [!]class2, ...] [-z stub]
[-z symbolcap] [-z target=sparc|x86]
[-z text | textwarn | textoff] [-z verbose] [-z wrap=symbol]
filename ...
```

## 描述

链接编辑器 `ld` 通过解析对符号定义的符号引用以及执行重定位来合并可重定位的目标文件。在所有情况下，`ld` 的缺省输出位于 `a.out` 文件中。请参见“附注”部分。

`ld` 有许多选项。“用法概要”部分中定义了与现代编程实践有关的选项，手册页后面的几个部分对这些选项进行了介绍。其他选项不太常用，“辅助选项”部分对这些选项进行了介绍。

`ld` 以两种模式之一运行：动态或静态。动态模式是缺省模式，但该模式可由 `-r` 或 `-d` 选项控制。在动态模式下，将合并作为参数提供的可重定位目标文件以生成动态可执行文件。此文件在运行时与作为参数提供的任何共享目标文件合并在一起。如果指定了 `-G` 选项，将合并可重定位目标文件以生成共享目标文件。此动态链接环境紧密结合了链接编辑器和运行时链接程序 `ld.so.1(1)` 的工作。这两个实用程序及其相关的技术和实用程序详细记录在《链接程序和库指南》中。

在静态模式下，将通过 `-r` 选项来合并作为参数提供的多个可重定位目标文件，以生成一个可重定位的目标文件。现代版本的 Oracle Solaris 操作系统不支持创建静态可执行文件。请参见 `Static Executables`。本部分中的所有讨论均假定使用 `-r` 选项进行动态链接或静态链接。

如果任何参数是库，当在命令行中遇到该库时，缺省情况下，`ld` 将仅搜索该库一次。库可以是共享目标文件也可以是可重定位归档文件。请参见 [ar.h\(3HEAD\)](#)。

共享目标文件由以前通过链接编辑一个或多个输入文件而生成的不可分割的整个单元组成。链接编辑器处理共享目标文件时，共享目标文件的所有内容将成为生成的输出文件映像的逻辑部分。在链接编辑期间，实际上不会复制共享目标文件，因为其实际包含将延迟到进程执行。包含此逻辑部分意味着，链接编辑过程可以使用在共享目标文件中定义的所有符号项。请参见《[链接程序和库指南](#)》中的第 4 章“共享目标文件”。

对于归档库，`ld` 仅装入定义未解析的外部引用的归档成员。`ld` 按顺序搜索归档库的符号表以解析可满足归档成员的外部引用。此搜索操作将重复执行，直到不存在可由归档解析的外部引用。因此，库中成员的顺序对功能没有影响，除非存在定义相同外部符号的多个归档成员。具有相互依赖关系的归档库可能需要多个命令行定义或使用 `-z rescanner` 选项之一。请参见《[链接程序和库指南](#)》中的“归档处理”。

`ld` 是交叉链接编辑器，能够链接 SPARC 或 x86 目标的 32 位目标文件或 64 位目标文件。`ld` 使用 ELF 类和命令行上第一个可重定位目标文件的机器类型来控制要运行的模式。不允许混用 32 位目标文件和 64 位目标文件。同样，只允许单个计算机类型的目标文件。请参见 `-32`、`-64` 和 `-z target` 选项以及 `LD_NOEXEC_64` 环境变量。

#### 静态可执行文件

许多发行版都建议不要创建静态可执行文件。实际上，Solaris 中从未提供过 64 位系统归档库。因为静态可执行文件是基于系统归档库生成的，所以这种可执行文件包含关于系统实现的详细信息。该自包含特性有许多缺点：

- 静态可执行文件无法利用以共享目标文件形式发布的系统修补程序。因此，必须重新生成静态可执行文件，才能利用众多的系统改进功能。
- 这种可执行文件是否能够在未来的发行版上运行可能会受到影响。
- 系统实现详细信息的重复会对系统性能造成负面影响。

从 Oracle Solaris 10 开始，不再提供 32 位系统归档库。如果没有这些库（特别是 `libc.a`），即没有专门的系统知识，将无法再创建静态可执行文件。但是，`ld` 处理静态链接选项以及归档库的功能保持不变。

#### 选项

支持以下选项。

`-32` | `-64`

创建 32 位或 64 位目标文件。

缺省情况下，要生成的目标文件的类根据从命令行处理的第一个 ELF 目标文件来确定。如果未指定任何目标文件，类由从命令行处理的第一个归档文件中遇到的第一个目标文件确定。如果没有目标文件或归档文件，链接编辑器将创建 32 位目标文件。

仅从 `mapfile` 创建 64 位目标文件时才需要 `-64` 选项。

-32 或 -64 选项还可用在下面的罕见情况中：完全从同时包含 32 和 64 位目标文件的归档文件链接。如果归档文件中的第一个目标文件不是需要创建的目标文件的类，-32 或 -64 选项可用来指向链接编辑器。

#### **-B direct | nodirect**

这些选项控制是否直接绑定。-B direct 通过记录符号引用与提供符号定义的依赖项之间的关系来建立直接绑定信息。此外，可以在符号引用与要创建的目标文件中的关联定义之间建立直接绑定信息。运行时链接程序使用此信息直接在关联目标文件中搜索符号而不是执行缺省符号搜索。

只能对使用链接编辑指定的依赖项建立直接绑定信息。因此，还应添加 -z defs 选项。希望在直接绑定环境中插入符号的目标文件应使用 -z interpose 选项将自身标识为插入项。使用 -B direct 还会对所有依赖项启用 -z lazyload。

-B nodirect 选项禁止直接绑定到要创建的目标文件所提供的接口。通过指定 -z direct 选项，可以将要创建的目标文件继续直接绑定到外部接口。请参见《链接程序和库指南》中的第 9 章“直接绑定”。

#### **-B dynamic | static**

这些选项用于控制库包含。-B dynamic 允许 -l 选项库搜索扩展至共享目标文件和归档库名称。此选项仅在动态模式下有效，并且是缺省选项。-B static 将 -l 选项库搜索限定为仅扩展至归档库名称。可以在命令行上任意次地指定这些选项以进行切换。如果指定了 -B static 选项，在切换为 -B dynamic 选项之前，不会接受任何共享目标文件。请参见 -l 选项。

#### **-B eliminate[=*mode*] | local[=*mode*]**

导致缺省设置为全局绑定的任何符号从符号表中被删除，或缩减为局部可见。具有 STV\_DEFAULT 可见性的全局符号可以由链接编辑器缩减为更具限制性的可见性。具有任何其他 STV\_ 可见性的全局符号被视为具有显式可见性。链接编辑器遵循显式可见性，不能修改。请参见《链接程序和库指南》中的“SYMBOL\_SCOPE / SYMBOL\_VERSION 指令”。符号可见性可以通过编译器指令或通过定义版本或接口定义的 mapfiles 显式定义。

Mapfile 版本和接口定义可以包含 *auto-elimination* 或 *auto-reduction* 指令。请参见《链接程序和库指南》中的“SYMBOL\_SCOPE / SYMBOL\_VERSION 指令”。这些指令将导致未在 mapfile 中显式定义或者没有定义显示可见性的符号分别被删除或缩减为局部。除了任何显式符号定义，符号的删除或缩减还会受所生成目标文件类型影响，如随后的段落所述。-B eliminate 选项与 mapfile *auto-elimination* 指令请求相同的符号删除。-B local 选项与 mapfile *auto-reduction* 指令请求相同的符号缩减。

任一选项均可通过模式 *external* 或 *noexternal* 来控制对删除全局符号或缩减全局符号的选择。此微调通常是不必要的，因为缺省情况下链接编辑器会根据所生成目标文件的类型来采用相应模式：*external* 适用于可执行文件，*noexternal* 适用于共享目标文件。

构建动态目标文件时，可能需要确保某些符号保持全局状态，以便可以从外部依赖项引用这些符号。对于动态可执行文件而言尤其如此。构建可执行文件时，可重定位目标文件由支持运行时进程初始化的编译环境提供。这些可重定位目标文件可以包含从系统依赖项引用的全局符号。无论采用任何自动删除或自动缩减符号技术，这些符号均应保持全局状态，以便不会影响运行时执行。

将 `mode` 定义为 `external` 时会分析任何外部依赖项，以确定依赖项中的任何符号引用是否可能绑定到所构建目标文件内的符号定义。任何符合此绑定的全局符号都不会被删除或缩减为局部。此模式是生成动态可执行文件时的缺省模式。

将 `mode` 定义为 `noexternal` 会绕过对任何外部依赖项的分析，这样，所有未在 `mapfile` 中显式定义或没有定义显式可见性的符号均会缩减。此模式是生成共享目标文件时的缺省模式。

另请参见 `-B reduce` 选项。

#### `-B reduce`

生成可重定位目标文件时，任何符号可见性属性所定义的符号信息或者通过定义版本或接口定义的 `mapfiles` 定义的符号信息会被缩减。缺省情况下，生成可重定位目标文件时，可见性属性、版本定义或接口定义仅会记录在输出映像中。创建动态可执行文件或共享目标文件时，可见性属性或 `mapfile` 版本或接口定义始终应用于任何符号信息。

#### `-c name`

记录配置文件的 `name` 以便在运行时使用。可以使用配置文件来改变缺省搜索路径，提供目录高速缓存以及提供备选目标文件依赖项。请参见 `crle(1)`。

#### `-C`

取消改编 C++ 符号名称显示在诊断消息中。

#### `-D [!]token1,[!]token2,...`

按每个 `token` 的指定将调试信息输出为标准错误。特殊标记 `help` 指示可用标记的完整列表。请参见《链接程序和库指南》中的“调试帮助”。

#### `-e epsym`

##### `--entry epsym`

将输出文件的入口点地址设置为符号 `epsym`。

#### `-f name`

##### `--auxiliary name`

仅当构建共享目标文件时才使用。指定将共享目标文件的符号表用作 `name` 所指定的共享目标文件的符号表上的辅助过滤器。允许使用此选项的多个实例。此选项不能与 `-F` 选项一起使用。请参见《链接程序和库指南》中的“生成辅助过滤器”。

#### `-F name`

##### `--filter name`

仅当构建共享目标文件时才使用。指定将共享目标文件的符号表用作 `name` 所指定的共享目标文件的符号表上的过滤器。允许使用此选项的多个实例。此选项不能与 `-f` 选项一起使用。请参见《链接程序和库指南》中的“生成标准过滤器”。



**-G**

**-shared**

仅在动态模式下，生成共享目标文件。允许使用未定义的符号。请参见《[链接程序和库指南](#)》中的第 4 章“共享目标文件”。

**-h *name***

**--soname *name***

仅在动态模式下，当构建共享目标文件时，在目标文件的动态部分中记录 *name*。*name* 记录在与此目标文件而不是目标文件的文件系统名链接的任何动态目标文件中。因此，*name* 由运行时链接程序用作共享目标文件的名称以在运行时进行搜索。请参见《[链接程序和库指南](#)》中的“记录共享目标文件名称”。

**-i**

忽略 LD\_LIBRARY\_PATH。当 LD\_LIBRARY\_PATH 设置实际影响运行时库搜索（这会干扰正在执行的链接编辑）时，此选项非常有用。

**-l *x***

**--library *x***

搜索 libx.so 或 libx.a 库，它们分别是共享目标文件和归档库的常规名称。在动态模式下，除非 -Bstatic 选项有效，否则 ld 将在库搜索路径中指定的每个目录中搜索 libx.so 或 libx.a 文件。目录搜索在包含任何一个文件的第一个目录处停止。ld 会选择以 .so 结尾的文件，条件是 -lx 扩展为两个文件，这两个文件的名称采用 libx.so 和 libx.a 形式。如果未找到 libx.so，ld 将接受 libx.a。在静态模式下，即当 -Bstatic 选项有效时，ld 将仅选择以 .a 结尾的文件。ld 在遇到该库时会搜索库，因此 -l 的位置非常重要。请参见《[链接程序和库指南](#)》中的“与其他库链接”。

**-L *path***

**--library-path *path***

将 *path* 添加到库搜索目录。ld 首先在 -L 选项指定的任何目录中搜索库，然后在标准目录中搜索。仅当此选项位于 -L 选项应用到的 -l 选项之前时，此选项才有用。请参见《[链接程序和库指南](#)》中的“链接编辑器搜索的目录”。

环境变量 LD\_LIBRARY\_PATH 可用于补充库搜索路径，但是建议使用 -L 选项，因为环境变量也会由运行时环境解释。请参见“环境变量”部分的 LD\_LIBRARY\_PATH。

**-m**

在标准输出中生成内存映射或输入/输出部分列表，以及任何非致命多重定义符号。

**-M *mapfile***

作为 ld 指令的文本文件读取 *mapfile*。可以多次指定此选项。如果 *mapfile* 是目录，将处理该目录中的所有正规文件（如 [stat\(2\)](#) 所定义）。请参见《[链接程序和库指南](#)》中的附录 B“System V 发行版 4（版本 1）Mapfile”。/usr/lib/ld 中提供了示例 mapfile。请参见“文件”部分。

**-N *string***

此选项会将 `DT_NEEDED` 条目添加到所生成的目标文件的 `.dynamic` 部分中。`DT_NEEDED` 字符串的值为在命令行上指定的 *string*。此选项与位置有关，且 `DT_NEEDED` `.dynamic` 条目与链接编辑行上搜索到的其他动态依赖项有关。当组合使用 `-dy` 和 `-r` 选项时，此选项适用于在设备驱动程序可重定位目标文件中指定依赖项。

**-o *outfile*****--output *outfile***

生成名为 *outfile* 的输出目标文件。缺省目标文件的名称为 `a.out`。

**-p *auditlib***

标识审计库 *auditlib*。此审计库用于审计在运行时创建的目标文件。共享目标文件被标识为要求使用 `-p` 选项进行审计，将该共享目标文件指定为依赖项的任何目标文件都将继承此项要求。请参见 `-P` 选项。请参见《链接程序和库指南》中的“运行时链接程序审计接口”。

**-P *auditlib***

标识审计库 *auditlib*。此审计库用于审计在运行时创建的目标文件的依赖项。依赖项审计还可以从标识为要求审计的依赖项继承。请参见 `-p` 选项和 `-z globalaudit` 选项。请参见《链接程序和库指南》中的“运行时链接程序审计接口”。

**-Q *y* | *n***

使用 `-Q y` 时，会将 `ident` 字符串添加到输出文件的 `.comment` 部分。此字符串标识用于创建文件的 `ld` 的版本。这将导致当存在多个链接步骤（例如使用 `ld -r`）时，出现多个 `ld ident`。此标识与 `cc` 命令的缺省操作相同。`-Q n` 将隐藏版本标识。`.comment` 部分可以通过 `mcs(1)` 实用程序来处理。

**-r****--relocatable**

合并可重定位目标文件以生成一个可重定位目标文件。`ld` 不受未解析引用的影响。此选项不能与 `-a` 选项一起使用。

**-R *path*****-rpath *path***

冒号分隔的目录列表，用于指定运行时链接程序的库搜索目录。如果路径存在且不为空，将在输出目标文件中记录路径并将其传递给运行时链接程序。此选项的多个实例与各个用冒号分隔的 *path* 串联在一起。请参见《链接程序和库指南》中的“运行时链接程序搜索的目录”。

在关联目标文件中使用 `runpath` 比设置全局搜索路径（例如通过 `LD_LIBRARY_PATH` 环境变量）更可取。仅应记录查找目标文件依赖项所需的 `runpaths`。`ldd(1)` 在与 `-U` 选项一起使用时，还可用于在动态目标文件中搜索未使用的 `runpaths`。

还可以通过 `runpath` 来提供各种标记，运行路径提供了标识系统功能或目标文件位置的灵活方式。请参见《链接程序和库指南》中的第 6 章“使用动态字符串标记建立依赖性”。`$ORIGIN` 标记特别适用于将动态目标文件重定位到文件系统中的其他位置。

- s
- strip-all  
从输出文件中分离任何符号信息。这些选项等效于组合使用 `-z strip-class` 选项与 `debug` 和 `symbol` 类标识符。另请参见 `-z redlocsym` 和 `-z noldynsym` 选项。
- S *supportlib*  
共享目标文件 *supportlib* 是使用 `ld` 以及有关链接进程的给定信息装入的。使用 `-S` 选项定义的共享目标文件还可通过使用 `SGS_SUPPORT` 环境变量来提供。请参见《[链接程序和库指南](#)》中的“[链接编辑器支持接口](#)”。
- t  
禁止对具有不同大小或不同对齐方式的多重定义符号发出警告。
- u *symname*  
--undefined *symname*  
在符号表中作为未定义的符号输入 *symname*。此选项适用于完全从归档库装入。在此实例中，需要使用未解析的引用来强制装入第一个例程。此选项在命令行上的位置非常重要。此选项必须置于定义符号的库之前。请参见《[链接程序和库指南](#)》中的“[使用 -u 选项定义其他符号](#)”。
- V  
--version  
输出一条消息，其中给出有关所使用的 `ld` 的版本的的信息。
- Y *P, dirlist*  
更改用于查找库的缺省目录。*dirlist* 是冒号分隔的路径列表。
- z *allextract* | *defaultextract* | *weakextract*  
--whole-archive | --no-whole-archive  
从跟随的任何归档中更改目标文件的提取条件。缺省情况下，将提取归档成员以满足未定义的引用并通过数据定义提升暂定的定义。弱符号引用不会触发提取。使用 `-z allextract` 或 `--whole-archive` 选项时，从归档中提取所有归档成员。使用 `-z weakextract` 时，弱引用触发归档提取。`-z defaultextract` 或 `--no-whole-archive` 选项提供一种在使用前面的提取选项之后返回缺省值的方法。请参见《[链接程序和库指南](#)》中的“[归档处理](#)”。
- z *altexec64*  
执行 64 位 `ld`。历史上，已执行的链接编辑器的类由所创建的 ELF 目标文件的类决定。现在，已执行的链接编辑器的类由底层系统的类决定。通常情况下，这是 64 位。此选项维持向后兼容性。
- z *ancillary*[=*outfile*]  
指定一个辅助输出文件，用于接收通常将添加到输出目标文件的不可分配部分。运行时不需要不可分配部分，它们主要由调试器和其他监测工具使用。如果 *outfile* 存在，将使用指定的名称创建辅助文件。如果 *outfile* 不存在，则辅助文件的名称为主输出文件的名称加 `.anc` 后缀。请参见《[链接程序和库指南](#)》中的第 2 章“[链接编辑器](#)”。

只有可执行文件和共享目标文件支持辅助目标文件。指定 `-r` 选项来构建可重定位目标文件时，不能使用 `-z ancillary` 选项。与 `-z stub` 选项一起使用时，`-z ancillary` 选项会被静默忽略。在以下情况下也会忽略 `-z ancillary` 选项：即在指定该选项时未指定 *outfile*，以及通过 `-o` 选项指定的输出文件是设备特殊文件（如 `/dev/null`）。

**-z aslr[=*mode*]**

指定可执行文件的地址空间布局随机化 (Address Space Layout Randomization, ASLR) 行为。可以将模式值设置为 `enabled` 或 `disabled`。如果省略模式，则启用 ASLR。

**-z assert-deflib[=*libname* ]**

允许为使用通过检查链接编辑器提供的缺省搜索路径找到的 `-l` 命令行选项指定的库发出警告消息。如果提供 *libname* 值，将启用缺省库警告功能，并将指定的库添加到不为其发出警告的库列表中。可以指定多个 `-z assert-deflib` 选项以指定多个不应为其发出警告的库。

*libname* 值应为链接编辑器找到的不带任何路径组件的库文件的名称。例如，以下选项允许发出缺省库警告，但标准 C 库除外。

```
ld ... -z assert-deflib=libc.so ...
```

`-z assert-deflib` 是专用选项，主要用于这样的版本环境：其中存在多个具有相同名称的目标文件且需要对所使用的库实行严格控制。该选项不用于一般用途。

**-z deferred | nodeferred**

允许或禁止将动态依赖项标记为延迟。标记为 `deferred` 的动态依赖项也将标记为延迟可装入，且不会在初始进程启动时装入。在首次绑定到延迟引用时，延迟依赖项的装入过程会延迟到进程执行。与基本的延迟可装入依赖项不同，不会在处理 `LD_BIND_NOW` 时处理延迟依赖项，也不会通过带有 `RTLD_NOW` 标记的 `dlopen(3C)` 来处理延迟依赖项。请参见《链接程序和库指南》中的“延迟装入动态依赖项”。

组合使用延迟依赖项以及 `dlsym(3C)` 和 `RTLD_PROBE` 句柄，可提供灵活机制和自然编码样式以供测试功能使用。

**-z defs | nodefs**

**--no-undefined**

`-z defs` 选项和 `--no-undefined` 选项会在链接末尾保留有任何未定义的符号时强制生成致命错误。此模式是构建可执行文件时的缺省模式。由于历史原因，此模式不是构建共享目标文件时的缺省模式。建议使用 `-z defs` 选项，因为此模式可保证所构建的目标文件是自包含的。自包含目标文件在内部解析所有符号引用，或解析到目标文件的直接依赖项。

`-z nodefs` 选项允许未定义的符号。由于历史原因，此模式是构建共享目标文件时的缺省模式。与可执行文件一起使用时，未指定对此类未定义符号的引用的行为。不建议使用 `-z nodefs` 选项。

**-z direct | nodirect**

允许或禁止直接绑定到命令行上跟在其后的任何依赖项。与全局等效 **-Bdirect** 相比，这些选项允许对直接绑定进行更精确的控制。**-z direct** 选项在以下方面也与 **-Bdirect** 选项不同。不会在符号引用与要创建的目标文件中的关联定义之间建立直接绑定信息。不支持延迟装入。

**-z discard-unused= item1, item2, ...**

缺省情况下，链接编辑器会放弃未使用的空部分。在链接编辑期间，可能会将其他类别的输入材料确定为未使用。使用 **-z discard-unused** 选项可自动删除此类项目。可以识别以下 *item* 标记。

**sections**

通过链接编辑创建的输出文件中未使用的部分将被放弃。

**files**

通过链接编辑创建的输出文件中未使用的可重定位目标文件将被放弃。

如果可重定位目标文件提供的所有可分配部分均未使用，则可将该输入可重定位目标文件确定为未使用。另请参见 **-z guidance** 选项中有关“不需要的可重定位目标文件”的论述。

**dependencies**

通过链接编辑创建的输出文件中不会记录未使用的显式共享目标文件依赖项。

显式依赖项是指使用路径名或通过使用 **-l** 选项（更为常用）在命令行上定义的依赖项。显式依赖项可以依赖于其他目标文件，后者被称为隐式依赖项。如果符合以下两个条件，则可将显式依赖项确定为未使用。

- 未从所构建的目标文件中引用该依赖项提供的任何全局符号。
- 该依赖项不补偿任何隐式依赖项的要求。

另请参见 **-z guidance** 选项中有关“不需要的或补偿性的依赖项”的论述。

**none**

禁用所有未使用的处理，包括删除未使用的空部分的缺省操作。

另请参见《Oracle Solaris 11.1 链接程序和库指南》中的“删除未使用的材料”。

**-z endfiltee**

标记 *filtee* 以便在过滤器进行处理时，*filtee* 可终止过滤器进行的任何进一步的 *filtee* 搜索。请参见《链接程序和库指南》中的“减少 *filtee* 搜索”。

**-z fatal-warnings | nofatal-warnings****--fatal-warnings | --no-fatal-warnings**

使用 **-z fatal-warnings** 和 **--fatal-warnings** 选项可使链接编辑器将警告视为致命错误。

使用 **-z nofatal-warnings** 和 **--no-fatal-warnings** 选项可使链接编辑器将警告视为非致命错误。这是缺省行为。

**-z finiarray=function**

在所构建的目标文件的 `.fini_array` 部分后附加一个条目。如果不存在 `.fini_array` 部分，将创建该部分。新条目将初始化为指向 `function`。请参见《链接程序和库指南》中的“初始化节和终止节”。

**-z globalaudit**

此选项对已使用 `-P` 选项记录的审计库定义进行补充。此选项仅在构建动态可执行文件时有意义。使用 `-P` 选项在目标文件中定义的审计库通常允许审计该目标文件的直接依赖项。`-z globalaudit` 将审计程序升级为全局审计程序，因此允许审计所有有依赖项。请参见《链接程序和库指南》中的“调用审计接口”。

使用 `-P` 选项和 `-z globalaudit` 选项建立的审计程序等效于使用 `LD_AUDIT` 环境变量建立的审计程序。请参见 `ld.so.1(1)`。

**-z guidance[=item1, item2, ...]**

提供指南消息来指示 `ld` 选项，这些选项可以改进所生成目标文件的质量，或者在其他方面被视为有益。提供的特定指南在系统升级时可能会发生更改。旧版本 `ld` 提供的过时指南在新版本中可能已删除。同样，可能会在新版本的 `ld` 中添加新指南。因此，指南始终代表当前最佳做法。

通过提供代表要禁止的指南类的 `item` 标记列表，可以启用指南但禁止特定指南消息。这样，可以禁止不需要的建议而不会失去其他指南的优点。`ld` 会静默忽略无法识别的 `item` 标记，从而允许在多种早期或新版本的 Solaris 上执行指定的 `ld` 命令。

下面是当前版本的 `ld` 提供的指南以及用于禁止这些消息的 `item` 标记。

**指定所需依赖项**

动态可执行文件和共享目标文件应明确定义它们所需的所有依赖项。如果在构建动态目标文件时有任何符号引用不能满足要求，指南建议使用 `-z defs` 选项。可以使用 `-z guidance=nodefs` 禁用此指南。

**不指定不需要的或补偿性的依赖项**

动态可执行文件和共享目标文件不应定义不满足动态目标文件建立的符号引用的任何显式依赖项。指南建议删除不需要的或未使用的依赖项。未使用的依赖项可以是以下两个类别中的一种。

- 不满足符号引用要求的显式依赖项。
- 不满足所构建动态目标文件的符号引用要求，但会补偿隐式依赖项的显式依赖项。请参见 `-z discard-unused` 选项中有关“依赖项”的论述。

可以通过 `-z guidance=nounused-dependencies` 或同义词 `-z guidance=nounused` 来禁用这两个类别的指南。可以通过 `-z guidance=nounused-compensators` 禁用补偿依赖项的指南。

另请参见《Oracle Solaris 11.1 链接程序和库指南》中的“删除未使用的材料”。



### 不指定不需要的可重定位目标文件

如果某个可重定位目标文件的可分配部分未由链接编辑所涉及的任何其他目标文件所引用，则所创建的输出文件不应包含该可重定位目标文件的任何信息。指南建议删除未使用的可重定位目标文件。可以使用 `-z guidance=nounused-files` 禁用此指南。

另请参见《Oracle Solaris 11.1 链接程序和库指南》中的“删除未使用的材料”。

### 延迟装入

应标识依赖项以进行延迟装入。如果在遇到 `-z lazyload` 或 `-z nolazyload` 选项之前要处理任何依赖项，指南建议使用 `-z lazyload` 选项。可以使用 `-z guidance=nolazyload` 禁用此指南。

### 直接绑定

应使用直接绑定来引用依赖项。如果在遇到 `-B direct` 或 `-z direct` 选项之一或者 `-z nodirect` 选项之前要处理任何依赖项，指南建议使用前两个选项中的一个。可以使用 `-z guidance=nodirect` 禁用此指南。

### 纯文本段

动态目标文件不应包含不可写但可分配部分的重定位。如果针对文本段的任何重定位持续，且未遇到 `-z textwarn` 和 `-z textoff` 选项，指南建议使用位置无关代码 (Position Independent Code, PIC) 编译目标文件。可以使用 `-z guidance=notext` 禁用此指南。

### Mapfile 语法

所有 mapfile 应使用版本 2 mapfile 语法。如果遇到任何使用版本 1 语法的 mapfile，指南将建议使用版本 2 语法。可以使用 `-z guidance=nomapfile` 禁用此指南。

### 库搜索路径

将静默忽略 ld 遇到的不合适的依赖项。例如，将忽略在生成 64 位目标文件时遇到的 32 位依赖项。这些依赖项可以由不正确的搜索路径设置产生，例如提供不正确的 `-L` 选项。尽管此依赖项是有利的，但其处理会造成浪费，而且可能会隐藏应解决的版本问题。指南建议删除任何不合适的依赖项。可以使用 `-z guidance=nolibpath` 禁用此指南。

此外，`-z guidance=noall` 可用来完全禁用指南功能。有关指南的更多信息和构建更好的目标文件的建议，请参见《链接程序和库指南》中的第 7 章“链接编辑器快速参考”。

`-z help`

`--help`

在标准输出中输出命令行选项的摘要并退出。

`-z ignore | record`

忽略或记录未作为链接编辑的一部分引用的共享目标文件依赖项。

`-z ignore` 和 `-z record` 是位置选项，用于切换 `ld` 处理命令行中遇到的未引用依赖项的方式。遇到 `-z ignore` 时，将静默忽略任何后续未引用依赖项。遇到 `-z record` 时，将记录所有依赖项，而不考虑依赖项是否被引用。

缺省情况下，`ld` 会记录所有依赖项而不考虑依赖项是否被引用。可以使用非位置 `-z discard-unused=dependencies` 选项来更改此初始缺省设置。确立初始设置后，可以使用 `-z ignore` 和 `-z record` 来更改缺省行为。

#### `-z initarray=function`

在所构建的目标文件的 `.init_array` 部分后附加一个条目。如果不存在 `.init_array` 部分，将创建该部分。新条目将初始化为指向 `function`。请参见《[链接程序和库指南](#)》中的“[初始化节和终止节](#)”。

#### `-z initfirst`

标记目标文件以便先对该目标文件进行运行时初始化，然后再对同时进入进程的任何其他目标文件进行运行时初始化。此外，在对该目标文件进行运行时终结化之前，先对同时从进程中删除的任何其他目标文件进行运行时终结化。此选项仅在构建共享目标文件时有意义。

#### `-z interpose`

将目标文件标记为插入项。在运行时，如果已使用 `-z interpose` 选项标记某个目标文件，则会将该目标文件标识为显式插入项。在使用 `LD_PRELOAD` 环境变量装入目标文件时，也会建立显式插入项。由于目标文件的装入顺序可能会出现隐含插入，但是，此隐含插入对于运行时链接程序是未知的。显式插入可确保不论目标文件的装入顺序如何都会进行插入。显式插入还可确保当直接绑定有效时，运行时链接程序在任何显式插入项中搜索符号。

#### `-z lazyload|nolazyload`

允许或禁止将动态依赖项标记为延迟装入。标记为 `lazyload` 的动态依赖项在初始进程启动时不会装入。这些依赖项将延迟到首次绑定目标文件时装入。**注意**：延迟装入要求对依赖项以及进程中使用的每个动态目标文件的关联 `runpaths` 进行正确声明。请参见《[链接程序和库指南](#)》中的“[延迟装入动态依赖项](#)”。

#### `-z ld32=arg1,arg2,...`

#### `-z ld64=arg1,arg2,...`

链接编辑器的类受所创建的输出文件的类以及底层操作系统的功能影响。`-z ld[32|64]` 选项提供定义任何链接编辑器参数的方法。定义的参数分别仅由链接编辑器的 32 位类或 64 位类解释。

例如，支持库是类特定的，因此可以使用以下选项确保支持库的类正确：

```
ld ... -z ld32=-Saudit32.so.1 -z ld64=-Saudit64.so.1 ...
```

#### `-z loadfltr`

标记过滤器以指示必须在运行时立即处理 `filtee`。通常，过滤器处理会延迟到某个符号引用绑定到该过滤器。包含此标志的目标文件的运行时处理将模拟 `LD_LOADFLTR` 环境变量生效时的运行时处理。请参见 `ld.so.1(1)`。



**-z mapfile-add=*name***

将 *name* 添加到已知的 mapfile 条件输入表达式谓词列表中。此选项等效于在链接编辑器读取的第一个 mapfile 的顶部放置以下行。

```
$mapfile_version 2
$add name
```

通过 **-z mapfile-add** 输入的名称可以与 `mapfile $if` 和 `$elif` 指令一起使用来有条件地处理 mapfile 输入。请参见《链接程序和库指南》中的第 10 章“Mapfile”。

**-z muldefs****--allow-multiple-definition**

允许多个符号定义。缺省情况下，在可重定位目标文件之间出现多个符号定义会导致致命错误。此选项将禁止错误条件出现，并允许采用第一个符号定义。

**-z nodelete**

将目标文件标记为在运行时不可删除。此模式类似于通过组合使用 `dlopen(3C)` 与 `RTLD_NODELETE` 模式将目标文件添加到进程中。

**-z nodlopen**

将目标文件标记为对 `dlopen(3C)` 不可用，即标记为 `dlopen()` 指定的目标文件或标记为 `dlopen()` 指定的目标文件所需的任何形式的依赖项。此选项仅在构建共享目标文件时有意义。

**-z nodump**

将目标文件标记为对 `dlDump(3C)` 不可用。

**-z parent=*object***

指定父目标文件，该目标文件可以是可执行文件或共享目标文件，针对该目标文件链接输出目标文件。通常，在创建要在运行时由可执行文件通过 `dlopen()` 函数装入的插件共享目标文件时，使用此选项。父目标文件的符号表用于满足来自插件目标文件的引用。请参见《链接程序和库指南》中的第 2 章“链接编辑器”。

**-z preinitarray=*function***

在所构建的目标文件的 `.preinit_array` 部分后附加一个条目。如果不存在 `.preinit_array` 部分，将创建该部分。新条目将初始化为指向 *function*。请参见《链接程序和库指南》中的“初始化节和终止节”。

**-z relaxreloc**

当使用引用已删除 COMDAT 部分的符号而遇到重定位时，`ld` 通常会发出致命错误。如果启用 **-z relaxreloc**，`ld` 将改为将此类重定位重定向到保留的 COMDAT 部分中的等效符号。**-z relaxreloc** 是专门选项，主要由编译器作者使用，不用于一般用途。

**-z rescanner-now**

这些选项重新扫描提供给链接编辑的归档文件。缺省情况下，当命令行上出现归档时，将立即处理归档。传统上来说，在命令行末尾指定归档，以便其符号定义可解析任何前述引用。但是，有必要多次指定归档以满足其自身的相互依赖项。

`-z rescan-now` 是位置选项，在命令行上遇到该选项时链接编辑器会立即处理该选项。将立即重新处理命令行上此时显示的所有归档以尝试定位解析符号引用的其他归档成员。将重复此归档重新扫描操作，直到通过归档传递时未提取新成员。

`-z rescan` 是与位置无关的选项。链接编辑器会将重新扫描操作推迟到在它处理整个命令行之后再行进行，然后开始对命令行上显示的所有归档执行最终重新扫描操作。`-z rescan` 操作可与包含初始化 (`.init`) 或终结化 (`.fini`) 部分的目标文件进行不正确地交互，从而禁止这些部分中的代码运行。由于此原因，`-z rescan` 已过时，建议使用 `-z rescan-now`。

`-z rescan-start ... -z rescan-end`

`--start-group ... --end-group`

`-( ... -)`

定义归档重新扫描组。这是位置构造，在遇到封闭分隔符选项时链接编辑器会立即进行处理。将在组分分隔符选项中的归档重新处理为组以尝试定位解析符号引用的其他归档成员。将重复此归档重新扫描操作，直到通过归档传递时未提取新成员。无法嵌套归档重新扫描组。

`-z strip-class=[!]class1,[!]class2,...`

从任何输入目标文件分离特定的部分类，禁止将这些部分添加到输出文件。通过此选项，可以对可在输出文件中忽略的部分进行细粒度控制。

以下分离类描述仅适用于不可分配的部分。

可以在每个类标记前加上“!”，以指示不应删除该类。此定义在与 `nonalloc` 类结合时非常有用。例如，使用 `"-z strip-class=nonalloc,!note"` 将删除所有不可分配部分（附注部分除外）。

分离的部分将从输出目标文件中彻底删除。使用 `-z ancillary` 选项可更改与非动态符号表 `.symtab` 有关的此行为以及与其相关的部分。缺省情况下，符号表会同时写入主目标文件和辅助目标文件。分离后，符号表将仅写入辅助目标文件，而在主目标文件中标记为不存在。如果在不使用 `-z ancillary` 的情况下从目标文件分离 `.symtab`，将以常规方式彻底删除该部分。

可以定义以下部分类。

**nonalloc**

分离任何不可分配部分。这些部分标识为不包括 `SHF_ALLOC` 节标志。此类可封装除 `symbol` 类以外的所有其他类。`nonalloc` 类往往本身即足以删除任何不需要的部分。

**annotate**

分离任何注释部分。这些部分提供内存访问工具和覆盖相关工具使用的信息。这些部分通过 `SHT_SUNW_ANNOTATE` 节类型来进行标识。

**comment**

分离任何注释部分。这些部分通过 `.comment` 节名称来进行标识。此外，`mcs(1)` 实用程序通常用于处理注释部分。

**debug**

分离通常用于包含调试数据的部分。这些部分通过 `.compcom`、`.line`、`.stab*` 或 `.debug*` 节名称来进行标识。这些部分还可通过 `SHT_SUNW_DEBUG*` 节类型来进行标识。

**exclude**

分离任何可排除的部分。这些部分通过 `SHF_EXCLUDE` 节标志来进行标识。在创建可重定位目标文件时此类非常有用。缺省情况下，当创建动态可执行文件或共享目标文件时会自动排除此类部分，在创建可重定位目标文件时将保留此类部分。

**说明**

分离任何注释部分。这些部分通过 `SHT_NOTE` 节类型来进行标识。

**符号**

如果输出文件不是可重定位目标文件，则分离任何不可分配符号表和字符串表部分。这些部分通过 `SHT_SYMTAB` 节类型来进行标识。任何关联的字符串表也将删除。

**-z stub**

生成桩共享目标文件。桩目标文件是可提供与实际目标文件相同的链接接口但不包含代码或数据的共享目标文件，完全由 `mapfile` 生成。桩目标文件不能在运行时使用。但是，可以根据桩目标文件构建应用程序，桩目标文件提供要在运行时使用的实际目标文件名称，然后在运行时使用实际目标文件。

只能为共享目标文件生成桩目标文件，必须提供定义要导出的全局符号的 `mapfile`。因此，在使用 `-z stub` 时，需要使用 `-G` 和 `-M` 选项。在生成桩目标文件时，链接编辑器将忽略在命令行中指定的任何目标文件或库文件，无需存在这些文件即可生成桩目标文件。由于可以省略编译步骤，并且相对而言链接编辑器只需进行少量操作，因此可以很快生成桩目标文件。

请参见《《链接程序和库指南》》中的““桩目标文件””。

**-z symbolcap**

将定义目标文件功能的可重定位目标文件转换为定义符号功能的可重定位目标文件。请参见《《链接程序和库指南》》中的“将目标文件功能转换为符号功能”。

**-z target=sparc|x86**

指定输出目标文件的机器类型。支持的目标有 `SPARC` 和 `x86`。使用指定目标的 32 位机器类型，除非存在 `-64` 选项，这种情况下使用相应的 64 位机器类型。缺省情况下，要生成的目标文件的机器类型根据在命令行处理的第一个 ELF 目标文件来决定。如果未指定任何目标文件，机器类型由在命令行处理的第一个归档文件中遇到的第一个目标文件决定。如果没有目标文件或归档文件，链接编辑器将假定使用本地机器。直接使用 `ld` 创建其输入完全来自 `mapfile` 的目标文件时，此选项非常有用。请参见 `-M` 选项。在极少数情况下，完全从包含不同机器类型目标文件的归档文件进行链接时（此时第一个目标文件不是所需的机器类型），此选项也非常有用。

**-z text | textoff | textwarn**

这些选项只能在动态模式下使用。如果针对不可写但可分配部分的任何重定位持续进行，`-z text` 选项将强制发出致命错误。由于历史原因，此模式不是构建可执行文件或共享目标文件时的缺省模式。但是，建议使用此模式以确保可在多个正在运行的进程之间共享所构建的动态目标文件的文本段。共享文本段可使在装入内存时重定位系统开销最少。请参见《[链接程序和库指南](#)》中的“与位置无关的代码”。

`-z textoff` 选项允许针对所有可分配部分（包括不可写部分）进行重定位。此模式是构建共享目标文件时的缺省模式。

如果针对不可写但可分配部分的任何重定位持续进行，`-z textwarn` 选项将列出警告。此模式是构建可执行文件时的缺省模式。

**-z verbose**

此选项在链接编辑期间提供附加警告诊断。当前，此选项支持以下警告。

- 对交换区重新定位的可疑使用。
- 构建共享目标文件时限制使用静态 TLS 重定位。
- 符号可见性不一致。

将来可能会增强此选项的功能，以提供被认为太混乱而不能在缺省情况下生成的其他诊断。

**-zwrap=symbol****-wrap= symbol****--wrap= symbol**

重命名对 *symbol* 的未定义引用以允许将包装代码链接到输出目标文件而不必修改源代码。如果指定 `-z wrap`，将修改对 *symbol* 的所有未定义引用以引用 `__wrap_symbol`，并修改对 `__real_symbol` 的所有引用以引用 *symbol*。您应提供包含 `__wrap_symbol` 函数的目标文件。此包装函数可以调用 `__real_symbol` 以引用要包装的实际函数。

以下是适用于 `malloc(3C)` 函数的包装示例。

```
void *
__wrap_malloc(size_t c)
{
 (void) printf("malloc called with %zu\n", c);
 return (__real_malloc(c));
}
```

如果使用 `-z wrap=malloc` 将其他代码链接到此文件以编译所有目标文件，则对 `malloc` 的所有调用将改为调用函数 `__wrap_malloc`。对 `__real_malloc` 的调用将调用 `malloc` 实函数。

应在单独的源文件中维护实函数和包装函数。否则，编译器或汇编程序可能会解析调用而不是将该操作留给链接编辑器执行，并且可能阻止进行包装。

## 辅助选项

以下选项不太常用。这些选项支持向后兼容性、非常专业化的功能或已被改进的选项变体取代的选项。

### -a

仅在静态模式下，生成静态可执行文件。不允许未定义的引用。此选项是静态模式的缺省行为。**-a** 选项不能与 **-r** 选项一起使用。请参见“描述”部分的“静态可执行文件”。

### -b

仅在动态模式下，不对引用共享目标文件中的符号的动态可执行文件重定位进行特殊处理。如果不提供 **-b** 选项，链接编辑器将在动态可执行文件中应用技术以便文本段可以保持只读状态。一种技术是为共享目标文件中定义的函数的引用创建特殊位置无关重定位。另一种技术安排在运行时将在共享目标文件中定义的数据目标文件复制到可执行文件的内存映像中。

**-b** 选项旨在用于专用动态目标文件，不建议用于一般用途。使用该选项将禁止执行确保目标文件可共享性时需要的所有专门处理操作，甚至可能会禁止重定位 64 位可执行文件。

### -B group

建立作为组的共享目标文件及其依赖项。组中的目标文件在运行时绑定到组的其他成员。此模式类似于通过组合使用 **dlopen(3C)** 和 **RTLD\_GROUP** 模式将目标文件添加到进程中。对标识为组的目标文件具有显式依赖性的目标文件会成为该组的成员。

由于该组必须为自包含，因此，使用 **-B group** 选项还将断言 **-z defs** 选项。

建立一个组可提供控制一组目标文件绑定的基本方式。不过，可以通过直接绑定实现更好的控制。请参见 **-B direct** 选项。

### -B symbolic

仅适用于动态模式。构建共享目标文件时，将该目标文件中全局符号的引用绑定到其定义（如果可用）。通常，在运行时才会绑定共享目标文件中全局符号的引用，即使有定义可用也是如此。此模式允许可执行文件或其他共享目标文件中的相同符号的定义覆盖该目标文件自己的定义。**ld** 将对未定义的符号发出警告，除非 **-z defs** 覆盖定义。

**-B symbolic** 选项旨在用于专用动态目标文件，不建议用于一般用途。要缩减目标文件所需的运行时重定位处理，建议创建版本定义并使用直接绑定。请参见 **-B direct** 选项。

### -dy|n

如果指定 **-dy**（缺省情况），**ld** 将使用动态链接。如果指定 **-dn**，**ld** 将使用静态链接。请参见“描述”部分的“静态可执行文件”以及 **-B dynamic|static**。

### -I name

#### --dynamic-linker name

在构建可执行文件时，使用 *name* 作为要写入程序头的解释程序的路径名。在静态模式下，缺省为没有解释程序。在动态模式下，缺省为运行时链接程序 **ld.so.1(1)**

的名称。任何一种情况都可以由 `-I name` 覆盖。`exec(2)` 在装入 `a.out` 时会装入此解释程序，并将控件传递给该解释程序而不是直接传递给 `a.out`。

**-z absexec**

仅当构建动态可执行文件时才使用。指定应立即解析外部绝对符号的引用而不是留到运行时进行解析。在非常专业化的情况下，此选项将删除可能导致可执行文件要求的交互空间过大的文本重定位。

**-z combrelloc | nocombrelloc**

缺省情况下，当构建可执行文件或共享目标文件时，`ld` 会合并多个重定位部分。此部分合并不同于可重定位目标文件合并，在部分合并中，重定位部分与必须应用重定位的部分维持一一对应的关系。`-z nocombrelloc` 选项禁止合并重定位部分，并保持在原始可重定位目标文件中找到的一一对应关系。

`ld` 按数据重定位部分条目的符号引用对这些条目进行排序。此排序会减少运行时符号查找。在合并多个重定位部分时，此排序在将目标文件装入内存时可能产生最少的重定位系统开销，并加快动态目标文件的运行时装入。

历史上，单个重定位部分已转入任何可执行文件或共享目标文件中，并且需要 `-z combrelloc` 选项才能启用前面所述的重定位部分合并。重定位部分合并现在是缺省设置。由于旧版本环境的好处，`-z combrelloc` 选项仍可接受，但该选项不是必需的，且不起作用。

**-z groupperm | nogroupperm**

分配或取消分配跟在唯一组后的每个依赖项。将依赖项分配给组的作用就如同使用 `-B group` 选项构建依赖项。

**-z nocompstrtab**

禁用 ELF 字符串表和注释部分的压缩。缺省情况下，字符串压缩适用于 `SHT_STRTAB` 部分、设置了 `SHF_MERGE` 和 `SHF_STRINGS` 节标志的 `SHT_PROGBITS` 部分以及注释部分。

`mcs(1)` 实用程序搭配 `-c` 选项时，可用于在构建目标文件后压缩注释部分。

**-z nodefaultlib**

标记目标文件以忽略在任何 `LD_LIBRARY_PATH` 或 `runpaths` 后使用的运行时缺省库搜索路径。此选项表明可以通过目标文件的 `runpath` 来满足目标文件的所有依赖项。

**-z noldynsym**

禁止在动态可执行文件或共享目标文件中包含 `.SUNW_ldynsym` 部分。`.SUNW_ldynsym` 部分通过为本地函数提供符号来增大 `.dynsym` 部分。本地函数符号允许调试器在精简程序的栈跟踪中显示本地函数名称。同样，`dladdr(3C)` 能够提供更准确的结果。

`-z noldynsym` 选项还禁止包含与 `.SUNW_ldynsym` 部分相关的两个符号排序部分。`.SUNW_dynsym` 部分提供对正则函数和变量符号的有序访问。`.SUNW_dyntls` 部分提供对线程本地存储 (TLS) 变量符号的有序访问。



成为所生成文件的可分配文本段一部分的 `.SUNW_ldynsym`、`.SUNW_dynsym` 和 `.SUNW_dyntlsort` 部分无法通过 `strip(1)` 删除。因此，`-z noldynsym` 选项是禁止其包含的唯一方式。

**-z nopartial**

在要生成的输出文件中展开在可重定位目标文件中定义的部分初始化的符号。

**-z now**

将目标文件标记为需要 `non-lazy` 运行时绑定。此模式类似于通过组合使用 `dlopen(3C)` 与 `RTLD_NOW` 模式将目标文件添加到进程中。此模式还类似于使 `LD_BIND_NOW` 环境变量起作用。请参见 `ld.so.1(1)`。

**-z origin**

将目标文件标记为需要在运行时立即进行 `$ORIGIN` 处理。维护此选项只是为了历史兼容性，因为现在缺省提供目标文件的运行时分析以进行 `$ORIGIN` 处理。

**-z redlocsym**

删除所有本地符号，但符号表 `SHT_SYMTAB` 中的 `SECT` 符号除外。引用本地符号的所有重定位将更新为引用相应的 `SECT` 符号。通过此选项，专用目标文件可显著减小其符号表大小。另请参见 `-z strip-class` 和 `-z noldynsym` 选项。

尽管 `-z redlocsym` 选项对特殊目标文件（例如在操作系统内核中使用的那些目标文件）很有用，但不建议将该选项用于一般用途。符号表 `SHT_SYMTAB` 的大小不影响运行时行为，而删除本地符号可能会对过程观察产生负面影响。删除本地符号可以减少使用编译器驱动程序 `-g` 选项生成的调试信息。删除本地符号还会删除正常写入到 `.SUNW_ldynsym` 部分中的信息，从而会降低调试器和工具（如 `pstack(1)` 和 `truss(1)`）的效用。

**-z rescanner**

一个与位置无关的选项，可导致重新扫描提供给链接编辑的归档文件。链接编辑器会将重新扫描操作推迟到在它处理整个命令行之后再行，然后开始对命令行上显示的所有归档执行最终重新扫描操作。`-z rescanner` 操作可与包含初始化 (`.init`) 或终结化 (`.fini`) 部分的目标文件进行不正确地交互，从而禁止这些部分中的代码运行。由于此原因，`-z rescanner` 已过时，建议使用 `-z rescanner-now`。

## 环境变量

**LD\_ALTEXEC**

备用链接编辑器路径名。`ld` 执行并传递对此备用链接编辑器的控制。此环境变量提供用于覆盖从各种编译器驱动程序调用的缺省链接编辑器的通用方法。请参见 `-z altexec64` 选项。

**LD\_LIBRARY\_PATH**

要在其中搜索使用 `-l` 选项指定的库的目录列表。多个目录之间用冒号分隔。在最普遍的情况下，此环境变量包含两个由分号分隔的目录列表。

*dirlist1;dirlist2*

如果使用出现的任意数量的 `-L` 来调用 `ld`，如下所示：

```
ld ... -Lpath1 ... -Lpathn ...
```

则搜索路径顺序为：

*dirlist1 path1 ... pathn dirlist2 LIBPATH*

如果目录列表不包含分号，该列表将被解释为 *dirlist2*。

`LD_LIBRARY_PATH` 环境变量还会影响动态依赖项的运行时链接程序搜索。

可以使用 `_32` 或 `_64` 后缀指定此环境变量。这会使环境变量分别特定于 32 位或 64 位进程，并覆盖任何有效的无后缀版的环境变量。

#### `LD_NOEXEC_64`

禁止自动执行 64 位链接编辑器。如果第一个可重定位目标文件的 ELF 类标识 64 位目标文件，缺省情况下链接编辑器将执行 64 位版本。32 位链接编辑器可以创建的 64 位映像具有一些限制。但是，某些链接编辑可能会发现使用 32 位链接编辑器的速度更快。

#### `LD_OPTIONS`

`ld` 的缺省选项集合。`LD_OPTIONS` 由 `ld` 解释，就好像已在命令行上设置其值，紧跟在用来调用 `ld` 的名称后，如下所示：

```
ld $LD_OPTIONS ... other-arguments ...
```

#### `LD_RUN_PATH`

用于指定链接编辑器的 `runpath` 的备用机制。请参见 `-R` 选项。如果同时指定 `LD_RUN_PATH` 和 `-R` 选项，`-R` 将取代另一个选项。

#### `SGS_SUPPORT`

提供使用链接编辑器装入的共享目标文件的冒号分隔列表以及有关链接进程的给定信息。可以使用 `_32` 或 `_64` 后缀指定此环境变量。这会使环境变量分别特定于 `ld` 的 32 位或 64 位类，并覆盖任何有效的无后缀版的环境变量。请参见 `-S` 选项。

请注意，以字符 "LD\_" 开头的环境变量名称是保留名称，供将来可能提升 `ld` 和 [ld.so.1\(1\)](#) 时使用。

## 文件

`libx.so` 共享目标文件库。

`libx.a` 归档库。

`a.out` 缺省输出文件。

`LIBPATH` 对于 32 位库，缺省搜索路径为 `/lib`，后跟 `/usr/lib`。对于 64 位库，缺省搜索路径为 `/lib/64`，后跟 `/usr/lib/64`。

`/usr/lib/ld` 包含多个可在链接编辑期间使用的 `mapfile` 的目录。这些 `mapfile` 提供多种功能，例如定义内存布局、对齐 `bss` 以及定义不可执行栈。

## 属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值           |
|------|---------------|
| 可用性  | system/linker |



| 属性类型  | 属性值             |
|-------|-----------------|
| 接口稳定性 | Committed (已确定) |

## 另请参见

`as(1)`、`crle(1)`、`gprof(1)`、`ld.so.1(1)`、`ldd(1)`、`mcs(1)`、`pvs(1)`、`strip(1)`、`exec(2)`、`stat(2)`

《链接程序和库指南》

## 附注

由于历史原因，维护由 `ld` 应用的缺省选项。在今天的编程环境中，动态目标文件占主导地位，备用缺省选项通常会更有意义。但是，必须维护历史缺省选项以确保与现有程序开发环境的兼容性。本手册中在可能的地方提到了历史缺省选项。有关最新推荐的选项的描述，请参见《链接程序和库指南》中的第 7 章“链接编辑器快速参考”。

如果由 `ld` 创建的文件已经存在，则在处理完所有输入文件后，该文件将解除链接。然后，将创建具有指定名称的新文件。这允许 `ld` 创建新版本的文件，同时允许正在访问旧文件内容的现有进程继续运行。如果旧文件没有其他链接，则当最后一个引用文件的进程终止时，将释放已删除文件的磁盘空间。

在 Oracle Solaris 11 中，对于所创建的文件已存在的情况，`ld` 的行为已更改。在早期版本中，将现有文件重写到适当的位置，这种方法可能会损坏任何使用该文件的正在运行的进程。此更改对于在文件系统中具有多个硬链接的输出文件有意义。以前，所有链接都保持不变，而且所有链接都可以访问新文件内容。新的 `ld` 行为**中断**了此类链接，其结果是只有指定的输出文件名才能引用新文件。所有其他链接继续引用旧文件。为了确保行为一致，依赖于指向链接程序输出文件的多个硬链接的应用程序应该明确删除其他文件名，然后再重新链接这些文件名。

**引用名** ldapdelete – ldap 删除条目工具

**用法概要**

```
ldapdelete [-n] [-v] [-c] [-d debuglevel] [-f file]
 [-D bindDN] [-w passwd | -j file] [-J [:criticality]]
 [-?] [-H] [-h ldaphost] [-V version] [-i locale]
 [-k path] [-P path] [-N certificate] [-y proxyid]
 [-p ldapport] [-O hoplimit] [-o attributename=value]
 [-W password] [dn]...
```

**描述** ldapdelete 实用程序可打开与 LDAP 服务器的连接，然后绑定并删除一个或多个条目。如果提供了一个或多个 *dn* 参数，则会删除那些具有标识名的条目。如果没有提供 *dn* 参数，则会从 *file*（如果指定了 -f 选项）或者从标准输入读取一组 DN。

**选项** 支持以下选项：

|                      |                                                                                                                                                                                                                            |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -a                   | 在删除分支时，绕过确认问题。                                                                                                                                                                                                             |
| -c                   | 连续运行模式。将会报告错误，但是 ldapdelete 将继续进行删除操作。缺省行为是在报告错误后退出。                                                                                                                                                                       |
| -d <i>debuglevel</i> | 设置 LDAP 调试级别。适用于 ldapdelete 的有用调试级别包括： <ul style="list-style-type: none"> <li>1 跟踪</li> <li>2 包</li> <li>4 参数</li> <li>32 过滤器</li> <li>128 访问控制</li> </ul> 要请求多个类别的调试信息，请将掩码相加。例如，要请求跟踪和过滤器信息，请将 <i>debuglevel</i> 指定为 33。 |
| -D <i>bindDN</i>     | 使用标识名 <i>bindDN</i> 绑定到目录。                                                                                                                                                                                                 |
| -E                   | 通过验证响应控制，要求服务器公开（报告）绑定标识。                                                                                                                                                                                                  |
| -f <i>file</i>       | 从 <i>file</i> 而不是从标准输入读取条目删除信息。                                                                                                                                                                                            |
| -?                   | 显示简要描述所有选项的用法帮助文本。                                                                                                                                                                                                         |

|                                                               |                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -H                                                            | 显示简要描述所有选项的用法帮助文本。                                                                                                                                                                                                                                                  |
| -h <i>ldaphost</i>                                            | 指定运行 LDAP 服务器的备用主机。                                                                                                                                                                                                                                                 |
| -i <i>locale</i>                                              | 指定用于命令行输入的字符集。缺省值是在 LANG 环境变量中指定的字符集。您可能希望使用此选项执行从指定的字符集到 UTF8 的转换，从而覆盖 LANG 设置。<br><br>使用此参数，可以采用指定的字符集输入绑定 DN 和目标 DN。ldapdelete 工具在处理搜索请求之前基于这些参数转换输入。例如，-i no 指示将以挪威语提供绑定 DN 和目标 DN。<br><br>此选项仅影响命令行输入。也就是说，如果指定了一个包含 DN 的文件（通过 -f 选项），ldapdelete 将不会转换该文件中的数据。 |
| -j <i>filename</i>                                            | 指定绑定 DN 的口令或 SSL 客户机密钥数据库的口令所在的文件。要保护口令，请在脚本中使用此选项，并将口令存放在安全文件中。此选项与 -w 和 -W 选项互斥。-j 选项是介于 -j 和 -w/-W 之间的一个较安全的备选项。                                                                                                                                                 |
| -J [ <i>:criticality[:value]::b64value b64value fileurl</i> ] | Criticality 是一个布尔值（缺省值是 false）。                                                                                                                                                                                                                                     |
| -k <i>path</i>                                                | 指定包含转换例程的目录路径。如果要指定目录服务器缺省情况下不支持的语言环境，则需要使用这些例程。这用于 NLS 支持。                                                                                                                                                                                                         |
| -M                                                            | 管理智能引用。当它们是操作的目标时，将删除包含引用的实际条目而不是删除通过跟踪引用获取的条目。                                                                                                                                                                                                                     |

- n** 显示应执行的操作，但不实际删除条目。可以与 **-v** 和 **-d** 选项一起用于调试。
- N *certificate*** 指定用于基于证书的客户端验证的证书名称。例如：**-N "Directory-Cert"**。
- o *attributename=value*** 用于 SASL 机制和其他选项，例如安全属性、运行模式、授权 ID、验证 ID 等。
- 各种属性名称及其值如下所示：
- secProp=*"number"*** 用于定义 SASL 安全属性。
- realm=*"value"*** 指定 SASL 领域（缺省值为 **realm=none**）。
- authzid=*"value"*** 指定用于 SASL 绑定的授权 ID 名称。
- authid=*"value"*** 指定用于 SASL 绑定的验证 ID。
- mech=*"value"*** 指定各种 SASL 机制。
- O *hopLimit*** 指定在查找要删除的条目时要遵循的引用跳数的最大数目。缺省情况下，没有任何限制。
- p *ldapport*** 指定 LDAP 服务器侦听的备用 TCP 端口。
- P *path*** 指定客户端证书数据库的路径和文件名。例如：
- P /home/uid/.netscape/cert7.db**
- 在与目录服务器相同的主机上使用命令时，可以使用服务器自己的证书数据库。例如：

|                                                          |                                                                                                                                                                                                                                                          |
|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-P installDir/\lapd-serverID/alias/cert7.db</code> | 单独使用 <code>-P</code> 选项将仅指定服务器验证。                                                                                                                                                                                                                        |
| <code>-v</code>                                          | 使用详细模式，将诊断信息写入到标准输出。                                                                                                                                                                                                                                     |
| <code>-v version</code>                                  | 指定要用于删除操作的 LDAP 协议版本号，2 或 3。LDAP v3 为缺省值。当连接到不支持 v3 的服务器时，请指定 LDAP v2。                                                                                                                                                                                   |
| <code>-W password</code>                                 | 指定在 <code>-P</code> 选项中给出的客户端密钥数据库的口令。对于基于证书的客户端验证，此选项是必需的。在命令行上指定 <code>password</code> 会有安全问题，因为系统上的其他人可以通过 <code>ps</code> 命令看到口令。请改用 <code>-j</code> 从文件中指定口令。此选项与 <code>-j</code> 互斥。                                                               |
| <code>-w passwd</code>                                   | 使用 <code>passwd</code> 作为用于对目录进行验证的口令。当使用 <code>-w passwd</code> 指定用于验证的口令时，系统的其他用户可以通过 <code>ps</code> 命令在脚本文件中或者在 <code>shell</code> 历史记录中看到口令。如果不使用此选项的情况下使用 <code>ldapdelete</code> 命令，则该命令将提示输入口令并从标准输入中读取口令。不与 <code>-w</code> 选项一起使用时，其他用户将看不到口令。 |
| <code>-Y proxyid</code>                                  | 指定要用于删除操作的代理 DN（被代理的授权 id），在 <code>shell</code> 中通常置于双引号（" "）中。                                                                                                                                                                                          |
| <code>-Z</code>                                          | 指定要用于提供基于证书的客户端验证的 SSL。此选项需要 <code>-N</code> 和 SSL 口令以及识别证书和密钥数据库所需的任何其他 SSL 选项。                                                                                                                                                                         |

## 操作数

支持下列操作数：

`dn` 指定要删除的条目的一个或多个标识名。

**示例**                   **示例 1 删除条目**  
 要删除紧跟在 XYZ 公司组织条目下的 commonName 为 Delete Me 的条目，请使用以下命令：

```
example% ldapdelete -D "cn=Administrator, o=XYZ, c=US" \
 "cn=Delete Me, o=XYZ, c=US"
```

**示例 2 删除使用 SASL 验证的条目**

要删除紧跟在 XYZ 公司组织条目下的 commonName 为 "Delete Me" 的条目，请使用以下命令：

```
example% ldapdelete -o mech=DIGEST-MD5 -o secProp=noanonymous \
 -o realm=none -o authid="dn:uid=foo,o=XYZ, c=US" \
 "cn=Delete Me, o=XYZ, c=US"
```

**属性**                   有关以下属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值             |
|-------|-----------------|
| 可用性   | system/core-os  |
| 接口稳定性 | Committed (已确定) |

**退出状态**            将返回以下退出值：

**0**                    成功完成。

**非零值**            出现错误。向标准错误写入一条诊断消息。

**另请参见**            [ldapadd\(1\)](#)、[ldapmodify\(1\)](#)、[ldapmodrdn\(1\)](#)、[ldapsearch\(1\)](#)、[ldap\\_get\\_option\(3LDAP\)](#)、[ldap\\_set\\_](#)

**附注**                    -M *authentication* 选项已过时。

- 引用名** ldaplist – 使用所配置的配置文件的 LDAP 目录搜索并列出命名信息
- 用法概要**
- ```
/usr/bin/ldaplist [-dlv] [-h LDAP_server[:serverPort]] [-M domainName]
  [-N profileName] [-a authenticationMethod] [-P certifPath]
  [-D bindDN] [-w bindPassword] [-j passwdFile]
  [database [key]...]
```
- ```
/usr/bin/ldaplist -g
```
- ```
/usr/bin/ldaplist -h
```
- 描述**
- 如果指定了 `-h LDAP_server[:serverPort]` 选项，则 `ldaplist` 将建立到该选项指定的服务器的连接，来获取 `-N` 选项指定的 `DUAProfile`。然后，`ldaplist` 从由所获得的配置描述的目录列出信息。
- 缺省情况下（如果未指定 `-h LDAP_server[:serverPort]` 选项），该实用程序将从 LDAP 目录服务搜索并列出命名信息，该目录服务是在客户端初始化阶段在 `ldapclient(1M)` 生成的 LDAP 配置文件中定义的。要以缺省模式使用该实用程序，必须预先设置 Oracle Solaris LDAP 客户端。
- `database` 是一个容器名称或一个数据库名称，如 `nsswitch.conf(4)` 中所定义。容器是包含命名服务信息的目录信息树 (Directory Information Tree, DIT) 中的非叶条目。容器名称是容器相对于 `defaultSearchBase` 的 LDAP 相对标识名 (Relative Distinguished Name, RDN)，如配置文件中所定义。例如，对于名为 `ou=people` 的容器，数据库名称是在 `nsswitch.conf` 中指定的数据库。该数据库映射到一个容器，例如，`passwd` 映射到 `ou=people`。如果指定了一个无效的数据库，则它将映射到一个通用容器，例如 `nisMapName=name`。
- `key` 是要在数据库中搜索的属性值。可以指定要在同一数据库中搜索的多个 `key`。可以通过两种方式中的任一方式来指定 `key`：`attribute=value` 或 `value`。在第一种情况下，`ldaplist` 将搜索 `key` 传递到服务器。在后一种情况下，将根据指定数据库的方式分配属性。如果 `database` 是一个容器名称，则会使用“`cn`”属性类型。如果 `database` 是如在 `nsswitch.conf` 中定义的一个有效数据库名称，则会使用预定义的属性类型（请参见下表）。如果 `database` 是一个无效的数据库名称，则 `cn` 将用作属性类型。
- `ldaplist` 实用程序依赖于在 RFC 2307bis（当前为 IETF 草稿）中定义的模式。LDAP 服务器上存储的数据必须基于该模式来存储，除非配置文件包含模式映射定义。有关模式映射的更多信息，请参见 `ldapclient(1M)`。下表列出了当没有在 `key` 中定义时，将使用的从数据库名称到容器的缺省映射、LDAP 对象类以及属性类型。

Database	Object Class	Attribute Type	Container
aliases	mailGroup	cn	ou=Aliases
automount	nisObject	cn	automountMapName=auto_*
bootparams	bootableDevice	cn	ou=Ethers
ethers	ieee802Device	cn	ou=Ethers
group	posixgroup	cn	ou=Group
hosts	ipHost	cn	ou=Hosts

ipnodes	ipHost	cn	ou=Hosts
netgroup	ipNetgroup	cn	ou=Netgroup
netmasks	ipNetwork	ipnetworknumber	ou=Networks
networks	ipNetwork	ipnetworknumber	ou=Networks
passwd	posixAccount	uid	ou=People
protocols	ipProtocol	cn	ou=Protocols
publickey	nisKeyObject	uidnumber	ou=People
		cn	ou=Hosts
rpc	oncRpc	cn	ou=Rpc
services	ipService	cn	ou=Services
printers	printerService	printer-uri	ou=printers
auth_attr	SolarisAuthAttr	nameT	ou=SolarisAuthAttr
prof_attr	SolarisProfAttr	nameT	ou=SolarisProfAttr
exec_attr	SolarisExecAttr	nameT	ou=SolarisProfAttr
user_attr	SolarisUserAttr	uidT	ou=people
projects	SolarisProject	SolarisProjectID	ou=projects

以下数据库只有当系统配置有 Trusted Extensions 时才可用：

tnrhtp	ipTnetTemplate	ipTnetTemplateName	ou=ipTnet
tnrhdb	ipTnetHost	ipTnetNumber	ou=ipTnet

- 对于 automount 数据库 auto_*, 在容器列中会呈现 auto_home、auto_direct 等等。
- 对于 publickey 数据库, 如果 key 以数字开头, 则它将被解释为一个 uid 编号。如果 key 以非数字开头, 则它将被解释为一个主机名。

ldaplist 实用程序支持通过在 key 中使用通配符 “*” 执行子字符串搜索。例如, “my*” 将匹配以 “my” 开头的任何字符串。在某些 shell 环境中, 包含通配符的 key 可能需要括在引号中。

如果没有指定 key, 则当前搜索 baseDN 中的所有容器都将列出。

选项

支持以下选项：

-a authenticationMethod 指定验证方法。缺省值是配置文件中配置的值。支持的验证方法包括：

```
simple
sasl/CRAM-MD5
sasl/DIGEST-MD5
tls:simple
tls:sasl/CRAM-MD5
tls:sasl/DIGEST-MD5
```

选择 simple 将导致口令以明文形式在网络中发送。强烈建议不要使用该方法。

- 此外，如果客户端配置有不使用验证的配置文件，也就是说，如果 *credentialLevel* 属性设置为 *anonymous* 或者 *authenticationMethod* 设置为 *none*，则用户必须使用此选项来提供验证方法。
- d 列出指定数据库的属性而不是条目。缺省情况下会列出条目。
 - D *bindDN* 指定对所请求的数据库具有读取权限的条目。
 - g 列出数据库映射。
 - h 列出数据库映射。
- 此选项已过时。
- h *LDAP_server[:serverPort]* 指定从中读取条目的 LDAP 服务器的地址（或名称）和端口。系统会使用 *nsswitch.conf* 文件中指定的当前命名服务。端口的缺省值为 389，除非在验证方法中指定了 TLS。在这种情况下，缺省 LDAP 服务器端口号为 636。
- 用于指定 IPv6 的地址和端口号的格式如下：
- [ipv6_addr]:port*
- 要为 IPv4 指定地址和端口号，请使用以下格式：
- ipv4_addr:port*
- 如果指定了主机名，请使用以下格式：
- host_name:port*
- j *passwdFile* 指定一个文件，该文件包含绑定 DN 的口令或包含 SSL 客户端密钥数据库的口令。要保护口令，请在脚本中使用此选项，并将口令存放在安全文件中。
- 此选项与 *-w* 选项互斥。
- l 列出与搜索条件匹配的每个条目的所有属性。缺省情况下，*ldaplist* 仅列出所找到的条目的标识名。
 - M *domainName* 指定由指定服务器提供服务的域名称。如果未指定此选项，则使用缺省域名称。
 - N *profileName* 指定 DUAProfile 名称。系统假定具有此名称的配置文件存在于由 *-H* 选项指定的服务器上。缺省值为 *default*。
 - p *certifPath* 指定通向证书数据库位置的证书路径。该值为安全数据库文件所在的路径。该值用于 TLS 支持，TLS 支持在

authenticationMethod 和 *serviceAuthenticationMethod* 属性中指定。缺省值为 `/var/ldap`。

-w bindPassword

要用于验证 *bindDN* 的口令。如果缺少此参数，则命令会提示输入口令。LDAP 不支持空口令。

当使用 **-w bind_password** 指定用于验证的口令时，系统的其他用户可以通过 `ps` 命令在脚本文件中或者在 shell 历史记录中看到口令。

如果提供 `-` 值作为口令，则命令会提示输入口令。

-v

设置详细模式。`ldaplist` 实用程序还输出用来搜索条目的过滤器。过滤器的前缀为“+++”。

示例

示例 1 列出主机数据库中的所有条目

以下示例列出 `hosts` 数据库中的所有条目：

```
example% ldaplist hosts
```

示例 2 列出非标准数据库 `ou=new` 中的所有条目

以下示例列出非标准数据库中的所有条目：

```
example% ldaplist ou=new
```

示例 3 在 `passwd` 数据库中查找 `user1`

以下示例在 `passwd` 数据库中查找 `user1`：

```
example% ldaplist passwd user1
```

示例 4 在 `services` 数据库中查找服务端口为 4045 的条目

以下示例在 `services` 数据库中查找服务端口为 4045 的条目：

```
example% ldaplist services ipServicePort=4045
```

示例 5 在 `passwd` 数据库中查找用户名以 `new` 开头的所有用户

以下示例在 `passwd` 数据库中查找用户名以 `new` 开头的所有用户：

```
example% ldaplist passwd 'new*'
```

示例 6 列出 `hosts` 数据库的属性

以下示例列出 `hosts` 数据库的属性：

```
example% ldaplist -d hosts
```

示例7 在 passwd 数据库中查找 user1

以下示例在 passwd 数据库中查找 user1。显式指定了一个 LDAP 服务器。

```
example% ldaplist -H 10.10.10.10:3890 \
    -M another.domain.name -N special_duaprofile \
    -D "cn=directory manager" -w secret \
    user1
```

退出状态

将返回以下退出值：

- 0 成功匹配了某些条目。
- 1 成功搜索了表但没有找到匹配项。
- 2 出现错误。输出了错误消息。

文件

/var/ldap/ldap_client_file

/var/ldap/ldap_client_cred

包含客户端的 LDAP 配置的文件。不要手动修改这些文件。其内容不保证是用户可读的。要更新这些文件，请使用 [ldapclient\(1M\)](#)

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/network/nis
接口稳定性	Committed (已确定)

另请参见

[ldapadd\(1\)](#)、[ldapdelete\(1\)](#)、[ldapmodify\(1\)](#)、[ldapmodrdn\(1\)](#)、[ldapsearch\(1\)](#)、[idsconfig\(1M\)](#)、

附注

RFC 2307bis 是一个处于草稿阶段的 IETF 信息文档，定义了将 LDAP 用作命名服务的方法。

当前，libldap.so.5 不支持 StartTLS，因此所提供的端口号指的是在打开 TLS 期间使用的端口，而不是用作 StartTLS 序列的一部分的端口。例如，`-h foo:1000 -a tls:simple` 指的是在主机 foo 上的原始 TLS 打开端口 1000，而不是在不安全端口 1000 上打开的 StartTLS 序列。如果端口 1000 是不安全的，则不会建立连接。

引用名 ldapmodify, ldapadd – ldap 条目添加和修改工具

用法概要

```
ldapmodify [-a] [-c] [-r] [-n] [-v] [-F] [-b] [-A] [-q]
  [-H] [-?] [-E] [-J] [-Z] [-M] [-d debuglevel]
  [-D bindDN] [-j filename] [-J [:criticality]]
  [-B baseDN] [-V version] [-Y proxyDN] [-O hopLimit]
  [-i locale] [-k path] [-e errorFile] [-P path]
  [-N certificate] [-w passwd] [-o attributename=value]
  [-h ldaphost] [-W password] [-p ldapport] [-f file]
  [-l nb-ldap-connections]
```

```
ldapadd [-c] [-n] [-v] [-F]
  [ [-b] [-A] [-q] [-H] [-?] [-E] [-J] [-Z] [-M] -d debuglevel]
  [-D bindDN] [-j filename] [-B baseDN] [-V version]
  [-Y proxyDN] [-O hopLimit] [-i locale] [-k path]
  [-e errorFile] [-P path] [-N certificate] [-w passwd]
  [-o attributename=value] [-h ldaphost] [-W password]
  [-p ldapport] [-f file] [-l nb-ldap-connections]
```

描述 ldapmodify 实用程序可打开与 LDAP 服务器的连接，绑定并修改或添加条目。条目信息是从标准输入或者从使用 -f 选项指定的 *file* 中读取的。ldapadd 实用程序是作为到 ldapmodify 工具的硬链接实现的。当作为 ldapadd 调用时，会自动打开 -a（添加新条目）选项。

ldapadd 和 ldapmodify 都拒绝同一条目的重复属性名/值对。

选项 支持以下选项：

-a	添加新条目。ldapmodify 的缺省行为是修改现有条目。如果作为 ldapadd 调用，则始终设置此选项。
-A	非 ASCII 模式：与 -v 选项一起使用，显示非 ASCII 值。
-b	处理二进制文件。ldapmodify 工具将扫描输入中的每个属性值以确定它是否为有效的文件引用。如果引用有效，它将使用文件的内容作为属性的值。此选项用来为属性输入二进制数据，例如 JPEG 图像。例如，对应的 LDIF 输入将是： jpegPhoto: /tmp/photo.jpg。ldapmodify 工具还支持用于直接包括文件内容的 LDIF :< URL 表示法。

<code>-B baseDN</code>	指定执行添加时的基 DN，在 shell 中通常置于双引号 (") 中。所有条目都将被置于该后缀下，从而提供批量导入功能。										
<code>-c</code>	指定连续运行模式。将会报告错误，但 <code>ldapmodify</code> 和 <code>ldapadd</code> 将继续进行修改操作。缺省行为是在报告错误后退出。										
<code>-D bindDN</code>	使用标识名 <code>bindDN</code> 绑定到目录。										
<code>-d debuglevel</code>	设置 LDAP 调试级别。适用于 <code>ldapmodify</code> 和 <code>ldapadd</code> 的有用调试级别包括： <table><tr><td>1</td><td>跟踪</td></tr><tr><td>2</td><td>包</td></tr><tr><td>4</td><td>参数</td></tr><tr><td>32</td><td>过滤器</td></tr><tr><td>128</td><td>访问控制</td></tr></table>	1	跟踪	2	包	4	参数	32	过滤器	128	访问控制
1	跟踪										
2	包										
4	参数										
32	过滤器										
128	访问控制										
<code>-e errorFile</code>	要请求多个类别的调试信息，请将掩码相加。例如，要请求跟踪和过滤器信息，请将 <code>debuglevel</code> 指定为 33。										
<code>-E</code>	输入中的无效更新语句将被复制到 <code>errorFile</code> 以进行调试。与 <code>-c</code> 选项一起使用可以更正处理大型 LDIF 输入时发生的错误。										
<code>-E</code>	通过验证响应控制，要求服务器公开（报告）绑定标识。										
<code>-F</code>	不管以 <code>replica:</code> 开头的输入行的内容为何，都强制应用所有更改。缺省情况下，将根据所使用的 LDAP 服务器主机和端口对 <code>replica:</code> 行进行比较，以确定是否应当应用 <code>repllog</code> 记录。										
<code>-f file</code>	从 <code>file</code> 而不是从标准输入读取条目修改信息。										

-?	显示简要描述所有选项的用法帮助文本。
-H	显示简要描述所有选项的用法帮助文本。
-h <i>ldaphost</i>	指定运行 LAPD 服务器的备用主机。
-i <i>locale</i>	指定用于 -f <i>LDIFfile</i> 或标准输入的字符集。缺省值是在 LANG 环境变量中指定的字符集。您可以选择使用此选项执行从指定字符集到 UTF8 的转换，从而覆盖 LANG 设置。
-j <i>filename</i>	指定绑定 DN 的口令或 SSL 客户机密钥数据库的口令所在的文件。要保护口令，请在脚本中使用此选项，并将口令存放在安全文件中。此选项与 -w 和 -W 选项互斥。
-J [:criticality[: value]::b64value b64value[: fileurl]]	Criticality 是一个布尔值（缺省值是 false）。
-k <i>path</i>	指定包含转换例程的目录路径。如果要指定目录服务器缺省情况下不支持的语言环境，则需要使用这些例程。这用于 NLS 支持。
-l <i>nb-ldap-connections</i>	指定 <i>ldapadd</i> 或 <i>ldapmodify</i> 将打开以处理目录中的修改的 LDAP 连接的数目。缺省值是一个连接。
-M	管理智能引用。当它们是操作的目标时，将修改包含引用的条目而不是修改通过跟踪引用获取的条目。
-n	预览修改，但不对条目进行修改。可以与 -v 和 -d 一起用于调试。
-N <i>certificate</i>	指定用于基于证书的客户端验证的证书名称。例如：-N "Directory-Cert"。
-o <i>attributename= value</i>	用于 SASL 机制和其他选项，例如安全属性、运行模式、授权 ID、验证 ID 等。

各种属性名称及其值如下所示：

<code>secProp="number"</code>	用于定义 SASL 安全属性。
<code>realm="value"</code>	指定 SASL 领域（缺省值为 <code>realm=none</code> ）。
<code>authzid="value"</code>	指定用于 SASL 绑定的授权 ID 名称。
<code>authid="value"</code>	指定用于 SASL 绑定的验证 ID。
<code>mech="value"</code>	指定各种 SASL 机制。

<code>-O hopLimit</code>	指定在查找要修改的条目时要遵循的引用跳数的最大数目。缺省情况下，没有任何限制。
<code>-p ldapport</code>	指定安全 LDAP 服务器侦听的备用 TCP 端口。
<code>-P path</code>	指定客户端证书数据库的路径和文件名。例如： <code>-P /home/uid/.netscape/cert7.db</code> 在与目录服务器相同的主机上使用命令时，可以使用服务器自己的证书数据库。例如：
<code>-P installDir/ldap-serverID/alias/cert7.db</code>	单独使用 <code>-P</code> 选项将仅指定服务器验证。
<code>-r</code>	用指定的值替换现有值。这是 <code>ldapmodify</code> 的缺省操作。当调用 <code>ldapadd</code> 时，或者如果指定了 <code>-a</code> 选项，则会忽略 <code>-r</code> 选项。
<code>-v</code>	使用详细模式，将诊断信息写入到标准输出。

<code>-v version</code>	指定要用于删除操作的 LDAP 协议版本号，2 或 3。LDAP v3 为缺省值。当连接到不支持 v3 的服务器时，请指定 LDAP v2。
<code>-W password</code>	指定在 <code>-p</code> 选项中给出的客户端密钥数据库的口令。对于基于证书的客户端验证，此选项是必需的。在命令行上指定 <code>password</code> 会有安全问题，因为系统上的其他人可以通过 <code>ps</code> 命令看到口令。请改用 <code>-j</code> 从文件中指定口令。此选项与 <code>-j</code> 互斥。
<code>-w passwd</code>	使用 <code>passwd</code> 作为用于对目录进行验证的口令。当使用 <code>-w passwd</code> 指定用于验证的口令时，系统的其他用户可以通过 <code>ps</code> 命令在脚本文件中或者在 <code>shell</code> 历史记录中看到口令。如果不使用此选项的情况下使用 <code>ldapmodify</code> 命令或 <code>ldapadd</code> 命令，则该命令将提示输入口令并从标准输入中读取口令。不与 <code>-w</code> 选项一起使用时，其他用户将看不到口令。
<code>-Y proxyid</code>	指定要用于修改操作的代理 DN（被代理的授权 id），在 <code>shell</code> 中通常置于双引号（" "）中。
<code>-Z</code>	指定要用于提供基于证书的客户端验证的 SSL。此选项需要 <code>-N</code> 和 SSL 口令以及识别证书和密钥数据库所需的任何其他 SSL 选项。

退出状态

将返回以下退出值：

- 0 成功完成。
- 非零值 出现错误。向标准错误写入一条诊断消息。

示例

以下示例中举例说明了 `file`（或标准输入，如果未指定 `-f` 选项）的内容格式。

示例 1 修改条目

文件 `/tmp/entrymods` 包含以下修改指令：

示例1 修改条目 (续)

```

dn: cn=Modify Me, o=XYZ, c=US
changetype: modify
replace: mail
mail: modme@atlanta.xyz.com
-
add: title
title: System Manager
-
add: jpegPhoto
jpegPhoto:< file:///tmp/modme.jpeg
-
delete: description
-

```

此命令：

```
example% ldapmodify -r -f /tmp/entrymods
```

如下所述修改 Modify Me 条目：

1. 将 mail 属性的当前值替换为值 modme@atlanta.xyz.com。
2. 添加一个值为 System Manager 的 title 属性。
3. 添加一个 jpegPhoto 属性（使用 /tmp/modme.jpeg 文件的内容作为属性值）。
4. 删除 description 属性。

示例2 创建新条目

文件 /tmp/newentry 包含用于创建新条目的以下信息：

```

dn: cn=Ann Jones, o=XYZ, c=US
objectClass: person
cn: Ann Jones
cn: Annie Jones
sn: Jones
title: Director of Research and Development
mail: ajones@londonrd.xyz.us.com
uid: ajones

```

以下命令

```
example% ldapadd -f /tmp/newentry
```

使用文件中的信息为 Ann Jones 添加一个新条目。

示例3 在 IPv6 服务器上创建一个新条目

文件 /tmp/newentry 包含用于在 IPv6 服务器上创建新条目的以下信息：

示例 3 在 IPv6 服务器上创建一个新条目 (续)

```
dn: cn=Ann Jones, o=XYZ, c=US
objectClass: person
cn: Ann Jones
cn: Annie Jones
sn: Jones
title: Director of Research and Development
mail: ajones@londonrd.xyz.us.com
uid: ajones
```

以下命令

```
example% ldapadd -c -v -h '[:fec0::111:a00:20ff:feaa:a364]':389 \
-D cn=Directory Manager -w secret \
-f /tmp/entry
```

使用文件中的信息为 Directory Manager 添加一个新条目。

示例 4 删除条目

文件 /tmp/badentry 包含关于要删除的条目的以下信息：

```
dn: cn=Ann Jones, o=XYZ, c=US
changetype: delete
```

此命令：

```
example% ldapmodify -f /tmp/badentry
```

删除 Ann Jones 的条目。

属性

有关以下属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed (已确定)

另请参见

[ldapdelete\(1\)](#)、[ldaplist\(1\)](#)、[ldapmodrdn\(1\)](#)、[ldapsearch\(1\)](#)、[ldapaddent\(1M\)](#)、[ldap_cachemgr\(1M\)](#)

引用名	ldapmodrdn – ldap 修改条目 RDN 工具											
用法概要	<pre>ldapmodrdn [-r] [-n] [-v] [-c] [-E] [-H] [-?] [-M] [-R] [-Z] [-V version] [-d debuglevel] [-D bindDN] [-w passwd] [-h ldaphost] [-i locale] [-j filename] [-J [:criticality]] [-k path] [-N certificate] [-O hopLimit] [-P path] [-W password] [-p ldapport] [-o attributename=value] [-f file] [-Y proxyDN] [dn rdn]</pre>											
描述	ldapmodrdn 可打开与 LDAP 服务器的连接，绑定并修改条目的 RDN。条目信息是从标准输入中读取的，通过使用 <code>-f</code> 选项从 <code>file</code> 中读取的，或者是从命令行对 <code>dn</code> 和 <code>rdn</code> 读取的。											
选项	<pre>-c -D bindDN -d debuglevel -E -f file -? -H</pre>	<p>连续运行模式。将会报告错误，但是 <code>ldapmodify</code> 将继续进行修改操作。缺省行为是在报告错误后退出。</p> <p>使用标识名 <code>binddn</code> 绑定到目录。</p> <p>设置 LDAP 调试级别。适用于 <code>ldapmodrdn</code> 的有用 <code>debuglevel</code> 值包括：</p> <table border="0"> <tr><td>1</td><td>跟踪</td></tr> <tr><td>2</td><td>包</td></tr> <tr><td>4</td><td>参数</td></tr> <tr><td>32</td><td>过滤器</td></tr> <tr><td>128</td><td>访问控制</td></tr> </table> <p>要请求多个类别的调试信息，请将掩码相加。例如，要请求跟踪和过滤器信息，请将 <code>debuglevel</code> 指定为 33。</p> <p>通过验证响应控制，要求服务器公开（报告）绑定标识。</p> <p>从 <code>file</code> 中而不是从标准输入或命令行读取条目修改信息。</p> <p>显示简要描述所有选项的用法帮助文本。</p> <p>显示简要描述所有选项的用法帮助文本。</p>	1	跟踪	2	包	4	参数	32	过滤器	128	访问控制
1	跟踪											
2	包											
4	参数											
32	过滤器											
128	访问控制											

-h <i>ldaphost</i>	指定运行 LDAP 服务器的备用主机。
-i <i>locale</i>	指定用于 -f <i>LDIFfile</i> 或标准输入的字符集。缺省值是在 LANG 环境变量中指定的字符集。您可以选择使用此选项执行从指定字符集到 UTF8 的转换，从而覆盖 LANG 设置。
-J [<i>:criticality[:value][:b64value b64value :fileurl]</i>]	Criticality 是一个布尔值（缺省值是 false）。
-j <i>filename</i>	指定绑定 DN 的口令或 SSL 客户机密钥数据库的口令所在的文件。要保护口令，请在脚本中使用此选项，并将口令存放在安全文件中。此选项与 -w 和 -W 选项互斥。
-k <i>path</i>	指定包含转换例程的目录路径。如果要指定目录服务器缺省情况下不支持的语言环境，则需要使用这些例程。这用于 NLS 支持。
-M	管理智能引用。当它们是操作的目标时，将修改包含引用的条目而不是修改通过跟踪引用获取的条目。
-n	预览修改，但不对条目进行修改。可以与 -v 和 -d 一起用于调试。
-N <i>certificate</i>	指定用于基于证书的客户端验证的证书名称。例如：-N "Directory-Cert"。
-n	显示应执行的操作，但不实际更改条目。可以与 -v 一起用于调试。
-o <i>attributename=value</i>	用于 SASL 机制和其他选项，例如安全属性、运行模式、授权 ID、验证 ID 等。
	各种属性名称及其值如下所示：
	<code>secProp="number"</code> 用于定义 SASL 安全属性。
	<code>realm="value"</code> 指定 SASL 领域（缺省值

	为 realm=none)。
<code>authzid="value"</code>	指定用于 SASL 绑定的 授权 ID 名 称。
<code>authid="value"</code>	指定用于 SASL 绑定的 验证 ID。
<code>mech="value"</code>	指定各种 SASL 机制。
<code>-O hopLimit</code>	指定在查找要修改的条目时要遵循 的引用跳数的最大数目。缺省情况 下，没有任何限制。
<code>-P path</code>	指定客户端证书数据库的路径和文 件名。例如： <code>-P /home/uid/.netscape/cert7.db</code> 在与目录服务器相同的主机上使用 命令时，可以使用服务器自己的证 书数据库。例如：
<code>-P installDir/lapd-serverID/alias/cert7.db</code>	单独使用 <code>-P</code> 选项将仅指定服务器验 证。
<code>-p ldapport</code>	指定安全 LAPD 服务器侦听的备用 TCP 端口。
<code>-R</code>	不自动跟踪在搜索时返回的引用。
<code>-r</code>	从条目中删除旧的 RDN 值。缺省情 况下会保留旧值。
<code>-v version</code>	指定要用于删除操作的 LDAP 协议 版本号，2 或 3。LDAP v3 为缺省 值。当连接到不支持 v3 的服务器 时，请指定 LDAP v2。
<code>-v</code>	使用详细模式，将诊断信息写入到 标准输出。
<code>-W password</code>	指定在 <code>-P</code> 选项中给出的客户端密钥 数据库的口令。对于基于证书的客

	<p>户端验证，此选项是必需的。在命令行上指定 <i>password</i> 会有安全问题，因为系统上的其他人可以通过 <i>ps</i> 命令看到口令。请改用 <i>-j</i> 从文件中指定口令。此选项与 <i>-j</i> 互斥。</p>
<i>-w password</i>	<p>使用 <i>password</i> 作为用于对目录进行验证的口令。当使用 <i>-w password</i> 指定用于验证的口令时，系统的其他用户可以通过 <i>ps</i> 命令在脚本文件中或者在 <i>shell</i> 历史记录中看到口令。如果不使用此选项的情况下使用 <i>ldapmodrdn</i> 命令，则该命令将提示输入口令并从标准输入中读取口令。不与 <i>-w</i> 选项一起使用时，其他用户将看不到口令。</p>
<i>-Y proxyid</i>	<p>指定要用于修改操作的代理 DN（被代理的授权 id），在 <i>shell</i> 中通常置于双引号（" "）中。</p>
<i>-Z</i>	<p>指定要用于提供基于证书的客户端验证的 SSL。此选项需要 <i>-N</i> 和 SSL 口令以及识别证书和密钥数据库所需的任何其他 SSL 选项。</p>

输入格式

如果给定了命令行参数 *dn* 和 *rdn*，则 *rdn* 将替换由 DN *dn* 指定的条目的 RDN。

否则，*file*（或标准输入，如果未指定 *-f* 选项）的内容必须包含一个或多个行对：

```
Distinguished Name (DN)
Relative Distinguished Name (RDN)
```

使用一个或多个空行来分隔每个 DN/RDN 对。

示例

文件 */tmp/entrymods* 包含：

```
cn=Modify Me, o=XYZ, c=US
cn=The New Me
```

此命令：

```
example% ldapmodify -r -f /tmp/entrymods
```

将 "Modify Me" 条目的 RDN 从 "Modify Me" 更改为 "The New Me"，并删除旧的 *cn* "Modify Me"。

属性

有关以下属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed（已确定）

另请参见

[ldapadd\(1\)](#)、[ldapdelete\(1\)](#)、[ldapmodify\(1\)](#)、[ldapsearch\(1\)](#)、[attributes\(5\)](#)

诊断

如果未发生错误，则退出状态为 0。错误会导致非零退出状态并且会将一条诊断消息写入到标准错误。

引用名 ldapsearch – ldap 搜索工具

用法概要 ldapsearch [-n] [-u] [-v] [-t] [-A] [-B] [-L] [-R] [-H] [-?] [-t] [-T] [-B] [-E] [-J] [-e] [-l] [-Z] [-r] [-M] [-d *debuglevel*] [-F *sep*] [-f *file*] [-D *bindDN*] [-j *filename*] [-V *version*] [-Y *proxyDN*] [-O *hopLimit*] [-i *locale*] [-k *path*] [-S [-] *attribute*] [-C *pattern*] [-c *authzid*] [-P *path*] [-N *certificate*] [-w *passwd*] [-h *ldaphost*] [-p *ldapport*] [-o *attributename=value*] [-b *searchbase*] [-s *scope*] [-a *deref*] [-l *timelimit*] [-z *sizelimit*] *filter* [*attrs*]...

描述 ldapsearch 实用程序可打开与 LDAP 服务器的连接，使用过滤器 *filter* 绑定并执行搜索。

如果 ldapsearch 找到一个或多个条目，则会检索由 *attrs* 指定的属性并且会将条目和值输出到标准输出。如果没有列出 *attrs*，则会返回所有属性。

输出格式 如果找到一个或多个条目，则会按以下格式将每个条目写入到标准输出：

```
dn: Distinguished Name (DN)
   attributename: value
   attributename: value
   attributename: value
...
```

多个条目以单个空行分隔。如果使用 -F 选项指定一个不同的分隔符字符，则会改用该字符而不使用 : 字符。如果使用 -t 选项，则会返回临时文件的名称而不是实际值。如果给定了 -A 选项，则只会返回“attributename”而不返回属性值。

选项 支持以下选项：

- A 仅检索属性（不检索值）。如果只想了解条目中是否存在某个属性而对具体值不感兴趣，则该选项非常有用。
- a *deref* 指定如何执行别名引用解除。*deref* 的可能值有 *never*、*always*、*search* 或 *find*，分别指定从不解除别名引用、始终解除别名引用、在搜索时解除别名引用，或只在查找要搜索的基对象时解除别名引用。缺省行为是从不解除别名引用。
- B 显示非 ASCII 值并使用旧的非 LDIF 格式。此选项禁用缺省的 -L 选项。
- b *searchbase* 使用 *searchbase* 而非缺省值作为搜索的起点。

-C <i>pattern</i>	持久性搜索。执行搜索时使连接保持打开状态，并且每当添加、修改或删除与搜索的范围或过滤器匹配的条目时都显示结果。使用此选项， <code>ldapsearch</code> 工具将无限期运行，必须通过键入 <code>Ctrl-c</code> 来停止它。 <i>pattern</i> 采用以下格式：										
<pre>ps:changeType[:changesOnly[:entryChangeControls]]</pre>											
-c <i>authzid</i>	指定 <code>getEffectiveRights</code> 控件 <i>authzid</i> 。例如： <pre>dn:uid=bjensen,dc=example,dc=com</pre>										
-D <i>bindDN</i>	使用标识名 <i>bindDN</i> 绑定到目录。										
-d <i>debuglevel</i>	设置 LDAP 调试级别。适用于 <code>ldapsearch</code> 的有用调试级别包括： <table border="0" style="margin-left: 20px;"> <tr><td>1</td><td>跟踪</td></tr> <tr><td>2</td><td>包</td></tr> <tr><td>4</td><td>参数</td></tr> <tr><td>32</td><td>过滤器</td></tr> <tr><td>128</td><td>访问控制</td></tr> </table> <p>要请求多个类别的调试信息，请将掩码相加。例如，要请求跟踪和过滤器信息，请将 <code>debuglevel</code> 指定为 33。</p>	1	跟踪	2	包	4	参数	32	过滤器	128	访问控制
1	跟踪										
2	包										
4	参数										
32	过滤器										
128	访问控制										
-E	通过验证响应控制，要求服务器公开（报告）绑定标识。										
-e	最小化值的 base-64 编码。										
-F <i>sep</i>	使用 <i>sep</i> 作为属性名称与值之间的字段分隔符。如果指定了此选项，则会忽略 -L 选项。										
-f <i>file</i>	从 <i>file</i> 中读取一系列行，针对每行执行一次 LDAP 搜索。在这种情况下，在命令行上给定的 <i>filter</i> 将被视为一个模式，其中， <code>%s</code> 的第一个实例将被替换为 <i>file</i> 中的一行。如果 <i>file</i> 是单个 - 字符，则会从标准输入中读取行。										

- G *pattern*** 虚拟列表视图。仅检索所有结果的一部分，具体由搜索目标的索引或值和为目标前后要返回的条目数目来确定。此选项始终要求使用 **-s** 和 **-x** 选项来指定服务器上的排序顺序。
- ?** 显示简要描述所有选项的用法帮助文本。
- H** 显示简要描述所有选项的用法帮助文本。
- h *ldaphost*** 指定运行安全 LDAP 服务器的备用主机。
- i *locale*** 指定用于命令行输入的字符集。缺省值是在 **LANG** 环境变量中指定的字符集。您可能希望使用此选项执行从指定的字符集到 UTF8 的转换，从而覆盖 **LANG** 设置。使用此参数，可以采用指定的字符集输入绑定 DN、基 DN 和搜索过滤器模式。**ldapsearch** 工具在处理搜索请求之前基于这些参数转换输入。例如，**-i no** 指示将以挪威语提供绑定 DN、基 DN 和搜索过滤器。此参数仅影响命令行输入。如果指定了包含搜索过滤器的文件（通过 **-f** 选项），则 **ldapsearch** 不会转换文件中的数据。
- j *filename*** 指定绑定 DN 的口令或 SSL 客户机密钥数据库的口令所在的文件。要保护口令，请在脚本中使用此选项，并将口令存放在安全文件中。此选项与 **-w** 和 **-W** 选项互斥。
- J [*criticality*[:*value*][:*b64value*|*b64value*][:*fileurl*]]** **Criticality** 是一个布尔值（缺省值是 **false**）。
- k *path*** 指定包含转换例程的目录路径。如果要指定目录服务器缺省情况下不支持的语言环境，则需要使用这些例程。这用于 NLS 支持。

- L 以 LDIF 格式显示搜索结果。此选项还会打开 -B 选项。此行为是缺省行为。
- l *timelimit* 最多等待 *timelimit* 秒后搜索完成。
- M 管理智能引用。当它们是操作的目标时，将搜索包含引用的条目而不是搜索通过跟踪引用获取的条目。
- N *certificate* 指定用于基于证书的客户端验证的证书名称。例如：-N “Directory-Cert”。
- n 显示应执行的操作，但不实际执行搜索。可以与 -v 和 -d 一起用于调试。
- O *hopLimit* 指定在查找要修改的条目时要遵循的引用跳数的最大数目。缺省情况下，没有任何限制。
- o *attributename=value* 用于 SASL 机制和其他选项，例如安全属性、运行模式、授权 ID、验证 ID 等。
- 各种属性名称及其值如下所示：
- | | |
|-------------------------------|--|
| <code>secProp=“number”</code> | 用于定义 SASL 安全属性。 |
| <code>realm=“value”</code> | 指定 SASL 领域（缺省值为 <code>realm=none</code> ）。 |
| <code>authzid=“value”</code> | 指定用于 SASL 绑定的授权 ID 名称。 |
| <code>authid=“value”</code> | 指定用于 SASL 绑定的验证 ID。 |
| <code>mech=“value”</code> | 指定各种 SASL 机制。 |
- P *path* 指定客户端证书数据库的路径和文件名。例如：

	-P /home/uid/.netscape/cert7.db
	在与目录服务器相同的主机上使用命令时，可以使用服务器自己的证书数据库。例如：
-P <i>installDir/ldap-serverID/alias/cert7.db</i>	单独使用 -P 选项将仅指定服务器验证。
-p <i>ldapport</i>	指定安全 LDAP 服务器侦听的备用 TCP 端口。
-R	不自动跟踪在搜索时返回的引用。
-r	以旧格式显示 ldapsearch 命令的输出。
-S [-] <i>attribute</i>	指定用于对搜索返回的条目进行排序的属性。排序条件按照属性值的字母顺序排序或按照格式 - <i>attribute</i> 的反向字母顺序排序。您可以给定多个 -S 选项来细化排序，例如： -S sn -S <i>givenname</i> 缺省情况下，这些条目未排序。使用 -x 选项执行服务器端排序。
-s <i>scope</i>	指定搜索的范围。 <i>scope</i> 的可能值有 base 、 one 或 sub ，分别用来指定基对象、一层或子树搜索。缺省值为 sub 。
-T	设置搜索结果的输出格式，以便在各个属性值内不使用换行符。
-t	将检索到的值写入到一组临时文件。这对于处理非 ASCII 值（例如 jpegPhoto 或 audio ）非常有用。
-U	URL 格式（仅在与 -t 选项一起使用时有效）。当使用临时文件输出时，该工具的标准输出包括文件的 URL 而不是属性值。例如： jpegPhoto:< file:/tmp/ldapsearch-jpegPhoto-YzaOMh

- u** 在输出中包括标识名 (Distinguished Name, DN) 的用户友好形式。
- v *version*** 指定要用于删除操作的 LDAP 协议版本号, 2 或 3。LDAP v3 为缺省值。当连接到不支持 v3 的服务器时, 请指定 LDAP v2。
- v** 在详细模式下运行, 将诊断信息写入到标准输出。
- W *password*** 指定在 **-P** 选项中给出的客户端密钥数据库的口令。对于基于证书的客户端验证, 此选项是必需的。在命令行上指定 *password* 会有安全问题, 因为系统上的其他人可以通过 **ps** 命令看到口令。请改用 **-j** 从文件中指定口令。此选项与 **-j** 互斥。
- w *passwd*** 使用 *passwd* 作为用于对目录进行验证的口令。当使用 **-w *passwd*** 指定用于验证的口令时, 系统的其他用户可以通过 **ps** 命令在脚本文件中或者在 **shell** 历史记录中看到口令。如果不使用此选项的情况下使用 **ldapsearch** 命令, 则该命令将提示输入口令并从标准输入中读取口令。当不与 **-w** 选项一起使用时, 其他用户将看不到口令。
- x** 与 **-s** 选项一起使用以指定搜索结果在服务器上进行排序而不是由在客户端上运行的 **ldapsearch** 命令进行排序。如果希望根据某个匹配规则进行排序 (例如使用国际搜索时), 这非常有用。如果受支持, 在服务器上进行排序通常比在客户端上进行排序更快。
- Y *proxyDN*** 指定要用于修改操作的代理 DN (被代理的授权 id), 在 **shell** 中通常置于双引号 ("") 中。
- Z** 指定要用于提供基于证书的客户端验证的 SSL。此选项需要 **-N** 和 SSL 口令以及识别证书和密钥数据库所需的任何其他 SSL 选项。

-z *sizelimit*

最多检索 *sizelimit* 个条目后搜索完成。

示例

示例1 执行子树搜索

以下命令执行子树搜索（使用缺省的搜索基）来查找 `commonName` 为“mark smith”的条目。将检索 `commonName` 和 `telephoneNumber` 值并将其输出到标准输出。使用 `-r` 选项将以旧格式显示此输出。

```
example% ldapsearch "cn=mark smith" cn telephoneNumber
```

输出结果看起来类似于以下内容：

```
dn: Mark D Smith, ou=Sales, ou=Atlanta, ou=People, o=XYZ, c=US
cn: Mark Smith
cn: Mark David Smith
cn: Mark D Smith 1
cn: Mark D Smith
telephoneNumber: +1 123 456-7890
```

```
dn: Mark C Smith, ou=Distribution, ou=Atlanta, ou=People, o=XYZ, c=US
cn: Mark Smith
cn: Mark C Smith 1
cn: Mark C Smith
telephoneNumber: +1 123 456-9999
```

示例2 使用缺省搜索基执行子树搜索

以下命令使用 `-r` 选项通过缺省搜索基执行子树搜索来以旧格式显示用户 `id` 为 `mcs` 的条目。将在包含 DN 本身的行后输出条目 DN 的用户友好形式，并且将检索 `jpegPhoto` 和 `audio` 值并将其写入到临时文件中。

```
ldapsearch -r -u -t "uid=mcs" -r jpegPhoto audio
```

如果找到了对于所请求的每个属性都有一个值的一个条目，则输出结果可能看起来类似于以下内容：

```
cn=Mark C Smith, ou=Distribution, ou=Atlanta, ou=People, o=XYZ, c=US
Mark C Smith, Distribution, Atlanta, People, XYZ, US
audio=/tmp/ldapsearch-audio-a19924
jpegPhoto=/tmp/ldapsearch-jpegPhoto-a19924
```

示例3 执行单层搜索

以下命令在 `organizationName` 以 `XY` 开头的所有组织的 `c=US` 层执行单层搜索。

```
example% ldapsearch -s one -b "c=US" "o=XY*" o description
```

将检索 `organizationName` 和 `description` 属性值并将其输出到标准输出，最后所得到的输出结果类似于以下内容：

示例 3 执行单层搜索 (续)

```
dn: o=XYZ      c=US
    o: XYZ
    description: XYZ Corporation

dn: o="XY Trading Company", c=US
    o: XY Trading Company
    description: Import and export specialists

dn: o=XYInternational, c=US
    o: XYInternational
    o: XYI
    o: XY International
```

示例 4 在 IPv6 服务器上执行子树搜索

以下命令使用缺省搜索基在 IPv6（即 -h）服务器上执行子树搜索来查找用户 id 为 mcs 的条目。

```
example% ldapsearch -u -h '['fec0::111:a00:20ff:fea3:edcf']' \
    -t "uid=mcs" jpegPhoto audio
```

退出状态

将返回以下退出值：

- 0 成功完成。
- >0 出现错误。向标准错误写入一条诊断消息。

属性

有关以下属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed（已确定）

另请参见

[ldapadd\(1\)](#)、[ldapdelete\(1\)](#)、[ldapmodify\(1\)](#)、[ldapmodrdn\(1\)](#)、[attributes\(5\)](#)

引用名 ldd – list dynamic dependencies of executable files or shared objects

用法概要 ldd [-d | -r] [-c] [-D] [-e *envar*] [-f] [-i] [-L] [-l] [-p]
[-s] [-U | -u] [-v] [-w] *filename*...

描述 The ldd utility lists the dynamic dependencies of executable files or shared objects. ldd uses the runtime linker, ld.so.1, to generate the diagnostics. The runtime linker takes the object being inspected and prepares the object as would occur in a running process. By default, ldd triggers the loading of any lazy dependencies, and deferred dependencies.

ldd lists the path names of all shared objects that would be loaded when *filename* is loaded. ldd expects the shared objects that are being inspected to have execute permission. If a shared object does not have execute permission, ldd issues a warning before attempting to process the file.

ldd processes its input one file at a time. For each file, ldd performs one of the following:

- Lists the object dependencies if the dependencies exist.
- Succeeds quietly if dependencies do not exist.
- Prints an error message if processing fails.

The dynamic objects that are inspected by ldd are not executed. Therefore, ldd does not list any shared objects explicitly attached using `dlopen(3C)`. To display all the objects in use by a process, or a core file, use `pldd(1)`.

选项 ldd can also check the compatibility of *filename* with the shared objects *filename* uses. With the following options, ldd prints warnings for any unresolved symbol references that would occur when *filename* is loaded.

- d Check *immediate* references.
- r Check both *immediate* references and *lazy* references.

Only one of the options -d or -r can be specified during any single invocation of ldd.

immediate references are typically to data items used by the executable or shared object code. *immediate* references are also pointers to functions, and even calls to functions made from a position *dependent* shared object. *lazy* references are typically calls to global functions made from a position *independent* shared object, or calls to external functions made from an executable. For more information on these types of reference, see When Relocations Are Performed in the 《链接程序和库指南》. Object loading can also be affected by relocation processing. See Lazy Loading under USAGE for more details.

Some unresolved symbol references are not reported by default. These unresolved references can be reported with the following options. These options are only useful when combined with either the -d or the -r options.

- p Expose any unresolved symbol errors to explicit *parent* and *external* references.
- w Expose any unresolved *weak* symbol references.

A shared object can make reference to symbols that should be supplied by the caller of the shared object. These references can be explicitly classified when the shared object is created, as being available from a *parent*, or simply as being *external*. See the `-Mmapfile` option of `ld(1)`, and the `PARENT` and `EXTERN` symbol definition keywords. When examining a dynamic executable, a *parent* or *external* reference that can not be resolved is flagged as an error. However by default, when examining a shared object, a *parent* or *external* reference that can not be resolved is not flagged as an error. The `-p` option, when used with either the `-d` or `-r` options, causes any unresolved *parent* or *external* reference to be flagged as a relocation error.

Symbols that are used by relocations may be defined as *weak* references. By default, if a weak symbol reference can not be resolved, the relocation is ignored and a zero written to the relocation offset. The `-w` option, when used with either the `-d` or the `-r` options, causes any unresolved relocation against a weak symbol reference to be flagged as a relocation error.

`ldd` can also check dependency use. With each of the following options, `ldd` prints warnings for any unreferenced, or unused dependencies that are loaded when *filename* is loaded. Only when a symbol reference is bound to a dependency, is that dependency deemed used. These options are therefore only useful when symbol references are being checked. If the `-r` option is not in effect, the `-d` option is enabled.

A dependency that is defined by an object but is not bound to from that object is an unreferenced dependency. A dependency that is not bound to by any other object when *filename* is loaded is an unused object.

Dependencies can be located in default system locations, or in locations that must be specified by search paths. Search paths may be specified globally, such as the environment variable `LD_LIBRARY_PATH`. Search paths can also be defined in dynamic objects as `runpaths`. See the `-R` option to `ld(1)`. Search paths that are not used to satisfy any dependencies cause unnecessary file system processing.

`-U` Displays any unreferenced, or unused dependencies. If an unreferenced dependency is not bound to by other objects loaded with *filename*, the dependency is also flagged as unused. Cyclic dependencies that are not bound to from objects outside of the cycle are also deemed unreferenced.

This option also displays any unused search paths.

`-u` Displays any unused objects.

Only one of the options `-U` or `-u` can be specified during any single invocation of `ldd`, although `-U` is a superset of `-u`. Objects that are found to be unreferenced, or unused when using the `-r` option, should be removed as dependencies. These objects provide no references, but result in unnecessary overhead when *filename* is loaded. When using the `-d` option, any objects that are found to be unreferenced, or unused are not immediately required when *filename* is loaded. These objects are candidates for lazy loading. See `Lazy Loading` under `USAGE` for more details.

The removal of unused dependencies reduces runtime-linking overhead. The removal of unreferenced dependencies reduces runtime-linking overhead to a lesser degree. However, the removal of unreferenced dependencies guards against a dependency being unused when combined with different objects, or as the other object dependencies evolve.

The removal of unused search paths can reduce the work required to locate dependencies. This can be significant when accessing files from a file server over a network. Note, a search path can be encoded within an object to satisfy the requirements of `dlopen(3C)`. This search path might not be required to obtain the dependencies of this object, and hence will look unused to `ldd`.

The following additional options are supported:

- c Disables any configuration file use. Configuration files can be employed to alter default search paths, and provide alternative object dependencies. See `crle(1)`.
- D Skip deferred dependency loading. By default, `ldd` forces the processing of both lazy dependencies and deferred dependencies. See also the `-L` option. During normal process execution, deferred dependencies are only loaded when the first runtime binding to a deferred reference is made. When using the `-D` option, the use of the `-d` or `-r` options do not trigger the loading of any deferred dependencies. See the `-z deferred` option of `ld(1)`.
- e *envar* Sets the environment variable *envar*.

This option is useful for experimenting with environment variables that are recognized by the runtime linker that can adversely affect `ldd`, for example, `LD_PRELOAD`.

This option is also useful for extracting additional information solely from the object under inspection, for example, `LD_DEBUG`. See `ld.so.1(1)` and `lari(1)`.
- f Forces `ldd` to check for an executable file that is not secure. When `ldd` is invoked by a superuser, by default `ldd` does not process any executable that is not secure. An executable is not considered secure if the interpreter that the executable specifies does not reside under `/lib` or `/usr/lib`. An executable is also not considered secure if the interpreter cannot be determined. See `Security` under `USAGE`.
- i Displays the order of execution of initialization sections. The order that is discovered can be affected by use of the `-d` or `-r` options. See `Initialization Order` under `USAGE`.
- L Enables lazy loading. By default, `ldd` forces the processing of both lazy dependencies and deferred dependencies. See also the `-D` option. During normal process execution, lazy loading is the default mode of operation. In this case, any lazy dependencies, or filters, are only loaded into the process when reference is

made to a symbol that is defined within the lazy object. The `-d` or `-r` options, together with the `-L` option, can be used to inspect the dependencies, and their order of loading as would occur in a running process. See the `-z lazyload` option of `ld(1)`.

- `-l` Forces the immediate processing of any filters so that all filtees, and their dependencies, are listed. The immediate processing of filters is now the default mode of operation for `ldd`. However, under this default any auxiliary filtees that cannot be found are silently ignored. Under the `-l` option, missing auxiliary filtees generate an error message.
- `-s` Displays the search path used to locate shared object dependencies.
- `-v` Displays all dependency relationships incurred when processing *filename*. This option also displays any dependency version requirements. See `pvs(1)`.

用法

Security

A superuser should use the `-f` option only if the executable to be examined is known to be trustworthy. The use of `-f` on an untrustworthy executable while superuser can compromise system security. If an executables trustworthyness is unknown, a superuser should temporarily become a regular user. Then invoke `ldd` as this regular user.

Untrustworthy objects can be safely examined with `dump(1)`, `elfdump(1)`, `elfedit(1)`, and with `mdb(1)`, as long as the `:r` subcommand is not used. In addition, a non-superuser can use either the `:r` subcommand of `mdb`, or `truss(1)` to examine an untrustworthy executable without too much risk of compromise. To minimize risk when using `ldd`, `mdb :r`, or `truss` on an untrustworthy executable, use the UID "nobody".

Lazy Loading

Lazy loading can be applied directly by specified lazy dependencies. See the `-z lazyload` option of `ld(1)`. Lazy loading can also be applied indirectly through filters. See the `-f` option and `-F` option of `ld(1)`. Objects that employ lazy loading techniques can experience variations in `ldd` output due to the options used. If an object expresses all its dependencies as lazy, the default operation of `ldd` lists all dependencies in the order in which the dependencies are recorded in that object:

```
example% ldd main
libelf.so.1 => /lib/libelf.so.1
libnsl.so.1 => /lib/libnsl.so.1
libc.so.1 => /lib/libc.so.1
```

The lazy loading behavior that occurs when this object is used at runtime can be enabled using the `-L` option. In this mode, lazy dependencies are loaded when reference is made to a symbol that is defined within the lazy object. Therefore, combining the `-L` option with use of the `-d` and `-r` options reveals the dependencies that are needed to satisfy the immediate, and lazy references respectively:

```

example% ldd -L main
example% ldd -d main
      libc.so.1 => /lib/libc.so.1
example% ldd -r main
      libc.so.1 => /lib/libc.so.1
      libelf.so.1 => /lib/libelf.so.1

```

Notice that in this example, the order of the dependencies that are listed is not the same as displayed from `ldd` with no options. Even with the `-r` option, the lazy reference to dependencies might not occur in the same order as would occur in a running program.

Observing lazy loading can also reveal objects that are not required to satisfy any references. These objects, in this example, `libnsl.so.1`, are candidates for removal from the link-line used to build the object being inspected.

Initialization Order

Objects that do not explicitly define their required dependencies might observe variations in the initialization section order displayed by `ldd` due to the options used. For example, a simple application might reveal:

```

example% ldd -i main
      libA.so.1 => ./libA.so.1
      libc.so.1 => /lib/libc.so.1
      libB.so.1 => ./libB.so.1

      init object=./libB.so.1
      init object=./libA.so.1
      init object=/lib/libc.so.1

```

whereas, when relocations are applied, the initialization section order is:

```

example% ldd -ir main
      .....

      init object=/lib/libc.so.1
      init object=./libB.so.1
      init object=./libA.so.1

```

In this case, `libB.so.1` makes reference to a function in `/usr/lib/libc.so.1`. However, `libB.so.1` has no explicit dependency on this library. Only after a relocation is discovered is a dependency then established. This implicit dependency affects the initialization section order.

Typically, the initialization section order established when an application is executed, is equivalent to `ldd` with the `-d` option. The optimum order can be obtained if all objects fully define their dependencies. Use of the `ld(1)` options `-z defs` and `-z ignore` when building dynamic objects is recommended.

Cyclic dependencies can result when one or more dynamic objects reference each other. Cyclic dependencies should be avoided, as a unique initialization sort order for these dependencies can not be established.

Users that prefer a more static analysis of object files can inspect dependencies using tools such as [dump\(1\)](#) and [elfdump\(1\)](#).

文件

`/usr/lib/lddstub` Fake 32-bit executable loaded to check the dependencies of shared objects.

`/usr/lib/64/lddstub` Fake 64-bit executable loaded to check the dependencies of shared objects.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/linker

另请参见

[crle\(1\)](#), [dump\(1\)](#), [elfdump\(1\)](#), [elfedit\(1\)](#), [lari\(1\)](#), [ld\(1\)](#), [ld.so.1\(1\)](#), [mdb\(1\)](#), [pldd\(1\)](#), [pvs\(1\)](#), [truss\(1\)](#), [dlopen\(3C\)](#), [attributes\(5\)](#)

《链接程序和库指南》

诊断

`ldd` prints the record of shared object path names to `stdout`. The optional list of symbol resolution problems is printed to `stderr`. If *filename* is not an executable file or a shared object, or if *filename* cannot be opened for reading, a non-zero exit status is returned.

附注

Use of the `-d` or `-r` option with shared objects can give misleading results. `ldd` does a worst case analysis of the shared objects. However, in practice, the symbols reported as unresolved might be resolved by the executable file referencing the shared object. The runtime linker's preloading mechanism can be employed to add dependencies to the object being inspected. See `LD_PRELOAD`.

`ldd` uses the same algorithm as the runtime linker to locate shared objects.

引用名 ld.so.1 – 动态目标文件的运行时链接程序

用法概要 /lib/ld.so.1

/lib/ld.so.1 [-e *envar*] *dynamic-object* [*object args*]...

描述 动态应用程序包含一个或多个动态目标文件。动态应用程序通常是指动态可执行文件，以及一个或多个共享目标文件依赖项。在动态应用程序的初始化和执行过程中，将调用**解释器**。该解释器可将应用程序与其共享目标文件依赖项绑定在一起。在 Solaris 中，该解释器也称为运行时链接程序。

在动态可执行文件的链接编辑期间，将创建一个特殊的 `.interp` 部分，以及一个关联的程序头。此节包含用于指定程序的解释器的路径名。将 `-I` 选项用于链接编辑器 `ld(1)` 来构造可执行文件时，可指定解释器路径名。由链接编辑器提供的缺省名称是运行时链接程序的名称 `ld.so.1`。

在动态可执行文件的执行过程中，内核将映射文件和所需解释器的定位。请参见 `exec(2)` 和 `mmapobj(2)`。内核将映射到该解释器中并将控制权传输给该解释器。将向解释器传递足够的信息，使之能够继续绑定，然后执行应用程序。

除了初始化应用程序，运行时链接程序还可以向应用程序提供相关服务以扩展其地址空间。此外，还可以映射其他共享目标文件并绑定到共享目标文件中的符号。

运行时链接程序可执行以下功能：

- 处理配置文件（如果存在）。配置文件可用于更改缺省搜索路径、提供目录高速缓存，以及提供备选目标文件依赖项。请参见 `crle(1)`。缺省情况下，对于 32 位目标文件，将使用配置文件 `/var/ld/ld.config`。对于 64 位目标文件，将使用缺省配置文件 `/var/ld/64/ld.config`。备选配置文件可通过 `LD_CONFIG` 环境变量指定。此外，还可以使用 `ld(1)` 的 `-c` 选项在动态可执行文件内对备选配置文件进行编码。
- 运行时链接程序可以分析应用程序的动态信息部分 `.dynamic`，从而确定哪些共享目标文件依赖项是必需的。
- 然后，运行时链接程序会定位并映射到这些依赖项中，并接着分析每个依赖项的动态信息部分，以便确定是否需要任何其他依赖项。
- 装入所有共享目标文件依赖项以后，运行时链接程序将执行任何必要的重定位操作。这些重定位操作可以绑定共享目标文件，为进程执行做准备。
- 将调用共享目标文件依赖项以及（可能）动态可执行文件提供的初始化函数。这些函数的调用顺序与依赖项的拓扑排序顺序相反。如果存在循环依赖项，将使用排序顺序调用初始化函数并删除循环。`ldd(1)` 可用于显示共享目标文件依赖项的初始化顺序。
- 控制权将传递给应用程序。
- 在应用程序执行期间，可调用运行时链接程序来执行任何延迟的函数绑定操作。
- 如果从进程中删除任何共享目标文件，将调用结束化函数。缺省情况下，这些函数的调用顺序与依赖项的拓扑排序顺序相同。

- 应用程序还可以使用 `dlopen(3C)` 来调用运行时链接程序的服务以获取其他共享目标文件。可以使用 `dlsym(3C)` 绑定到这些目标文件提供的符号。

《链接程序和库指南》中提供了此前每一个主题的更多详细信息。

运行时链接程序使用规定的搜索路径定位目标文件的动态依赖项。缺省搜索路径为目标文件中记录的 `runpath`，后跟一系列缺省值。32 位目标文件的缺省搜索路径为 `/lib`，后跟 `/usr/lib`。64 位目标文件的缺省搜索路径为 `/lib/64`，后跟 `/usr/lib/64`。这些缺省部分可以使用通过 `crle(1)` 创建的配置文件进行修改。将 `-R` 选项用于 `ld(1)` 以构造动态目标文件时，将指定 `runpath`。可以使用环境变量 `LD_LIBRARY_PATH` 来表示将先于缺省目录进行搜索的目录。

命令行使用

通常情况下，运行时链接程序是通过执行动态可执行文件间接调用的，动态可执行文件将运行时链接程序声明为它的解释器。此外，也可通过命令行直接执行运行时链接程序。该机制最常用来尝试运行时链接程序的新的实现。命令行上提供的参数包含适用于运行时链接程序的选项。这些选项后跟要执行的动态目标文件的名称，以及该目标文件所需的任何选项。事实上，运行时链接程序将替换动态目标文件所指定的任何解释器。

支持以下选项：

`-e envvar` 指定特定于运行时链接程序的环境变量。请参见“环境变量”部分。使用该选项设置的变量将优先于任何环境变量或具有相同名称的配置文件变量。可指定变量 `LD_NOENVIRON` 来指示在 `-e` 选项处理后不应处理任何环境变量。

环境变量

每个环境变量均可使用 `_32` 或 `_64` 后缀进行指定。这样，环境变量将分别特定于 32 位或 64 位进程。该环境变量将覆盖任何可能有效的环境变量的无后缀版本。如果环境变量指定时没有提供值且其具有 `_32` 或 `_64` 后缀，则事实上会取消任何关联的通用环境变量设置。

`LD_AUDIT`、`LD_AUDIT_32` 和 `LD_AUDIT_64`

运行时链接程序装入的目标文件的冒号分隔列表。装入每个目标文件时，将检查目标文件的链接审计接口例程。将按照《链接程序和库指南》中所述的链接审计接口中指定的那样调用存在的例程。另请参见 `ld(1)` 的 `-p` 和 `-P` 选项。

`LD_BIND_LAZY`、`LD_BIND_LAZY_32` 和 `LD_BIND_LAZY_64`

可以将环境变量 `LD_BIND_LAZY` 设置为任意非空值，从而强制实施运行时链接程序执行延迟绑定的缺省模式。此设置使运行时链接程序对所有已装入进程的目标文件都只执行延迟引用重定位。各目标文件可以请求在装入目标文件时执行延迟引用重定位。请参见 `ld(1)` 的 `-z now` 选项，以及模式为 `RTLD_NOW` 的 `dlopen(3C)`。 `LD_BIND_LAZY` 有效时，将抑制用于执行延迟引用重定位的此类请求。

如果同时指定 `LD_BIND_LAZY` 和 `LD_BIND_NOW`，则 `LD_BIND_NOW` 优先。

`LD_BIND_NOW`、`LD_BIND_NOW_32` 和 `LD_BIND_NOW_64`

可以将环境变量 `LD_BIND_NOW` 设置为任意非空值，从而覆盖运行时链接程序执行延迟绑定的缺省模式。此设置使运行时链接程序对所有已装入进程的非延迟目标文件

执行立即引用和延迟引用重定位。各目标文件可以请求在装入目标文件时执行非延迟的延迟引用重定位。请参见 `ld(1)` 的 `-z now` 选项，以及模式为 `RTLD_NOW` 的 `dlopen(3C)`。延迟的依赖项不受 `LD_BIND_NOW` 或 `RTLD_NOW` 的影响。请参见 `ld(1)` 的 `-z deferred` 选项。

如果同时指定 `LD_BIND_NOW` 和 `LD_BIND_LAZY`，则 `LD_BIND_NOW` 优先。

`LD_CAP_FILES`、`LD_CAP_FILES_32` 和 `LD_CAP_FILES_64`

应针对任何备选功能进行验证的文件的逗号分隔列表。请参见 `LD_PLATCAP`、`LD_MACHCAP`、`LD_HWCAP` 和 `LD_SFCAP`。

`LD_CONFIG`、`LD_CONFIG_32` 和 `LD_CONFIG_64`

提供备选配置文件。配置文件可用来更改缺省搜索路径、提供目录高速缓存，以及提供备用目标文件依赖项。请参见 `crle(1)`。

`LD_DEBUG`、`LD_DEBUG_32` 和 `LD_DEBUG_64`

提供以逗号或冒号分隔的标记列表，使运行时链接程序将调试信息输出到标准错误中。特殊标记 `help` 指示可用标记的完整列表。还可提供环境变量 `LD_DEBUG_OUTPUT` 来指定接收调试信息的文件。文件名的后缀为生成调试信息的应用程序的进程 ID。请参见 `lari(1)`。

`LD_DEMANGLE`、`LD_DEMANGLE_32` 和 `LD_DEMANGLE_64`

在诊断消息中使用的任何符号名称都将按 ELF 文件中定义的那样进行显示。将 `LD_DEMANGLE` 设置为任意非空值时，运行时链接程序将尝试解码（取消改编）任何 C++ 符号名称。

`LD_FLAGS`、`LD_FLAGS_32` 和 `LD_FLAGS_64`

提供环境变量信息的备选提供方式。任何 `LD_XXX` 环境变量均可指定为 `xxx` 标记。可提供多个标记，以逗号分隔。请参见“示例”部分。

`LD_HWCAP`、`LD_HWCAP_32` 和 `LD_HWCAP_64`

标识替代硬件功能值。

```
LD_HWCAP=[+-]{token | number},...
```

通过“+”前缀可以将后面的功能添加到备选功能中。通过“-”前缀可以将后面的功能从备选功能中删除。缺少“+”时，后面的功能将替换备选功能。

`LD_LIBRARY_PATH`、`LD_LIBRARY_PATH_32` 和 `LD_LIBRARY_PATH_64`

`LD_LIBRARY_PATH` 环境变量在设置的情况下将用于增强搜索路径，运行时链接程序使用该路径查找动态依赖项。`LD_LIBRARY_PATH` 指定将先于缺省目录进行搜索的目录的冒号分隔列表。还请注意，`LD_LIBRARY_PATH` 为 `ld(1)` 添加了其他语义。

`LD_LOADFLTR`、`LD_LOADFLTR_32` 和 `LD_LOADFLTR_64`

过滤器是共享目标文件的一种形式。过滤器允许在运行时选择备选共享目标文件，提供在过滤器内定义的任何符号的实现。请参见 `ld(1)` 的 `-f` 和 `-F` 选项。缺省情况下，备选共享目标文件的处理将推迟到对过滤器进行符号解析的时候。将 `LD_LOADFLTR` 设置为任意非空值时，装入任何过滤器后会立即对其进行处理。另请参见 `ld(1)` 的 `-z loadfltr` 选项。

LD_MACHCAP、LD_MACHCAP_32 和 LD_MACHCAP_64

标识替代计算机硬件名称。

LD_NOAUDIT、LD_NOAUDIT_32 和 LD_NOAUDIT_64

本地审计库可在应用程序和共享目标文件内进行定义。请参见 [ld\(1\)](#) 的 `-p` 和 `-P` 选项。将 `LD_NOAUDIT` 设置为任意非空值时，运行时链接程序将忽略任何本地审计库。

LD_NOAUXFLTR、LD_NOAUXFLTR_32 和 LD_NOAUXFLTR_64

辅助过滤器是共享目标文件的一种形式。辅助过滤器允许在运行时选择备选共享目标文件，提供在过滤器内定义的任何符号的实现。请参见 [ld\(1\)](#) 的 `-f` 选项。将 `LD_NOAUXFLTR` 设置为任意非空值时，运行时链接程序将禁用该备选共享目标文件查找。

LD_NOCONFIG、LD_NOCONFIG_32 和 LD_NOCONFIG_64

缺省情况下，运行时链接程序将尝试打开并处理一个配置文件。将 `LD_NOCONFIG` 设置为任意非空值时，运行时链接程序将禁用该配置文件处理。

LD_NODIRCONFIG、LD_NODIRCONFIG_32 和 LD_NODIRCONFIG_64

提供 `LD_NOCONFIG` 的子集，从而将忽略配置文件中提供的任何目录高速缓存信息。

LD_NODIRECT、LD_NODIRECT_32 和 LD_NODIRECT_64

直接绑定信息可指导运行时链接程序直接搜索关联目标文件中的符号。请参见 [ld\(1\)](#) 的 `-Bdirect` 选项。在没有直接绑定的情况下，由运行时链接程序执行的符号搜索将遵循缺省模式。将 `LD_NODIRECT` 设置为任意非空值时，运行时链接程序将忽略任何直接绑定信息。

LD_NOENVCONFIG、LD_NOENVCONFIG_32 和 LD_NOENVCONFIG_64

提供 `LD_NOCONFIG` 的子集，从而将忽略配置文件中提供的任何环境变量。

LD_NOLAZYLOAD、LD_NOLAZYLOAD_32 和 LD_NOLAZYLOAD_64

加标签进行延迟装入的依赖项不会装入到内存中，直到对该依赖项进行了显式引用。请参见 [ld\(1\)](#) 的 `-z lazyload` 选项。将 `LD_NOLAZYLOAD` 设置为任意非空值时，运行时链接程序将忽略依赖项延迟装入标签并立即装入依赖项。

LD_NOOBJALTER、LD_NOOBJALTER_32 和 LD_NOOBJALTER_64

提供 `LD_NOCONFIG` 的子集，从而将忽略配置文件中提供的任何备选目标文件依赖项。

LD_NOVERSION、LD_NOVERSION_32 和 LD_NOVERSION_64

缺省情况下，运行时链接程序将验证主要可执行文件及其所有依赖项的版本依赖项。将 `LD_NOVERSION` 设置为任意非空值时，运行时链接程序将禁用该版本检查。

LD_ORIGIN、LD_ORIGIN_32 和 LD_ORIGIN_64

将环境变量 `LD_ORIGIN` 设置为任意非空值可触发 `$ORIGIN` 的即时处理。在 Solaris 9 之前，对于先调用 [chdir\(2\)](#)，然后定位使用 `$ORIGIN` 字符串标记的依赖项的应用程序，该选项非常有用。现在，运行时链接程序会在缺省情况下建立当前工作目录，因此使该选项变为冗余。

LD_PLATCAP、**LD_PLATCAP_32** 和 **LD_PLATCAP_64**

标识替代平台名称。

LD_PRELOAD、**LD_PRELOAD_32** 和 **LD_PRELOAD_64**

提供由空格分隔的共享目标文件列表。这些目标文件将装入在正在执行的程序之后，但在该程序引用的任何其他共享目标文件之前。预装入目标文件提供的符号定义可以插入程序引用的共享目标文件所进行的引用。预装入目标文件提供的符号定义不会插入程序所提供的符号定义。

LD_PROFILE、**LD_PROFILE_32** 和 **LD_PROFILE_64**

定义运行时链接程序将要分析的共享目标文件。启用分析时，将创建并映射分析缓冲区文件。缓冲区文件的名称为要进行分析的共享目标文件的名称，带有 `.profile` 扩展名。缺省情况下，该缓冲区置于 `/var/tmp` 下。还可提供环境变量 `LD_PROFILE_OUTPUT` 来指示可在其中放置分析缓冲区的备选目录。

分析缓冲区包含 `profil(2)` 和调用计数信息。该信息类似于由那些通过 `cc` 的 `-xpg` 选项链接的程序所生成的 `gmon.out` 信息。使用命名的共享目标文件并在设置该环境变量时运行的任何应用程序都会在分析缓冲区中累积数据。另请参见“附注”部分。可以使用 `gprof(1)` 检查分析缓冲区信息。

`LD_PROFILE` 分析技术是编译系统提供的其他技术的一种替代技术。进行分析的共享目标文件不必采用任何方式进行检测，而 `LD_PROFILE` 不应与已分析检测的应用程序结合使用。有关共享目标文件分析的更多信息，请参见《[链接程序和库指南](#)》。

LD_SFCAP、**LD_SFCAP_32** 和 **LD_SFCAP_64**

标识替代软件功能值。

```
LD_SFCAP=[+-]{token | number},...
```

通过“+”前缀可以将后面的功能添加到备选功能中。通过“-”前缀可以将后面的功能从备选功能中删除。缺少“+-”时，后面的功能将替换备选功能。

LD_SIGNAL、**LD_SIGNAL_32** 和 **LD_SIGNAL_64**

提供运行时链接程序在出现严重运行时错误的情况下用来中止进程的**数字信号**编号。请参见 `thr_kill(3C)`。缺省情况下，将使用 `SIGKILL`。例如，如果备选信号编号为 6 (`SIGABRT`)，即可利用该编号创建核心文件来协助调试。另请参见针对 `dldinfo(3C)` 的 `RTLD_DI_SETSIGNAL` 请求。

请注意，以“LD_”字符开头的环境变量名是为未来可能推出的针对 `ld(1)` 和 `ld.so.1` 的增强功能而保留的。

安全

安全进程在评估其依赖项及 `runpaths` 时会应用某些限制，以免出现恶意依赖项替换或符号插入的情况。

对于某个进程来说，如果 `issetugid(2)` 系统调用返回的结果为 `True`，运行时链接程序会将该进程归类为安全进程。

对于 32 位目标文件，对运行时链接程序已知的缺省可信目录为 `/lib/secure` 和 `/usr/lib/secure`。对于 64 位目标文件，缺省可信目录为 `/lib/secure/64` 和

`/usr/lib/secure/64`。实用程序 `crle(1)` 可用于指定适用于安全应用程序的其他可信目录。使用此技术的管理员应确保已对目标目录进行了适当的保护，以防受到恶意入侵。

如果安全进程的 `LD_LIBRARY_PATH` 系列环境变量有效，则仅使用该变量所指定的可信目录来扩充运行时链接程序的搜索规则。

在安全进程中，将使用由应用程序或其任意依赖项提供的 `runpath` 组件，前提是该组件使用全路径名，即路径名以 `"/` 开头。

在安全进程中，仅当 `$ORIGIN` 字符串扩展扩展到某个可信目录时才允许该字符串的扩展。但是，如果 `$ORIGIN` 扩展与一个已提供依赖项的目录匹配，则该目录是隐式安全的。该目录可用于提供其他依赖项。

在安全进程中，将忽略 `LD_CONFIG`，但是，会使用安全应用程序中记录的配置文件。请参见 `ld(1)` 的 `-c` 选项。记录的配置文件必须使用全路径名，即路径名必须以 `"/` 开头。使用 `$ORIGIN` 字符串的已记录配置文件仅限于已知的可信目录。在安全应用程序中记录配置文件的开发者应确保配置文件目录受到适当的保护，以避免恶意入侵。缺少记录的配置文件时，安全进程将使用缺省配置文件（如果存在）。请参见 `crle(1)`。

在安全进程中，将忽略 `LD_SIGNAL`。

可以使用 `LD_PRELOAD` 或 `LD_AUDIT` 环境变量来与安全进程一起装入其他目标文件。必须将这些目标文件指定为全路径名或简单文件名。全路径名仅限于已知的可信目录。不含 `"/` 的简单文件名在定位时将遵循前述搜索路径限制。简单文件名只能解析到已知的可信目录。

在安全进程中，使用前述路径名限制处理包含简单文件名的任何依赖项。以全路径名或相对路径名表示的依赖项按原样使用。因此，安全进程的开发者应确保对以全路径名或相对路径名依赖项形式引用的目标目录进行适当的保护，以免遭恶意入侵。

创建安全进程时，不应使用相对路径名来表示依赖项或构建 `dlopen(3C)` 路径名。此限制应适用于应用程序及所有依赖项。

示例

示例 1 使用 `LD_FLAGS` 对环境变量信息进行分组

对 `LD_FLAGS` 的下述用法相当于为 32 位应用程序设置单个环境变量 `LD_BIND_NOW` 和 `LD_LIBRARY_PATH`：

```
example% LD_FLAGS_32=bind_now,library_path=/lib/one:/lib/two
```

对 `LD_FLAGS` 的下述用法相当于为 64 位应用程序设置单个环境变量 `LD_LIBRARY_PATH` 和 `LD_PRELOAD`：

```
example% LD_FLAGS_64=library_path=/lib/one/64,preload=foo.so
```

文件	<code>/lib/ld.so.1</code>	缺省运行时链接程序。
	<code>/lib/libc.so.1</code>	可实现 SVID ABI 兼容性的备选解释器。
	<code>/usr/lib/0@0.so.1</code>	一种兼容性库，支持空字符指针。请参见“附注”部分。
	<code>/lib/secure</code> 和 <code>/usr/lib/secure</code>	安全应用程序的 <code>LD_PRELOAD</code> 位置。
	<code>/lib/secure/64</code> 和 <code>/usr/lib/secure/64</code>	安全的 64 位应用程序的 <code>LD_PRELOAD</code> 位置。
	<code>/lib/64/ld.so.1</code>	64 位应用程序的缺省运行时链接程序。
	<code>/usr/lib/64/0@0.so.1</code>	一种 64 位的兼容性库，支持空字符指针。请参见“附注”部分。
	<code>/var/ld/ld.config</code>	32 位应用程序的缺省配置文件。
	<code>/var/ld/64/ld.config</code>	64 位应用程序的缺省配置文件。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/linker

另请参见

[crle\(1\)](#)、[gprof\(1\)](#)、[lari\(1\)](#)、[ld\(1\)](#)、[ldd\(1\)](#)、[exec\(2\)](#)、[issetugid\(2\)](#)、[mmapobj\(2\)](#)、[profil\(2\)](#)、[dlad](#)

《链接程序和库指南》

附注

将 `LD_PROFILE` 与其他进程监视技术结合使用时应慎重，如使用 [proc\(4\)](#) 的用户。多进程监视技术可能会导致死锁条件，从而使分析缓冲区处于锁定状态。锁定的缓冲区将阻止任何尝试记录分析信息的进程。为了降低这种可能性，运行时链接程序的配置文件实现会确定在启动时进程是否受到监视。如果是，将在无提示情况下禁用进程的分析。不过，此机制无法捕捉在进程执行过程中附加到该进程的监视进程。

用户兼容性库 `/usr/lib/0@0.so.1` 提供了一种机制，可以在位置 0 建立值 0。存在一些应用程序，其错误地假定空字符指针应视为与指向空字符串的指针相同。访问空字符指针时，这些应用程序中会出现段违规。如果在运行时使用 `LD_PRELOAD` 将该库添加到此类应用程序，该库将提供一个与该错误行为对应的环境。然而，用户兼容性库既不是用来允许生成此类应用程序，也不是用来支持此特定编程实践。

在很多情况下，`/usr/lib/0@0.so.1` 的存在并没有危险，并可将其预装入不需要它的程序中，但也有例外。诸如 JVM（Java Virtual Machine，Java 虚拟机）之类的某些应用程序要求从空指针访问生成段违规。诸如 JVM 之类的应用程序不应预装入 `/usr/lib/0@0.so`。

引用名 let – shell built-in function to evaluate one or more arithmetic expressions

用法概要

ksh88 let *arg*...

ksh let [*expr*...]

描述

ksh88 Each *arg* is a separate arithmetic expression to be evaluated.

ksh let evaluates each *expr* in the current shell environment as an arithmetic expression using ANSI C syntax. Variable names are shell variables and they are recursively evaluated as arithmetic expressions to get numerical values. let has been made obsolete by the ((...)) syntax of [ksh\(1\)](#) which does not require quoting of the operators to pass them as command arguments.

退出状态

ksh88 ksh88 returns the following exit values:

- 0 The value of the last expression is non-zero.
- 1 The value of the last expression is zero.

ksh ksh returns the following exit values:

- 0 The last *expr* evaluates to a non-zero value.
- >0 The last *expr* evaluates to 0 or an error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[ksh\(1\)](#), [ksh88\(1\)](#), [set\(1\)](#), [typeset\(1\)](#), [attributes\(5\)](#)

引用名	lex – generate programs for lexical tasks
用法概要	lex [-cntv] [-e -w] [-V -Q [y n]] [<i>file</i>]. . .
描述	The <code>lex</code> utility generates C programs to be used in lexical processing of character input, and that can be used as an interface to <code>yacc</code> . The C programs are generated from <code>lex</code> source code and conform to the ISO C standard. Usually, the <code>lex</code> utility writes the program it generates to the file <code>lex.yy.c</code> . The state of this file is unspecified if <code>lex</code> exits with a non-zero exit status. See EXTENDED DESCRIPTION for a complete description of the <code>lex</code> input language.
选项	<p>The following options are supported:</p> <ul style="list-style-type: none">-c Indicates C-language action (default option).-e Generates a program that can handle EUC characters (cannot be used with the <code>-w</code> option). <code>yytext[]</code> is of type <code>unsigned char[]</code>.-n Suppresses the summary of statistics usually written with the <code>-v</code> option. If no table sizes are specified in the <code>lex</code> source code and the <code>-v</code> option is not specified, then <code>-n</code> is implied.-t Writes the resulting program to standard output instead of <code>lex.yy.c</code>.-v Writes a summary of <code>lex</code> statistics to the standard error. (See the discussion of <code>lex</code> table sizes under the heading Definitions in lex.) If table sizes are specified in the <code>lex</code> source code, and if the <code>-n</code> option is not specified, the <code>-v</code> option can be enabled.-w Generates a program that can handle EUC characters (cannot be used with the <code>-e</code> option). Unlike the <code>-e</code> option, <code>yytext[]</code> is of type <code>wchar_t[]</code>.-V Prints out version information on standard error.-Q[y n] Prints out version information to output file <code>lex.yy.c</code> by using <code>-Qy</code>. The <code>-Qn</code> option does not print out version information and is the default.
操作数	<p>The following operand is supported:</p> <p><i>file</i> A pathname of an input file. If more than one such <i>file</i> is specified, all files is concatenated to produce a single <code>lex</code> program. If no <i>file</i> operands are specified, or if a <i>file</i> operand is <code>-</code>, the standard input is used.</p>
Output	The <code>lex</code> output files are described below.
Stdout	If the <code>-t</code> option is specified, the text file of C source code output of <code>lex</code> is written to standard output.
Stderr	If the <code>-t</code> option is specified informational, error and warning messages concerning the contents of <code>lex</code> source code input is written to the standard error.
	If the <code>-t</code> option is not specified:

1. Informational error and warning messages concerning the contents of lex source code input is written to either the standard output or standard error.
2. If the `-v` option is specified and the `-n` option is not specified, lex statistics is also written to standard error. These statistics can also be generated if table sizes are specified with a `%` operator in the `Definitions in lex` section (see `EXTENDED DESCRIPTION`), as long as the `-n` option is not specified.

Output Files

A text file containing C source code is written to `lex.yy.c`, or to the standard output if the `-t` option is present.

扩展描述

Each input file contains lex source code, which is a table of regular expressions with corresponding actions in the form of C program fragments.

When `lex.yy.c` is compiled and linked with the lex library (using the `-l lex` operand with `c89` or `cc`), the resulting program reads character input from the standard input and partitions it into strings that match the given expressions.

When an expression is matched, these actions occur:

- The input string that was matched is left in `yytext` as a null-terminated string; `yytext` is either an external character array or a pointer to a character string. As explained in `Definitions in lex`, the type can be explicitly selected using the `%array` or `%pointer` declarations, but the default is `%array`.
- The external `int yyleng` is set to the length of the matching string.
- The expression's corresponding program fragment, or action, is executed.

During pattern matching, lex searches the set of patterns for the single longest possible match. Among rules that match the same number of characters, the rule given first is chosen.

The general format of lex source is:

Definitions

%%

Rules

%%

User Subroutines

The first %% is required to mark the beginning of the rules (regular expressions and actions); the second %% is required only if user subroutines follow.

Any line in the `Definitions in lex` section beginning with a blank character is assumed to be a C program fragment and is copied to the external definition area of the `lex.yy.c` file. Similarly, anything in the `Definitions in lex` section included between delimiter lines containing only `{` and `}` is also copied unchanged to the external definition area of the `lex.yy.c` file.

Any such input (beginning with a blank character or within `{` and `}` delimiter lines) appearing at the beginning of the *Rules* section before any rules are specified is written to `lex.yy.c` after the declarations of variables for the `yyllex` function and before the first line of code in `yyllex`. Thus, user variables local to `yyllex` can be declared here, as well as application code to execute upon entry to `yyllex`.

The action taken by `lex` when encountering any input beginning with a blank character or within `{` and `}` delimiter lines appearing in the *Rules* section but coming after one or more rules is undefined. The presence of such input can result in an erroneous definition of the `yyllex` function.

Definitions in lex

Definitions in `lex` appear before the first `%` delimiter. Any line in this section not contained between `{` and `}` lines and not beginning with a blank character is assumed to define a `lex` substitution string. The format of these lines is:

name substitute

If a *name* does not meet the requirements for identifiers in the ISO C standard, the result is undefined. The string *substitute* replaces the string `{ name }` when it is used in a rule. The *name* string is recognized in this context only when the braces are provided and when it does not appear within a bracket expression or within double-quotes.

In the Definitions in `lex` section, any line beginning with a `%` (percent sign) character and followed by an alphanumeric word beginning with either `s` or `S` defines a set of start conditions. Any line beginning with a `%` followed by a word beginning with either `x` or `X` defines a set of exclusive start conditions. When the generated scanner is in a `%s` state, patterns with no state specified also active; in a `%x` state, such patterns are not active. The rest of the line, after the first word, is considered to be one or more blank-character-separated names of start conditions. Start condition names are constructed in the same way as definition names. Start conditions can be used to restrict the matching of regular expressions to one or more states as described in *Regular expressions in lex*.

Implementations accept either of the following two mutually exclusive declarations in the Definitions in `lex` section:

`%array` Declare the type of *yytext* to be a null-terminated character array.

`%pointer` Declare the type of *yytext* to be a pointer to a null-terminated character string.

When using the `%pointer` option, you cannot also use the `yylless` function to alter *yytext*.

`%array` is the default. If `%array` is specified (or neither `%array` nor `%pointer` is specified), then the correct way to make an external reference to *yytext* is with a declaration of the form:

```
extern char yytext[ ]
```

If `%pointer` is specified, then the correct external reference is of the form:


```
extern char *yytext;
```

lex accepts declarations in the `Definitions in lex` section for setting certain internal table sizes. The declarations are shown in the following table.

Table Size Declaration in lex

Declaration	Description	Default
%pn	Number of positions	2500
%nn	Number of states	500
%an	Number of transitions	2000
%en	Number of parse tree nodes	1000
%kn	Number of packed character classes	10000
%on	Size of the output array	3000

Programs generated by lex need either the `-e` or `-w` option to handle input that contains EUC characters from supplementary codesets. If neither of these options is specified, `yytext` is of the type `char[]`, and the generated program can handle only ASCII characters.

When the `-e` option is used, `yytext` is of the type `unsigned char[]` and `yylen` gives the total number of *bytes* in the matched string. With this option, the macros `input()`, `unput(c)`, and `output(c)` should do a byte-based I/O in the same way as with the regular ASCII lex. Two more variables are available with the `-e` option, `yywtext` and `yywlen`, which behave the same as `yytext` and `yylen` would under the `-w` option.

When the `-w` option is used, `yytext` is of the type `wchar_t[]` and `yylen` gives the total number of *characters* in the matched string. If you supply your own `input()`, `unput(c)`, or `output(c)` macros with this option, they must return or accept EUC characters in the form of wide character (`wchar_t`). This allows a different interface between your program and the lex internals, to expedite some programs.

Rules in lex

The `Rules in lex` source files are a table in which the left column contains regular expressions and the right column contains actions (C program fragments) to be executed when the expressions are recognized.

```
ERE action
ERE action
...
```

The extended regular expression (ERE) portion of a row is separated from *action* by one or more blank characters. A regular expression containing blank characters is recognized under one of the following conditions:

- The entire expression appears within double-quotes.
- The blank characters appear within double-quotes or square brackets.
- Each blank character is preceded by a backslash character.

User Subroutines in lex Anything in the user subroutines section is copied to `lex.yy.c` following `yylex`.

Regular Expressions in lex The `lex` utility supports the set of Extended Regular Expressions (EREs) described on [regex\(5\)](#) with the following additions and exceptions to the syntax:

- . . . Any string enclosed in double-quotes represents the characters within the double-quotes as themselves, except that backslash escapes (which appear in the following table) are recognized. Any backslash-escape sequence is terminated by the closing quote. For example, "`\ 01"1"` represents a single string: the octal value 1 followed by the character 1.

`<state>r`

`<state1, state2, ... >r` The regular expression `r` is matched only when the program is in one of the start conditions indicated by `state`, `state1`, and so forth. For more information, see [Actions in lex](#). As an exception to the typographical conventions of the rest of this document, in this case `<state>` does not represent a metavariable, but the literal angle-bracket characters surrounding a symbol. The start condition is recognized as such only at the beginning of a regular expression.

`r/x`

The regular expression `r` is matched only if it is followed by an occurrence of regular expression `x`. The token returned in `yytext` is only matched `r`. If the trailing portion of `r` matches the beginning of `x`, the result is unspecified. The `r` expression cannot include further trailing context or the `$` (match-end-of-line) operator; `x` cannot include the `^` (match-beginning-of-line) operator, nor trailing context, nor the `$` operator. That is, only one occurrence of trailing context is allowed in a `lex` regular expression, and the `^` operator only can be used at the beginning of such an expression. A further restriction is that the trailing-context operator `/` (slash) cannot be grouped within parentheses.

`{name}`

When `name` is one of the substitution symbols from the [Definitions](#) section, the string, including the enclosing braces, is replaced by the `substitute` value. The `substitute` value is treated in the extended regular expression as if it were enclosed in parentheses. No substitution occurs if `{name}` occurs within a bracket expression or within double-quotes.

Within an ERE, a backslash character (`\\`, `\ a`, `\ b`, `\ f`, `\ n`, `\ r`, `\ t`, `\ v`) is considered to begin an escape sequence. In addition, the escape sequences in the following table is recognized.

A literal newline character cannot occur within an ERE; the escape sequence `\ n` can be used to represent a newline character. A newline character cannot be matched by a period operator.

Escape Sequences in `lex`

Escape Sequences in <code>lex</code>		
Escape Sequence	Description	Meaning
<code>\digits</code>	A backslash character followed by the longest sequence of one, two or three octal-digit characters (01234567). If all of the digits are 0, (that is, representation of the NUL character), the behavior is undefined.	The character whose encoding is represented by the one-, two- or three-digit octal integer. Multi-byte characters require multiple, concatenated escape sequences of this type, including the leading <code>\</code> for each byte.
<code>\xdigits</code>	A backslash character followed by the longest sequence of hexadecimal-digit characters (01234567abcdefABCDEF). If all of the digits are 0, (that is, representation of the NUL character), the behavior is undefined.	The character whose encoding is represented by the hexadecimal integer.
<code>\c</code>	A backslash character followed by any character not described in this table. (<code>\</code> , <code>\a</code> , <code>\b</code> , <code>\f</code> , <code>\n</code> , <code>\r</code> , <code>\t</code> , <code>\v</code>).	The character <code>c</code> , unchanged.

The order of precedence given to extended regular expressions for `lex` is as shown in the following table, from high to low.

The escaped characters entry is not meant to imply that these are operators, but they are included in the table to show their relationships to the true operators. The start condition, trailing context and anchoring notations have been omitted from the table because of the placement restrictions described in this section; they can only appear at the beginning or ending of an ERE.

ERE Precedence in <code>lex</code>	
<i>collation-related bracket symbols</i>	<code>[=] [: :] [. .]</code>
<i>escaped characters</i>	<code>\<special character></code>
<i>bracket expression</i>	<code>[]</code>
<i>quoting</i>	<code>" . . . "</code>
<i>grouping</i>	<code>()</code>

ERE Precedence in lex	
<i>definition</i>	{ <i>name</i> }
<i>single-character RE duplication</i>	* + ?
<i>concatenation</i>	
<i>interval expression</i>	{ <i>m,n</i> }
<i>alternation</i>	

The ERE anchoring operators (`^` and `$`) do not appear in the table. With `lex` regular expressions, these operators are restricted in their use: the `^` operator can only be used at the beginning of an entire regular expression, and the `$` operator only at the end. The operators apply to the entire regular expression. Thus, for example, the pattern `(^abc) | (def$)` is undefined; it can instead be written as two separate rules, one with the regular expression `^abc` and one with `def$`, which share a common action via the special `|` action (see below). If the pattern were written `^abc|def$`, it would match either of `abc` or `def` on a line by itself.

Unlike the general ERE rules, embedded anchoring is not allowed by most historical `lex` implementations. An example of embedded anchoring would be for patterns such as `(^)foo($)` to match `foo` when it exists as a complete word. This functionality can be obtained using existing `lex` features:

```
^foo/[ \ n]|
"foo"/[ \ n] /* found foo as a separate word */
```

Notice also that `$` is a form of trailing context (it is equivalent to `/\ n` and as such cannot be used with regular expressions containing another instance of the operator (see the preceding discussion of trailing context).

The additional regular expressions trailing-context operator `/` (slash) can be used as an ordinary character if presented within double-quotes, `" / "`; preceded by a backslash, `\ /`; or within a bracket expression, `[/]`. The start-condition `<` and `>` operators are special only in a start condition at the beginning of a regular expression; elsewhere in the regular expression they are treated as ordinary characters.

The following examples clarify the differences between `lex` regular expressions and regular expressions appearing elsewhere in this document. For regular expressions of the form `r/x`, the string matching `r` is always returned; confusion can arise when the beginning of `x` matches the trailing portion of `r`. For example, given the regular expression `a*b/cc` and the input `aaabcc`, `yytext` would contain the string `aaab` on this match. But given the regular expression `x*/xy` and the input `xxxxy`, the token `xxx`, not `xx`, is returned by some implementations because `xxx` matches `x*`.

In the rule ab^*/bc , the b^* at the end of r extends r 's match into the beginning of the trailing context, so the result is unspecified. If this rule were ab/bc , however, the rule matches the text ab when it is followed by the text bc . In this latter case, the matching of r cannot extend into the beginning of x , so the result is specified.

Actions in lex

The action to be taken when an ERE is matched can be a C program fragment or the special actions described below; the program fragment can contain one or more C statements, and can also include special actions. The empty C statement `;` is a valid action; any string in the `lex.yy.c` input that matches the pattern portion of such a rule is effectively ignored or skipped. However, the absence of an action is not valid, and the action `lex` takes in such a condition is undefined.

The specification for an action, including C statements and special actions, can extend across several lines if enclosed in braces:

```
ERE <one or more blanks> { program statement
program statement }
```

The default action when a string in the input to a `lex.yy.c` program is not matched by any expression is to copy the string to the output. Because the default behavior of a program generated by `lex` is to read the input and copy it to the output, a minimal `lex` source program that has just `%%` generates a C program that simply copies the input to the output unchanged.

Four special actions are available:

```
|      ECHO;      REJECT;      BEGIN
```

| The action `|` means that the action for the next rule is the action for this rule. Unlike the other three actions, `|` cannot be enclosed in braces or be semicolon-terminated. It must be specified alone, with no other actions.

ECHO; Writes the contents of the string *yytext* on the output.

REJECT; Usually only a single expression is matched by a given string in the input. REJECT means *continue to the next expression that matches the current input*, and causes whatever rule was the second choice after the current rule to be executed for the same input. Thus, multiple rules can be matched and executed for one input string or overlapping input strings. For example, given the regular expressions `xyz` and `xy` and the input `xyz`, usually only the regular expression `xyz` would match. The next attempted match would start after `z`. If the last action in the `xyz` rule is REJECT, both this rule and the `xy` rule would be executed. The REJECT action can be implemented in such a fashion that flow of control does not continue after it, as if it were equivalent to a `goto` to another part of `yylex`. The use of REJECT can result in somewhat larger and slower scanners.

BEGIN The action:

```
BEGIN newstate;
```

switches the state (start condition) to *newstate*. If the string *newstate* has not been declared previously as a start condition in the `Definitions` in `lex` section, the results are unspecified. The initial state is indicated by the digit `0` or the token `INITIAL`.

The functions or macros described below are accessible to user code included in the `lex` input. It is unspecified whether they appear in the C code output of `lex`, or are accessible only through the `-ll` operand to `c89` or `cc` (the `lex` library).

<code>int yylex(void)</code>	Performs lexical analysis on the input; this is the primary function generated by the <code>lex</code> utility. The function returns zero when the end of input is reached; otherwise it returns non-zero values (tokens) determined by the actions that are selected.
<code>int yymore(void)</code>	When called, indicates that when the next input string is recognized, it is to be appended to the current value of <i>yytext</i> rather than replacing it; the value in <i>yyleng</i> is adjusted accordingly.
<code>int yyless(int n)</code>	Retains <i>n</i> initial characters in <i>yytext</i> , NUL-terminated, and treats the remaining characters as if they had not been read; the value in <i>yyleng</i> is adjusted accordingly.
<code>int input(void)</code>	Returns the next character from the input, or zero on end-of-file. It obtains input from the stream pointer <i>yyin</i> , although possibly via an intermediate buffer. Thus, once scanning has begun, the effect of altering the value of <i>yyin</i> is undefined. The character read is removed from the input stream of the scanner without any processing by the scanner.
<code>int unput(int c)</code>	Returns the character <i>c</i> to the input; <i>yytext</i> and <i>yyleng</i> are undefined until the next expression is matched. The result of using <i>unput</i> for more characters than have been input is unspecified.

The following functions appear only in the `lex` library accessible through the `-ll` operand; they can therefore be redefined by a portable application:

<code>int yywrap(void)</code>	Called by <code>yylex</code> at end-of-file; the default <code>yywrap</code> always returns 1. If the application requires <code>yylex</code> to continue processing with another source of input, then the application can include a function <code>yywrap</code> , which associates another file with the external variable <code>FILE *yyin</code> and returns a value of zero.
<code>int main(int argc, char *argv[])</code>	Calls <code>yylex</code> to perform lexical analysis, then exits. The user code can contain <code>main</code> to perform application-specific operations, calling <code>yylex</code> as applicable.

The reason for breaking these functions into two lists is that only those functions in `libl.a` can be reliably redefined by a portable application.

Except for `input`, `unput` and `main`, all external and static names generated by `lex` begin with the prefix `yy` or `YY`.

用法

Portable applications are warned that in the `Rules in lex` section, an ERE without an action is not acceptable, but need not be detected as erroneous by `lex`. This can result in compilation or run-time errors.

The purpose of `input` is to take characters off the input stream and discard them as far as the lexical analysis is concerned. A common use is to discard the body of a comment once the beginning of a comment is recognized.

The `lex` utility is not fully internationalized in its treatment of regular expressions in the `lex` source code or generated lexical analyzer. It would seem desirable to have the lexical analyzer interpret the regular expressions given in the `lex` source according to the environment specified when the lexical analyzer is executed, but this is not possible with the current `lex` technology. Furthermore, the very nature of the lexical analyzers produced by `lex` must be closely tied to the lexical requirements of the input language being described, which is frequently locale-specific anyway. (For example, writing an analyzer that is used for French text is not automatically be useful for processing other languages.)

示例

示例1 Using `lex`

The following is an example of a `lex` program that implements a rudimentary scanner for a Pascal-like syntax:

```
%{
/* need this for the call to atof() below */
#include <math.h>
/* need this for printf(), fopen() and stdin below */
#include <stdio.h>
}%

DIGIT    [0-9]
ID       [a-z][a-z0-9]*
%%

{DIGIT}+  {
            printf("An integer: %s (%d)\n", yytext,
                atoi(yytext));
        }

{DIGIT}+".{DIGIT}*  {
            printf("A float: %s (%g)\n", yytext,
                atof(yytext));
        }
```

示例1 Usinglex (续)

```

    }

if|then|begin|end|procedure|function      {
    printf("A keyword: %s\n", yytext);
}

{ID}          printf("An identifier: %s\n", yytext);

"+"|"-"|"*"|"\/"      printf("An operator: %s\n", yytext);

"{[^}\n]*}"      /* eat up one-line comments */

[ \t\n]+          /* eat up white space */

.              printf("Unrecognized character: %s\n", yytext);

%%

int main(int argc, char *argv[ ])
{
    ++argv, --argc; /* skip over program name */
    if (argc > 0)
        yyin = fopen(argv[0], "r");
    else
        yyin = stdin;

    yylex();
}

```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of lex: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities
Interface Stability	Committed

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Standard	See standards(5) .

另请参见

[yacc\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [regex\(5\)](#), [standards\(5\)](#)

附注

If routines such as `yyback()`, `yywrap()`, and `yylock()` in `.l` (ell) files are to be external C functions, the command line to compile a C++ program must define the `__EXTERN_C__` macro. For example:

```
example% CC -D__EXTERN_C__ ... file
```

引用名 lgrpinfo – 显示有关地址组的信息

用法概要

```
lgrpinfo [-aceGLmrt] [-u unit] [-C | -P] lgrp ...
lgrpinfo -h
lgrpinfo -I [-c] [-G] [-C | -P] lgrp ...
lgrpinfo [-T] [-aceGLmr] [-u unit]
lgrpinfo -d device_path
```

描述

`lgrpinfo` 输出有关地址组 (`lgroup`) 分层结构及其内容的信息。

`lgroup` 表示彼此之间最多相隔某个间隔（等待时间）的 CPU 和内存之类的硬件设备的集合。系统中的所有 `lgroup` 都由一个称作 `lgroup ID` 的唯一整数进行标识。

`lgroup` 组织到分层结构中，以便于查找最近的资源。每个叶 `lgroup` 包含一组彼此之间最近的（本地的）资源。分层结构中的每个父 `lgroup` 都包含其子 `lgroup` 的资源以及其最邻近的资源。最后，`root lgroup` 包含域中位于最大等待时间范围内的所有资源。

统一内存访问 (Uniform Memory Access, UMA) 计算机由根 `lgroup` 来简明地表示。非统一内存访问 (Non Uniform Memory Access, NUMA) 计算机由 `lgroup` 分层结构表示以显示地址的相应级别。例如，具有两个等待时间（本地和远程）的 NUMA 计算机具有由叶和根两个级别组成的 `lgroup` 层次结构。

每个应用程序线程都分配有一个主 `lgroup`。当系统需要为线程分配 CPU 或内存资源时，它将从线程的主 `lgroup` 中搜索 `lgroup` 分层结构以查找离线程的主 `lgroup` 最近的可用资源。有关详细信息，请参见 `plgrp(1)`。

不使用参数时，`lgrpinfo` 将输出关于系统中所有 `lgroup` 的常规信息。如果在命令行上指定了任何 `lgroup ID`，该命令将仅输出有关指定的 `lgroup` 的信息。各个选项用于控制要显示哪些 `lgroup` 以及为每个 `lgroup` 输出的确切信息。

可以在命令行上通过 `lgroup ID` 或通过使用特定的关键字来指定 `lgroup`。请参见“操作数”部分。

选项

可以组合使用这些选项，指定选项时的顺序不重要。小写字母选项用于选择应输出关于 `lgroup` 的哪些信息。

不带参数调用 `lgrpinfo` 等效于以下命令：

```
lgrpinfo -c -e -l -m -r -t all
```

支持以下选项：

`-a` 输出拓扑、CPU、内存、负荷和等待时间信息。

此选项是以下内容的简略表达方法

```
lgrpinfo -t -c -e -m -r -l -L
```

- 除非还指定了 `-T`。如果指定了 `-T`，将不包括 `-t` 选项。
- `-c` 输出 CPU 信息。
这是缺省值。
- `-C` 将列表中的每个 `lgroup` 替换为其子项。
此选项不能与 `-P` 或 `-T` 选项一起使用。如果未指定任何参数，则此选项应用于缺省情况下显示的 `lgroup`。
- `-d device_path` 输出与指定的 I/O 设备最近的 `lgroup` 的 ID。`device_path` 是表示设备路径的字符串。
- `-e` 输出 `lgroup` 平均负荷值。只会为叶 `lgroups` 显示 `lgroup` 平均负荷值。
这是缺省值。
- `-G` 输出 `lgroup` 分层结构的操作系统视图。
缺省情况下，会显示调用者的 `lgroup` 分层结构视图，其中仅包括调用者可以使用的内容，例如，仅显示调用者的处理器集中的 CPU。有关操作系统和调用者的视图，请参见 `lgrp_init(3LGRP)`。
- `-h` 输出简短帮助消息并退出。
- `-I` 仅输出匹配的 ID。
此选项用于脚本，并可以与 `-c`、`-G` 以及 `-c` 或 `-P` 一起使用。如果指定了 `-c`，将输出所有匹配 `lgroup` 中包含的 CPU 列表。否则，将显示匹配的 `lgroup` 的 ID。请参见“示例”部分。
未指定任何参数时，此选项应用于所显示的 `lgroup`，缺省情况下是所有 `lgroup`。
- `-l` 输出有关 `lgroup` 等待时间的信息。
为每个 `lgroup` 指定的等待时间值是由操作系统定义的，并特定于平台。它只能用于对正在运行的系统上的 `lgroup` 进行相对比较。它不一定表示硬件设备之间的实际等待时间，可能不适合跨平台使用。
- `-L` 输出 `lgroup` 等待时间表。`lgroup` 等待时间表显示每个 `lgroup` 相对于其他各个 `lgroups`（包括自身）的相对等待时间。
- `-m` 输出内存信息。

除非还指定了 `-u` 选项，否则将根据情况对内存大小的单位进行调整，使最后生成的值为 0 到 1023 之间的整数。只会为小于 10 的值显示数字的小数部分。此行为类似于使用 `ls(1)` 或 `df(1M)` 的 `-h` 选项显示人类可阅读的格式。

这是缺省值。

`-P` 将列表中的每个 `lgroup` 替换为其父项。

此选项无法与 `-C` 或 `-T` 选项一起使用。未指定任何参数时，此选项应用于所显示的 `lgroup`。缺省情况下是所有 `lgroup`。

`-r` 输出有关 `lgroup` 资源的信息。

资源由一组 `lgroup` 来表示，其中每个成员 `lgroup` 直接包含 CPU 和内存资源。如果还指定了 `-T`，则只会显示有关中间 `lgroup` 的资源的信息。

`-t` 输出有关 `lgroup` 拓扑的信息。

这是缺省值。

`-T` 以图形方式将系统的 `lgroup` 拓扑输出为树的形式。此选项只能与 `-a`、`-c`、`-e`、`-G`、`-l`、`-L`、`-m`、`-r` 和 `-u` 选项一起使用。与 `-r` 一起使用时，它只输出中间 `lgroup` 的 `lgroup` 资源。当 `-T` 与 `-a` 一起使用时，`-t` 选项会被忽略。不会输出 `root lgroup` 的任何信息，除非它是唯一的 `lgroup`。

`-u units` 指定内存单位。单位应该为 `b`、`k`、`m`、`g`、`t`、`p` 或 `e`，分别表示字节、千字节、兆字节、千兆字节、兆兆字节、千兆兆字节、艾字节。只会为小于 10 的值显示数字的小数部分。此行为类似于使用 `ls(1)` 或 `df(1M)` 的 `-h` 选项显示人类可阅读的格式。

操作数

支持下列操作数：

`lgrp` 可以在命令行上通过 `lgroup` ID 或使用以下关键字之一来指定 `lgroup`：

`all` 所有 `lgroup`。

这是缺省值。

`intermediate` 所有中间 `lgroup`。中间 `lgroup` 是具有父项和子项的 `lgroup`。

`leaves` 所有叶 `lgroup`。叶 `lgroup` 指的是在 `lgroup` 分层结构中没有子代的 `lgroup`。

`root` 根 `lgroup`。根 `lgroup` 包含域中位于最大等待时间范围内的所有资源，且没有父 `lgroup`。

如果指定的 `lgroup` 无效，`lgrpinfo` 命令将在标准错误上输出一条显示有无效 ID 的消息，然后继续处理命令行中指定的其他 `lgroup`。如果指定的 `lgroup` 都无效，则 `lgrpinfo` 将以退出状态 2 退出。

示例

示例 1 输出有关 `lgroup` 的信息

以下示例输出有关系统中的 `lgroup` 的常规信息。

在此示例中，系统为具有两个节点的 2 CPU AMD Opteron 计算机，每个节点具有一个 CPU 和 2 千兆字节的内存。每个节点由一个叶 `lgroup` 来表示。根 `lgroup` 包含计算机中的所有资源：

```
$ lgrpinfo
  lgroup 0 (root):
    Children: 1 2
    CPUs: 0 1
    Memory: installed 4.0G, allocated 2.2G, free 1.8G
    Lgroup resources: 1 2 (CPU); 1 2 (memory)
    Latency: 83
  lgroup 1 (leaf):
    Children: none, Parent: 0
    CPU: 0
    Memory: installed 2.0G, allocated 1.2G, free 788M
    Lgroup resources: 1 (CPU); 1 (memory)
    Load: 0.793
    Latency: 56
  lgroup 2 (leaf):
    Children: none, Parent: 0
    CPU: 1
    Memory: installed 2.0G, allocated 1017M, free 1.0G
    Lgroup resources: 2 (CPU); 2 (memory)
    Load: 0.817
    Latency: 56
```

示例 2 输出 `lgroup` 拓扑

以下示例输出了 4 CPU AMD Opteron 计算机上的 `lgroup` 拓扑树：

```
$ lgrpinfo -T
0
|-- 5
|  '-- 1
|-- 6
|  '-- 2
|-- 7
|  '-- 3
'-- 8
    '-- 4
```

示例3 输出 lgroup 拓扑

以下示例输出了 2 CPU AMD Opteron 计算机上的 lgroup 拓扑树、资源、内存和 CPU 信息：

```
$ lgrpinfo -Ta
0
|-- 1
|   CPU: 0
|   Memory: installed 2.0G, allocated 1.2G, free 790M
|   Load: 0.274
|   Latency: 56
'-- 2
    CPU: 1
    Memory: installed 2.0G, allocated 1019M, free 1.0G
    Load: 0.937
    Latency: 56
```

Lgroup latencies:

```
-----
| 0 1 2
-----
0 | 83 83 83
1 | 83 56 83
2 | 83 83 56
-----
```

示例4 输出 lgroup ID

以下示例输出了根 lgroup 的子代的 lgroup ID：

```
$ lgrpinfo -I -C root
1 2
```

示例5 输出 CPU ID

以下示例输出了 lgroup 1 中的所有 CPU 的 CPU ID：

```
$ lgrpinfo -c -I 1
0
```

示例6 输出有关 lgroup 等待时间的信息

以下示例输出了有关 lgroup 等待时间的信息：

```
$ lgrpinfo -l
lgroup 0 (root):
    Latency: 83
lgroup 1 (leaf):
    Latency: 56
```

示例 6 输出有关 lgroup 等待时间的信息 (续)

```
lgroup 2 (leaf):
  Latency: 5
```

示例 7 输出与给定设备最近的 lgroup 的 ID

以下示例表明 lgroup 2 和 6 与给定设备最近。

```
$ lgrpinfo -d /dev/dsk/c9t0d0s0
  lgroup IDs : 2 6
```

退出状态

将返回以下退出值：

- 0 成功完成。
- 1 无法从系统获取 lgroup 信息。
- 2 指定的所有 lgroups 或 *device_path* 无效。
- 3 语法无效。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

人类可读的输出是 "Uncommitted" (未确定)。

另请参见

[ls\(1\)](#)、[plgrp\(1\)](#)、[pmap\(1\)](#)、[proc\(1\)](#)、[ps\(1\)](#)、[df\(1M\)](#)、[prstat\(1M\)](#)、[lgrp_init\(3LGRP\)](#)、[liblgr](#)

引用名 limit, ulimit, unlimit – set or get limitations on the system resources available to the current shell and its descendents

用法概要 /usr/bin/ulimit [-f] [*blocks*]

sh ulimit [- [HS] [a | cdfnstv]]

 ulimit [- [HS] [c | d | f | n | s | t | v]] *limit*

csH limit [-h] [*resource* [*limit*]]

 unlimit [-h] [*resource*]

ksh88 ulimit [-HSacdfnstv] [*limit*]

ksh ulimit [-HSacdfmnpstv] [*limit*]

描述

/usr/bin/ulimit The `ulimit` utility sets or reports the file-size writing limit imposed on files written by the shell and its child processes (files of any size can be read). Only a process with appropriate privileges can increase the limit.

sh The Bourne shell built-in function, `ulimit`, prints or sets hard or soft resource limits. These limits are described in [getrlimit\(2\)](#).

If *limit* is not present, `ulimit` prints the specified limits. Any number of limits can be printed at one time. The `-a` option prints all limits.

If *limit* is present, `ulimit` sets the specified limit to *limit*. The string `unlimited` requests that the current limit, if any, be removed. Any user can set a soft limit to any value less than or equal to the hard limit. Any user can lower a hard limit. Only a user with appropriate privileges can raise or remove a hard limit. See [getrlimit\(2\)](#).

The `-H` option specifies a hard limit. The `-S` option specifies a soft limit. If neither option is specified, `ulimit` sets both limits and prints the soft limit.

The following options specify the resource whose limits are to be printed or set. If no option is specified, the file size limit is printed or set.

- c Maximum core file size (in 512-byte blocks)
- d Maximum size of data segment or heap (in Kbytes)
- f Maximum file size (in 512-byte blocks)
- n Maximum file descriptor plus 1
- s Maximum size of stack segment (in Kbytes)
- t Maximum CPU time (in seconds)
- v Maximum size of virtual memory (in Kbytes)

csh

The C-shell built-in function, `limit`, limits the consumption by the current process or any process it spawns, each not to exceed *limit* on the specified *resource*. The string `unlimited` requests that the current limit, if any, be removed. If *limit* is omitted, prints the current limit. If *resource* is omitted, displays all limits.

`-h` Use hard limits instead of the current limits. Hard limits impose a ceiling on the values of the current limits. Only the privileged user can raise the hard limits.

resource is one of:

`cputime` Maximum CPU seconds per process.

`filesize` Largest single file allowed. Limited to the size of the filesystem and capabilities of the filesystem. See `df(1M)`.

`datasize` The maximum size of a process's heap in kilobytes.

`stacksize` Maximum stack size for the process. The default stack size is 2^{13} .

`coredumpsize` Maximum size of a core dump (file). This is limited to the size of the filesystem.

`descriptors` Maximum number of file descriptors. Run the `sysdef(1M)` command to obtain the maximum possible limits for your system. The values reported by `sysdef` are in hexadecimal, but can be translated into decimal numbers using the `bc(1)` command.

`memorysize` Maximum size of virtual memory.

limit is a number, with an optional scaling factor, as follows:

`nh` Hours (for `cputime`).

`nk` *n* kilobytes. This is the default for all but `cputime`.

`nm` *n* megabytes or minutes (for `cputime`).

`mm:ss` Minutes and seconds (for `cputime`).

`unlimit` removes a limitation on *resource*. If no *resource* is specified, then all resource limitations are removed. See the description of the `limit` command for the list of resource names.

`-h` Remove corresponding hard limits. Only the privileged user can do this.

ksh88

The Korn shell built-in function, `ulimit`, sets or displays a resource limit. The available resources limits are listed below. Many systems do not contain one or more of these limits. The limit for a specified resource is set when *limit* is specified. The value of *limit* can be a number in the unit specified below with each resource, or the value `unlimited`. The string `unlimited` requests that the current limit, if any, be removed. The `-H` and `-S` flags specify whether the hard limit or the soft limit for the specified resource is set. A hard limit cannot be

increased once it is set. A soft limit can be increased up to the value of the hard limit. If neither the `-H` or `-S` options is specified, the limit applies to both. The current resource limit is printed when *limit* is omitted. In this case, the soft limit is printed unless `-H` is specified. When more than one resource is specified, then the limit name and unit is printed before the value.

- a Lists all of the current resource limits.
- c The number of 512-byte blocks on the size of core dumps.
- d The number of K-bytes on the size of the data area.
- f The number of 512-byte blocks on files written by child processes (files of any size can be read).
- n The number of file descriptors plus 1.
- s The number of K-bytes on the size of the stack area.
- t The number of seconds (CPU time) to be used by each process.
- v The number of K-bytes for virtual memory.

If no option is specified, `-f` is assumed.

Per-Shell Memory Parameters

The `heapsize`, `datasize`, and `stacksize` parameters are not system tunables. The only controls for these are hard limits, set in a shell startup file, or system-wide soft limits, which, for the current version of the Solaris OS, is 2^{13} bytes.

ksh

`ulimit` sets or displays resource limits. These limits apply to the current process and to each child process created after the resource limit has been set. If *limit* is specified, the resource limit is set, otherwise, its current value is displayed on standard output.

Increasing the limit for a resource usually requires special privileges. Some systems allow you to lower resource limits and later increase them. These are called soft limits. Once a hard limit is set the resource cannot be increased.

Different systems allow you to specify different resources and some restrict how much you can raise the limit of the resource.

The value of *limit* depends on the unit of the resource listed for each resource. In addition, *limit* can be “unlimited” to indicate no limit for that resource.

If you do not specify `-H` or `-S`, `-S` is used for listing and both `-S` and `-H` are used for setting resources.

If you do not specify any resource, the default is `-f`.

The following options are available for `ulimit` in ksh:

- a Displays all current resource limits.

-b	
--sbsize	Specifies the socket buffer size in bytes.
-c	
--core	Specifies the core file size in blocks.
-d	
--data	Specifies the data size in kbytes.
-f	
--fsize	Specifies the file size in blocks.
-H	Displays or sets a hard limit.
-L	
--locks	Specifies the number of file locks.
-l	
--memlock	Specifies the locked address space in Kbytes.
-M	
--as	Specifies the address space limit in Kbytes.
-n	
--nofile	Specifies the number of open files.
-p	
--pipe	Specifies the pipe buffer size in bytes.
-m	
--rss	Specifies the resident set size in Kbytes
-S	Displays or sets a soft limit.
-s	
--stack	Specifies the stack size in Kbytes.
-T	
--threads	Specifies the number of threads.
-t	
--cpu	Specifies the CPU time in seconds.
-u	
--nproc	Specifies the number of processes.
-v	
--vmem	Specifies the process size in Kbytes.

选项

The following option is supported by `/usr/bin/ulimit`:

-f Sets (or reports, if no *blocks* operand is present), the file size limit in blocks. The -f option is also the default case.

操作数

The following operand is supported by `/usr/bin/ulimit`:

blocks The number of 512-byte blocks to use as the new file size limit.

示例

`/usr/bin/ulimit`

示例 1 Limiting the Stack Size

The following example limits the stack size to 512 kilobytes:

```
example% ulimit -s 512
example% ulimit -a
time(seconds)          unlimited
file(blocks)           100
data(kbytes)           523256
stack(kbytes)          512
coredump(blocks)       200
nofiles(descriptors)   64
memory(kbytes)         unlimited
```

`sh/ksh88`

示例 2 Limiting the Number of File Descriptors

The following command limits the number of file descriptors to 12:

```
example$ ulimit -n 12
example$ ulimit -a
time(seconds)          unlimited
file(blocks)           41943
data(kbytes)           523256
stack(kbytes)          8192
coredump(blocks)       200
nofiles(descriptors)   12
vmemory(kbytes)        unlimited
```

`csh`

示例 3 Limiting the Core Dump File Size

The following command limits the size of a core dump file size to 0 kilobytes:

```
example% limit coredumpsize 0
example% limit
cputime                unlimited
filesize               unlimited
datasize               523256 kbytes
stacksize              8192 kbytes
coredumpsize           0 kbytes
descriptors            64
memorysize             unlimited
```

示例 4 Removing the limitation for core file size

The following command removes the above limitation for the core file size:

```
example% unlimit coredumpsize
example% limit
cputime          unlimited
filesize         unlimited
datasize        523256 kbytes
stacksize       8192 kbytes
coredumpsize    unlimited
descriptors     64
memorysize      unlimited
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `ulimit`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned by `ulimit`:

- `0` Successful completion.
- `>0` A request for a higher limit was rejected or an error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/ulimit, csh,
ksh88, sh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

ksh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

另请参见

[bc\(1\)](#), [csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [df\(1M\)](#), [su\(1M\)](#), [swap\(1M\)](#), [sysdef\(1M\)](#), [getrlimit\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注

Be aware of possible unexpected consequences when using `ulimit` in conjunction with other Solaris resource-limiting features, such as [prctl\(1\)](#). See [resource_controls\(5\)](#).

With the use of the project resource controls described in [resource_controls\(5\)](#), you should use [prctl\(1\)](#) to get an accurate observation of the limits in effect at any given time.

引用名 line – read one line

用法概要 line

描述 The `line` utility copies one line (up to and including a new-line) from the standard input and writes it on the standard output. It returns an exit status of 1 on EOF and always prints at least a new-line. It is often used within shell files to read from the user's terminal.

退出状态 Exit status is:

0 Successful completion

>0 End-of-file on input.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [sh\(1\)](#), [read\(2\)](#), [attributes\(5\)](#)

引用名	list_devices – 列出可分配的设备												
用法概要	list_devices [-s] [-U uid] [-z zonename] [-a [-w]] -l -n -u [device] [-l -n -u] -c dev-class list_devices [-s] -d dev-type												
描述	list_devices 实用程序可依据指定资格列出系统中可分配的设备。 会列出 设备 以及与该设备相关联的所有设备特殊文件。设备参数是可选的，如果不存在，会列出所有相关设备。如果存在 <i>dev-class</i> ，会列出属于指定 <i>dev-class</i> 的设备。没有缺省的 <i>dev-class</i> 。												
选项	支持以下选项： <table border="0" style="margin-left: 2em;"> <tr> <td style="vertical-align: top;">-l [-c dev-class device]</td> <td>列出与可分配给当前进程的 <i>device</i> 相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i>，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i>，则只列出与指定设备相关联的文件。</td> </tr> <tr> <td style="vertical-align: top;">-n [-c dev-class device]</td> <td>列出与可分配给当前进程但当前没有分配的设备相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i>，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i>，则只列出与指定设备相关联的文件。</td> </tr> <tr> <td style="vertical-align: top;">-s</td> <td>无提示。抑制任何诊断信息的输出。</td> </tr> <tr> <td style="vertical-align: top;">-u [-c dev-class device]</td> <td>列出与分配给当前进程的所有者的设备相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i>，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i>，则只列出与指定设备相关联的文件。</td> </tr> <tr> <td style="vertical-align: top;">-U uid</td> <td>执行 list_devices 操作时，使用用户 ID <i>uid</i>，而不是当前进程的实际用户 ID。只有拥有 solaris.device.revoke 授权的用户可使用此选项。</td> </tr> </table> <p>以下选项在系统上配置有 Trusted Extensions 时受支持：</p> <table border="0" style="margin-left: 2em;"> <tr> <td style="vertical-align: top;">-a</td> <td>列出属性，如授权、清理程序和与设备相关联的标签。 此列表是每个设备的一行以分号 (;) 分隔的键=值对，格式为：</td> </tr> </table>	-l [-c dev-class device]	列出与可分配给当前进程的 <i>device</i> 相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i> ，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i> ，则只列出与指定设备相关联的文件。	-n [-c dev-class device]	列出与可分配给当前进程但当前没有分配的设备相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i> ，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i> ，则只列出与指定设备相关联的文件。	-s	无提示。抑制任何诊断信息的输出。	-u [-c dev-class device]	列出与分配给当前进程的所有者的设备相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i> ，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i> ，则只列出与指定设备相关联的文件。	-U uid	执行 list_devices 操作时，使用用户 ID <i>uid</i> ，而不是当前进程的实际用户 ID。只有拥有 solaris.device.revoke 授权的用户可使用此选项。	-a	列出属性，如授权、清理程序和与设备相关联的标签。 此列表是每个设备的一行以分号 (;) 分隔的 键=值 对，格式为：
-l [-c dev-class device]	列出与可分配给当前进程的 <i>device</i> 相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i> ，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i> ，则只列出与指定设备相关联的文件。												
-n [-c dev-class device]	列出与可分配给当前进程但当前没有分配的设备相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i> ，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i> ，则只列出与指定设备相关联的文件。												
-s	无提示。抑制任何诊断信息的输出。												
-u [-c dev-class device]	列出与分配给当前进程的所有者的设备相关联的设备特殊文件的路径名。 如果指定了 <i>dev-class</i> ，则只列出与指定设备类的所有设备相关联的文件。 如果指定了 <i>device</i> ，则只列出与指定设备相关联的文件。												
-U uid	执行 list_devices 操作时，使用用户 ID <i>uid</i> ，而不是当前进程的实际用户 ID。只有拥有 solaris.device.revoke 授权的用户可使用此选项。												
-a	列出属性，如授权、清理程序和与设备相关联的标签。 此列表是每个设备的一行以分号 (;) 分隔的 键=值 对，格式为：												

```
device=device-name;type=device-type;\
auths=auths;clean=device-exec;\
device-attributes;\
files=device-list
```

其中，*device-attributes* 是 `device_allocate(4)` 的 `reserved1` 字段的内容。字段是以冒号(:)分隔的。

有关这些属性及其格式的说明，请参见 `device_allocate(4)`。

-a 输出包含以下键：

auths 指定授权的列表。如 `device_allocate(4)` 中所述，此值是 `auths`。

clean 指定设备清理脚本。如 `device_allocate(4)` 中所述，此值是 `device-exec`。

device 指定设备名称。如 `device_allocate(4)` 中所述，此值是 `device-name`。

files 指定设备文件路径。如 `device_maps(4)` 中所述，此值是 `device-list`。

type 指定设备类型。如 `device_allocate(4)` 中所述，此值是 `device-type`。

-d 为设备分配管理的设备类型显示系统提供的缺省属性。如果指定 *dev-type*，只会列出此设备类型的缺省属性。

-w 此选项与 -a 一起使用可以键值对**所有者=值**的形式列出设备的当前所有者。**值**是设备当前所有者的 `uid`。如果设备未被分配，则值为 `/FREE`。如果设备处于错误状态，则值为 `/ERROR`。此选项还会抑制任何诊断输出。

-z *zonename* 如果使用 -l 选项指定，只会列出其标签范围包含区域名称标签的未分配设备，而对于已分配设备，只会列出在与**区域名称**的标签相同的标签处分配的那些设备。

如果使用 -n 选项指定的、，只会列出其标签范围包含**区域名称**的标签的未分配设备。

如果使用 -u 选项指定，只会列出在与**区域名称**相同的标签处分配的设备。

示例

示例1 列出所有设备

以下示例列出可供调用者用于分配的所有设备。

```
% list_devices -l
device: audio type: audio \
```


示例1 列出所有设备 (续)

```
files: /dev/audio /dev/audiocctl /dev/sound/0 /dev/sound/0ctl
```

示例2 列出所有设备的属性

在配置有 Trusted Extensions 的系统上，以下示例会列出可供调用者用于分配的所有设备的属性：

```
% list_devices -al
device=audio1;type=audio;\
auths=solaris.device.allocate;\
clean=/etc/security/lib/audio_clean;\
minlabel=admin_low:maxlabel=admin_high;\
files=/dev/audio1 /dev/audio1ctl /dev/sound/1 /dev/sound/1ctl
```

示例3 列出包括设备所有者的属性

在配置有 Trusted Extensions 的系统上，以下示例会列出包括分配给用户的所有设备的设备所有者的属性：

```
% list_devices -auw
device=audio2;type=audio;auths=solaris.device.allocate;\
clean=/etc/security/lib/audio_clean;\
minlabel=admin_low:maxlabel=admin_high:zone=public;\
owner=1234;\
files=/dev/audio2 /dev/audio2ctl /dev/sound/2 /dev/sound/2ctl
```

退出状态

将返回以下退出值：

- 0 成功完成。
- 20 对于指定的设备没有任何项。
- 其他值** 出现错误。

文件

```
/etc/security/device_allocate
/etc/security/device_maps
/etc/security/dev/*
/usr/security/lib/*
```

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

属性类型	属性值
接口稳定性	请参见下文。

调用为 "Uncommitted"（未确定）。选项为 "Uncommitted"（未确定）。-a 和 -w 选项的输出为 "Uncommitted"（未确定）。所有其他输出为“非接口”。

另请参见

[allocate\(1\)](#)、[deallocate\(1\)](#)、[device_allocate\(1M\)](#)、[dminfo\(1M\)](#)、[mkdevalloc\(1M\)](#)、[mkdevmaps\(1M\)](#)

“控制对设备的访问”

附注

只有 Solaris 审计功能启用后，本手册页中描述的功能才可用。

只有 [device_allocate\(1M\)](#) 服务启用后，本手册页中描述的功能才可用。

在配置有 Trusted Extensions 的系统上，该功能是缺省启用的。

Solaris 操作环境的将来发行版可能不再支持 `/etc/security/dev`、[mkdevalloc\(1M\)](#) 和 [mkdevmaps\(1M\)](#)。

引用名 listusers – 列出用户登录信息

用法概要 listusers [-g *groups*] [-l *logins*]

描述 不使用任何选项执行时，此命令可按登录排序列出所有用户登录。输出会显示 `/etc/nsswitch.conf` 所指定的系统口令数据库中的登录 ID 和帐户字段值。

选项 支持以下选项：

- g *groups* 显示属于 *group* 的所有用户登录，按登录排序。可以逗号分隔的列表形式指定多个组。
- l *logins* 列出由 *logins* 指定的用户登录，按登录排序。可以逗号分隔的列表形式指定多个登录。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [nsswitch.conf\(4\)](#)、[attributes\(5\)](#)

附注 用户登录是指具有 100 或更大的 UID 的登录。

-l 和 -g 选项可以合并。用户登录只会列出一行，即使它们属于多个选定组。

- 引用名** llc2_autoconfig – 生成 LLC2 配置文件
- 用法概要** /usr/lib/llc2/llc2_autoconfig [-f]
- 描述** llc2_autoconfig 实用程序用于生成 LLC2 配置文件 (/etc/llc2/default/llc2.*)。如果 /etc/llc2_default/ 中没有配置文件，则它将检测系统中的所有可用接口并生成相应的缺省配置文件。
- 如果 /etc/llc2_default/ 中有现有的配置文件，则它将检查在文件中定义的那些接口是否仍然存在。如果它们在系统中不存在，则它会将那些文件中的 llc2_on 设置为 0。在这之后，它将检测系统中是否存在新的接口。如果存在，它将为它们生成配置文件。
- 选项** 支持以下选项：
- f 删除 /etc/llc2/default/ 中的所有配置文件。然后，检测系统中的所有可用接口并生成相应的缺省配置文件。请谨慎使用此选项。
- 文件** /etc/llc2/default/llc2.* LLC2 配置文件
- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/network/llc2

另请参见 [llc2_config\(1\)](#)、[llc2\(4\)](#)、[attributes\(5\)](#)、[llc2\(7D\)](#)

引用名	llc2_config – 配置 LLC2 接口参数
用法概要	<code>/usr/lib/llc2/llc2_config</code> <code>[-P -U -d -q -i <i>ppa</i> -r <i>ppa</i>]</code>
描述	llc2_config 实用程序用于启动/停止 LLC2 子系统和配置 LLC2 接口参数。
选项	支持以下选项： <ul style="list-style-type: none"> -d 打开调试模式。将输出额外的调试信息。 -<i>ppa</i> 使用文件 <code>/etc/llc2/default/llc2.<i>ppa</i></code> 初始化相应的接口。 -P 在所有 <code>/etc/llc2/default/llc2.*</code> 配置文件中读取，打开文件中定义的那些设备，并设置 LLC2 要使用那些设备所需的流。在执行此操作之前，llc2_config -q 不会显示任何内容。 -q 查询 LLC2 子系统。将为 LLC2 模块下可用的所有 PPA（Physical Point of Attachment，物理挂接点）输出类似于以下示例的信息： <pre>PPA State ID MACAddr Type MaxSDU MinSDU Mode 0 up 0000 0800208a217e ethernet 1500 0 3</pre> <p>所显示的字段如下所述：</p> <p>PPA 接口的相对逻辑位置。</p> <p>State 接口的状态：</p> <ul style="list-style-type: none"> <i>up</i> 接口已初始化并正常运行。 <i>down</i> 接口已被 LLC2 驱动程序“发现”，已传递了其引导诊断信息并且正在等待初始化。 <i>bad</i> LLC2 驱动程序已知道该接口，但是在引导时执行的一个或多个完整性检查失败。这可能包括检测到中断请求和共享内存冲突或者在执行 0 级诊断时检测到故障。 <p>ID 接口 ID。</p> <p>MACAddr 接口的当前有效 MAC 地址。</p> <p>Type MAC 类型。当前支持的类型包括：</p> <ul style="list-style-type: none"> <i>csma/cd</i> 10 兆位以太网 <i>ethernet</i> 以太网类型设备 <i>tkn-ring</i> 4/16 兆位令牌环 <i>fddi</i> 100 兆位光纤分布式数据接口 <p>MaxSDU 在此接口上传送的最大服务数据单元大小。</p>

- Mode** 此接口支持的服务模式。此字段包括受支持模式（也在 `/usr/include/sys/dlpi.h` 中进行了定义）的按位逻辑或。
- r ppa** 取消初始化相应的接口。通过使用此选项然后使用 `-i` 选项，可以更改与某个接口关联的参数。
- U** 销毁 LLC2 子系统使用的所有流。这是 `-P` 选项的反向操作。执行此操作后，`llc2_config -q` 将不显示任何内容。

文件 `/etc/llc2/default/llc2.*` LLC2 配置文件

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/network/llc2

另请参见 [llc2_autoconfig\(1\)](#)、[llc2\(4\)](#)、[attributes\(5\)](#)、[llc2\(7D\)](#)

引用名	llc2_stats – LLC2 工作站、SAP 和连接统计信息
用法概要	llc2_stats <i>ppa</i> [-r] [-s <i>sap</i>] [-c <i>connection</i>]
描述	llc2_stats 命令用于从 LLC2 驱动程序的基于主机的逻辑链路控制 2 类组件检索统计信息。统计信息是为工作站、SAP（Service Access Point，服务访问点）和连接组件保留的。
选项	支持以下选项： <ul style="list-style-type: none"> -c <i>connection</i> 指定关注的连接。其值是以不带前导 0x 的十六进制表示法输入的。 -r 在读取指定的计数器后将其重置为零。仅当 root 用户执行该命令时此选项才有效。 -ssap 指定此请求的 SAP。它是一个单字节值，以不带前导 0x 的十六进制表示法表示。例如，NetBIOS sap 240 (0xf0) 将输入为：-s f0。
操作数	支持下列操作数： <ul style="list-style-type: none"> <i>ppa</i> 用来对适配器进行寻址的逻辑编号。PPA（Physical Point of Attachment，物理挂接点）必须是第一个参数。
示例	<p>示例1 工作站统计信息</p> <p>以下命令将显示 PPA 4 的工作站统计信息。在示例后提供了每个字段的简要描述。</p> <pre>example% /usr/lib/llc2/llc2_stats 4</pre> <pre>Station values received: ppa = 0x00000004 clearFlag = 0x00 # of saps (hex) = 0x0002 saps (hex) = 02 aa state = 0x01 nullSapXidCmdRcvd = 0x00000000 nullSapXidRspSent = 0x00000000 nullSapTestCmdRcvd = 0x00000000 nullSapTestRspSent = 0x00000000 outOfState = 0x00000000 allocFail = 0x00000000 protocolError = 0x00000000</pre> <p>这些字段如下所述：</p> <ul style="list-style-type: none"> <i>ppa</i> 用来对适配器进行寻址的逻辑编号。 <i>clearFlag</i> 此标志指示是否在读取后将统计信息重置为零（设置为 1 时）或者统计信息是否为只读（设置为 0 时）。 <i># of saps</i> 此工作站上当前绑定的 SAP 的数目。

示例1 工作站统计信息 (续)

saps	LLC 与其相邻层之间的工作站服务访问点 (Service Access Point, SAP) 逻辑接口值的数组。
state	用于指示工作站组件当前状态的数字 (0 = 关闭, 1 = 正常运行)。
nullSapXidCmdRcvd	为空 SAP 地址 (sap = 0x00) 接收的 XID 命令协议数据单元 (Protocol Data Unit, PDU) 的数目。
nullSapXidRspSent	为响应针对空 SAP 地址接收的 XID 命令 PDU 而发送的 XID 响应 PDU 的数目。
nullSapTestCmdRcvd	为空 SAP 地址接收的 TEST 命令 PDU 的数目。
nullSapTestRspSent	为响应针对空 SAP 地址接收的 TEST 命令 PDU 而发送的 TEST 响应 PDU 的数目。
outOfState	接收的处于无效状态的事件数目。
allocFail	缓冲区分配故障的数目。
protocolError	LLC 协议错误数, 即, 收到的格式错误的 PDU 数目或者预期收到帧 Y 时收到的是帧 X 的次数。

示例2 SAP统计信息

在上面的显示中, 有两个活动的 SAP: 0x02 和 0xaa。下面是用于接收 SAP 02 的统计信息的命令示例以及对所提供的每个字段的简要解释。

```
example% /usr/lib/llc2/llc2_stats 4 -s 02
```

```
Sap values received:
ppa                = 0x00000004  clearFlag = 0x00
sap                = 0x02
state              = 0x01
# of cons (hex)    = 0x0000000a
connections (hex)  = 0000 0001 0002 0003 0004 0005 0006 0007 0008 0009
xidCmdSent         = 0x00000000
xidCmdRcvd        = 0x00000000
xidRspSent        = 0x00000000
xidRspRcvd        = 0x00000000
testCmdSent       = 0x00000000
testCmdRcvd      = 0x00000000
testRspSent       = 0x00000000
testRspRcvd      = 0x00000000
uiSent            = 0x00000000
uiRcvd           = 0x00000000
outOfState        = 0x00000000
```


示例2 SAP统计信息 (续)

```
allocFail      = 0x00000000
protocolError  = 0x00000000
```

这些字段如下所述：

ppa	用来对适配器进行寻址的逻辑编号。
clearFlag	此标志指示是否在读取后将统计信息重置为零（设置为1时）或者统计信息是否为只读（设置为0时）。
sap	工作站的指定服务访问点 (Service Access Point, SAP) 逻辑接口值。
state	用于指示 SAP 组件当前状态的数字（0 = 非活动，1 = 活动）。
# of cons	此 SAP 上的活动连接数目。
connections	活动连接索引的数组。
xidCmdSent	发送的 XID 命令 PDU 数（源 SAP = 此 sap）。
xidCmdRcvd	接收的 XID 命令 PDU 数（目标 SAP = 此 sap）。
xidRspSent	发送的 XID 响应 PDU 数（源 SAP = 此 sap）。
xidRspRcvd	接收的 XID 响应 PDU 数（源 SAP = 此 sap）。
testCmdSent	发送的 TEST 命令 PDU 数（源 SAP = 此 sap）。
testCmdRcvd	接收的 TEST 命令 PDU 数（目标 SAP = 此 sap）。
testRspSent	发送的 TEST 响应 PDU 数（源 SAP = 此 sap）。
testRspRcvd	接收到的 TEST 响应 PDU 数（源 SAP = 此 sap）。
uiSent	发送的无编号信息帧数目。
uiRcvd	接收的无编号信息帧数目。
outOfState	接收的处于无效状态的事件数目。
allocFail	缓冲区分配故障的数目。
protocolError	LLC 协议错误数，即，收到的格式错误的 PDU 数目或者预期收到帧 Y 时收到的是帧 X 的次数。

示例3 连接统计信息

十个已建立的连接与此 SAP 相关联。要检索连接 1 的统计信息，请输入以下命令：

```
example% /usr/lib/llc2/llc2_stats 4 -s 2 -c 1
Connection values received:
ppa          = 0x0004  clearFlag    = 0x00
sap          = 0x02      con          = 0x0001  sid          = 0x0201
```

示例3 连接统计信息 (续)

```

stateOldest = 0x00      stateOlder = 0x00      stateOld = 0x01
state       = 0x08
dl_nodeaddr = 0x0080d84008c2      dl_sap      = 0x04
flag       = 0x50      dataFlag   = 0x00      timerOn   = 0x18
vs         = 0x29      vr = 0x1e      nrRcvd    = 0x29      k = 0x14
retryCount = 0x0000      numToBeAcked = 0x0000      numToResend = 0x0000
macOutSave = 0x0000      macOutDump  = 0x0000
iSent      = 0x0ba9      iRcvd       = 0x001e
frmrSent   = 0x0000      frmrRcvd    = 0x0000
rrSent     = 0x016a      rrRcvd      = 0x00c1
rnrSent    = 0x0000      rnrRcvd     = 0x06fb
rejSent    = 0x0000      rejRcvd     = 0x0000
sabmeSent  = 0x0000      sabmeRcvd   = 0x0001
uaSent     = 0x0001      uaRcvd      = 0x0000      discSent   = 0x0000
outOfState = 0x0000      allocFail   = 0x0000      protocolError = 0x0000
localBusy  = 0x0000      remoteBusy  = 0x00b5      maxRetryFail = 0x0000
ackTimerExp = 0x0000      pollTimerExp = 0x0000      rejTimerExp = 0x0000
remBusyTimerExp = 0x0000
inactTimerExp = 0x0000
sendAckTimerExp = 0x0000

```

ppa	用来对适配器进行寻址的逻辑编号。
clearFlag	此标志指示是否在读取后将统计信息重置为零（设置为 1 时）或者统计信息是否为只读（设置为 0 时）。
sap	工作站的指定服务访问点 (Service Access Point, SAP) 逻辑接口值。
con	SAP 的指定连接索引值。
stateOldest	用于表示 stateOlder 之前的连接组件状态的数字。
stateOlder	用于表示 stateOld 之前的连接组件状态的数字。
stateOld	用于表示 state 之前的连接组件状态的数字。
state	用于表示连接组件最新状态的数字。请参见 Table 1。
sid	由 SAP（高位字节）和连接索引（低位字节）组成的工作站标识符。
dl_nodeaddr	数据链路节点地址。这是目标节点的 MAC 地址。
dl_sap	目标节点的 SAP。
flag	连接组件处理标志。请参见 Table 3。
dataFlag	用于表示接收到的 I 帧 PDU 中数据单元状态的数字（0 = 未丢弃，1 = 已丢弃，2 = 进入了具有未决 REJ PDU 的繁忙状态）。

示例3 连接统计信息 (续)

timerOn	用于表示计时器活动标志的数字，每个位表示此连接的活动计时器。有关计时器定义，请参见Table 2。
vs	要发送的下一个 I 帧 PDU 的序列号。
vr	要接收的下一个 I 帧 PDU 的预期序列号。
nrRcvd	远程节点确认的上次发送的 I 帧 PDU 的序列号加 1。
k	传送窗口大小。
retryCount	每当计时器失效时，retryCount 都会递增。这些计时器可保护传出帧。
numToBeAcked	等待确认的传出 I 帧数目。
numToResend	要重新传送的传出 I 帧数目。
macOutSave	不再使用。
macOutDump	不再使用。
iSent	发送的 I 帧数目。
iRcvd	接收的 I 帧数目。
frmrSent	发送的帧拒绝 PDU (FRMR) 的数目。
frmrRcvd	接收的帧拒绝 PDU (FRMR) 的数目。
rrSent	发送的接收器就绪 PDU (RR) 的数目。
rrRcvd	接收的接收器就绪 PDU (RR) 的数目。
rnrSent	发送的接收器未就绪 PDU (RNR) 的数目。
rnrRcvd	接收的接收器未就绪 PDU (RNR) 的数目。
rejSent	发送的拒绝 PDU (REJ) 的数目。
rejRcvd	接收的拒绝 PDU (REJ) 的数目。
sabmeSent	发送的设置异步平衡模式扩展 PDU (SABME) 的数目。
sabmeRcvd	接收的设置异步平衡模式扩展 PDU (SABME) 的数目。
uaSent	发送的无编号确认 PDU (UA) 的数目。
uaRcvd	接收的无编号确认 PDU (UA) 的数目。
discSent	发送的断开连接 PDU (DISC) 的数目。
outOfState	接收的处于无效状态的事件数目。

示例3 连接统计信息 (续)

allocFail	缓冲区分配故障的数目。
protocolError	LLC 协议错误数，即，收到的格式错误的 PDU 数目或者预期收到帧 Y 时收到的是帧 X 的次数。
localBusy	此组件处于本地繁忙状态且无法接受 I 帧的次数。
remoteBusy	远程连接组件处于繁忙状态且无法接受 I 帧的次数。
maxRetryFail	由于达到 maxRetry 而失败的次数。
ackTimerExp	确认计时器失效的次数。
pollTimerExp	轮询计时器失效的次数。
rejTimerExp	拒绝计时器失效的次数。
remBusyTimerExp	远程繁忙计时器失效的次数。
inactTimerExp	非活动计时器失效的次数。
sendAckTimerExp	发送确认计时器失效的次数。

表 1 : LLC2 States

工作站	
~~DOWN	0x00
~~UP	0x01
SAP	
~~INACTIVE	0x00
~~ACTIVE	0x01
连接	
~~ADM	0x00
~~CONN	0x01
~~RESET_WAIT	0x02
~~RESET_CHECK	0x03
~~SETUP	0x04
~~RESET	0x05
~~D_CONN	0x06

示例3 连接统计信息 (续)

表 1 : LLC2 States	
~~ERROR	0x07
~~NORMAL	0x08
~~BUSY	0x09
~~REJECT	0x0a
~~AWAIT	0x0b
~~AWAIT_BUSY	0x0c
~~AWAIT_REJECT	0x0d

表 2 : timers0n	
确认	0x80
轮询	0x40
拒绝	0x20
删除繁忙	0x10
非活动	0x08
发送确认	0x04

表 3 : LLC2 Flags	
P_FLAG	0x80
F_FLAG	0x40
S_FLAG	0x20
REMOTE_BUSY	0x10
RESEND_PENDING	0x08

属性

有关下列属性的说明, 请参见 [attributes\(5\)](#) :

属性类型	属性值
可用性	system/network/llc2

文件 /dev/llc2 克隆设备

另请参见 [attributes\(5\)](#)

附注 有关 LLC2 组件、状态和标志的详细信息，请参见国际标准化组织文档 ISO 8802-2:1994 第 7 部分。

引用名	ln – make hard or symbolic links to files
用法概要	<pre> /usr/bin/ln [-fns] source_file [target] /usr/bin/ln [-fns] source_file... target /usr/xpg4/bin/ln [-fs] source_file [target] /usr/xpg4/bin/ln [-fs] source_file... target </pre>
描述	<p>In the first synopsis form, the <code>ln</code> utility creates a new directory entry (link) for the file specified by <i>source_file</i>, at the destination path specified by <i>target</i>. If <i>target</i> is not specified, the link is made in the current directory. This first synopsis form is assumed when the final operand does not name an existing directory; if more than two operands are specified and the final is not an existing directory, an error will result.</p> <p>In the second synopsis form, the <code>ln</code> utility creates a new directory entry for each file specified by a <i>source_file</i> operand, at a destination path in the existing directory named by <i>target</i>.</p> <p>The <code>ln</code> utility may be used to create both hard links and symbolic links. A hard link is a pointer to a file and is indistinguishable from the original directory entry. Any changes to a file are effective independent of the name used to reference the file. Hard links may not span file systems and may not refer to directories.</p> <p><code>ln</code> by default creates hard links. <i>source_file</i> is linked to <i>target</i>. If <i>target</i> is a directory, another file named <i>source_file</i> is created in <i>target</i> and linked to the original <i>source_file</i>.</p> <p>If <i>target</i> is an existing file and the <code>-f</code> option is not specified, <code>ln</code> will write a diagnostic message to standard error, do nothing more with the current <i>source_file</i>, and go on to any remaining <i>source_files</i>.</p> <p>A symbolic link is an indirect pointer to a file; its directory entry contains the name of the file to which it is linked. Symbolic links may span file systems and may refer to directories.</p> <p>File permissions for <i>target</i> may be different from those displayed with an <code>-l</code> listing of the <code>ls(1)</code> command. To display the permissions of <i>target</i>, use <code>ls -lL</code>. See <code>stat(2)</code> for more information.</p>
/usr/bin/ln	If <code>/usr/bin/ln</code> determines that the mode of <i>target</i> forbids writing, it prints the mode (see <code>chmod(1)</code>), asks for a response, and reads the standard input for one line. If the response is affirmative, the link occurs, if permissible. Otherwise, the command exits.
/usr/xpg4/bin/ln	When creating a hard link, and the source file is itself a symbolic link, the target will be a hard link to the file referenced by the symbolic link, not to the symbolic link object itself (<i>source_file</i>).
选项	<p>The following options are supported for both <code>/usr/bin/ln</code> and <code>/usr/xpg4/bin/ln</code>:</p> <ul style="list-style-type: none"> -f Links files without questioning the user, even if the mode of <i>target</i> forbids writing. This is the default if the standard input is not a terminal.

-s Creates a symbolic link.

If the -s option is used with two arguments, *target* may be an existing directory or a non-existent file. If *target* already exists and is not a directory, an error is returned. *source_file* may be any path name and need not exist. If it exists, it may be a file or directory and may reside on a different file system from *target*. If *target* is an existing directory, a file is created in directory *target* whose name is *source_file* or the last component of *source_file*. This file is a symbolic link that references *source_file*. If *target* does not exist, a file with name *target* is created and it is a symbolic link that references *source_file*.

If the -s option is used with more than two arguments, *target* must be an existing directory or an error will be returned. For each *source_file*, a link is created in *target* whose name is the last component of *source_file*. Each new *source_file* is a symbolic link to the original *source_file*. The files and *target* may reside on different file systems.

/usr/bin/ln

The following option is supported for /usr/bin/ln only:

-n If *target* is an existing file, writes a diagnostic message to stderr and goes on to any remaining *source_files*. The -f option overrides this option. This is the default behavior for /usr/bin/ln and /usr/xpg4/bin/ln, and is silently ignored.

操作数

The following operands are supported:

source_file A path name of a file to be linked. This can be either a regular or special file. If the -s option is specified, *source_file* can also be a directory.

target The path name of the new directory entry to be created, or of an existing directory in which the new directory entries are to be created.

用法

See [largefile\(5\)](#) for the description of the behavior of ln when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of ln: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLS_PATH.

退出状态

The following exit values are returned:

0 All the specified files were linked successfully

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/ln

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Enabled

/usr/xpg4/bin/ln

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[chmod\(1\)](#), [ls\(1\)](#), [stat\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注

A symbolic link to a directory behaves differently than you might expect in certain cases. While an [ls\(1\)](#) command on such a link displays the files in the pointed-to directory, entering `ls -l` displays information about the link itself:

```
example% ln -s dir link
example% ls link
file1 file2 file3 file4
example% ls -l link
lrwxrwxrwx  1 user          7 Jan 11 23:27 link -> dir
```

When you change to a directory (see [cd\(1\)](#)) through a symbolic link, using `/usr/bin/sh` or `/usr/bin/csh`, you wind up in the pointed-to location within the file system. This means that the parent of the new working directory is not the parent of the symbolic link, but rather, the parent of the pointed-to directory. This will also happen when using `cd` with the `-P` option from `/usr/bin/ksh` or `/usr/xpg4/bin/sh`. For instance, in the following case, the final working directory is `/usr` and not `/home/user/linktest`.

```
example% pwd
/home/user/linktest
example% ln -s /usr/tmp symlink
example% cd symlink
example% cd . .
example% pwd
/usr
```

C shell users can avoid any resulting navigation problems by using the `pushd` and `popd` built-in commands instead of `cd`.

引用名 ln – make hard or symbolic links to files

用法概要 `/usr/ucb/ln [-fs] filename [linkname]`
`/usr/ucb/ln [-fs] pathname... directory`

描述 The `/usr/ucb/ln` utility creates an additional directory entry, called a link, to a file or directory. Any number of links can be assigned to a file. The number of links does not affect other file attributes such as size, protections, data, etc.

filename is the name of the original file or directory. *linkname* is the new name to associate with the file or filename. If *linkname* is omitted, the last component of *filename* is used as the name of the link.

If the last argument is the name of a directory, symbolic links are made in that directory for each *pathname* argument; `/usr/ucb/ln` uses the last component of each *pathname* as the name of each link in the named *directory*.

A hard link (the default) is a standard directory entry just like the one made when the file was created. Hard links can only be made to existing files. Hard links cannot be made across file systems (disk partitions, mounted file systems). To remove a file, all hard links to it must be removed, including the name by which it was first created; removing the last hard link releases the inode associated with the file.

A symbolic link, made with the `-s` option, is a special directory entry that points to another named file. Symbolic links can span file systems and point to directories. In fact, you can create a symbolic link that points to a file that is currently absent from the file system; removing the file that it points to does not affect or alter the symbolic link itself.

A symbolic link to a directory behaves differently than you might expect in certain cases. While an `ls(1)` on such a link displays the files in the pointed-to directory, an `'ls -l'` displays information about the link itself:

```
example% /usr/ucb/ln -s dir link
example% ls link
file1 file2 file3 file4
example% ls -l link
lrwxrwxrwx  1 user          7 Jan 11 23:27 link -> dir
```

When you use `cd(1)` to change to a directory through a symbolic link, you wind up in the pointed-to location within the file system. This means that the parent of the new working directory is not the parent of the symbolic link, but rather, the parent of the pointed-to directory. For instance, in the following case the final working directory is `/usr` and not `/home/user/linktest`.

```
example% pwd
/home/user/linktest
example% /usr/ucb/ln -s /var/tmp symlink
example% cd symlink
```

```
example% cd . .
example% pwd
/usr
```

C shell users can avoid any resulting navigation problems by using the `pushd` and `popd` built-in commands instead of `cd`.

选项

- f Force a hard link to a directory. This option is only available to the super-user, and should be used with extreme caution.
- s Create a symbolic link or links.

用法

See [largefile\(5\)](#) for the description of the behavior of `ln` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例1 The `/usr/ucb/ln` command

The commands below illustrate the effects of the different forms of the `/usr/ucb/ln` command:

```
example% /usr/ucb/ln file link
example% ls -F file link
file link
example% /usr/ucb/ln -s file symlink
example% ls -F file symlink
file symlink@
example% ls -li file link symlink
 10606 -rw-r--r--  2 user          0 Jan 12 00:06 file
 10606 -rw-r--r--  2 user          0 Jan 12 00:06 link
 10607 lrwxrwxrwx  1 user          4 Jan 12 00:06 symlink -> file
example% /usr/ucb/ln -s nonesuch devoid
example% ls -F devoid
devoid@
example% cat devoid
devoid: No such file or directory
example% /usr/ucb/ln -s /proto/bin/* /tmp/bin
example% ls -F /proto/bin /tmp/bin
/proto/bin:
x*      y*      z*

/tmp/bin:
x@      y@      z@
```

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见

[cp\(1\)](#), [ls\(1\)](#), [mv\(1\)](#), [rm\(1\)](#), [link\(2\)](#), [readlink\(2\)](#), [stat\(2\)](#), [symlink\(2\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

附注

When the last argument is a directory, simple basenames should not be used for *pathname* arguments. If a basename is used, the resulting symbolic link points to itself:

```
example% /usr/ucb/ln -s file /tmp
example% ls -l /tmp/file
lrwxrwxrwx  1 user          4 Jan 12 00:16 /tmp/file -> file
example% cat /tmp/file
/tmp/file: Too many levels of symbolic links
```

To avoid this problem, use full pathnames, or prepend a reference to the PWD variable to files in the working directory:

```
example% rm /tmp/file
example% /usr/ucb/ln -s $PWD/file /tmp
lrwxrwxrwx  1 user 4      Jan 12 00:16 /tmp/file ->
/home/user/subdir/file
```

引用名 loadkeys, dumpkeys – load and dump keyboard translation tables

用法概要 loadkeys [*filename*]

dumpkeys

描述

loadkeys reads the file specified by *filename*, and modifies the keyboard streams module's translation tables. If no file is specified, loadkeys loads the file: `/usr/share/lib/keytables/type_tt/layout_dd`, where *tt* is the value returned by the `KIOCTYPE ioctl`, and *dd* is the value returned by the `KIOCLAYOUT ioctl` (see [kb\(7M\)](#)). These keytable files specify only the entries that change between the specified layout and the default layout for the particular keyboard type. On self-identifying keyboards, the value returned by the `KIOCLAYOUT ioctl` is set from the DIP switches.

dumpkeys writes the current contents of the keyboard streams module's translation tables, in the format specified by [keytables\(4\)](#), to the standard output.

文件

`/usr/share/lib/keytables/layout_dd` default keytable files

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed

另请参见

[kbd\(1\)](#), [keytables\(4\)](#), [attributes\(5\)](#), [kb\(7M\)](#), [usbkbm\(7M\)](#)

引用名 locale – get locale-specific information

用法概要 locale [-a | -m]

locale [-ck] *name*...

描述 The `locale` utility writes information about the current locale environment, or all public locales, to the standard output. For the purposes of this section, a *public locale* is one provided by the implementation that is accessible to the application.

When `locale` is invoked without any arguments, it summarizes the current locale environment for each locale category as determined by the settings of the environment variables.

When invoked with operands, it writes values that have been assigned to the keywords in the locale categories, as follows:

- Specifying a keyword name selects the named keyword and the category containing that keyword.
- Specifying a category name selects the named category and all keywords in that category.

选项 The following options are supported:

- a Writes information about all available public locales. The available locales include POSIX, representing the POSIX locale.
- c Writes the names of selected locale categories. The `-c` option increases readability when more than one category is selected (for example, via more than one keyword name or via a category name). It is valid both with and without the `-k` option.
- k Writes the names and values of selected keywords. The implementation may omit values for some keywords; see OPERANDS.
- m Writes names of available charmaps; see [localedef\(1\)](#).

操作数 The following operand is supported:

name The name of a locale category, the name of a keyword in a locale category, or the reserved name `charmap`. The named category or keyword is selected for output. If a single *name* represents both a locale category name and a keyword name in the current locale, the results are unspecified; otherwise, both category and keyword names can be specified as *name* operands, in any sequence.

示例 示例 1 Examples of the locale utility

In the following examples, the assumption is that locale environment variables are set as follows:

```
LANG=locale_x LC_COLLATE=locale_y
```

The command `locale` would result in the following output:

示例 1 Examples of the locale utility (续)

```
LANG=locale_x
LC_CTYPE="locale_x"
LC_NUMERIC="locale_x"
LC_TIME="locale_x"
LC_COLLATE=locale_y
LC_MONETARY="locale_x"
LC_MESSAGES="locale_x"
LC_ALL=
```

The command

```
LC_ALL=POSIX locale -ck decimal_point
```

would produce:

```
LC_NUMERIC
decimal_point="."
```

The following command shows an application of `locale` to determine whether a user-supplied response is affirmative:

```
if printf "%s\n" "$response" | /usr/xpg4/bin/grep -Eq\
    "${locale yesexpr}"
then
    affirmative processing goes here
else
    non-affirmative processing goes here
fi
```

环境变量

See [environ\(5\)](#) for the descriptions of `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

The `LANG`, `LC_*`, and `NLSPATH` environment variables must specify the current locale environment to be written out. These environment variables are used if the `-a` option is not specified.

退出状态

The following exit values are returned:

- 0 All the requested information was found and output successfully.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[localedef\(1\)](#), [attributes\(5\)](#), [charmap\(5\)](#), [environ\(5\)](#), [locale\(5\)](#), [locale_alias\(5\)](#), [standards\(5\)](#)

附注

If `LC_CTYPE` or keywords in the category `LC_CTYPE` are specified, only the values in the range `0x00-0x7f` are written out.

If `LC_COLLATE` or keywords in the category `LC_COLLATE` are specified, no actual values are written out.

The locale names shown at `locale -a` output are restricted to canonical locale names. For the accepted and supported locale name aliases, see [locale_alias\(5\)](#)

引用名	localedef – define locale environment
用法概要	<pre>localedef [-c] [-C <i>compiler_options</i>] [-f <i>charmap</i>] [-i <i>sourcefile</i>] [-L <i>linker_options</i>] [-m <i>model</i>] [-u <i>code_set_name</i>] [-W <i>cc, arg</i>] [-x <i>extensions_file</i>] <i>localename</i></pre>
描述	<p>The <code>localedef</code> utility converts source definitions for locale categories into a format usable by the functions and utilities whose operational behavior is determined by the setting of the locale environment variables; see environ(5).</p> <p>The utility reads source definitions for one or more locale categories belonging to the same locale from the file named in the <code>-i</code> option (if specified) or from standard input.</p> <p>Each category source definition is identified by the corresponding environment variable name and terminated by an <code>END <i>category-name</i></code> statement. The following categories are supported.</p> <p><code>LC_CTYPE</code> Defines character classification and case conversion.</p> <p><code>LC_COLLATE</code> Defines collation rules.</p> <p><code>LC_MONETARY</code> Defines the format and symbols used in formatting of monetary information.</p> <p><code>LC_NUMERIC</code> Defines the decimal delimiter, grouping and grouping symbol for non-monetary numeric editing.</p> <p><code>LC_TIME</code> Defines the format and content of date and time information.</p> <p><code>LC_MESSAGES</code> Defines the format and values of affirmative and negative responses.</p>
选项	<p>The following options are supported:</p> <p><code>-c</code> Creates permanent output even if warning messages have been issued.</p> <p><code>-C <i>compiler_options</i></code> Passes the <i>compiler_options</i> to the C compiler (<code>cc</code>). If more than one option is specified, then the options must be enclosed in quotes (" ").</p> <p style="padding-left: 40px;">This is an old option. Use the <code>-W <i>cc,arg</i></code> option instead.</p> <p><code>-f <i>charmap</i></code> Specifies the pathname of a file containing a mapping of character symbols and collating element symbols to actual character encodings. This option must be specified if symbolic names (other than collating symbols defined in a <code>collating-symbol</code> keyword) are used. If the <code>-f</code> option is not present, the default character mapping will be used.</p> <p><code>-i <i>sourcefile</i></code> The path name of a file containing the source definitions. If this option is not present, source definitions will be read from standard input.</p>

- L linker_options* Passes the *linker_options* to the C compiler (cc) that follows the C source filename. If more than one option is specified, then the options must be enclosed in quotes (" ").
- This is an old option. Use the *-W cc,arg* option instead.
- m model* Specifies whether `localedef` will generate a 64-bit or a 32-bit locale object.
- Specify *model* as `ilp32` to generate a 32-bit locale object. Specify `lp64` to generate a 64-bit locale object. If the *-m* option is not specified, `localedef` generates a 32-bit locale object. And if no other options than *-c*, *-f*, and *-i* options are specified and if the system running `localedef` supports the 64-bit environment, `localedef` additionally generates a 64-bit locale object.
- u code_set_name* Specifies the name of a codeset used as the target mapping of character symbols and collating element symbols whose encoding values are defined in terms of the ISO/IEC 10646-1: 2000 standard position constant values. See NOTES.
- W cc,arg* Passes *arg* options to the C compiler. Each argument must be separated from the preceding by only a comma. A comma can be part of an argument by escaping it with an immediately preceding backslash character; the backslash is removed from the resulting argument.
- Use this option instead of the *-C* and *-L* options.
- x extensions_file* Specifies the name of an extension file where various `localedef` options are listed. See [locale\(5\)](#).

操作数

The following operand is supported:

- localename* Identifies the locale. If the name contains one or more slash characters, *localename* will be interpreted as a path name where the created locale definitions will be stored. This capability may be restricted to users with appropriate privileges. (As a consequence of specifying one *localename*, although several categories can be processed in one execution, only categories belonging to the same locale can be processed.)

Output

`localedef` creates a temporary C source file that represents the locale's data. `localedef` then calls the C compiler to compile this C source file into a shared object.

If the *-m ilp32* option is specified, `localedef` calls the C compiler for generating 32-bit objects and generates a 32-bit locale object. If the *-m lp64* option is specified, `localedef` calls the C compiler for generating 64-bit objects and generates a 64-bit locale object.

If the `-m` option is not specified, `localedef` calls the C compiler for generating 32-bit objects and generates a 32-bit locale object. If no other options than `-c`, `-f`, and `-i` options are specified and if the system running `localedef` supports the 64-bit environment, `localedef` additionally calls the C compiler for generating 64-bit objects and generates a 64-bit locale object.

If no option to the C compiler is explicitly specified using the `-W`, `-C`, or `-L` options, `localedef` calls the C compiler with appropriate C compiler options to generate a locale object or objects.

If the `-m ilp32` option is specified, `localedef` generates a 32-bit locale object named:

localename.so.version_number

If the `-m lp64` option is specified, `localedef` generates a 64-bit locale object named:

localename.so.version_number

If the `-m` option is not specified, `localedef` generates a 32-bit locale object named:

localename.so.version_number

and, if appropriate, generates a 64-bit locale object named:

64-bit_architecture_name/localename.so.version_number

The shared object for the 32-bit environment must be moved to:

/usr/lib/locale/localename/localename.so.version_number

The shared object for the 64-bit environment on SPARC must be moved to:

/usr/lib/locale/localename/sparcv9/localename.so.version_number

The shared object for the 64-bit environment on AMD64 must be moved to:

/usr/lib/locale/<localename>/amd64/<localename>.so.<version_number>

`localedef` also generates a text file named *localename* that is used for information only.

环境变量

See [environ\(5\)](#) for definitions of the following environment variables that affect the execution of `localedef`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

- 0 No errors occurred and the locales were successfully created.
- 1 Warnings occurred and the locales were successfully created.

- 2 The locale specification exceeded implementation limits or the coded character set or sets used were not supported by the implementation, and no locale was created.
- 3 The capability to create new locales is not supported by the implementation.
- >3 Warnings or errors occurred and no output was created.

If an error is detected, no permanent output will be created.

文件

`/usr/lib/localedef/extensions/generic_eucbc.x`

Describes what a generic EUC locale uses in the system. This file is used by default.

`/usr/lib/localedef/extensions/single_byte.x`

Describes a generic single-byte file used in the system.

`/usr/lib/locale/locaename/locaename.so.version_number`

The shared object for the 32-bit environment.

`/usr/lib/locale/locaename/sparcv9/locaename.so.version_number`

The shared object for the 64-bit environment on SPARC.

`/usr/lib/locale/<locaename>/amd64/<locaename>.so.<version_number>`

The shared object for the 64-bit environment on AMD64.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[locale\(1\)](#), [iconv_open\(3C\)](#), [nl_langinfo\(3C\)](#), [strftime\(3C\)](#), [attributes\(5\)](#), [charmap\(5\)](#), [environ\(5\)](#), [extensions\(5\)](#), [locale\(5\)](#), [standards\(5\)](#)

警告

If warnings occur, permanent output will be created if the `-c` option was specified. The following conditions will cause warning messages to be issued:

- If a symbolic name not found in the *charmap* file is used for the descriptions of the LC_CTYPE or LC_COLLATE categories (for other categories, this will be an error conditions).
- If optional keywords not supported by the implementation are present in the source.

附注

When the `-u` option is used, the *code_set_name* option-argument is interpreted as a name of a codeset to which the ISO/IEC 10646-1: 2000 standard position constant values are converted. Both the ISO/IEC 10646-1: 2000 standard position constant values and other formats (decimal, hexadecimal, or octal) are valid as encoding values within the *charmap* file. The codeset can be any codeset that is supported by the [iconv_open\(3C\)](#) function on the system.

When conflicts occur between the charmap specification of *code_set_name*, *mb_cur_max*, or *mb_cur_min* and the corresponding value for the codeset represented by the `-u` option-argument *code_set_name*, the `localedef` utility fails as an error.

When conflicts occur between the charmap encoding values specified for symbolic names of characters of the portable character set and the character encoding values defined by the US-ASCII, the result is unspecified.

If a non-printable character in the charmap has a width specified that is not `-1`, `localedef` generates a warning.

引用名 logger – add entries to the system log

用法概要 logger [-i] [-f *file*] [-p *priority*] [-t *tag*] [*message*] ...

描述 The logger command provides a method for adding one-line entries to the system log file from the command line. One or more *message* arguments can be given on the command line, in which case each is logged immediately. If this is unspecified, either the file indicated with -f or the standard input is added to the log. Otherwise, a *file* can be specified, in which case each line in the file is logged. If neither is specified, logger reads and logs messages on a line-by-line basis from the standard input.

选项 The following options are supported:

- f *file* Uses the contents of *file* as the message to log.
- i Logs the process ID of the logger process with each line.
- p *priority* Enters the message with the specified *priority*. The message priority can be specified numerically, or as a *facility.level* pair. For example, '-p local3.info' assigns the message priority to the info level in the local3 facility. The default priority is user.notice.
- t *tag* Marks each line added to the log with the specified *tag*.

操作数 The following operand is supported:

message One of the string arguments whose contents are concatenated together, in the order specified, separated by single space characters.

示例 示例 1 Examples of the logger command

The following example:

```
example% logger System rebooted
```

logs the message 'System rebooted' to the default priority level notice to be treated by syslogd as are other messages to the facility user.

The next example:

```
example% logger -p local0.notice -t HOSTIDM -f /dev/idmc
```

reads from the file /dev/idmc and logs each line in that file as a message with the tag 'HOSTIDM' at priority level notice to be treated by syslogd as are other messages to the facility local0.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of logger: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLS_PATH.

退出状态 The following exit values are returned:

- 0 Successful completion.

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[mailx\(1\)](#), [write\(1\)](#), [syslogd\(1M\)](#), [syslog\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名 logger – add entries to the system log

用法概要 /usr/ucb/logger [-f *filename*] [-i] [-p *priority*] [-t *tag*] mm
[*message*]...

描述 The logger utility provides a method for adding one-line entries to the system log file from the command line. One or more *message* arguments can be given on the command line, in which case each is logged immediately. If *message* is unspecified, either the file indicated with -f or the standard input is added to the log. Otherwise, a *filename* can be specified, in which case each line in the file is logged. If neither is specified, logger reads and logs messages on a line-by-line basis from the standard input.

选项 The following options are supported:

- i Log the process ID of the logger process with each line.
- f *filename* Use the contents of *filename* as the message to log.
- p *priority* Enter the message with the specified *priority*. The message priority can be specified numerically, or as a *facility.level* pair. For example, '-p local3.info' assigns the message priority to the info level in the local3 facility. The default priority is user.notice.
- t *tag* Mark each line added to the log with the specified *tag*.

示例 示例1 Logging a message

The command:

```
example% logger System rebooted
```

will log the message 'System rebooted' to the facility at priority notice to be treated by syslogd as other messages to the facility notice are.

示例2 Logging messages from a file

The command:

```
example% logger -p local0.notice -t HOSTIDM -f /dev/idmc
```

will read from the file /dev/idmc and will log each line in that file as a message with the tag 'HOSTIDM' at priority notice to be treated by syslogd as other messages to the facility local0 are.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见

[syslogd\(1M\)](#), [syslog\(3C\)](#), [attributes\(5\)](#)

引用名 login – sign on to the system

用法概要 login [-p] [-d *device*] [-R *repository*] [-s *service*]
[-t *terminal*] [-u *identity*] [-U *ruser*]
[-h *hostname* [*terminal*] | -r *hostname*]
[*name* [*environ*]...]

描述 The `login` command is used at the beginning of each terminal session to identify oneself to the system. `login` is invoked by the system when a connection is first established, after the previous user has terminated the login shell by issuing the `exit` command.

Login cannot be invoked as a command, except by the superuser.

If `login` is invoked as a command, it must replace the initial command interpreter. To invoke `login` in this fashion, type:

exec login

from the initial shell. The C shell and Korn shell have their own built-ins of `login`. See [ksh\(1\)](#), [ksh88\(1\)](#), and [csh\(1\)](#) for descriptions of login built-ins and usage.

`login` asks for your user name, if it is not supplied as an argument, and your password, if appropriate. Where possible, echoing is turned off while you type your password, so it does not appear on the written record of the session.

If you make any mistake in the login procedure, the message:

```
Login incorrect
```

is printed and a new login prompt appears. If you make five incorrect login attempts, all five can be logged in `/var/adm/loginlog`, if it exists. The TTY line is dropped.

If password aging is turned on and the password has aged (see [passwd\(1\)](#) for more information), the user is forced to change the password. In this case the `/etc/nsswitch.conf` file is consulted to determine password repositories (see [nsswitch.conf\(4\)](#)). The password update configurations supported are limited to the following five cases.

- `passwd: files`
- `passwd: files nis`
- `passwd: compat (==> files nis)`

Failure to comply with the configurations prevents the user from logging onto the system because [passwd\(1\)](#) fails. If you do not complete the login successfully within a certain period of time, it is likely that you are silently disconnected.

After a successful login, accounting files are updated. Device owner, group, and permissions are set according to the contents of the `/etc/logindevperm` file, and the time you last logged in is printed (see [logindevperm\(4\)](#)).

The user-ID, group-ID, supplementary group list, and working directory are initialized, and the command interpreter is started.

The basic *environment* is initialized to:

```
HOME=your-login-directory
LOGNAME=your-login-name
PATH=/usr/bin:
SHELL=last-field-of-passwd-entry
MAIL=/var/mail/
TZ=timezone-specification
```

For Bourne shell and Korn shell logins, the shell executes `/etc/profile` and `$HOME/.profile`, if it exists.

For the ksh Korn shell, an interactive shell then executes `/etc/ksh.kshrc`, followed by the file specified by the ENV environment variable. If ENV is not set, this defaults to `$HOME/.kshrc`. For the ksh and `/usr/xpg4/bin/sh` Korn Shell, an interactive shell executes the file named by ENV (no default).

For C shell logins, the shell executes `/etc/.login`, `$HOME/.cshrc`, and `$HOME/.login`. The default `/etc/profile` and `/etc/.login` files check quotas (see [quota\(1M\)](#)), print `/etc/motd`, and check for mail. None of the messages are printed if the file `$HOME/.hushlogin` exists. The name of the command interpreter is set to `-` (dash), followed by the last component of the interpreter's path name, for example, `-sh`.

If the *login-shell* field in the password file (see [passwd\(4\)](#)) is empty, then the default command interpreter, `/usr/bin/sh`, is used. If this field is `*` (asterisk), then the named directory becomes the root directory. At that point, `login` is re-executed at the new level, which must have its own root structure.

The environment can be expanded or modified by supplying additional arguments to `login`, either at execution time or when `login` requests your login name. The arguments can take either the form `xxx` or `xxx=yyy`. Arguments without an `=` (equal sign) are placed in the environment as:

```
Ln=xxx
```

where *n* is a number starting at 0 and is incremented each time a new variable name is required. Variables containing an `=` (equal sign) are placed in the environment without modification. If they already appear in the environment, then they replace the older values.

There are two exceptions: The variables PATH and SHELL cannot be changed. This prevents people logged into restricted shell environments from spawning secondary shells that are not restricted. `login` understands simple single-character quoting conventions. Typing a `\` (backslash) in front of a character quotes it and allows the inclusion of such characters as spaces and tabs.

Alternatively, you can pass the current environment by supplying the `-p` flag to `login`. This flag indicates that all currently defined environment variables should be passed, if possible, to

the new environment. This option does not bypass any environment variable restrictions mentioned above. Environment variables specified on the login line take precedence, if a variable is passed by both methods.

To enable remote logins by root, edit the `/etc/default/login` file by inserting a # (pound sign) before the `CONSOLE=/dev/console` entry. See FILES.

安全

For accounts in the files (`passwd(4)` and `shadow(4)`) name service, or the `ldap` name service, when configured with `enableShadowUpdate true`, the account can be configured to be automatically locked if successive failed login attempts equals or exceeds the configured value. See `ldapclient(1M)`, `user_attr(4)`, `policy.conf(4)`, and `pam_unix_auth(5)`.

The `login` command uses `pam(3PAM)` for authentication, account management, session management, and password management. The PAM configuration policy, listed in either `/etc/pam.conf` or `/etc/pam.d/login`, specifies the modules to be used for `login`. Here is a partial `pam.conf` file with entries for the `login` command using the UNIX authentication, account management, and session management modules:

```
login auth      required pam_authtok_get.so.1
login auth      required pam_dhkeys.so.1
login auth      required pam_unix_auth.so.1
login auth      required pam_dial_auth.so.1

login account   requisite pam_roles.so.1
login account   required pam_unix_account.so.1

login session   required pam_unix_session.so.1
```

The equivalent PAM configuration in `/etc/pam.d/` would be the following entries in `/etc/pam.d/login`:

```
auth      required pam_authtok_get.so.1
auth      required pam_dhkeys.so.1
auth      required pam_unix_auth.so.1
auth      required pam_dial_auth.so.1
account   requisite pam_roles.so.1
account   required pam_unix_account.so.1

session   required pam_unix_session.so.1
```

The Password Management stack in `/etc/pam.conf` typically looks like the following:

```
other password required pam_dhkeys.so.1
other password requisite pam_authtok_get.so.1
other password requisite pam_authtok_check.so.1
other password required pam_authtok_store.so.1
```

If there are no entries for a PAM service in `/etc/pam.conf` and `/etc/pam.d/service` then the entries for the “other” service in `/etc/pam.conf` are used. If there are not any entries in

`/etc/pam.conf` for the “other” service, then the entries in `/etc/pam.d/other` will be used. If multiple authentication modules are listed, then the user can be prompted for multiple passwords.

When `login` is invoked through `rlogind` or `telnetd`, the service name used by PAM is `rlogin` or `telnet`, respectively.

选项

The following options are supported:

-d *device*

`login` accepts a device option, *device*. *device* is taken to be the path name of the TTY port `login` is to operate on. The use of the device option can be expected to improve `login` performance, since `login` does not need to call `ttyname(3C)`. The `-d` option is available only to users whose UID and effective UID are root. Any other attempt to use `-d` causes `login` to quietly exit.

-h *hostname* [*terminal*]

Used by `in.telnetd(1M)` to pass information about the remote host and terminal type.

Terminal type as a second argument to the `-h` option should not start with a hyphen (`-`).

-p

Used to pass environment variables to the `login` shell.

-r *hostname*

Used by `in.rlogind(1M)` to pass information about the remote host.

-R *repository*

Used to specify the PAM repository that should be used to tell PAM about the “identity” (see option `-u` below). If no “identity” information is passed, the repository is not used.

-s *service*

Indicates the PAM service name that should be used. Normally, this argument is not necessary and is used only for specifying alternative PAM service names. For example: “`ktelnet`” for the Kerberized telnet process.

-u *identity*

Specifies the “identity” string associated with the user who is being authenticated. This usually is *not* be the same as that user's Unix login name. For Kerberized login sessions, this is the Kerberos principal name associated with the user.

-U *ruser*

Indicates the name of the person attempting to login on the remote side of the `rlogin` connection. When `in.rlogind(1M)` is operating in Kerberized mode, that daemon processes the terminal and remote user name information prior to invoking `login`, so the “`ruser`” data is indicated using this command line parameter. Normally (non-Kerberos authenticated `rlogin`), the `login` daemon reads the remote user information from the client.

退出状态

The following exit values are returned:

0
Successful operation.

non-zero
Error.

文件

`$HOME/.cshrc`
Initial commands for each csh.

`$HOME/.hushlogin`
Suppresses login messages.

`$HOME/.kshrc`
User's commands for interactive ksh, if `$ENV` is unset; executes after `/etc/ksh.kshrc`.

`$HOME/.login`
User's login commands for csh.

`$HOME/.profile`
User's login commands for sh and ksh.

`$HOME/.rhosts`
Private list of trusted hostname/username combinations.

`/etc/.login`
System-wide csh login commands.

`/etc/issue`
Issue or project identification.

`/etc/ksh.kshrc`
System-wide commands for interactive ksh.

`/etc/logindevperm`
Login-based device permissions.

`/etc/motd`
Message-of-the-day.

`/etc/nologin`
Message displayed to users attempting to login during machine shutdown.

`/etc/passwd`
Password file.

`/etc/profile`
System-wide sh and ksh login commands.

`/etc/shadow`
List of users' encrypted passwords.

`/usr/bin/sh`
User's default command interpreter.

`/var/adm/lastlog`
Time of last login.

`/var/adm/loginlog`
Record of failed login attempts.

`/var/adm/utmpx`
Accounting.

`/var/adm/wtmpx`
Accounting.

`/var/mail/your-name`
Mailbox for user *your-name*.

`/etc/default/login`
Default value can be set for the following flags in `/etc/default/login`. Default values are specified as comments in the `/etc/default/login` file, for example, `TIMEZONE=EST5EDT`.

TIMEZONE
Sets the TZ environment variable of the shell (see [environ\(5\)](#)).

HZ
Sets the HZ environment variable of the shell.

ULIMIT
Sets the file size limit for the login. Units are disk blocks. Default is zero (no limit).

CONSOLE
If set, root can login on that device only. This does not prevent execution of remote commands with [rsh\(1\)](#). Comment out this line to allow login by root.

PASSREQ
Determines if login requires a non-null password.

ALTSHELL
Determines if login should set the SHELL environment variable.

PATH
Sets the initial shell PATH variable.

SUPATH
Sets the initial shell PATH variable for root.

TIMEOUT
Sets the number of seconds (between 0 and 900) to wait before abandoning a login session.

UMASK
Sets the initial shell file creation mode mask. See [umask\(1\)](#).

SYSLOG

Determines whether the [syslog\(3C\)](#) LOG_AUTH facility should be used to log all root logins at level LOG_NOTICE and multiple failed login attempts at LOG_CRIT.

DISABLETIME

If present, and greater than zero, the number of seconds that login waits after RETRIES failed attempts or the PAM framework returns PAM_ABORT. Default is 20 seconds. Minimum is 0 seconds. No maximum is imposed.

SLEEPTIME

If present, sets the number of seconds to wait before the login failure message is printed to the screen. This is for any login failure other than PAM_ABORT. Another login attempt is allowed, providing RETRIES has not been reached or the PAM framework is returned PAM_MAXTRIES. Default is 4 seconds. Minimum is 0 seconds. Maximum is 5 seconds.

Both [su\(1M\)](#) and [slogin\(1M\)](#) are affected by the value of SLEEPTIME.

RETRIES

Sets the number of retries for logging in (see [pam\(3PAM\)](#)). The default is 5. The maximum number of retries is 15. For accounts configured with automatic locking (see SECURITY above), the account is locked and login exits. If automatic locking has not been configured, login exits without locking the account.

SYSLOG_FAILED_LOGINS

Used to determine how many failed login attempts are allowed by the system before a failed login message is logged, using the [syslog\(3C\)](#) LOG_NOTICE facility. For example, if the variable is set to 0, login logs *all* failed login attempts.

Of the flags listed in `/etc/default/login`, [sshd\(1M\)](#) (which see) uses:

- TIMEZONE
- HZ
- ALTSHELL
- PATH
- SUPATH
- CONSOLE
- PASSREQ
- UMASK
- ULIMIT
- RETRIES
- SYSLOG_AFTER_FAILED_LOGINS

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

另请参见

[csh\(1\)](#), [exit\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [mail\(1\)](#), [mailx\(1\)](#), [newgrp\(1\)](#), [passwd\(1\)](#), [rlogin\(1\)](#), [rsh\(1\)](#), [sh\(1\)](#), [shell_builtins\(1\)](#), [telnet\(1\)](#), [umask\(1\)](#), [in.rlogind\(1M\)](#), [in.telnetd\(1M\)](#), [user_attr\(4\)](#) and [policy.conf\(4\)](#), [logins\(1M\)](#), [quota\(1M\)](#), [sshd\(1M\)](#), [su\(1M\)](#), [sulogin\(1M\)](#), [syslogd\(1M\)](#), [useradd\(1M\)](#), [userdel\(1M\)](#), [pam\(3PAM\)](#), [rcmd\(3SOCKET\)](#), [syslog\(3C\)](#), [ttyname\(3C\)](#), [auth_attr\(4\)](#), [exec_attr\(4\)](#), [hosts.equiv\(4\)](#), [issue\(4\)](#), [logindevperm\(4\)](#), [loginlog\(4\)](#), [nologin\(4\)](#), [nsswitch.conf\(4\)](#), [pam.conf\(4\)](#), [passwd\(4\)](#), [policy.conf\(4\)](#), [profile\(4\)](#), [shadow\(4\)](#), [user_attr\(4\)](#), [utmpx\(4\)](#), [wtmpx\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [pam_unix_account\(5\)](#), [pam_unix_auth\(5\)](#), [pam_unix_session\(5\)](#), [pam_authok_check\(5\)](#), [pam_authok_get\(5\)](#), [pam_authok_store\(5\)](#), [pam_dhkeys\(5\)](#), [pam_passwd_auth\(5\)](#), [termio\(7I\)](#)

诊断

Login incorrect

The user name or the password cannot be matched.

Not on system console

Root login denied. Check the `CONSOLE` setting in `/etc/default/login`.

No directory! Logging in with home=/ The user's home directory named in the [passwd\(4\)](#) database cannot be found or has the wrong permissions. Contact your system administrator.

No shell

Cannot execute the shell named in the [passwd\(4\)](#) database. Contact your system administrator.

NO LOGINS: System going down in *N* minutes

The machine is in the process of being shut down and logins have been disabled.

警告

Users with a UID greater than 76695844 are not subject to password aging, and the system does not record their last login time.

If you use the `CONSOLE` setting to disable root logins, you should arrange that remote command execution by root is also disabled. See [rsh\(1\)](#), [rcmd\(3SOCKET\)](#), and [hosts.equiv\(4\)](#) for further details.

引用名 logname – return user's login name

用法概要 /usr/bin/logname

描述 logname writes the user's login name to standard output. The login name is the string that is returned by the [getlogin\(3C\)](#) function. If `getlogin()` does not return successfully, the name corresponding to the real user ID of the calling process is used instead.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of logname: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态 The following error values are returned:

0 Successful completion.

>0 An error occurred.

文件 /etc/profile Environment for user at login time

/var/adm/utmpx User and accounting information

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [env\(1\)](#), [login\(1\)](#), [getlogin\(3C\)](#), [utmpx\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名 logout – shell built-in function to exit from a login session

用法概要

csH logout

描述

csH Terminate a login shell.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [csH\(1\)](#), [login\(1\)](#), [attributes\(5\)](#)

引用名 look – find words in the system dictionary or lines in a sorted list

用法概要 /usr/bin/look [-d] [-f] [-tc] *string* [*filename*]

描述 The look command consults a sorted *filename* and prints all lines that begin with *string*.

If no *filename* is specified, look uses /usr/share/lib/dict/words with collating sequence -df.

look limits the length of a word to search for to 256 characters.

- 选项**
- d Dictionary order. Only letters, digits, TAB and SPACE characters are used in comparisons.
 - f Fold case. Upper case letters are not distinguished from lower case in comparisons.
 - tc Set termination character. All characters to the right of *c* in *string* are ignored.

文件 /usr/share/lib/dict/words spelling list

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/extended-system-utilities

另请参见 [grep\(1\)](#), [sort\(1\)](#), [attributes\(5\)](#)

引用名 lookbib – find references in a bibliographic database

用法概要 lookbib *database*

描述

A bibliographic reference is a set of lines, constituting fields of bibliographic information. Each field starts on a line beginning with a '%', followed by a key-letter, then a blank, and finally the contents of the field, which may continue until the next line starting with '%'.

The lookbib utility uses an inverted index made by `indxbib` to find sets of bibliographic references. It reads keywords typed after the '>' prompt on the terminal, and retrieves records containing all these keywords. If nothing matches, nothing is returned except another '>' prompt.

It is possible to search multiple databases, as long as they have a common index made by `indxbib(1)`. In that case, only the first argument given to `indxbib` is specified to `lookbib`.

If `lookbib` does not find the index files (the `.i[abc]` files), it looks for a reference file with the same name as the argument, without the suffixes. It creates a file with a `.ig` suffix, suitable for use with `fgrep` (see `grep(1)`). `lookbib` then uses this `fgrep` file to find references. This method is simpler to use, but the `.ig` file is slower to use than the `.i[abc]` files, and does not allow the use of multiple reference files.

文件

`x.ia`

`x.ib`

`x.ic` index files

`x.ig` reference file

属性

See `attributes(5)` for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

另请参见

`addbib(1)`, `grep(1)`, `indxbib(1)`, `refer(1)`, `roffbib(1)`, `sortbib(1)`, `attributes(5)`

已知问题

Probably all dates should be indexed, since many disciplines refer to literature written in the 1800s or earlier.

引用名 lorder – find ordering relation for an object or library archive

用法概要 lorder *filename...*

描述 The input is one or more object or library archive *filenames* (see [ar\(1\)](#)). The standard output is a list of pairs of object file or archive member names; the first file of the pair refers to external identifiers defined in the second. The output may be processed by [tsort\(1\)](#) to find an ordering of a library suitable for one-pass access by `ld`. Note that the link editor `ld` is capable of multiple passes over an archive in the portable archive format (see [ar.h\(3HEAD\)](#)) and does not require that `lorder` be used when building an archive. The usage of the `lorder` command may, however, allow for a more efficient access of the archive during the link edit process.

The following example builds a new library from existing `.o` files.

```
ar -cr library `lorder *.o | tsort `
```

文件 TMPDIR/*symref temporary files

TMPDIR/*symdef temporary files

TMPDIR usually `/var/tmp` but can be redefined by setting the environment variable `TMPDIR`. See `tempnam()` in [tmpnam\(3C\)](#).

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities

另请参见 [ar\(1\)](#), [ld\(1\)](#), [tsort\(1\)](#), [tmpnam\(3C\)](#), [ar.h\(3HEAD\)](#), [attributes\(5\)](#)

附注 `lorder` will accept as input any object or archive file, regardless of its suffix, provided there is more than one input file. If there is but a single input file, its suffix must be `.o`.

The length of the filename for `TMPDIR` is limited to whatever `sed` allows.

引用名

ls – list contents of directory

用法概要

```

/usr/bin/ls [-aAbcCdEfFghHiklLmnopqrRsStuUvwVx1@]
           [-/ c | -/v] [-% atime | ctime | mtime | all]
           [--block-size size] [--color[=when]] [--file-type]
           [--si] [--time-style style] [file]...

/usr/xpg4/bin/ls [-aAbcCdEfFghHiklLmnopqrRsStuUvwVx1@]
                [-/ c | -/v] [-% atime | ctime | mtime | all]
                [--block-size size] [--color[=when]] [--file-type]
                [--si] [--time-style style] [file]...

/usr/xpg6/bin/ls [-aAbcCdEfFghHiklLmnopqrRsStuUvwVx1@]
                [-/ c | -/v] [-% atime | ctime | mtime | all]
                [--block-size size] [--color[=when]] [--file-type]
                [--si] [--time-style style] [file]...

```

描述

For each *file* that is a directory, `ls` lists the contents of the directory. For each *file* that is an ordinary file, `ls` repeats its name and any other information requested. The output is sorted alphabetically by default. When no argument is given, the current directory (`.`) is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments appear before directories and their contents.

There are three major listing formats. The default format for output directed to a terminal is multi-column with entries sorted down the columns. The `-1` option allows single column output and `-m` enables stream output format. In order to determine output formats for the `-C`, `-x`, and `-m` options, `ls` uses an environment variable, `COLUMNS`, to determine the number of character positions available on one output line. If this variable is not set, the [terminfo\(4\)](#) database is used to determine the number of columns, based on the environment variable, `TERM`. If this information cannot be obtained, 80 columns are assumed. If the `-w` option is used, the argument overrides any other column width.

The mode printed when the `-e`, `-E`, `-g`, `-l`, `-n`, `-o`, `-v`, `-V`, or `-@` option is in effect consists of eleven characters. The first character can be one of the following:

- d
The entry is a directory.
- D
The entry is a door.
- l
The entry is a symbolic link.
- b
The entry is a block special file.
- c
The entry is a character special file.

p
The entry is a FIFO (or “named pipe”) special file.

P
The entry is an event port.

s
The entry is an AF_UNIX address family socket.

—
The entry is an ordinary file.

The next 9 characters are interpreted as three sets of three bits each. The first set refers to the owner's permissions; the next to permissions of others in the user-group of the file; and the last to all others. Within each set, the three characters indicate permission to read, to write, and to execute the file as a program, respectively. For a directory, execute permission is interpreted to mean permission to search the directory for a specified file. The character after permissions is an ACL or extended attributes indicator. This character is an @ if extended attributes are associated with the file and the -@ option is in effect. Otherwise, this character is a plus sign (+) character if a non-trivial ACL is associated with the file or a space character if not.

If -/ and/or -% are in effect, then the extended system attributes are printed when filesystem supports extended system attributes. The display looks as follows:

```
$ls -/ c file
-rw-r--r-- 1 root    root          0 May 10 14:17 file
              {AHRSadim-u}
```

```
$ls -/ v file
-rw-r--r-- 1 root    root          0 May 10 14:17 file
              {archive,hidden,readonly,system,appendonly\
              nodump,immutable, av_modified,\
              noav_quarantined,nounlink}
```

```
$ls -l -% all file
-rw-r--r-- 1 root    root          0 May 10 14:17 file
              timestamp: atime   Jun 25 12:56:44 2007
              timestamp: ctime   May 10 14:20:23 2007
              timestamp: mtime   May 10 14:17:56 2007
              timestamp: crtime  May 10 14:17:56 2007
```

See the option descriptions of the -/ and -% option for details.

ls -l (the long list) prints its output as follows for the POSIX locale:

```
-rwxrwxrwx+ 1 smith dev  10876  May 16 9:42 part2
```

Reading from right to left, you see that the current directory holds one file, named part2. Next, the last time that file's contents were modified was 9:42 A.M. on May 16. The file contains 10,876 characters, or bytes. The owner of the file, or the user, belongs to the group

dev (perhaps indicating *development*), and his or her login name is smith. The number, in this case 1, indicates the number of links to file part2 (see [cp\(1\)](#)). The plus sign indicates that there is an ACL associated with the file. If the `-@` option has been specified, the presence of extended attributes supersede the presence of an ACL and the plus sign is replaced with an 'at' sign (@). Finally, the dash and letters tell you that user, group, and others have permissions to read, write, and execute part2.

The execute (x) symbol occupies the third position of the three-character sequence. A – in the third position would have indicated a denial of execution permissions.

The permissions are indicated as follows:

r	The file is readable.
w	The file is writable.
x	The file is executable.
–	The indicated permission is <i>not</i> granted.
s	The set-user-ID or set-group-ID bit is on, and the corresponding user or group execution bit is also on.
S	Undefined bit-state (the set-user-ID or set-group-id bit is on and the user or group execution bit is off). For group permissions, this applies only to non-regular files.
t	The 1000 (octal) bit, or sticky bit, is on (see chmod(1)), and execution is on.
T	The 1000 bit is turned on, and execution is off (undefined bit-state).
/usr/bin/ls	l Mandatory locking occurs during access (on a regular file, the set-group-ID bit is on and the group execution bit is off).
/usr/xpg4/bin/ls and /usr/xpg6/bin/ls	L Mandatory locking occurs during access (on a regular file, the set-group-ID bit is on and the group execution bit is off).

For user and group permissions, the third position is sometimes occupied by a character other than x or -. s or S also can occupy this position, referring to the state of the set-ID bit, whether

it be the user's or the group's. The ability to assume the same ID as the user during execution is, for example, used during login when you begin as root but need to assume the identity of the user you login as.

In the case of the sequence of group permissions, `l` can occupy the third position. `l` refers to mandatory file and record locking. This permission describes a file's ability to allow other files to lock its reading or writing permissions during access.

For others permissions, the third position can be occupied by `t` or `T`. These refer to the state of the sticky bit and execution permissions.

选项

The following options are supported:

`/usr/bin/ls`,
`/usr/xpg4/bin/ls`, and
`/usr/xpg6/bin/ls`

The following options are supported for all three versions:

- a
--all
Lists all entries, including those that begin with a dot (`.`), which are normally not listed.
- A
--almost-all
Lists all entries, including those that begin with a dot (`.`), with the exception of the working directory (`.`) and the parent directory (`..`).
- b
--escape
Forces printing of non-printable characters to be in the octal `\ddd` notation.
- B
--ignore-backups
Do not display any files ending with a tilde (`~`).
- c
Uses time of last modification of the i-node (file created, mode changed, and so forth) for sorting (`-t`) or printing (`-l` or `-n`).
- C
Multi-column output with entries sorted down the columns. This is the default output format.
- d
If an argument is a directory, lists only its name (not its contents). Often used with `-l` to get the status of a directory.
- e
The same as `-l`, except displays time to the second, and with one format for all files regardless of age: `mmm dd hh:mm:ss yyyy`.

-E

The same as -l, except displays time to the nanosecond and with one format for all files regardless of age: *yyyy-mm-dd hh:mm:ss.nnnnnnnnnn* (ISO 8601:2000 format).

In addition, this option displays the offset from UTC in ISO 8601:2000 standard format (*+hhmm* or *-hhmm*) or no characters if the offset is indeterminable. The offset reflects the appropriate standard or alternate offset in force at the file's displayed date and time, under the current timezone.

-f

Forces each argument to be interpreted as a directory and list the name found in each slot. This option turns off -l, -t, -s, -S, and -r, and turns on -a. The order is the order in which entries appear in the directory.

-F

--classify

Append a symbol after certain types of files to indicate the file type. The following symbols are used:

/
Directory

>
Door file

|
Named pipe (FIFO)

@
Symbolic link

=
Socket

*
Executable

-g

The same as -l, except that the owner is not printed.

-h

--human-readable

All sizes are scaled to a human readable format, for example, 14K, 234M, 2.7G, or 3.0T. Scaling is done by repetitively dividing by 1024. The last --si or -h option determines the divisor used.

-H

--dereference-command-line

If an argument is a symbolic link that references a directory, this option evaluates the file information and file type of the directory that the link references, rather than those of the link itself. However, the name of the link is displayed, rather than the referenced directory.

-i

--inode

For each file, prints the i-node number in the first column of the report.

-k

All sizes are printed in kbytes. Equivalent to --block-size=1024.

-l

Lists in long format, giving mode, ACL indication, number of links, owner, group, size in bytes, and time of last modification for each file (see above). If the file is a special file, the size field instead contains the major and minor device numbers. If the time of last modification is greater than six months ago, it is shown in the format 'month date year' for the POSIX locale. When the LC_TIME locale category is not set to the POSIX locale, a different format of the time field can be used. Files modified within six months show 'month date time'. If the file is a symbolic link, the filename is printed followed by "→" and the path name of the referenced file.

-L

--dereference

If an argument is a symbolic link, this option evaluates the file information and file type of the file or directory that the link references, rather than those of the link itself. However, the name of the link is displayed, rather than the referenced file or directory.

-m

Streams output format. Files are listed across the page, separated by commas.

-n

--numeric-uid-gid

The same as -l, except that the owner's UID and group's GID numbers are printed, rather than the associated character strings.

-o

--no-group

The same as -l, except that the group is not printed.

-p

Puts a slash (/) after each filename if the file is a directory.

-q

--hide-control-chars

Forces printing of non-printable characters in file names as the character question mark (?).

-r

--reverse

Reverses the order of sort to get reverse alphabetic, oldest first, or smallest file size first as appropriate.

-R

--recursive

Recursively lists subdirectories encountered.

-s

--size

Indicate the total number of file system blocks consumed by each file displayed.

-S

Sort by file size (in decreasing order) and for files with the same size by file name (in increasing alphabetic order) instead of just by name.

-t

Sorts by time stamp (latest first) instead of by name. The default is the last modification time. See -c, -u and -%.

-u

Uses time of last access instead of last modification for sorting (with the -t option) or printing (with the -l option).

-U

Output is unsorted.

-v

The same as -l, except that verbose ACL information is displayed as well as the -l output. ACL information is displayed even if the file or directory doesn't have an ACL.

-V

The same as -l, except that compact ACL information is displayed after the -l output.

The -V option is only applicable to file systems that support NFSv4 ACLs, such as the Solaris ZFS file system.

The format of the displayed ACL is as follows:

entry_type : *permissions* : *inheritance_flags* : *access_type*

entry_type is displayed as one of the following:

user:username

Additional user access for *username*.

group:groupname

Additional group access for group *groupname*.

owner@

File owner.

group@

File group owner.

everyone@

Everyone access, including file owner and file group owner. This is not equivalent to the POSIX other class.

The following permissions, supported by the NFSv4 ACL model, are displayed by using the -v or -V options:

read_data (r)

Permission to read the data of a file.

list_directory (r)

Permission to list the contents of a directory.

write_data (w)

Permission to modify a file's data. anywhere in the file's offset range.

add_file (w)

Permission to add a new file to a directory.

append_data (p)

The ability to modify a file's data, but only starting at EOF.

add_subdirectory (p)

Permission to create a subdirectory to a directory.

read_xattr (R)

Ability to read the extended attributes of a file.

write_xattr (W)

Ability to create extended attributes or write to the extended attribute directory.

execute (x)

Permission to execute a file.

read_attributes (a)

The ability to read basic attributes (non-ACLs) of a file.

write_attributes (A)

Permission to change basic attributes (non-ACLs) of a file.

delete (d)

Permission to delete a file.

delete_child (D)

Permission to delete a file within a directory.

read_acl (c)

Permission to read the ACL of a file.

`write_acl (C)`

Permission to write the ACL of a file.

`write_owner (o)`

Permission to change the owner of a file.

`synchronize (s)`

Permission to access file locally at server with synchronize reads and writes.

-

No permission granted

The following inheritance flags, supported by the NFSv4 ACL model, are displayed by using the `-v` or `-V` options:

`file_inherit (f)`

Inherit to all newly created files.

`dir_inherit (d)`

Inherit to all newly created directories.

`inherit_only (i)`

When placed on a directory, do not apply to the directory, only to newly created files and directories. This flag requires that either `file_inherit` and or `dir_inherit` is also specified.

`no_propagate (n)`

Indicates that ACL entries should be inherited to objects in a directory, but inheritance should stop after descending one level. This flag is dependent upon either `file_inherit` and or `dir_inherit` also being specified.

`successful_access (S)`

Indicates if an alarm or audit record should be initiated upon successful accesses. Used with audit/alarm ACE types.

`failed_access (F)`

Indicates if an alarm or audit record should be initiated when access fails. Used with audit/alarm ACE types.

`inherited (I)`

ACE was inherited.

-

No permission granted.

`access_type` is displayed as one of the following types:

`alarm` Permission field that specifies permissions that should trigger an alarm.

`allow` Permission field that specifies allow permissions.

`audit` Permission field that specifies permissions that should be audited.

`deny` Permission field that specifies deny permissions.

For example:

```
$ ls -dV /sandbox/dir.1
drwxr-xr-x+ 2 root    root          2 Jan 17 15:09 dir.1
      user:marks:r-----:fd-----:allow
      owner@:-----:-----:deny
      owner@:rwxp---A-W-Co-:-----:allow
      group@:-w-p-----:-----:deny
      group@:r-x-----:-----:allow
      everyone@:-w-p---A-W-Co-:-----:deny
      everyone@:r-x---a-R-c-s:-----:allow

$
|||||+ inherited access
|||||+ failed access
|||||+--success access
|||||+-- no propagate
|||||+--- inherit only
|||||+---- directory inherit
|||||+----- file inherit
|||||
|||||+ sync
|||||+ change owner
|||||+-- write ACL
|||||+--- read ACL
|||||+---- write extended attributes
|||||+----- read extended attributes
|||||+----- write attributes
|||||+----- read attributes
|||||+----- delete child
|||||+----- delete
|||||+----- append
|||||+----- execute
|||+----- write data
|+----- read data
```

`-w cols`

`--width cols`

Multi-column output where the column width is forced to *cols*.

`-x`

Multi-column output with entries sorted across rather than down the page.

`-1`

Prints one entry per line of output.

-@

The same as -l, except that extended attribute information overrides ACL information. An @ is displayed after the file permission bits for files that have extended attributes.

-/

The -/ option supports two option arguments c (compact mode) and v (verbose mode). Displays the long listing, same as -l. In addition, displays the extended system attributes associated with the file when extended system attributes are fully supported by the underlying file system.

appendonly

Allows a file to be modified only at offset EOF. Attempts to modify a file at a location other than EOF fails with EPERM.

archive

Indicates if a file has been modified since it was last backed up. Whenever the modification time (mtime) of a file is changed the archive attribute is set.

av_modified

ZFS sets the anti-virus attribute which whenever a file's content or size changes or when the file is renamed.

av_quarantined

Anti-virus software sets to mark a file as quarantined.

crttime

Timestamp when a file is created.

hidden

Marks a file as hidden.

immutable

Prevents the content of a file from being modified. Also prevents all metadata changes, except for access time updates. When placed on a directory, prevents the deletion and creation of files in the directories. Attempts to modify the content of a file or directory marked as immutable fail with EPERM. Attempts to modify any attributes (with the exception of access time and, with the proper privileges, the immutable) of a file marked as immutable fails with EPERM.

nodump

Solaris systems have no special semantics for this attribute.

nounlink

Prevents a file from being deleted. On a directory, the attribute also prevents any changes to the contents of the directory. That is, no files within the directory can be removed or renamed. The errno EPERM is returned when attempting to unlink or rename files and directories that are marked as nounlink.

readonly

Marks a file as **readonly**. Once a file is marked as **readonly** the content data of the file cannot be modified. Other metadata for the file can still be modified.

sparse

This attribute is available to users and applications to indicate that a file can be interpreted as **sparse**. It does not indicate whether or not the file is actually **sparse** and it has no special semantics on the Solaris operating system. The **sparse** attribute will be cleared if the file is truncated to zero length.

system

Solaris systems have no special semantics for this attribute.

The display characters used in compact mode (**-/ c**) are as follows:

Attribute Name	Display
archive	A
hidden	H
readonly	R
system	S
appendonly	a
nodump	d
immutable	i
av_modified	m
av_quarantined	q
sparse	s
nounlink	u

The display in verbose mode (**/ v**) uses full attribute names when it is set and the name prefixed by 'no' when it is not set.

The attribute name **ctime** and all other timestamps are handled by the option **-%** with the respective timestamp option arguments and also with **all** option argument. The display positions are as follows: The display in verbose mode (**-/ v**) uses full attribute names when it is set and the name prefixed by **no** when it is not set. The attribute name **ctime** and all other timestamps are handled by the option **-%** with the respective timestamp option arguments and also with **all** option argument.

The display positions are as follows:

```
{|}|}|}|}|}|}|}|
|}|}|}|}|}|}|}|+ - s (sparse)
|}|}|}|}|}|}|}|+ - - 0 (offline)
|}|}|}|}|}|}|}|+ - - - u (nounlink)
|}|}|}|}|}|}|}|+ - - - - q (av_quarantined)
|}|}|}|}|}|}|}|+ - - - - - m (av_modified)
|}|}|}|}|}|}|}|+ - - - - - - i (immutable)
|}|}|}|}|}|}|}|+ - - - - - - - d (nodump)
|}|}|}|}|}|}|}|+ - - - - - - - - a (appendonly)
```

```

|||+----- S (system)
||+----- R (readonly)
|+----- H (hidden)
+----- A (archive)

-% atime | ctime | mtime | all

```

atime

Equivalent to `-u`.

ctime

Uses the creation time of the file for sorting or printing.

ctime

Equivalent to `-c`.

mtime

Uses the last modification time of the file contents for sorting or printing.

If extended system attributes are not supported or if the user does not have read permission on the file or if the `ctime` extended attribute is not set, `ctime` is treated as a synonym for `mtime`.

When option argument `-all` is specified, all available timestamps are printed which includes `-atime`, `-ctime`, `-mtime` and on the extended system attribute supporting file systems, `-ctime` (create time). The option `-%all` does not effect which timestamp is displayed in long format and does not affect sorting.

--block-size *size*

Display sizes in multiples of *size*. Size can be scaled by suffixing one of `YyZzEePpTtGgMmKk`. Additionally, a `B` can be placed at the end to indicate powers of 10 instead of 2. For example, `.10mB` means blocks of 1000000 bytes while `10m` means blocks of $10 \cdot 2^{20}$ -- 10485760 -- bytes. This is mutually exclusive with the `-h` option.

--color [=when]

--colour [=when]

Display filenames using color on color-capable terminals. *when* is an optional argument that determines when to display color output.

Possible values for *when* are:

`always`

`yes`

`force`

Always use color.

`auto`

`tty`

`if-tty`

Use color if a terminal is present.

no
never
none
Never use color. This is the default

See the `Color Output` section of this manual page for information on how to control the output colors.

`--file-type`

Display a suffix after a file depending on its type, similar to the `-F` option, except `*` is not appended to executable files.

`--si`

Display human scaled sizes similar to the `-h` option, except values are repeatedly divided by 1000 instead of 1024. The last option `--si` or `-h` determines the divisor used.

`--time-style style`

Display times using the specified style. This does not effect the times displayed for extended attributes (`-%`).

Possible values for *style* are:

`full-iso`

Equivalent to `-E`.

`long-iso`

Display in `YYYY-MM-DD HH:MM` for all files.

`iso`

Display older files using `YYYY-MM-DD` and newer files with `MM-DD HH:MM`.

`locale`

Use the default locale format for old and new files. This is the default.

`+FORMAT`

Use a custom format. Values are the same as described in `strftime(3C)`. If a `NEWLINE` appears in the string, the first line is used for older files and the second line is used for newer files. Otherwise, the given format is used for all files.

`/usr/bin/ls`

`-F`

Marks directories with a trailing slash (`/`), doors with a trailing greater-than sign (`>`), executable files with a trailing asterisk (`*`), FIFOs with a trailing vertical bar (`|`), symbolic links with a trailing “at” sign (`@`), and `AF_UNIX` address family sockets with a trailing equals sign (`=`). Follows `symlinks` named as operands.

`--file-type`

Marks entries as with `-F` with the exception of executable files. Executable files are not marked. Follows `symlinks` named as operands.

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: `-C` and `-l` (ell), `-m` and `-l` (ell), `-x` and `-l` (ell), `-@` and `-l` (ell). The `-l` option overrides the other option specified in each pair.

Specifying more than one of the options in the following mutually exclusive groups is not considered an error: `-C` and `-l` (one), `-H` and `-L`, `-c` and `-u`, and `-e` and `-E`, and `-t` and `-S`. The last option specifying a specific timestamp (`-c`, `-u`, `-% atime`, `-% ctime`, `-% mtime`, and `-% mtime`) determines the timestamps used for sorting or in long format listings. The last option `-t`, `-S`, or `-U` determines the sorting behavior.

/usr/xpg4/bin/ls

`-F`

Marks directories with a trailing slash (/), doors with a trailing greater-than sign (>), executable files with a trailing asterisk (*), FIFOs with a trailing vertical bar (|), symbolic links with a trailing “at” sign (@), and AF_UNIX address family sockets with a trailing equals sign (=). Follows symlinks named as operands.

`--file-type`

Marks entries as with `-F` with the exception of executable files. Executable files are not marked. Follows symlinks named as operands.

Specifying more than one of the options in the following groups of mutually exclusive options is not considered an error: `-C` and `-l` (ell), `-m` and `-l` (ell), `-x` and `-l` (ell), `-@` and `-l` (ell), `-C` and `-l` (one), `-H` and `-L`, `-c` and `-u`, `-e` and `-E`, `-t` and `-S` and `-U`. The last option specifying a specific timestamp (`-c`, `-u`, `-% atime`, `-% ctime`, `-% mtime`, and `-% mtime`) determines the timestamps used for sorting or in long format listings. The last `-t`, `-S`, or `-U` option determines the sorting behavior.

/usr/xpg6/bin/ls

`-F`

Marks directories with a trailing slash (/), doors with a trailing greater-than sign (>), executable files with a trailing asterisk (*), FIFOs with a trailing vertical bar (|), symbolic links with a trailing “at” sign (@), and AF_UNIX address family sockets with a trailing equals sign (=). Does not follow symlinks named as operands unless the `-H` or `-L` option is specified.

`--file-type`

Marks entries as with `-F` with the exception of executable files. Executable files are not marked. Does not follow symlinks named as operands unless the `-H` or `-L` option is specified.

Specifying more than one of the options in the following mutually exclusive pairs is not considered an error: `-C` and `-l` (ell), `m` and `-l` (ell), `-x` and `-l` (ell), `-@` and `-l` (ell), `-C` and `-l` (one), `-H` and `--L`, `-c` and `-u`, `-e` and `-E`, `-t` and `-S` and `-U`. The last option specifying a specific timestamp (`-c`, `-u`, `-% atime`, `-% ctime`, `-% mtime`, and `-% mtime`) determines the timestamps used for sorting or in long format listings. The last `-t`, `-S`, or `-U` option determines the sorting behavior.

操作数

The following operand is supported:

file

A path name of a file to be written. If the file specified is not found, a diagnostic message is output on standard error.

Color Output

If color output is enabled, the environment variable `LS_COLORS` is checked. If it exists, its contents are used to control the colors used to display filenames. If it is not set, a default list of colors is used. The format of `LS_COLORS` is a colon separated list of attribute specifications. Each attribute specification is of the format

filespec=attr[;attr...]

filespec is either of the form **.SUFFIX*, for example, **.jar* or **.Z*, or one of the following file types:

no	Normal file
fi	Regular file
di	Directory
ln	Symbolic link
pi	FIFO or named pipe
so	Socket
do	Door file
bd	Block device
cd	Character device
ex	Execute bit (either user, group, or other) set
po	Event port
st	Sticky bit set
or	Orphaned symlink
sg	setgid binary
su	setuid binary
ow	world writable
tw	Sticky bit and world writable

attr is a semicolon delimited list of color and display attributes which are combined to determine the final output color. Any combination of *attr* values can be specified. Possible *attr* values are:

00	All attributes off (default terminal color)
----	---

01	Display text in bold
04	Display text with an underscore
05	Display text in bold
07	Display text with foreground and background colors reversed
08	Display using concealed text.

One of the following values can be chosen. If multiple values are specified, the last specified value is used.

30	Set foreground to black.
31	Set foreground to red.
32	Set foreground to green.
33	Set foreground to yellow.
34	Set foreground to blue.
35	Set foreground to magenta (purple).
36	Set foreground to cyan.
37	Set foreground to white.
39	Set foreground to default terminal color.

One of the following can be specified. If multiple values are specified, the last value specified is used.

40	Set foreground to black.
41	Set foreground to red.
42	Set foreground to green.
43	Set foreground to yellow.
44	Set foreground to blue.
45	Set foreground to magenta (purple).
46	Set foreground to cyan.
47	Set foreground to white.
49	Set foreground to default terminal color.

On some terminals, setting the bold attribute causes the foreground colors to be high-intensity, that is, brighter. In such cases the low-intensity yellow is often displayed as a brown or orange color.

At least one attribute must be listed for a file specification.

The appropriate color codes are chosen by selecting the most specific match, starting with the file suffixes and proceeding with the file types until a match is found. The no (normal file) type matches any file.

用法

See [largefile\(5\)](#) for the description of the behavior of `ls` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Viewing File Permissions

The following example shows how to display detailed information about a file.

```
% ls -l file.1
-rw-r--r--  1 gozer    staff    206663 Mar 14 10:15 file.1
```

The permissions string above (`-rw-r--r--`) describes that the file owner has read and write permissions, the owning group has read permissions, and others have read permissions.

The following example shows how to display detailed information about a directory.

```
% ls -ld test.dir
drwxr-xr-x  2 gozer    staff          2 Mar 14 10:17 test.dir
```

The permissions string above (`drwxr-xr-x`) describes that the directory owner has read, write, and search permissions, the owning group has read and search permissions, and others have read and search permissions.

Another example of listing file permissions is as follows:

```
% ls -l file.2
-rw-rwl---  1 gozer    staff    206663 Mar 14 10:47 file.2
```

The permissions string above (`-rw-rwl---`) describes that the file owner has read and write permissions, the owning group has read and write permissions, and the file can be locked during access.

示例 2 Displaying ACL Information on Files and Directories

The following example shows how to display verbose ACL information on a ZFS file.

```
% ls -v file.1
-rw-r--r--  1 marks    staff    206663 Mar 14 10:15 file.1
  0:owner@:execute:deny
  1:owner@:read_data/write_data/append_data/write_xattr/write_attributes
    /write_acl/write_owner:allow
```


示例 2 Displaying ACL Information on Files and Directories (续)

```
2:group@:write_data/append_data/execute:deny
3:group@:read_data:allow
4:everyone@:write_data/append_data/write_xattr/execute/write_attributes
  /write_acl/write_owner:deny
5:everyone@:read_data/read_xattr/read_attributes/read_acl/synchronize
  :allow
```

The following example shows how to display compact ACL information on a ZFS directory.

```
% ls -dV test.dir
drwxr-xr-x  2 marks  staff          2 Mar 14 10:17 test.dir
      owner@:-----:-----:deny
      owner@:rwxp---A-W-Co:-----:allow
      group@:-w-p-----:-----:deny
      group@:r-x-----:-----:allow
      everyone@:-w-p---A-W-Co:-----:deny
      everyone@:r-x---a-R-c--s:-----:allow
```

The following example illustrates the `ls -v` behavior when listing ACL information on a UFS file.

```
$ ls -v file.3
-rw-r--r--  1 root  root          2703 Mar 14 10:59 file.3
      0:user::rw-
      1:group::r--          #effective:r--
      2:mask:r--
      3:other:r--
```

示例 3 Printing the Names of All Files

The following example prints the names of all files in the current directory, including those that begin with a dot (.), which normally do not print:

```
example% ls -a
```

示例 4 Providing File Information

The following example provides file information:

```
example% ls -aisn
```

This command provides information on all files, including those that begin with a dot (a), the *i*-number, the memory address of the *i*-node associated with the file—printed in the left-hand column (*i*); the size (in blocks) of the files, printed in the column to the right of the *i*-numbers (*s*); finally, the report is displayed in the numeric version of the long list, printing the UID (instead of user name) and GID (instead of group name) numbers associated with the files.

示例 4 Providing File Information (续)

When the sizes of the files in a directory are listed, a total count of blocks, including indirect blocks, is printed.

示例 5 Providing Extended System Attributes Information

```
example% ls -/ c file (extended system attribute in compact mode)
-rw-r--r-- 1 root root 0 May 10 14:17 file
           {AHSadim-u}
```

In this example, `av_quarantined` is not set.

```
example% ls -/ v file (extended system attribute in verbose mode)
-rw-r--r-- 1 root root 0 May 10 14:17 file
           {archive,hidden,readonly,system,appendonly\
           nodump,immutable,av_modified,\
           noav_quarantined,nounlink}
```

```
example% ls -/ v file (no extended system attribute)
-rw-r--r-- 1 root staff 0 May 16 14:48 file
           {}
```

```
example% ls -/ c file (extended system attribute
                       supported file system)
-rw-r--r-- 1 root staff 3 Jun 4 22:04 file
           {A-----m--}
```

`archive` and `av_modified` attributes are set by default on an extended system attribute supported file.

```
example% ls -/ c -%ctime file
-rw-r--r-- root root 0 May 10 14:17 file
           {AHSadim-u}
```

This example displays the timestamp as the creation time:

```
example% ls -l -%all file
-rw-r--r-- 1 root root 0 May 10 14:17 file
           timestamp: atime Jun 14 08:47:37 2007
           timestamp: ctime May 10 14:20:23 2007
           timestamp: mtime May 10 14:17:56 2007
           timestamp: ctime May 10 14:17:56 2007
```

```
example% ls -%ctime -tl file*
```

```

-rw-r--r--  1 foo      staff      3 Jun  4 22:04 file1
-rw-r--r--  1 root     root       0 May 10 14:17 file
-rw-r--r--  1 foo      staff      0 May  9 13:49 file.1

```

In this example the files are sorted by creation time.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `ls`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_TIME`, `LC_MESSAGES`, `NLS_PATH`, and `TZ`.

COLUMNS

Determines the user's preferred column position width for writing multiple text-column output. If this variable contains a string representing a decimal integer, the `ls` utility calculates how many path name text columns to write (see `-C`) based on the width provided. If `COLUMNS` is not set or is invalid, 80 is used. The column width chosen to write the names of files in any given directory is constant. File names are not be truncated to fit into the multiple text-column output.

LS_COLORS

Determines the coloring scheme used when displaying color output. If not set and color output is specified, a default scheme is used. If `TERM` is not set, no color output is used.

TERM

Determine the terminal type. If this variable is unset or `NULL`, no color output is generated regardless of the value of the `--color` option.

退出状态

0 All information was written successfully.

>0 An error occurred.

文件

`/etc/group` group IDs for `ls -l` and `ls -g`

`/etc/passwd` user IDs for `ls -l` and `ls -o`

`/usr/share/lib/terminfo/??/*` terminal information database

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/ls`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See below.

For all options except `-A`, `-b`, `-e`, `-E`, `-h`, `-S`, `U -v`, `-V`, `-@`, `-/`, `-%`, `--all`, `--almost-all`, `--block-size`, `--classify`, `--color`, `--colour`, `--dereference`, `--dereference-command-line`, `--escape`, `--file-type`, `--full-time`, `--human-readable`, `--ignore-backups`, `--inode`, `--no-group`, `--numeric-uid-gid`, `--reverse`, `--recursive`, `--si`, `--size`, and `--time-style`, see [standards\(5\)](#).

/usr/xpg4/bin/ls

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See below.

For all options except `-A`, `-b`, `-e`, `-E`, `-h`, `-S`, `U -v`, `-V`, `-@`, `-/`, `-%`, `--all`, `--almost-all`, `--block-size`, `--classify`, `--color`, `--colour`, `--dereference`, `--dereference-command-line`, `--escape`, `--file-type`, `--full-time`, `--human-readable`, `--ignore-backups`, `--inode`, `--no-group`, `--numeric-uid-gid`, `--reverse`, `--recursive`, `--si`, `--size`, and `--time-style`, see [standards\(5\)](#).

/usr/xpg6/bin/ls

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu6
CSI	Enabled
Interface Stability	Committed
Standard	See below.

For all options except `-A`, `-b`, `-e`, `-E`, `-h`, `-S`, `U -v`, `-V`, `-@`, `-/`, `-%`, `--all`, `--almost-all`, `--block-size`, `--classify`, `--color`, `--colour`, `--dereference`, `--dereference-command-line`, `--escape`, `--file-type`, `--full-time`, `--human-readable`, `--ignore-backups`, `--inode`, `--no-group`, `--numeric-uid-gid`, `--reverse`, `--recursive`, `--si`, `--size`, and `--time-style`, see [standards\(5\)](#).

另请参见

[chmod\(1\)](#), [cp\(1\)](#), [setfacl\(1\)](#), [fgetattr\(3C\)](#), [strftime\(3C\)](#), [terminfo\(4\)](#), [acl\(5\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注

Unprintable characters in file names can confuse the columnar output options.

The total block count is incorrect if there are hard links among the files.

The sort order of `ls` output is affected by the locale and can be overridden by the `LC_COLLATE` environment variable. For example, if `LC_COLLATE` equals `C`, dot files appear first, followed by

names beginning with upper-case letters, then followed by names beginning with lower-case letters. But if LC_COLLATE equals en_US.ISO8859-1, then leading dots as well as case are ignored in determining the sort order.

引用名	ls – list the contents of a directory
用法概要	<code>/usr/ucb/ls [-aAcCdfFgiLLqRstu1] file...</code>
描述	<p>For each <i>filename</i> that is a directory, <code>ls</code> lists the contents of the directory; for each <i>filename</i> that is a file, <code>ls</code> repeats its name and any other information requested. By default, the output is sorted alphabetically. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments are processed before directories and their contents.</p>
Permissions Field	<p>The mode printed under the <code>-l</code> option contains 10 characters interpreted as follows. If the first character is:</p> <ul style="list-style-type: none">d Entry is a directory.D Entry is a door.b Entry is a block-type special file.c Entry is a character-type special file.l Entry is a symbolic link.p Entry is a FIFO (also known as “named pipe”) special file.s Entry is an AF_UNIX address family socket.– Entry is a plain file. <p>The next 9 characters are interpreted as three sets of three bits each. The first set refers to owner permissions; the next refers to permissions to others in the same user-group; and the last refers to all others. Within each set, the three characters indicate permission respectively to read, to write, or to execute the file as a program. For a directory, “execute” permission is interpreted to mean permission to search the directory. The permissions are indicated as follows:</p> <ul style="list-style-type: none">r The file is readable.w The file is writable.x The file is executable.– The indicated permission is not granted. <p>The group-execute permission character is given as <code>s</code> if the file has the set-group-id bit set; likewise the owner-execute permission character is given as <code>s</code> if the file has the set-user-id bit set.</p> <p>The last character of the mode (normally <code>x</code> or <code>-</code>) is <code>t</code> <code>rue</code> if the 1000 bit of the mode is on. See chmod(1) for the meaning of this mode. The indications of set-ID and 1000 bits of the mode are capitalized (<code>S</code> and <code>T</code>, respectively) if the corresponding execute permission is <i>not</i> set.</p>

A plus sign (+) appended to the list of permissions indicates that an ACL is associated with the file.

When the sizes of the files in a directory are listed, a total count of blocks, including indirect blocks, is printed.

选项

The following options are supported:

- a Lists all entries; in the absence of this option, entries whose names begin with a '.' are *not* listed (except for the privileged user, for whom `ls` normally prints even files that begin with a '.').
- A Same as -a, except that '.' and './' are not listed.
- c Uses time of last edit (or last mode change) for sorting or printing.
- C Forces multi-column output, with entries sorted down the columns; for `ls`, this is the default when output is to a terminal.
- d If argument is a directory, lists only its name (not its contents); often used with `-l` to get the status of a directory.
- f Forces each argument to be interpreted as a directory and lists the name found in each slot. This option turns off `-l`, `-t`, `-s`, and `-r`, and turns on `-a`; the order is the order in which entries appear in the directory.
- F Marks directories with a trailing slash (/), doors with a trailing greater-than sign (>), executable files with a trailing asterisk (*), FIFOs with a trailing vertical bar (|), symbolic links with a trailing at-sign (@), and AF_UNIX address family sockets with a trailing equals sign (=).
- g For `ls`, shows the group ownership of the file in a long output.
- i For each file, prints the i-node number in the first column of the report.
- l Lists in long format, giving mode, ACL indication, number of links, owner, size in bytes, and time of last modification for each file. If the file is a special file the size field will instead contain the major and minor device numbers. If the time of last modification is greater than six months ago, it is shown in the format '*month date year*'; files modified within six months show '*month date time*'. If the file is a symbolic link, the pathname of the linked-to file is printed preceded by '->'.
 If argument is a symbolic link, lists the file or directory the link references rather than the link itself.
- L If argument is a symbolic link, lists the file or directory the link references rather than the link itself.
- q Displays non-graphic characters in filenames as the character ?; for `ls`, this is the default when output is to a terminal.
- r Reverses the order of sort to get reverse alphabetic or oldest first as appropriate.
- R Recursively lists subdirectories encountered.

- s Indicate the total number of file system blocks consumed by each file displayed.
- t Sorts by time modified (latest first) instead of by name.
- u Uses time of last access instead of last modification for sorting (with the -t option) and/or printing (with the -l option).
- 1 Forces one entry per line output format; this is the default when output is not to a terminal.

操作数

The following operand is supported:

file A path name of a file to be listed. If the file specified is not found, a diagnostic message is output on standard error.

用法

See [largefile\(5\)](#) for the description of the behavior of `ls` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

文件

`/etc/group` to get group ID for '`ls -g`'

`/etc/passwd` to get user IDs for '`ls -l`' and '`ls -o`'

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见

[ls\(1\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

附注

NEWLINE and TAB are considered printing characters in filenames.

The output device is assumed to be 80 columns wide.

The option setting based on whether the output is a teletype is undesirable as '`ls -s`' is much different than '`ls -s | lpr`'. On the other hand, not doing this setting would make old shell scripts which used `ls` almost certain losers.

Unprintable characters in file names can confuse the columnar output options.

引用名	m4 – macro processor
用法概要	<pre>/usr/bin/m4 [-e] [-s] [-B int] [-H int] [-S int] [-T int] [-Dname [=val]] ... [-U name] ... [file]... /usr/xpg4/bin/m4 [-e] [-s] [-B int] [-H int] [-S int] [-T int] [-Dname [...=val]] [-U name] ... [file]...</pre>
描述	<p>The m4 utility is a macro processor intended as a front end for C, assembler, and other languages. Each of the argument files is processed in order. If there are no files, or if a file is <code>–</code>, the standard input is read. The processed text is written on the standard output. <i>Note:</i> m4 cannot include more than nine nested files and writes a diagnostic message if that number is exceeded.</p>
Macro Syntax	<p>Macro calls have the form:</p> <pre>name(arg1,arg2, ..., argn)</pre> <p>The open parenthesis character, (, must immediately follow the name of the macro. If the name of a defined macro is not followed by a (, it is deemed to be a call of that macro with no arguments. Potential macro names consist of alphanumeric characters and underscore (_), where the first character is not a digit.</p> <p>Leading unquoted blanks, TABs, and NEWLINES are ignored while collecting arguments. Left and right single quotes are used to quote strings. The value of a quoted string is the string stripped of the quotes.</p>
Macro Processing	<p>When a macro name is recognized, its arguments are collected by searching for a matching right parenthesis. If fewer arguments are supplied than are in the macro definition, the trailing arguments are taken to be NULL. Macro evaluation proceeds normally during the collection of the arguments, and any commas or right parentheses that happen to turn up within the value of a nested call are as effective as those in the original input text. After argument collection, the value of the macro is pushed back onto the input stream and rescanned.</p>
选项	<p>The options and their effects are as follows:</p> <ul style="list-style-type: none"> -Bint Changes the size of the push-back and argument collection buffers from the default of 4,096. Values of size less than or equal to zero are ignored and the default value is used. -e Operates interactively. Interrupts are ignored and the output is unbuffered. -Hint Changes the size of the symbol table hash array from the default of 199. For better performance, the size should be prime. Values of size less than or equal to zero are ignored and the default value is used. -s Enables line sync output for the C preprocessor (#line . . .) -Sint Changes the size of the call stack from the default of 100 slots. Macros take three slots, and non-macro arguments take one. Values of size less than or equal to zero are ignored and the default value is used.

`-Tint` Changes the size of the token buffer from the default of 512 bytes. Values of size less than or equal to zero are ignored and the default value is used.

To be effective, the above flags must appear before any file names and before any `-D` or `-U` flags:

`-D name[=val]` Defines *name* to *val* or to NULL in *val*'s absence.

`-Uname` Undefines *name*.

操作数

The following operand is supported:

file A path name of a text file to be processed. If no *file* is given, or if it is `-`, the standard input is read.

用法

The `m4` utility makes available the following built-in macros. These macros can be redefined, but once this is done the original meaning is lost. Their values are NULL unless otherwise stated.

`changequote` Change quote symbols to the first and second arguments. The symbols can be up to five characters long. `changequote` without arguments restores the original values (that is, `' '`).

`changecom` Change left and right comment markers from the default `#` and `NEWLINE`. With no arguments, the comment mechanism is effectively disabled. With one argument, the left marker becomes the argument and the right marker becomes `NEWLINE`. With two arguments, both markers are affected. Comment markers can be up to five characters long.

`decr` Returns the value of its argument decremented by 1.

`define` The second argument is installed as the value of the macro whose name is the first argument. Each occurrence of `$n` in the replacement text, where *n* is a digit, is replaced by the *n*-th argument. Argument 0 is the name of the macro; missing arguments are replaced by the null string; `$#` is replaced by the number of arguments; `$*` is replaced by a list of all the arguments separated by commas; `$@` is like `$*`, but each argument is quoted (with the current quotes).

`defn` Returns the quoted definition of its argument(s). It is useful for renaming macros, especially built-ins.

`divert` `m4` maintains 10 output streams, numbered 0-9. The final output is the concatenation of the streams in numerical order. Initially stream 0 is the current stream. The `divert` macro changes the current output stream to its (digit-string) argument. Output diverted to a stream other than 0 through 9 is discarded.

`divnum` Returns the value of the current output stream.

`dnl` Reads and discards characters up to and including the next `NEWLINE`.

<code>dumpdef</code>	Prints current names and definitions, for the named items, or for all if no arguments are given.
<code>errprint</code>	Prints its argument on the diagnostic output file.
<code>ifdef</code>	If the first argument is defined, the value is the second argument, otherwise the third. If there is no third argument, the value is NULL. The word <code>unix</code> is predefined.
<code>ifelse</code>	This macro has three or more arguments. If the first argument is the same string as the second, then the value is the third argument. If not, and if there are more than four arguments, the process is repeated with arguments 4, 5, 6 and 7. Otherwise, the value is either the fourth string, or, if it is not present, NULL.
<code>include</code>	Returns the contents of the file named in the argument.
<code>incr</code>	Returns the value of its argument incremented by 1. The value of the argument is calculated by interpreting an initial digit-string as a decimal number.
<code>index</code>	Returns the position in its first argument where the second argument begins (zero origin), or <code>-1</code> if the second argument does not occur.
<code>len</code>	Returns the number of characters in its argument.
<code>m4exit</code>	This macro causes immediate exit from m4. Argument 1, if given, is the exit code; the default is <code>0</code> .
<code>m4wrap</code>	Argument 1 is pushed back at final EOF. Example: <code>m4wrap('cleanup()')</code>
<code>maketemp</code>	Fills in a string of "X" characters in its argument with the current process ID.
<code>popdef</code>	Removes current definition of its argument(s), exposing the previous one, if any.
<code>pushdef</code>	Like <code>define</code> , but saves any previous definition.
<code>shift</code>	Returns all but its first argument. The other arguments are quoted and pushed back with commas in between. The quoting nullifies the effect of the extra scan that is subsequently be performed.
<code>sinclude</code>	This macro is identical to <code>include</code> , except that it says nothing if the file is inaccessible.
<code>substr</code>	Returns a substring of its first argument. The second argument is a zero origin number selecting the first character; the third argument indicates the length of the substring. A missing third argument is taken to be large enough to extend to the end of the first string.

	<code>syscmd</code>	This macro executes the command given in the first argument. No value is returned.
	<code>sysval</code>	This macro is the return code from the last call to <code>syscmd</code> .
	<code>translit</code>	Transliterates the characters in its first argument from the set given by the second argument to the set given by the third. No abbreviations are permitted.
	<code>traceon</code>	This macro with no arguments, turns on tracing for all macros (including built-ins). Otherwise, turns on tracing for named macros.
	<code>traceoff</code>	Turns off trace globally and for any macros specified.
	<code>undefine</code>	Removes the definition of the macro named in its argument.
	<code>undivert</code>	This macro causes immediate output of text from diversions named as arguments, or all diversions if no argument. Text can be undiverted into another diversion. Undiverting discards the diverted text.
<code>/usr/bin/m4</code>	<code>eval</code>	Evaluates its argument as an arithmetic expression, using 32-bit signed-integer arithmetic. The following operators are supported: parentheses, unary -, unary +, !, ~, *, /, %, +, -, relationals, bitwise &, , &&, and . Octal and hex numbers can be specified as in C. The second argument specifies the radix for the result; the default is 10. The third argument can be used to specify the minimum number of digits in the result.
<code>/usr/xpg4/bin/m4</code>	<code>eval</code>	Evaluates its argument as an arithmetic expression, using 32-bit signed-integer arithmetic. The following operators are supported: parentheses, unary -, unary +, !, ~, *, /, %, +, -, <<, >>, relationals, bitwise &, , &&, and . Precedence and associativity are as in C. Octal and hex numbers can also be specified as in C. The second argument specifies the radix for the result; the default is 10. The third argument can be used to specify the minimum number of digits in the result.

示例

示例1 Examples of m4 files

If the file `m4src` contains the lines:

The value of 'VER' is "VER".

```
ifdef('VER', "VER' is defined to be VER., VER is not defined.)
ifelse(VER, 1, "VER'' is 'VER'.)
ifelse(VER, 2, "VER'' is 'VER'., "VER'' is not 2.)
end
```

then the command:

```
m4 m4src
```

or the command:

```
m4 -U VER m4src
```

示例1 Examples of m4 files (续)

produces the output:

```
The value of VER is "VER".
    VER is not defined.

    VER is not 2.
end
```

The command:

```
m4 -D VER m4src
```

produces the output:

```
The value of VER is "".
    VER is defined to be .

    VER is not 2.
end
```

The command:

```
m4 -D VER=1 m4src
```

produces the output:

```
The value of VER is "1".
    VER is defined to be 1.
    VER is 1.
    VER is not 2.
end
```

The command:

```
m4 -D VER=2 m4src
```

produces the output:

```
The value of VER is "2".
    VER is defined to be 2.

    VER is 2.
end
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of m4: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred

If the `m4exit` macro is used, the exit value can be specified by the input file.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/m4

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

/usr/xpg4/bin/m4

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[as\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名	mac – 计算输入的消息验证代码
用法概要	<pre> /usr/bin/mac -l /usr/bin/mac [-v] -a algorithm [-k keyfile -K key_label [-T token_spec]] [file]...</pre>
描述	<p>mac 实用程序可使用指定算法计算给定文件或 stdin 的消息验证代码 (message authentication code, MAC)。</p> <p>如果给定了多个文件，每个输出行就是单个文件的 MAC。</p>
选项	<p>支持以下选项：</p> <p>-a algorithm 指定加密或解密过程中要使用的算法的名称。有关详细信息，请参见“用法”部分的“算法”。注意：不支持用于生成一般长度 MAC 的算法。</p> <p>-k keyfile 指定包含用于加密算法的密钥值的文件。每种算法都具有特定的密钥材料要求，如 PKCS#11 规范中所述。如果未指定 -k，mac 会使用 <code>getpassphrase(3C)</code> 提示提供密钥材料。</p> <p>有关生成密钥文件的信息，请参见 <code>pktool(1)</code>、<code>dd(1M)</code> 或《Oracle Solaris 11.1 管理：安全服务》。</p> <p>-K key_label 指定 PKCS#11 令牌中的对称令牌密钥的标签。</p> <p>-l 显示系统上可用的算法列表。此列表可依加密框架的配置而变化。以位为单位显示密钥大小。</p> <p>-T token_spec 指定 PKCS#11 令牌，而不使用指定 -K 时的缺省软令牌对象存储。</p> <p><i>token_spec</i> 的格式为：</p> <pre>token_name [:manuf_id [:serial_no]]</pre> <p>当令牌标签包含结尾空格时，为方便起见，此选项不要求用户键入这些空格。</p> <p>使用冒号分隔令牌标识字符串。如果任一部分中包含冒号 (:) 文本字符，必须使用反斜杠 (\) 对其进行转义。如果未找到冒号 (:)，则将整个字符串（最多 32 个字符）视为令牌标签。如果仅找到一个冒号 (:)，则该字符串是令牌标签和生产商。</p> <p>-v 提供详细信息。</p>
用法	
算法	<p>可使用 -l 选项显示受支持的算法。这些算法由加密框架提供。每个受支持的算法都是特定算法类型的最常用和受限最小版本的别名。例如，<code>md5_hmac</code> 是 <code>CKM_MD5_HMAC</code> 的别名。</p>

这些别名与 `-a` 选项一起使用，并且区分大小写。

口令短语

若在加密和解密任务期间未使用 `-k` 选项，则会提示用户提供口令短语。可使用 PKCS #5 中指定的 PBKDF2 算法将口令短语处理为更安全的密钥。

示例

示例 1 列出可用算法

以下示例列出了可用的算法：

```
example$ mac -l
Algorithm      Keysize:  Min   Max
-----
des_mac                64    64
sha1_hmac             8    512
md5_hmac              8    512
sha224_hmac           8    512
sha256_hmac           8    512
sha384_hmac           8   1024
sha512_hmac           8   1024
```

示例 2 获取消息验证代码

以下示例获取文件的消息验证代码：

```
example$ mac -v -k mykey -a sha1_hmac /export/foo
sha1_hmac (/export/foo) = 913ced311df10f1708d9848641ca8992f4718057
```

示例 3 使用令牌密钥获取消息验证代码

以下示例使用软令牌 `keystore` 中的通用令牌密钥获取消息验证代码。可使用 [pktool\(1\)](#) 生成通用令牌密钥：

```
encrypt -v -a sha1_hmac -K my_generic_key \
-T "Sun Software PKCS#11 softtoken" /export/foo
Enter pin for Sun Software PKCS#11 softtoken:
sha1_hmac (/etc/foo) = c2ba5c38458c092a68940081240d22b670182968
```

退出状态

将返回以下退出值：

- 0 成功完成。
- >0 出现错误。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed (已确定)

另请参见

`digest(1)`、`pktool(1)`、`dd(1M)`、`getpassphrase(3C)`、`libpkcs11(3LIB)`、`attributes(5)`、`pkcs11(5)`

《Oracle Solaris 11.1 管理：安全服务》

RSA PKCS#11 v2.20 与 RSA PKCS#5 v2.0. <http://www.rsasecurity.com>

- 引用名** mach – 显示当前主机的处理器类型
- 用法概要** mach
- 描述** mach 命令可显示当前主机的处理器类型。
- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [arch\(1\)](#)、[uname\(1\)](#)、[attributes\(5\)](#)

附注 mach 和 `uname -p` 返回等效值；因此鼓励独立软件供应商 (Independent Software Vendor, ISV) 和其他需要确定处理器类型的人将 `uname` 与 `-p` 选项配合使用，而不使用 mach 命令。提供 mach 命令是为了与之前的发行版兼容，但一般不鼓励使用它。

引用名 machid, sun, i386, i486, sparc – 获取处理器类型实际值

用法概要

```
sun
i386
sparc
```

描述 如果您使用命令名称指示的指令集，以下命令将返回 `true` 值（退出代码 0）。

`sun` 如果您处于 Sun 系统上，则结果为 `true`。

`i386` 如果您处于使用 iAPX386 处理器的计算机上，则结果为 `true`。

`sparc` 如果您处于使用 SPARC 系列处理器的计算机上，则结果为 `true`。

不适用的命令将返回 `false`（非零）值。这些命令常在 `makefile`（请参见 [make\(1S\)](#)）和 `shell` 脚本（请参见 [sh\(1\)](#)）内使用以增加可移植性。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [make\(1S\)](#)、[sh\(1\)](#)、[test\(1\)](#)、[true\(1\)](#)、[uname\(1\)](#)、[attributes\(5\)](#)

附注 命令的 `machid` 系列已过时。请使用 `uname -p` 和 `uname -m` 作为代替。

引用名	madv.so.1 – madv 库
用法概要	/usr/lib/madv.so.1
描述	<p>madv.so.1 共享目标文件为已启动的进程及其子孙进程提供一种选择性配置 VM 建议的方法。要启用 <code>madv.so.1</code>，环境中需要存在以下字符串（请参见 <code>ld.so.1(1)</code>）以及一个或多个 MADV 环境变量：</p> <p>LD_PRELOAD=\$LD_PRELOAD:madv.so.1</p>
环境变量	<p>如果在 LD_PRELOAD 中指定了 <code>madv.so.1</code> 共享目标文件，<code>madv</code> 共享目标文件会读取以下环境变量，以确定哪些已创建进程应用指定建议。</p> <p>MADV=advice MADV 指定用于进程地址空间中所有堆、共享内存以及 <code>mmap</code> 区域的 VM 建议。此建议将应用于所有已创建进程。</p> <p><i>advice</i> 的值对应于 <code>madvise(3C)</code> 中使用的 <code><sys/mman.h></code> 中的值，以指定内存访问模式：</p> <p>normal random sequential access_lwp access_many access_many_pset access_default</p> <p>MADVCFGFILE=config-file <i>config-file</i> 是包含一个或多个以下形式的 <code>madv</code> 配置项的文本文件。</p> <p><i>exec-name exec-args:advice-opts</i></p> <p><i>config-file</i> 中指定的建议优先于 MADV 环境变量所指定的建议。没有设置 MADVCFGFILE 时，会从文件 <code>/etc/madv.conf</code> 中提取建议（如果存在）。</p> <p><i>exec-name</i> 指定应用程序或可执行文件的名称。对应的建议是为与文件中找到的第一个 <i>exec-name</i> 相匹配的新建进程（请参见 <code>getexecname(3C)</code>）而设置的。</p> <p><i>exec-name</i> 可能为全路径名、基名或模式字符串。有关模式匹配的讨论，请参见 <code>sh(1)</code> 中的“生成文件名”。</p> <p><i>exec-args</i> 是与参数相匹配的选择性指定模式字符串。仅当未指定 <i>exec-args</i> 或其出现在 <i>exec-name</i> 的参数内时，才会设置建议。</p>

advice-opts 是逗号分隔的列表，为不同内存区域指定建议：

madv=advice 应用于进程地址空间中的所有堆、共享内存和 *mmap* 区域。

heap=advice 堆被定义为 *brk* 区域（请参见 [brk\(2\)](#)）。应用于现有堆以及将来分配的任何其他堆内存。

shm=advice

ism=advice

dism=advice

分别使用任何标志、标志 *SHM_SHARE_MMU* 或标志 *SHM_PAGEABLE* 附加的共享内存区段（请参见 [shmat\(2\)](#)）。选项 *ism* 和 *dism* 优先于选项 *shm*。

map=advice

mapshared=advice

mapprivate=advice

mapanon=advice

分别使用任何标志、标志 *MAP_SHARED*、标志 *MAP_PRIVATE* 或标志 *MAP_ANON* 通过 [mmap\(2\)](#) 建立的映射。选项 *mapshared*、*mapprivate* 和 *mapanon* 优先于选项 *map*。选项 *mapanon* 优先于 *mapshared* 和 *mapprivate*。

MADVERRFILE=pathname 缺省情况下，使用级别 *LOG_ERR* 和工具 *LOG_USER* 通过 [syslog\(3C\)](#) 记录错误消息。如果 *MADVERRFILE* 包含有效的路径名（例如 */dev/stderr*），将在其中记录错误消息。

示例

示例 1 将建议应用于所有 ISM 段

以下配置将建议应用于应用程序 */usr/bin/foo* 的所有 ISM 段：

```
example$ LD_PRELOAD=$LD_PRELOAD:madv.so.1
```

```
example$ MADVCFGFILE=madvcfg
```

```
example$ export LD_PRELOAD MADVCFGFILE
```

```
example$ cat $MADVCFGFILE
/usr/bin/foo:ism=access_lwp
```

示例 2 为所有应用程序设置建议且具有例外

以下配置为所有应用程序设置建议，除 *ls* 外。

```
example$ LD_PRELOAD=$LD_PRELOAD:madv.so.1
```

```
example$ MADV=access_many
```

```
example$ MADVCFGFILE=madvcfg
```

示例2 为所有应用程序设置建议且具有例外 (续)

```
example$ export LD_PRELOAD MADV MADVCFGFILE
example$ cat $MADVCFGFILE
ls:
```

示例3 优先级规则 (从“示例2”继续)

由于 MADVCFGFILE 优先于 MADV，为最后一个 madv 配置项的 *exec-name* 指定 '*' (模式全匹配) 相当于设置 MADV。以下等效于示例2：

```
example$ LD_PRELOAD=$LD_PRELOAD:madv.so.1
example$ MADVCFGFILE=madvcfg
example$ export LD_PRELOAD MADVCFGFILE
example$ cat $MADVCFGFILE
ls:
*:madv=access_many
```

示例4 为不同区域应用建议

以下配置为 mmap 区域应用一种类型的建议，并为具有执行名称 (以 foo 开头) 的一组选定应用程序的堆与共享内存区域应用不同建议。

```
example$ LD_PRELOAD=$LD_PRELOAD:madv.so.1
example$ MADVCFGFILE=madvcfg
example$ export LD_PRELOAD MADVCFGFILE
example$ cat $MADVCFGFILE
foo*:madv=access_many,heap=sequential,shm=access_lwp
```

示例5 选择性地应用建议

以下配置为具有 ora1 作为参数并以 ora 开始的应用程序堆应用建议。

```
example$ LD_PRELOAD=$LD_PRELOAD:madv.so.1
example$ MADVCFGFILE=madvcfg
example$ export LD_PRELOAD MADVCFGFILE
example$ cat $MADVCFGFILE
ora* ora1:heap=access_many
```

文件 /etc/madv.conf 配置文件

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/extended-system-utilities
接口稳定性	Uncommitted (未确定)

另请参见

[cat\(1\)](#)、[ld.so.1\(1\)](#)、[proc\(1\)](#)、[sh\(1\)](#)、[brk\(2\)](#)、[exec\(2\)](#)、[fork\(2\)](#)、[mmap\(2\)](#)、[mencntl\(2\)](#)、[shmat](#)

附注

建议将被继承。子进程具有与其父进程相同的建议。有关 `exec()`（请参见 [exec\(2\)](#)），建议会重新设置为缺省系统建议，除非通过 `madv` 共享目标文件配置了不同建议。

建议只应用于由用户程序显式创建的 `mmap` 区域。由运行时链接程序或由进行直接系统调用（例如用于线程栈的 `libthread` 分配）的系统库建立的那些区域不会受到影响。

引用名 mail, rmail – read mail or send mail to users

用法概要

Sending Mail mail [-tw] [-m *message_type*] *recipient*...

rmail [-tw] [-m *message_type*] *recipient*...

Reading Mail mail [-ehpPqr] [-f *file*]

Debugging mail [-x *debug_level*] [*other_mail_options*] *recipient*...

描述

A *recipient* is usually a domain style address (“*user@machine*”) or a user name recognized by [login\(1\)](#). When *recipients* are named, mail assumes a message is being sent. It reads from the standard input up to an end-of-file (Control-d) or, if reading from a terminal device, until it reads a line consisting of just a period. When either of those indicators is received, mail adds the *letter* to the *mailfile* for each *recipient*.

A *letter* is composed of some *header lines* followed by a blank line followed by the *message content*. The *header lines* section of the letter consists of one or more UNIX postmarks:

From *sender date_and_time* [remote from *remote_system_name*]

followed by one or more standardized message header lines of the form:

keyword-name: [*printable text*]

where *keyword-name* is comprised of any printable, non-whitespace characters other than colon (:). A MIME-version: header line indicates that the message is formatted as described in RFC 2045. A Content-Length: header line, indicating the number of bytes in the *message content*, is always present unless the letter consists of only header lines with no message content. A Content-Type: header line that describes the type of the *message content* (such as text/plain, application/octet-stream, and so on) is also present, unless the letter consists of only header lines with no message content. Header lines may be continued on the following line if that line starts with white space.

选项

Sending Mail The following command-line arguments affect sending mail:

-m *message_type* A Message-Type: line is added to the message header with the value of *message_type*.

-t A To: line is added to the message header for each of the intended *recipients*.

-w A letter is sent to a remote recipient without waiting for the completion of the remote transfer program.

If a letter is found to be undeliverable, it is returned to the sender with diagnostics that indicate the location and nature of the failure. If mail is interrupted during input, the message is saved in the file `dead.letter` to allow editing and resending. `dead.letter` is always

appended to, thus preserving any previous contents. The initial attempt to append to (or create) `dead.letter` is in the current directory. If this fails, `dead.letter` is appended to (or created in) the user's login directory. If the second attempt also fails, no `dead.letter` processing is done.

`rmail` only permits the sending of mail; `uucp(1C)` uses `rmail` as a security precaution. Any application programs that generate mail messages should be sure to invoke `rmail` rather than `mail` for message transport and/or delivery.

If the local system has the Basic Networking Utilities installed, mail can be sent to a recipient on a remote system. There are numerous ways to address mail to recipients on remote systems depending on the transport mechanisms available to the local system. The two most prevalent addressing schemes are Domain-style and UUCP-style.

Domain-style addressing	Remote recipients are specified by appending an '@' and domain (and possibly sub-domain) information to the recipient name (such as <code>user@sf.att.com</code>). (The local system administrator should be consulted for details on which addressing conventions are available on the local system.)
UUCP-style addressing	Remote recipients are specified by prefixing the recipient name with the remote system name and an exclamation point, such as <code>sysa!user</code> . If <code>cs(1)</code> is the default shell, <code>sysa!user</code> should be used. A series of system names separated by exclamation points can be used to direct a letter through an extended network (such as <code>sysa!sysb!sysc!user</code> or <code>sysa!sysb!sysc!user</code>).

Reading Mail

The following command-line arguments affect reading mail:

- e Test for the presence of mail. `mail` prints nothing.
An exit status of 0 is returned if the user has mail. Otherwise, an exit status of 1 is returned.
- E Similar to -e, but tests only for the presence of *new* mail.
An exit status of 0 is returned if the user has new mail to read, an exit status of 1 is returned if the user has no mail, or an exit status of 2 is returned if the user has mail which has already been read.
- h A window of headers are initially displayed rather than the latest message. The display is followed by the ? prompt.
- p All messages are printed without prompting for disposition.
- P All messages are printed with *all* header lines displayed, rather than the default selective header line display.

- q mail terminates after interrupts. Normally an interrupt causes only the termination of the message being printed.
- r Messages are printed in first-in, first-out order.
- f *file* mail uses *file* (such as mbox) instead of the default *mailfile*.

mail, unless otherwise influenced by command-line arguments, prints a user's mail messages in last-in, first-out order. The default mode for printing messages is to display only those header lines of immediate interest. These include, but are not limited to, the UNIX From and >From postmarks, From:, Date:, Subject:, and Content-Length: header lines, and any recipient header lines such as To:, Cc:, Bcc:, and so forth. After the header lines have been displayed, mail displays the contents (body) of the message only if it contains no unprintable characters. Otherwise, mail issues a warning statement about the message having binary content and not display the content. This can be overridden by means of the p command.

For each message, the user is prompted with a ? and a line is read from the standard input. The following commands are available to determine the disposition of the message:

- # Print the number of the current message.
- Print previous message.
- <new-line>,+, or n Print the next message.
- !*command* Escape to the shell to do *command*.
- a Print message that arrived during the mail session.
- d, or dp Delete the current message and print the next message.
- d *n* Delete message number *n*. Do not go on to next message.
- dq Delete message and quit mail.
- h Display a window of headers around current message.
- h *n* Display a window of headers around message number *n*.
- h a Display headers of all messages in the user's *mailfile*.
- h d Display headers of messages scheduled for deletion.
- m [*persons*] Mail (and delete) the current message to the named *persons*.
- n* Print message number *n*.
- p Print current message again, overriding any indications of binary (that is, unprintable) content.
- P Override default brief mode and print current message again, displaying all header lines.

q, or Control-d	Put undeleted mail back in the <i>mailfile</i> and quit <code>mail</code> .
r [<i>users</i>]	Reply to the sender, and other <i>users</i> , then delete the message.
s [<i>files</i>]	Save message in the named <i>files</i> (mbox is default) and delete the message.
u [<i>n</i>]	Undelete message number <i>n</i> (default is last read).
w [<i>files</i>]	Save message contents, without any header lines, in the named <i>files</i> (mbox is default) and delete the message.
x	Put all mail back in the <i>mailfile</i> unchanged and exit <code>mail</code> .
y [<i>files</i>]	Same as -w option.
?	Print a command summary.

When a user logs in, the presence of mail, if any, is usually indicated. Also, notification is made if new mail arrives while using `mail`.

The permissions of *mailfile* can be manipulated using `chmod(1)` in two ways to alter the function of `mail`. The other permissions of the file can be read-write (0666), read-only (0664), or neither read nor write (0660) to allow different levels of privacy. If changed to other than the default (mode 0660), the file is preserved even when empty to perpetuate the desired permissions. (The administrator can override this file preservation using the `DEL_EMPTY_MAILFILE` option of `mailcnfg`.)

The group ID of the mailfile must be `mail` to allow new messages to be delivered, and the mailfile must be writable by group `mail`.

Debugging

The following command-line arguments cause `mail` to provide debugging information:

`-x debug_level` `mail` creates a trace file containing debugging information.

The `-x` option causes `mail` to create a file named `/tmp/MLDBGprocess_id` that contains debugging information relating to how `mail` processed the current message. The absolute value of `debug_level` controls the verbosity of the debug information. 0 implies no debugging. If `debug_level` is greater than 0, the debug file is retained *only* if `mail` encountered some problem while processing the message. If `debug_level` is less than 0, the debug file is always be retained. The `debug_level` specified via `-x` overrides any specification of `DEBUG` in `/etc/mail/mailcnfg`. The information provided by the `-x` option is esoteric and is probably only useful to system administrators.

Delivery Notification

Several forms of notification are available for mail by including one of the following lines in the message header.

Transport-Options: [*/options*]

Default-Options: [/options]

>To: *recipient* [/options]

Where the “/options” can be one or more of the following:

/delivery	Inform the sender that the message was successfully delivered to the <i>recipient's</i> mailbox.
/nodelivery	Do not inform the sender of successful deliveries.
/ignore	Do not inform the sender of failed deliveries.
/return	Inform the sender if mail delivery fails. Return the failed message to the sender.
/report	Same as /return except that the original message is not returned.

The default is /nodelivery/return. If contradictory options are used, the first is recognized and later, conflicting, terms are ignored.

操作数

The following operand is supported for sending mail:

recipient A domain style address (“*user@machine*”) or user login name recognized by [login\(1\)](#).

用法

See [largefile\(5\)](#) for the description of the behavior of mail and rmail when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of mail: LC_CTYPE, LC_MESSAGES, and NLSPATH.

TZ Determine the timezone used with date and time strings.

退出状态

The following exit values are returned:

0	Successful completion when the user had mail.
1	The user had no mail or an initialization error occurred.
>1	An error occurred after initialization.

文件

dead.letter	unmailable text
/etc/passwd	to identify sender and locate <i>recipients</i>
\$HOME/mbox	saved mail
\$MAIL	variable containing path name of <i>mailfile</i>
/tmp/MLDBG*	debug trace file
/var/mail/*.lock	lock for mail directory

`/var/mail/:saved` directory for holding temp files to prevent loss of data in the event of a system crash

`/var/mail/user` incoming mail for *user*; that is, the *mailfile*

`var/tmp/ma*` temporary file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[chmod\(1\)](#), [csh\(1\)](#), [login\(1\)](#), [mailx\(1\)](#), [uucp\(1C\)](#), [uuencode\(1C\)](#), [vacation\(1\)](#), [write\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#)

《Solaris Advanced User's Guide》

附注

The interpretation and resulting action taken because of the header lines described in the Delivery Notifications section only occur if this version of `mail` is installed on the system where the delivery (or failure) happens. Earlier versions of `mail` might not support any types of delivery notification.

Conditions sometimes result in a failure to remove a lock file.

After an interrupt, the next message might not be printed. Printing can be forced by typing a `p`.

引用名 mail, Mail – interactive message processing system

用法概要 /usr/ucb/mail ...

/usr/ucb/Mail ...

描述 /usr/ucb/mail and /usr/ucb/Mail are provided as links to /usr/bin/mailx. See [mailx\(1\)](#) for more information on the usage of these commands.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/ucb/mail

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

/usr/ucb/Mail

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [mailx\(1\)](#), [attributes\(5\)](#)

引用名	mailcompat – provide SunOS compatibility for Solaris mailbox format
描述	<p>mailcompat is a program to provide SunOS 4.x compatibility for the Solaris mailbox format. You would typically run mailcompat to be able to read mail on a workstation running SunOS 4.x when your mail server is running Solaris.</p> <p>Enabling mailcompat creates an entry in your .forward file, if it exists. If this file does not exist, mailcompat will create it. Disabling mailcompat will remove the entry from the .forward file, and if this was the only entry, will remove the entire file.</p> <p>To execute mailcompat, log onto the Solaris mail server and enter mailcompat on the command line. Answer the queries provided by the program.</p>
用法	<p>See largefile(5) for the description of the behavior of mailcompat when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).</p> <p>Depending on the security settings in your system, mailcompat can fail to create a .forward file. If you encounter such a failure, take the steps outlined below.</p> <p>As root user or user with comparable privileges, enter:</p> <pre>% sendmail -bv \$USER</pre> <p>You receive one of the following two messages. The string fwtest is for example purposes only.</p> <pre>fwtest... deliverable: mailer local, user fwtest</pre> <pre>" /usr/bin/mailcompat fwtest"... deliverable: mailer prog,\ user " /usr/bin/mailcompat fwtest"</pre> <p>The first message, with the local designation, indicates that mailcompat was not able to create the .forward file. The second message, with the prog designation, means that mailcompat succeeded in creating .forward.</p> <p>As a further check, invoke:</p> <pre>% check-permissions \$USER</pre> <p>You should get the message:</p> <pre>No unsafe directories found.</pre> <p>If you get a message different from this, it might indicate a reason for mailcompat failure. You should correct any conditions pointed out by the message before retrying mailcompat.</p>
示例	<p>示例1 Examples of the mailcompat feature.</p> <p>The following example enables the mailcompat feature for the user john.</p> <pre>example% mailcompat</pre> <p>This program can be used to store your mail in a format that you can read with SunOS 4.X based mail readers</p>

示例 1 Examples of the mailcompat feature. (续)

```
To enable the mailcompat feature a ".forward" file is created.
Would you like to enable the mailcompat feature? Y
Mailcompat feature ENABLED.Run mailcompat with no arguments to remove it
example%
```

The following example disables the mailcompat feature for the user john.

```
example% mailcompat
This program can be used to store your mail in a format
that you can read with SunOS 4.X based mail readers
You have a .forward file in your home directory containing:
    "|/usr/bin/mailcompat johns"
Would you like to remove it and disable the mailcompat feature? y
Back to normal reception of mail.
example%
```

文件 ~/.forward list of recipients for forwarding messages

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [mailx\(1\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

引用名	mailp, digestp, filep, newsp, filofaxp, franklinp, timemanp, timesysp – frontends to the mp Text to PDL (Printer Description Language) pretty print filter																
用法概要	<pre>mailp [options] filename... newsp [options] filename... digestp [options] filename... filep [options] filename... filofaxp [options] filename... franklinp [options] filename... timemanp [options] filename... timesysp [options] filename...</pre>																
描述	<p>The <code>mailp</code> utility is a frontend to the <code>mp(1)</code> program. It uses different names to provide various mp options:</p> <table border="0"> <tr> <td style="padding-right: 20px;"><code>mailp</code></td> <td>Prints out mail messages.</td> </tr> <tr> <td><code>newsp</code></td> <td>Prints out USENET news articles.</td> </tr> <tr> <td><code>digestp</code></td> <td>Prints out USENET digest files.</td> </tr> <tr> <td><code>filep</code></td> <td>Prints out ordinary ASCII files.</td> </tr> <tr> <td><code>filofaxp</code></td> <td>Prints out in Filofax personal organiser format.</td> </tr> <tr> <td><code>franklinp</code></td> <td>Prints out in Franklin Planner personal organiser format.</td> </tr> <tr> <td><code>timemanp</code></td> <td>Prints out in Time Manager personal organiser format.</td> </tr> <tr> <td><code>timesysp</code></td> <td>Prints out in Time/System International personal organiser format.</td> </tr> </table> <p><code>mailp</code> and the associated programs read each <i>filename</i> in sequence and generate a prettified version of the contents. If no filename arguments are provided, <code>mailp</code> reads the standard input.</p> <p><code>mailp</code> works in two ways. With the <code>-D</code> option, it will work as an X print server client to produce the PDL of the target printer and spool it. With the <code>-d</code> or <code>-P</code> option, it will generate and spool PostScript™ output.</p>	<code>mailp</code>	Prints out mail messages.	<code>newsp</code>	Prints out USENET news articles.	<code>digestp</code>	Prints out USENET digest files.	<code>filep</code>	Prints out ordinary ASCII files.	<code>filofaxp</code>	Prints out in Filofax personal organiser format.	<code>franklinp</code>	Prints out in Franklin Planner personal organiser format.	<code>timemanp</code>	Prints out in Time Manager personal organiser format.	<code>timesysp</code>	Prints out in Time/System International personal organiser format.
<code>mailp</code>	Prints out mail messages.																
<code>newsp</code>	Prints out USENET news articles.																
<code>digestp</code>	Prints out USENET digest files.																
<code>filep</code>	Prints out ordinary ASCII files.																
<code>filofaxp</code>	Prints out in Filofax personal organiser format.																
<code>franklinp</code>	Prints out in Franklin Planner personal organiser format.																
<code>timemanp</code>	Prints out in Time Manager personal organiser format.																
<code>timesysp</code>	Prints out in Time/System International personal organiser format.																
选项	<p>The following options are supported:</p> <table border="0"> <tr> <td style="padding-right: 20px;"><code>-d printer</code></td> <td>Sends output to the named printer. Otherwise, sends output to the printer named in the <code>PRINTER</code> environment variable.</td> </tr> <tr> <td><code>-D</code></td> <td>Generates the PDL for the target printer and spools it to the printer.</td> </tr> </table>	<code>-d printer</code>	Sends output to the named printer. Otherwise, sends output to the printer named in the <code>PRINTER</code> environment variable.	<code>-D</code>	Generates the PDL for the target printer and spools it to the printer.												
<code>-d printer</code>	Sends output to the named printer. Otherwise, sends output to the printer named in the <code>PRINTER</code> environment variable.																
<code>-D</code>	Generates the PDL for the target printer and spools it to the printer.																

- F Instead of printing who the mail article is *for*, the top header will contain who the mail article is *from*. This is a useful option for people with their own personal printer.
- h Banner printing is disabled. Most of the information that typically appears on the banner sheet is output in the mp banners.
- l Formats output in landscape mode. Two pages of text will be printed per sheet of paper.
- P *printer* Same as -d option.
- s *subject* Uses *subject* as the new subject for the printout. If you are printing ordinary ASCII files which have been specified on the command line, the subject will default to the name of each of these files.

操作数

The following operand is supported:

filename The name of the file to be read.

环境变量

If none of the -d, -D, or -P options is used, mailp uses the PRINTER environment variable to determine the printer to which the output from the mp(1) program is sent. If the PRINTER variable is not found, the default destination is the PostScript™ printer.

退出状态

The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	print/mp

另请参见

[mp\(1\)](#), [attributes\(5\)](#)

附注

The -P option, which spools the PDL directly to the target printer in mp(1), produces PostScript™ when used in mailp so as to be backward compatible.

引用名	mailq – print the mail queue
用法概要	<code>/usr/bin/mailq [-Ac] [-q <i>subarg</i>] [-v]</code>
描述	<p>The mailq utility displays a summary of the mail messages queued for future delivery.</p> <p>The first line displayed for each mail message shows the internal identifier used on this host for the message, the size of the message in bytes, the date and time the message was accepted into the queue, and the envelope sender of the message. The second line of the display shows the error message that caused this message to be retained in the queue. This line will not be displayed if the message is being processed for the first time.</p> <p>The mailq utility used to be identical to <code>sendmail -bp</code>. Now it checks for the authorization attribute, <code>solaris.mail.mailq</code>. If the check for the invoking user succeeds, <code>sendmail -bp</code> is executed with the remaining argument vector. Otherwise, an error message is printed. This authorization attribute is by default enabled for all users. It can be disabled by modifying the Basic Solaris User entry in <code>prof_attr(4)</code>.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -Ac Like <code>sendmail(1M)</code>, this flag tells mailq to use <code>submit.cf</code> rather than <code>sendmail.cf</code> even if the operation mode does not indicate an initial mail submission. This will result in the client queue <code>/var/spool/clientmqueue</code> being displayed rather than the default server queue <code>/var/spool/mqueue</code>. -qp[<i>time</i>] Similar to <code>-qtime</code>, except that instead of periodically forking a child to process the queue, <code>sendmail</code> forks a single persistent child for each queue that alternates between processing the queue and sleeping. The sleep time is given as the argument. The sleep time default is 1 second. The process will always sleep at least 5 seconds if the queue was empty in the previous queue run. -qf Processes saved messages in the queue once and does not <code>fork()</code>, but runs in the foreground. -qG <i>name</i> Processes jobs in the queue group called <i>name</i> only. -q[!]I <i>substr</i> Limits processed jobs to those containing <i>substr</i> as a substring of the queue id, or not when ! is specified. -q[!]R <i>substr</i> Limits processed jobs to those containing <i>substr</i> as a substring of one of the recipients, or not when ! is specified. -q[!]S <i>substr</i> Limits processed jobs to those containing <i>substr</i> as a substring of the sender, or not when ! is specified. -v Prints verbose information. This adds the priority of the message and a single character indicator (+ or blank) indicating whether a warning message has been sent on the first line of the message. Additionally, extra lines may be intermixed with the recipients that indicate the "controlling user"

information. This shows who will own any programs that are executed on behalf of this message and the name of the alias this command is expanded from, if any.

退出状态 0 Successful completion.

>0 An error occurred.

文件 /etc/security/prof_attr local source for execution profile attributes

/var/spool/mqueue default server queue

/var/spool/clientmqueue client queue

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail

另请参见 [sendmail\(1M\)](#), [prof_attr\(4\)](#), [attributes\(5\)](#)

引用名 mailstats – print statistics collected by sendmail

用法概要 mailstats [-o] [-c] [-C *configfile*] [-f *statisticsfile*]
[-p] [-P]

描述 The mailstats utility prints out the statistics collected by the [sendmail\(1M\)](#) program on mailer usage. These statistics are collected if the file indicated by the `StatusFile` configuration option of sendmail (defined in `/etc/mail/sendmail.cf`) exists. The default statistics file is `/etc/mail/statistics`.

To enable mailstats, you must, as root, touch `/etc/mail/statistics`. See the `StatusFile` processing option in [sendmail\(1M\)](#).

mailstats first prints the time that the statistics file was created and the last time it was modified. Then, the statistics for each mailer are displayed on a single line, each with the following whitespace-separated fields:

M	The mailer number.
msgsfrr	Number of messages from the mailer.
bytes_from	Kbytes from the mailer.
msgsto	Number of messages to the mailer.
bytes_to	Kbytes to the mailer.
msgsrrej	Number of messages rejected by the mailer.
msgsrdis	Number of messages discarded by the mailer.
msgsrqur	Number of messages quarantined by the mailer.
Mailer	The name of the mailer.

The display of statistics described above is followed by a separation line containing only equal sign (=) characters. After the separation line, a line preceded with a “T” and totaling the values for all of the mailers is displayed. This is followed by another line preceded with a “C” that lists the number of TCP connections.

To reinitialize the statistics file once a night, add an entry to root's [crontab\(1\)](#):

```
mailstats -p > /dev/null
```

选项 The following options are supported:

-c	Try to use <code>submit.cf</code> instead of the default sendmail configuration file.
-C <i>configfile</i>	Specify a sendmail configuration file.
-f <i>statisticsfile</i>	Specify a sendmail statistics file.
-o	Do not display the name of the mailer in the output.

- p Output information in program-readable mode and clear statistics.
- P Output information in program-readable mode without clearing statistics.

用法 See [largefile\(5\)](#) for the description of the behavior of `mailstats` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

文件

- `/dev/null` Zero-lined file
- `/etc/mail/statistics` Default `sendmail` statistics file
- `/etc/mail/sendmail.cf` Default `sendmail` configuration file

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail
Interface Stability	The output is uncommitted.

另请参见 [crontab\(1\)](#), [cron\(1M\)](#), [sendmail\(1M\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

引用名	mailx – interactive message processing system
用法概要	<pre>mailx [-BdeHiInNURvV~] [-f [file +folder]] [-T file] [-u user] mailx [-BdFintUv~] [-b bcc] [-c cc] [-h number] [-r address] [-s subject] recipient... /usr/ucb/mail ... /usr/ucb/Mail ...</pre>
描述	<p>The mail utilities listed above provide a comfortable, flexible environment for sending and receiving mail messages electronically.</p> <p>When reading mail, the mail utilities provide commands to facilitate saving, deleting, and responding to messages. When sending mail, the mail utilities allow editing, reviewing and other modification of the message as it is entered.</p> <p>Incoming mail is stored in a standard file for each user, called the mailbox for that user. When the mail utilities are called to read messages, the mailbox is the default place to find them. As messages are read, they are marked to be moved to a secondary file for storage, unless specific action is taken, so that the messages need not be seen again. This secondary file is called the mbox and is normally located in the user's HOME directory (see MBOX in ENVIRONMENT VARIABLES for a description of this file). Messages can be saved in other secondary files named by the user. Messages remain in a secondary file until forcibly removed.</p> <p>The user can access a secondary file by using the -f option. Messages in the secondary file can then be read or otherwise processed using the same Commands as in the primary mailbox. This gives rise within these pages to the notion of a current mailbox.</p>
选项	<p>On the command line options start with a dash (-). Any other arguments are taken to be destinations (recipients). If no recipients are specified, mailx attempts to read messages from the mailbox.</p> <ul style="list-style-type: none"> -B Do not buffer standard input or standard output. -b <i>bcc</i> Set the blind carbon copy list to <i>bcc</i>. <i>bcc</i> should be enclosed in quotes if it contains more than one name. -c <i>cc</i> Set the carbon copy list to <i>cc</i>. <i>cc</i> should be enclosed in quotes if it contains more than one name. -d Turn on debugging output. (Neither particularly interesting nor recommended.) -e Test for the presence of mail. mailx prints nothing and exits with a successful return code if there is mail to read. -F Record the message in a file named after the first recipient. Overrides the record variable, if set (see Internal Variables).

- f [*file*] Read messages from *file* instead of mailbox. If no *file* is specified, the mailbox is used.
- f [+*folder*] Use the file *folder* in the folder directory (same as the folder command). The name of this directory is listed in the folder variable.
- H Print header summary only.
- h *number* The number of network “hops” made so far. This is provided for network software to avoid infinite delivery loops. This option and its argument are passed to the delivery program.
- I Include the newsgroup and article-id header lines when printing mail messages. This option requires the -f option to be specified.
- i Ignore interrupts. See also ignore in Internal Variables.
- N Do not print initial header summary.
- n Do not initialize from the system default mailx.rc or Mail.rc file. See USAGE.
- r *address* Use *address* as the return address when invoking the delivery program. All tilde commands are disabled. This option and its argument is passed to the delivery program.
- s *subject* Set the Subject header field to *subject*. *subject* should be enclosed in quotes if it contains embedded white space.
- T *file* Message-id and article-id header lines are recorded in *file* after the message is read. This option also sets the -I option.
- t Scan the input for To:, Cc:, and Bcc: fields. Any recipients on the command line will be ignored.
- U Convert UUCP-style addresses to internet standards. Overrides the conv environment variable.
- u *user* Read *user*'s mailbox. This is only effective if *user*'s mailbox is not read protected.
- V Print the mailx version number and exit.
- v Pass the -v flag to sendmail(1M).
- ~ Interpret tilde escapes in the input even if not reading from a tty.

操作数

The following operands are supported:

recipient Addressee of message.

用法

Starting Mail

At startup time, mailx executes the system startup file `/etc/mail/mailx.rc`. If invoked as `mail` or `Mail`, the system startup file `/etc/mail/Mail.rc` is used instead.

The system startup file sets up initial display options and alias lists and assigns values to some internal variables. These variables are flags and valued parameters which are set and cleared using the `set` and `unset` commands. See [Internal Variables](#).

With the following exceptions, regular commands are legal inside startup files: `!`, `Copy`, `edit`, `followup`, `Followup`, `hold`, `mail`, `preserve`, `reply`, `Reply`, `shell`, and `visual`. An error in the startup file causes the remaining lines in the file to be ignored.

After executing the system startup file, the mail utilities execute the optional personal startup file `$HOME/.mailrc`, wherein the user can override the values of the internal variables as set by the system startup file.

If the `-n` option is specified, however, the mail utilities do not execute the system startup file.

Many system administrators include the commands

```
set appenddeadletter
unset replyall
unset pipeignore
```

in the system startup files (to be compatible with past Solaris behavior), but this does not meet standards requirements for mailx. To get standard behavior for mailx, users should use the `-n` option or include the following commands in a personal startup file:

```
unset appenddeadletter
set replyall
set pipeignore
```

When reading mail, the mail utilities are in *command mode*. A header summary of the first several messages is displayed, followed by a prompt indicating the mail utilities can accept regular commands (see [Commands](#) below). When sending mail, the mail utilities are in *input mode*. If no subject is specified on the command line, and the `asksub` variable is set, a prompt for the subject is printed.

As the message is typed, the mail utilities read the message and store it in a temporary file. Commands may be entered by beginning a line with the tilde (`~`) escape character followed by a single command letter and optional arguments. See [Tilde Escapes](#) for a summary of these commands.

Reading Mail

Each message is assigned a sequential number, and there is at any time the notion of a current message, marked by a right angle bracket (`>`) in the header summary. Many commands take

an optional list of messages (*message-list*) to operate on. In most cases, the current message is set to the highest-numbered message in the list after the command is finished executing.

The default for *message-list* is the current message. A *message-list* is a list of message identifiers separated by spaces, which may include:

<i>n</i>	Message number <i>n</i> .
.	The current message.
^	The first undeleted message.
\$	The last message.
*	All messages.
+	The next undeleted message.
-	The previous undeleted message.
<i>n-m</i>	An inclusive range of message numbers.
<i>user</i>	All messages from <i>user</i> .
<i>/string</i>	All messages with <i>string</i> in the Subject line (case ignored).
<i>:c</i>	All messages of type <i>c</i> , where <i>c</i> is one of:
d	deleted messages
n	new messages
o	old messages
r	read messages
u	unread messages

Notice that the context of the command determines whether this type of message specification makes sense.

Other arguments are usually arbitrary strings whose usage depends on the command involved. Filenames, where expected, are expanded using the normal shell conventions (see [sh\(1\)](#)). Special characters are recognized by certain commands and are documented with the commands below.

Sending Mail

Recipients listed on the command line may be of three types: login names, shell commands, or alias groups. Login names may be any network address, including mixed network addressing. If mail is found to be undeliverable, an attempt is made to return it to the sender's mailbox. If the recipient name begins with a pipe symbol (|), the rest of the name is taken to be a shell command to pipe the message through. This provides an automatic interface with any

program that reads the standard input. Groups are set by the `alias` command (see `Commands` below) or in a system startup file (for example, `$HOME/.mailrc`). Aliases are lists of recipients of any type.

Forwarding Mail To forward a specific message, include it in a message to the desired recipients with the `~f` or `~m` tilde escapes. See `Tilde Escapes` below. To forward mail automatically, add a comma-separated list of addresses for additional recipients to the `.forward` file in your home directory. This is different from the format of the `alias` command, which takes a space-separated list instead. *Note:* Forwarding addresses must be valid, or the messages will “bounce.” You cannot, for instance, reroute your mail to a new host by forwarding it to your new address if it is not yet listed in the NIS aliases domain.

Commands Regular commands are of the form

```
[ command ] [ message-list ] [ arguments ]
```

In *input mode*, commands are recognized by the escape character, tilde (~), and lines not treated as commands are taken as input for the message. If no command is specified in *command mode*, next is assumed. The following is a complete list of mailx commands:

<code>!shell-command</code>	Escape to the shell. See SHELL in ENVIRONMENT VARIABLES.
<code># comment</code>	NULL command (comment). Useful in mailrc files.
<code>=</code>	Print the current message number.
<code>?</code>	Prints a summary of commands.
<code>alias <i>alias name</i> . . .</code> <code>group <i>alias name</i> . . .</code>	Declare an alias for the given names. The names are substituted when <code>alias</code> is used as a recipient. Useful in the mailrc file. With no arguments, the command displays the list of defined aliases.
<code>alternates <i>name</i> . . .</code>	Declare a list of alternate names for your login. When responding to a message, these names are removed from the list of recipients for the response. With no arguments, print the current list of alternate names. See also <code>allnet</code> in <code>Internal Variables</code> .

<code>cd</code> [<i>directory</i>] <code>chdir</code> [<i>directory</i>]	Change directory. If <i>directory</i> is not specified, \$HOME is used.
<code>copy</code> [<i>file</i>] <code>copy</code> [<i>message-list</i>] <i>file</i>	Copy messages to the file without marking the messages as saved. Otherwise equivalent to the save command.
<code>Copy</code> [<i>message-list</i>]	Save the specified messages in a file whose name is derived from the author of the message to be saved, without marking the messages as saved. Otherwise equivalent to the Save command.
<code>delete</code> [<i>message-list</i>]	Delete messages from the mailbox. If <code>autoprint</code> is set, the next message after the last one deleted is printed (see <code>Internal Variables</code>).
<code>discard</code> [<i>header-field. . .</i>] <code>ignore</code> [<i>header-field. . .</i>]	Suppress printing of the specified header fields when displaying messages on the screen. Examples of header fields to ignore are <code>Status</code> and <code>Received</code> . The fields are included when the message is saved, unless the <code>alwaysignore</code> variable is set. The <code>More</code> , <code>Page</code> , <code>Print</code> , and <code>Type</code> commands override this command. If no header is specified, the current list of header fields being ignored is printed. See also the <code>undiscard</code> and <code>unignore</code> commands.
<code>dp</code> [<i>message-list</i>] <code>dt</code> [<i>message-list</i>]	Delete the specified messages from the mailbox and print the next message after the last one deleted. Roughly equivalent to a delete command followed by a print command.
<code>echo</code> <i>string. . .</i>	Echo the given strings (like <code>echo(1)</code>).
<code>edit</code> [<i>message-list</i>]	Edit the given messages. Each message is placed in a temporary file and the

	program named by the EDITOR variable is invoked to edit it (see ENVIRONMENT VARIABLES). Default editor is <code>ed(1)</code> .
<code>exit</code>	
<code>xit</code>	Exit from <code>mailx</code> , without changing the mailbox. No messages are saved in the <code>mbox</code> (see also <code>quit</code>).
<code>field [message-list] header-file</code>	Display the value of the header field in the specified message.
<code>file [file]</code>	
<code>folder [file]</code>	Quit from the current file of messages and read in the specified file. Several special characters are recognized when used as file names:
	<code>%</code> the current mailbox.
	<code>%user</code> the mailbox for <i>user</i> .
	<code>#</code> the previous mail file.
	<code>&</code> the current mbox.
	<code>+file</code> The named file in the <i>folder</i> directory (listed in the <code>folder</code> variable).
	With no arguments, print the name of the current mail file, and the number of messages and characters it contains.
<code>folders</code>	Print the names of the files in the directory set by the <code>folder</code> variable (see Internal Variables).
<code>Followup [message]</code>	Respond to a message, recording the response in a file whose name is derived from the author of the message. Overrides the <code>record</code> variable, if set. If the <code>replyall</code> variable is set, the actions of <code>Followup</code> and <code>followup</code> are reversed. See also the <code>followup</code> , <code>Save</code> , and <code>Copy</code> commands and <code>outfolder</code> in Internal Variables, and the Starting Mail section in USAGE above.

followup [*message-list*]

Respond to the first message in the *message-list*, sending the message to the author of each message in the *message-list*. The subject line is taken from the first message and the response is recorded in a file whose name is derived from the author of the first message. If the `replyall` variable is set, the actions of `followup` and `Followup` are reversed. See also the `Followup`, `Save`, and `Copy` commands and `outfolder` in `Internal Variables`, and the `Starting Mail` section in `USAGE` above.

from [*message-list*]

Print the header summary for the specified messages. If no messages are specified, print the header summary for the current message.

group *alias name* . . .
alias *alias name* . . .

Declare an alias for the given names. The names are substituted when `alias` is used as a recipient. Useful in the `mailrc` file.

headers [*message*]

Print the page of headers which includes the message specified. The `screen` variable sets the number of headers per page (see `Internal Variables`). See also the `z` command.

help

Print a summary of commands.

hold [*message-list*]

preserve [*message-list*]

Hold the specified messages in the mailbox.

ifs|r|t
mail-commands
else
mail-commands
endif

Conditional execution, where *s* executes following *mail-commands*, up to an `else` or `endif`, if the program is in *send* mode, *r* causes the *mail-commands* to be

	executed only in <i>receive</i> mode, and <i>t</i> causes the <i>mail-commands</i> to be executed only if <i>mailx</i> is being run from a terminal. Useful in the <i>mailrc</i> file.
<code>inc</code>	Incorporate messages that arrive while you are reading the system mailbox. The new messages are added to the message list in the current <i>mail</i> session. This command does not commit changes made during the session, and prior messages are not renumbered.
<code>ignore [header-field...]</code> <code>discard [header-field...]</code>	Suppress printing of the specified header fields when displaying messages on the screen. Examples of header fields to ignore are <i>Status</i> and <i>Cc</i> . All fields are included when the message is saved. The <i>More</i> , <i>Page</i> , <i>Print</i> and <i>Type</i> commands override this command. If no header is specified, the current list of header fields being ignored is printed. See also the <i>undiscard</i> and <i>unignore</i> commands.
<code>list</code>	Print all commands available. No explanation is given.
<code>load</code>	<code>[message] file</code> The specified message is replaced by the message in the named file. <code>file</code> should contain a single mail message including mail headers (as saved by the <i>save</i> command).
<code>mail recipient...</code>	Mail a message to the specified recipients.
<code>Mail recipient</code>	Mail a message to the specified recipients, and record it in a file whose name is derived from the author of the message. Overrides the <i>record</i> variable, if set. See also the <i>Save</i> and <i>Copy</i> commands and <i>outfolder</i> in <i>Internal Variables</i> .

`mbox` [*message-list*]

Arrange for the given messages to end up in the standard mbox save file when mailx terminates normally. See MBOX in ENVIRONMENT VARIABLES for a description of this file. See also the exit and quit commands.

`more` [*message-list*]

`page` [*message-list*]

Print the specified messages. If `crt` is set, the messages longer than the number of lines specified by the `crt` variable are paged through the command specified by the `PAGER` variable. The default command is `pg(1)` or if the `bsdcompat` variable is set, the default is `more(1)`. See ENVIRONMENT VARIABLES. Same as the print and type commands.

`More` [*message-list*]

`Page` [*message-list*]

Print the specified messages on the screen, including all header fields. Overrides suppression of fields by the ignore command. Same as the Print and Type commands.

`new` [*message-list*]

`New` [*message-list*]

`unread` [*message-list*]

`Unread`

[*message-list*] Take a message list and mark each message as *not* having been read.

`next` [*message*]

Go to the next message matching *message*. If *message* is not supplied, this command finds the next message that was not deleted or saved. A *message-list* may be specified, but in this case the first valid message in the list is the only one used. This is useful for jumping to the next message from a specific user, since the name would be taken as a command in the absence of a real command. See


```
pipe [message-list] [shell-command]
| [message-list] [shell-command]
```

the discussion of *message-list* above for a description of possible message specifications.

Pipe the message through the given *shell-command*. The message is treated as if it were read. If no arguments are given, the current message is piped through the command specified by the value of the `cmd` variable. If the `page` variable is set, a form feed character is inserted after each message (see `Internal Variables`).

```
preserve [message-list]
hold [message-list]
```

Preserve the specified messages in the mailbox.

```
print [message-list]
type [message-list]
```

Print the specified messages. If `crt` is set, the messages longer than the number of lines specified by the `crt` variable are paged through the command specified by the `PAGER` variable. The default command is `pg(1)` or if the `bsdcompat` variable is set, the default is `more(1)`. See `ENVIRONMENT VARIABLES`. Same as the `more` and `page` commands.

```
Print [message-list]
Type [message-list]
```

Print the specified messages on the screen, including all header fields. Overrides suppression of fields by the `ignore` command. Same as the `More` and `Page` commands.

```
put [file]
put [message-list] file
```

Save the specified message in the given file. Use the same conventions as the `print` command for which header fields are ignored.

Put [*file*]

Put [*message-list*] *file*

Save the specified message in the given file. Overrides suppression of fields by the ignore command.

quit

Exit from mailx, storing messages that were read in mbox and unread messages in the mailbox. Messages that have been explicitly saved in a file are deleted unless the keepsave variable is set.

reply [*message-list*]

respond [*message-list*]

replysender [*message-list*]

Send a response to the author of each message in the *message-list*. The subject line is taken from the first message. If record is set to a file, a copy of the reply is added to that file. If the replyall variable is set, the actions of Reply/Respond and reply/respond are reversed. The replysender command is not affected by the replyall variable, but sends each reply only to the sender of each message. See the Starting Mail section in USAGE above.

Reply [*message*]

Respond [*message*]

replyall [*message*]

Reply to the specified message, including all other recipients of that message. If the variable record is set to a file, a copy of the reply added to that file. If the replyall variable is set, the actions of Reply/Respond and reply/respond are reversed. The replyall command is not affected by the replyall variable, but always sends the reply to all recipients of the message. See the Starting Mail section in USAGE above.

retain

Add the list of header fields named to the *retained list*. Only the header fields in the retain list are shown on your terminal when you print a message. All

Save [*message-list*]

save [*file*]
save [*message-list*] *file*

set
set *variable*
set *variable*=*string*
set *variable*=*number*

other header fields are suppressed. The set of retained fields specified by the retain command overrides any list of ignored fields specified by the ignore command. The Type and Print commands can be used to print a message in its entirety. If retain is executed with no arguments, it lists the current set of retained fields.

Save the specified messages in a file whose name is derived from the author of the first message. The name of the file is taken to be the author's name with all network addressing stripped off. See also the Copy, followup, and Followup commands and outfolder in Internal Variables.

Save the specified messages in the given file. The file is created if it does not exist. The file defaults to mbox. The message is deleted from the mailbox when mailx terminates unless keepsave is set (see also Internal Variables and the exit and quit commands).

Define a *variable*. To assign a *value* to *variable*, separate the variable name from the value by an '=' (there must be no space before or after the '='). A variable may be given a null, string, or numeric *value*. To embed SPACE characters within a *value*, enclose it in quotes.

With no arguments, set displays all defined variables and any values they might have. See Internal Variables for a description of all predefined mail variables.

shell	Invoke an interactive shell. See also SHELL in ENVIRONMENT VARIABLES.
size [<i>message-list</i>]	Print the size in characters of the specified messages.
source <i>file</i>	Read commands from the given file and return to command mode.
top [<i>message-list</i>]	Print the top few lines of the specified messages. If the topLines variable is set, it is taken as the number of lines to print (see Internal Variables). The default is 5.
touch [<i>message-list</i>]	Touch the specified messages. If any message in <i>message-list</i> is not specifically saved in a file, it is placed in the mbox, or the file specified in the MBOX environment variable, upon normal termination. See exit and quit.
Type [<i>message-list</i>] Print [<i>message-list</i>]	Print the specified messages on the screen, including all header fields. Overrides suppression of fields by the ignore command.
type [<i>message-list</i>] print [<i>message-list</i>]	Print the specified messages. If crt is set, the messages longer than the number of lines specified by the crt variable are paged through the command specified by the PAGER variable. The default command is pg(1) . See ENVIRONMENT VARIABLES.
unalias [<i>alias</i>] ... ungroup [<i>alias</i>] ...	Remove the definitions of the specified aliases.
undelete [<i>message-list</i>]	Restore the specified deleted messages. Will only restore messages deleted in the current mail session. If autoprint is set, the last message of those restored is printed (see Internal Variables).

<p>undiscard [<i>header-field</i> . . .] unignore [<i>header-field</i> . . .]</p>	<p>Remove the specified header fields from the list being ignored. If no header fields are specified, all header fields are removed from the list being ignored.</p>
<p>unretain [<i>header-field</i> . . .]</p>	<p>Remove the specified header fields from the list being retained. If no header fields are specified, all header fields are removed from the list being retained.</p>
<p>unread [<i>message-list</i>] Unread [<i>message-list</i>] Same as the new command.</p>	
<p>unset <i>variable</i> . . .</p>	<p>Erase the specified variables. If the variable was imported from the environment (that is, an environment variable or exported shell variable), it cannot be unset from within mailx.</p>
<p>version</p>	<p>Print the current version and release date of the mailx utility.</p>
<p>visual [<i>message-list</i>]</p>	<p>Edit the given messages with a screen editor. Each messages is placed in a temporary file and the program named by the VISUAL variable is invoked to edit it (see ENVIRONMENT VARIABLES). Notice that the default visual editor is vi.</p>
<p>write [<i>message-list</i>] <i>file</i></p>	<p>Write the given messages on the specified file, minus the header and trailing blank line. Otherwise equivalent to the save command.</p>
<p>xit exit</p>	<p>Exit from mailx, without changing the mailbox. No messages are saved in the mbox (see also quit).</p>
<p>z[+ -]</p>	<p>Scroll the header display forward or backward one screen–full. The number of headers displayed is set by the screen variable (see Internal Variables).</p>

Tilde Escapes

The following tilde escape commands can be used when composing mail to send. These may be entered only from *input mode*, by beginning a line with the tilde escape character (~). See escape in Internal Variables for changing this special character. The escape character can be entered as text by typing it twice.

~ ! <i>shell-command</i>	Escape to the shell. If present, run <i>shell-command</i> .
~.	Simulate end of file (terminate message input).
~ : <i>mail-command</i>	
~_ <i>mail-command</i>	Perform the command-level request. Valid only when sending a message while reading mail.
~?	Print a summary of tilde escapes.
~A	Insert the autograph string Sign into the message (see Internal Variables).
~a	Insert the autograph string sign into the message (see Internal Variables).
~b <i>name</i> . . .	Add the <i>names</i> to the blind carbon copy (Bcc) list. This is like the carbon copy (Cc) list, except that the names in the Bcc list are not shown in the header of the mail message.
~c <i>name</i> . . .	Add the <i>names</i> to the carbon copy (Cc) list.
~d	Read in the dead-letter file. See DEAD in ENVIRONMENT VARIABLES for a description of this file.
~e	Invoke the editor on the partial message. See also EDITOR in ENVIRONMENT VARIABLES.
~f [<i>message-list</i>]	Forward the specified message, or the current message being read. Valid only when sending a message while reading mail. The messages are inserted into the message without alteration (as opposed to the ~m escape).
~F [<i>message-list</i>]	Forward the specified message, or the current message being read, including all header fields. Overrides the suppression of fields by the ignore command.
~h	Prompt for Subject line and To, Cc, and Bcc lists. If the field is displayed with an initial value, it may be edited as if you had just typed it.
~i <i>variable</i>	Insert the value of the named variable into the text of the message. For example, ~A is equivalent to '~i Sign.' Environment variables set and exported in the shell are also accessible by ~i.

<code>~m [message-list]</code>	Insert the listed messages, or the current message being read into the letter. Valid only when sending a message while reading mail. The text of the message is shifted to the right, and the string contained in the <code>indentprefix</code> variable is inserted as the leftmost characters of each line. If <code>indentprefix</code> is not set, a TAB character is inserted into each line.
<code>~M [message-list]</code>	Insert the listed messages, or the current message being read, including the header fields, into the letter. Valid only when sending a message while reading mail. The text of the message is shifted to the right, and the string contained in the <code>indentprefix</code> variable is inserted as the leftmost characters of each line. If <code>indentprefix</code> is not set, a TAB character is inserted into each line. Overrides the suppression of fields by the <code>ignore</code> command.
<code>~p</code>	Print the message being entered.
<code>~q</code>	Quit from input mode by simulating an interrupt. If the body of the message is not null, the partial message is saved in <code>dead-letter</code> . See <code>DEAD</code> in <code>ENVIRONMENT VARIABLES</code> for a description of this file.
<code>~R</code>	Mark message for return receipt.
<code>~r file</code>	Read in the specified file. If the argument begins with an exclamation point (!), the rest of the string is taken as an arbitrary shell command and is executed, with the standard output inserted into the message.
<code>~< file</code>	
<code>~< ! shell-command</code>	
<code>~s string ...</code>	Set the subject line to <i>string</i> .
<code>~t name ...</code>	Add the given <i>names</i> to the To list.
<code>~v</code>	Invoke a preferred screen editor on the partial message. The default visual editor is <code>vi(1)</code> . See also <code>VISUAL</code> in <code>ENVIRONMENT VARIABLES</code> .
<code>~w file</code>	Write the message into the given file, without the header.
<code>~x</code>	Exit as with <code>~q</code> except the message is not saved in <code>dead-letter</code> .
<code>~ shell-command</code>	Pipe the body of the message through the given <i>shell-command</i> . If the <i>shell-command</i> returns a successful exit status, the output of the command replaces the message.

Internal Variables

The following variables are internal variables. They may be imported from the execution environment or set using the `set` command at any time. The `unset` command may be used to erase variables.

allnet	All network names whose last component (login name) match are treated as identical. This causes the <i>message-list</i> message specifications to behave similarly. Disabled by default. See also the <code>alternates</code> command and the <code>metoo</code> and <code>fuzzymatch</code> variables.
alwaysignore	Ignore header fields with <code>ignore</code> everywhere, not just during print or type. Affects the <code>save</code> , <code>Save</code> , <code>copy</code> , <code>Copy</code> , <code>top</code> , <code>pipe</code> , and <code>write</code> commands, and the <code>~m</code> and <code>~f</code> tilde escapes. Enabled by default.
append	Upon termination, append messages to the end of the <code>mbox</code> file instead of prepending them. Although disabled by default, <code>append</code> is set in the system startup file (which can be suppressed with the <code>-n</code> command line option).
appenddeadletter	Append to the deadletter file rather than overwrite it. Although disabled by default, <code>appenddeadletter</code> is frequently set in the system startup file. See <code>Starting Mail</code> in <code>USAGE</code> above.
askbcc	Prompt for the Bcc list after the Subject is entered if it is not specified on the command line with the <code>-b</code> option. Disabled by default.
askcc	Prompt for the Cc list after the Subject is entered if it is not specified on the command line with the <code>-c</code> option. Disabled by default.
asksub	Prompt for subject if it is not specified on the command line with the <code>-s</code> option. Enabled by default.
autoinc	Automatically incorporate new messages into the current session as they arrive. This has an affect similar to issuing the <code>inc</code> command every time the command prompt is displayed. Disabled by default, but <code>autoinc</code> is set in the default system startup file for <code>mailx</code> ; it is not set for <code>/usr/ucb/mail</code> or <code>/usr/ucb/Mail</code> .
autoprint	Enable automatic printing of messages after <code>delete</code> and <code>undelete</code> commands. Disabled by default.
bang	Enable the special-casing of exclamation points (!) in shell escape command lines as in <code>vi(1)</code> . Disabled by default.
bsdcompat	Set automatically if <code>mailx</code> is invoked as <code>mail</code> or <code>Mail</code> . Causes <code>mailx</code> to use <code>/etc/mail/Mail.rc</code> as the system startup file. Changes the default pager to <code>more(1)</code> .

<code>cmd=shell-command</code>	Set the default command for the pipe command. No default value.
<code>conv=conversion</code>	Convert uucp addresses to the specified address style, which can be either: <ul style="list-style-type: none"> <code>internet</code> This requires a mail delivery program conforming to the RFC822 standard for electronic mail addressing. <code>optimize</code> Remove loops in <code>uucp(1C)</code> address paths (typically generated by the reply command). No rerouting is performed; <code>mail</code> has no knowledge of UUCP routes or connections. <p>Conversion is disabled by default. See also <code>sendmail(1M)</code> and the <code>-U</code> command-line option.</p>
<code>crt[=number]</code>	Pipe messages having more than <i>number</i> lines through the command specified by the value of the PAGER variable (<code>pg(1)</code> or <code>more(1)</code> by default). If <i>number</i> is not specified, the current window size is used. Disabled by default.
<code>debug</code>	Enable verbose diagnostics for debugging. Messages are not delivered. Disabled by default.
<code>dot</code>	Take a period on a line by itself, or EOF during input from a terminal as end-of-file. Disabled by default, but <code>dot</code> is set in the system startup file (which can be suppressed with the <code>-n</code> command line option).
<code>fcc</code>	By default, <code>mailx</code> will treat any address containing a slash (/) character as a local send to file address. By unsetting this option, this behavior is disabled. Enabled by default.
<code>flipr</code>	Reverse the effect of the <code>followup/Followup</code> and <code>reply/Reply</code> command pairs. If both <code>flipr</code> and <code>replyall</code> are set, the effect is as if neither was set.
<code>from</code>	Extract the author listed in the header summary from the <code>From:</code> header instead of the UNIX <code>From</code> line. Enabled by default.
<code>fuzzymatch</code>	The <code>from</code> command searches for messages from the indicated sender. By default, the full sender address must be specified. By setting this option, only a sub-string of the sender address need be specified. Disabled by default.

escape=c	Substitute <i>c</i> for the ~ escape character. Takes effect with next message sent.
folder= <i>directory</i>	The directory for saving standard mail files. User-specified file names beginning with a plus (+) are expanded by preceding the file name with this directory name to obtain the real file name. If <i>directory</i> does not start with a slash (/), \$HOME is prepended to it. There is no default for the folder variable. See also out folder below.
header	Enable printing of the header summary when entering mailx. Enabled by default.
hold	Preserve all messages that are read in the mailbox instead of putting them in the standard mbox save file. Disabled by default.
ignore	Ignore interrupts while entering messages. Handy for noisy dial-up lines. Disabled by default.
ignoreeof	Ignore end-of-file during message input. Input must be terminated by a period (.) on a line by itself or by the ~. command. See also dot above. Disabled by default.
indentprefix= <i>string</i>	When indentprefix is set, <i>string</i> is used to mark indented lines from messages included with ~m. The default is a TAB character.
keep	When the mailbox is empty, truncate it to zero length instead of removing it. Disabled by default.
iprompt= <i>string</i>	The specified prompt string is displayed before each line on input is requested when sending a message.
keepsave	Keep messages that have been saved in other files in the mailbox instead of deleting them. Disabled by default.
makeremote	When replying to all recipients of a message, if an address does not include a machine name, it is assumed to be relative to the sender of the message. Normally not needed when dealing with hosts that support RFC822.
metoo	If your login appears as a recipient, do not delete it from the list. Disabled by default.
mustbang	Force all mail addresses to be in bang format.
onehop	When responding to a message that was originally sent to several recipients, the other recipient addresses are normally forced to be relative to the originating author's machine for the response. This flag disables alteration of the recipients'

addresses, improving efficiency in a network where all machines can send directly to all other machines (that is, one hop away). Disabled by default.

<code>outfolder</code>	Locate the files used to record outgoing messages in the directory specified by the <code>folder</code> variable unless the path name is absolute. Disabled by default. See <code>folder</code> above and the <code>Save</code> , <code>Copy</code> , <code>followup</code> , and <code>Followup</code> commands.
<code>page</code>	Used with the <code>pipe</code> command to insert a form feed after each message sent through the pipe. Disabled by default.
<code>pipeignore</code>	Omit ignored header when outputting to the <code>pipe</code> command. Although disabled by default, <code>pipeignore</code> is frequently set in the system startup file. See <code>Starting Mail</code> in <code>USAGE</code> above.
<code>postmark</code>	Your real name to be included in the <code>From</code> line of messages you send. By default this is derived from the <code>comment</code> field in your <code>passwd(4)</code> file entry.
<code>prompt=string</code>	Set the <i>command mode</i> prompt to <i>string</i> . Default is “?” , unless the <code>bsdcompat</code> variable is set, then the default is “&”.
<code>quiet</code>	Refrain from printing the opening message and version when entering <code>mailx</code> . Disabled by default.
<code>record=file</code>	Record all outgoing mail in <i>file</i> . Disabled by default. See also <code>outfolder</code> above.
<code>replyall</code>	Reverse the effect of the <code>reply</code> and <code>Reply</code> and <code>followup</code> and <code>Followup</code> commands. Although set by default, <code>replyall</code> is frequently unset in the system startup file. See <code>flipr</code> and <code>Starting Mail</code> in <code>USAGE</code> above.
<code>returnaddr=string</code>	The default sender address is that of the current user. This variable can be used to set the sender address to any arbitrary value. Set with caution.
<code>save</code>	Enable saving of messages in dead-letter on interrupt or delivery error. See <code>DEAD</code> for a description of this file. Enabled by default.
<code>screen=number</code>	Sets the number of lines in a screen-full of headers for the <code>headers</code> command. <i>number</i> must be a positive number.

The default is set according to baud rate or window size. With a baud rate less than 1200, *number* defaults to 5, if baud rate is

exactly 1200, it defaults to 10. If you are in a window, *number* defaults to the default window size minus 4. Otherwise, the default is 20.

<code>sendmail=shell-command</code>	Alternate command for delivering messages. <i>Note:</i> In addition to the expected list of recipients, <code>mail</code> also passes the <code>-i</code> and <code>-m</code> , flags to the command. Since these flags are not appropriate to other commands, you may have to use a shell script that strips them from the arguments list before invoking the desired command. Default is <code>/usr/bin/rmail</code> .
<code>sendwait</code>	Wait for background mailer to finish before returning. Disabled by default.
<code>showname</code>	Causes the message header display to show the sender's real name (if known) rather than their mail address. Disabled by default, but <code>showname</code> is set in the <code>/etc/mail/mailx.rc</code> system startup file for <code>mailx</code> .
<code>showto</code>	When displaying the header summary and the message is from you, print the recipient's name instead of the author's name.
<code>sign=string</code>	The variable inserted into the text of a message when the <code>~a</code> (autograph) command is given. No default (see also <code>~i</code> in Tilde Escapes).
<code>Sign=string</code>	The variable inserted into the text of a message when the <code>~A</code> command is given. No default (see also <code>~i</code> in Tilde Escapes).
<code>toplines=number</code>	The number of lines of header to print with the <code>top</code> command. Default is 5.
<code>verbose</code>	Invoke <code>sendmail(IM)</code> with the <code>-v</code> flag.
<code>translate</code>	The name of a program to translate mail addresses. The program receives mail addresses as arguments. The program produces, on the standard output, lines containing the following data, in this order: <ul style="list-style-type: none">▪ the postmark for the sender (see the <code>postmark</code> variable)▪ translated mail addresses, one per line, corresponding to the program's arguments. Each translated address will replace the corresponding address in the mail message being sent.▪ a line containing only <code>y</code> or <code>n</code>. if the line contains <code>y</code> the user will be asked to confirm that the message should be sent.

The translate program will be invoked for each mail message to be sent. If the program exits with a non-zero exit status, or fails to produce enough output, the message is not sent.

Large File Behavior See [largefile\(5\)](#) for the description of the behavior of mailx when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of mailx: HOME, LANG, LC_CTYPE, LC_TIME, LC_MESSAGES, NLSPATH, and TERM.

DEAD	The name of the file in which to save partial letters in case of untimely interrupt. Default is <code>\$HOME/dead.letter</code> .
EDITOR	The command to run when the edit or <code>~e</code> command is used. Default is ed(1) .
LISTER	The command (and options) to use when listing the contents of the folder directory. The default is ls(1) .
MAIL	The name of the initial mailbox file to read (in lieu of the standard system mailbox). The default is <code>/var/mail/username</code> .
MAILRC	The name of the startup file. Default is <code>\$HOME/.mailrc</code> .
MAILX_HEAD	The specified string is included at the beginning of the body of each message that is sent.
MAILX_TAIL	The specified string is included at the end of the body of each message that is sent.
MBOX	The name of the file to save messages which have been read. The exit command overrides this function, as does saving the message explicitly in another file. Default is <code>\$HOME/mbox</code> .
PAGER	The command to use as a filter for paginating output. This can also be used to specify the options to be used. Default is pg(1) , or if the <code>bsdcompat</code> variable is set, the default is more(1) . See Internal Variables .
SHELL	The name of a preferred command interpreter. Default is sh(1) .
VISUAL	The name of a preferred screen editor. Default is vi(1) .

退出状态 When the `-e` option is specified, the following exit values are returned:

- 0 Mail was found.
- >0 Mail was not found or an error occurred.

Otherwise, the following exit values are returned:

- 0 Successful completion. Notice that this status implies that all messages were *sent*, but it gives no assurances that any of them were actually *delivered*.

>0 An error occurred

文件

\$HOME/.mailrc	personal startup file
\$HOME/mbox	secondary storage file
\$HOME/.Maillock	lock file to prevent multiple writers of system mailbox
/etc/mail/mailx.rc	optional system startup file for mailx only
/etc/mail/Mail.rc	BSD compatibility system-wide startup file for /usr/ucb/mail and /usr/ucb/Mail
/tmp/R[emqsx]*	temporary files
/usr/share/lib/mailx/mailx.help*	help message files
/var/mail/*	post office directory

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[biff\(1B\)](#), [echo\(1\)](#), [ed\(1\)](#), [ex\(1\)](#), [fmt\(1\)](#), [ls\(1\)](#), [mail\(1\)](#), [mail\(1B\)](#), [mailcompat\(1\)](#), [more\(1\)](#), [pg\(1\)](#), [sh\(1\)](#), [uucp\(1C\)](#), [vacation\(1\)](#), [vi\(1\)](#), [newaliases\(1M\)](#), [sendmail\(1M\)](#), [aliases\(4\)](#), [passwd\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注

Where *shell-command* is shown as valid, arguments are not always allowed. Experimentation is recommended.

Internal variables imported from the execution environment cannot be unset.

The full internet addressing is not fully supported by mailx. The new standards need some time to settle down.

Replies do not always generate correct return addresses. Try resending the errant reply with `onehop set`.

mailx does not lock your record file. So, if you use a record file and send two or more messages simultaneously, lines from the messages may be interleaved in the record file.

The format for the `alias` command is a space-separated list of recipients, while the format for an alias in either the `.forward` or `/etc/aliases` is a comma-separated list.

To read mail on a workstation running Solaris 1.x when your mail server is running Solaris 2.x, first execute the `mailcompat(1)` program.

引用名

make – maintain, update, and regenerate related programs and files

用法概要

```
/usr/bin/make [-d] [-dd] [-D] [-DD] [-e] [-i] [-k] [-n]
               [-p] [-P] [-q] [-r] [-s] [-S] [-t] [-u][ -w] [-V]
               [-f makefile]... [-K statefile]... [target]...
               [macro = value...]

/usr/xpg4/bin/make [-d] [-dd] [-D] [-DD] [-e] [-i] [-k]
                   [-n] [-p] [-P] [-q] [-r] [-s] [-S] [-t] [-u][ -w] [-V]
                   [-f makefile]... [target]... [macro = value...]
```

描述

The make utility executes a list of shell commands associated with each *target*, typically to create or update a file of the same name. *makefile* contains entries that describe how to bring a target up to date with respect to those on which it depends, which are called *dependencies*. Since each dependency is a target, it can have dependencies of its own. Targets, dependencies, and sub-dependencies comprise a tree structure that make traces when deciding whether or not to rebuild a *target*.

The make utility recursively checks each *target* against its dependencies, beginning with the first target entry in *makefile* if no *target* argument is supplied on the command line. If, after processing all of its dependencies, a target file is found either to be missing, or to be older than any of its dependencies, make rebuilds it. Optionally with this version of make, a target can be treated as out-of-date when the commands used to generate it have changed since the last time the target was built.

To build a given target, make executes the list of commands, called a *rule*. This rule can be listed explicitly in the target's makefile entry, or it can be supplied implicitly by make.

If no *target* is specified on the command line, make uses the first target defined in *makefile*.

If a *target* has no makefile entry, or if its entry has no rule, make attempts to derive a rule by each of the following methods, in turn, until a suitable rule is found. Each method is described under [用法](#) below.

- Pattern matching rules.
- Implicit rules, read in from a user-supplied makefile.
- Standard implicit rules (also known as suffix rules), typically read in from the file `/usr/share/lib/make/make.rules`.
- SCCS retrieval. make retrieves the most recent version from the SCCS history file (if any). See the description of the `.SCCS_GET`: special-function target for details.
- The rule from the `.DEFAULT`: target entry, if there is such an entry in the makefile.

If there is no makefile entry for a *target*, if no rule can be derived for building it, and if no file by that name is present, make issues an error message and halts.

选项

The following options are supported:

- d Displays the reasons why `make` chooses to rebuild a target. `make` displays any and all dependencies that are newer. In addition, `make` displays options read in from the `MAKEFLAGS` environment variable.
- dd Displays the dependency check and processing in vast detail.
- D Displays the text of the makefiles read in.
- DD Displays the text of the makefiles, `make.rules` file, the state file, and all hidden-dependency reports.
- e Environment variables override assignments within makefiles.
- f *makefile* Uses the description file *makefile*. A – as the *makefile* argument denotes the standard input. The contents of *makefile*, when present, override the standard set of implicit rules and predefined macros. When more than one -f *makefile* argument pair appears, `make` uses the concatenation of those files, in order of appearance.

When no *makefile* is specified, `/usr/bin/make` tries the following in sequence, except when in POSIX mode (see `.POSIX` in 用法):

- If there is a file named `makefile` in the working directory, `make` uses that file. If, however, there is an SCCS history file (`SCCS/s.makefile`) which is newer, `make` attempts to retrieve and use the most recent version.
- In the absence of the above file(s), if a file named `Makefile` is present in the working directory, `make` attempts to use it. If there is an SCCS history file (`SCCS/s.Makefile`) that is newer, `make` attempts to retrieve and use the most recent version.

When no *makefile* is specified, `/usr/bin/make` in POSIX mode and `/usr/xpg4/bin/make` try the following files in sequence:

- `./makefile`, `./Makefile`
- `s.makefile`, `SCCS/s.makefile`
- `s.Makefile`, `SCCS/s.Makefile`

- i Ignores error codes returned by commands. Equivalent to the special-function target `.IGNORE:.`
- k When a nonzero error status is returned by a rule, or when `make` cannot find a rule, abandons work on the current target, but continues with other dependency branches that do not depend on it.
- K *statefile* Uses the state file *statefile*. A – as the *statefile* argument denotes the standard input. The contents of *statefile*, when present, override the standard set of implicit rules and predefined macros. When more than one -K *statefile*

argument pair appears, make uses the concatenation of those files, in order of appearance. (See also `.KEEP_STATE` and `.KEEP_STATE_FILE` in the [Special-Function Targets](#) section).

- n No execution mode. Prints commands, but does not execute them. Even lines beginning with an @ are printed. However, if a command line contains a reference to the `$(MAKE)` macro, that line is always executed (see the discussion of `MAKEFLAGS` in [Reading Makefiles and the Environment](#)). When in POSIX mode, lines beginning with a “+” are executed.
- p Prints out the complete set of macro definitions and target descriptions.
- P Merely reports dependencies, rather than building them.
- q Question mode. make returns a zero or nonzero status code depending on whether or not the target file is up to date. When in POSIX mode, lines beginning with a “+” are executed.
- r Does not read in the default makefile `/usr/share/lib/make/make.rules`.
- s Silent mode. Does not print command lines before executing them. Equivalent to the special-function target `.SILENT:`.
- S Undoes the effect of the `-k` option. Stops processing when a non-zero exit status is returned by a command.
- t Touches the target files (bringing them up to date) rather than performing their rules. *Warning:* This can be *dangerous* when files are maintained by more than one person. When the `.KEEP_STATE:` target appears in the makefile, this option updates the state file just as if the rules had been performed. When in POSIX mode, lines beginning with a “+” are executed.
- u Unconditional build of targets. Even if a target is up to date, it is rebuilt. This might be useful for rebuilding all targets without cleaning.
- V Puts make into SysV mode. Refer to [sysv-make\(1\)](#) for respective details.
- w Print a message containing the working directory before and after other processing. This can be useful for tracking down errors from complicated nests of recursive make commands.
- x Puts make into the specified compatibility mode. The following compatibility modes are supported:
 1. Compatibility with POSIX:
 - x `SUN_MAKE_COMPAT_MODE=POSIX`
 2. Compatibility with SUN make:
 - x `SUN_MAKE_COMPAT_MODE=SUN`

3. Compatibility with GNU make (partially supported):

-x SUN_MAKE_COMPAT_MODE=GNU

4. Compatibility with `/usr/lib/svr4.make`:

-x SUN_MAKE_COMPAT_MODE=SVR4

操作数

The following operands are supported:

target Target names, as defined in [用法](#).

macro=value Macro definition. This definition overrides any regular definition for the specified macro within the makefile itself, or in the environment. However, this definition can still be overridden by conditional macro assignments.

用法

The usage of make is described below:

Reading Makefiles and the Environment

When make first starts, it reads the MAKEFLAGS environment variable to obtain any of the following options specified present in its value: -d, -D, -e, -i, -k, -n, -p, -q, -r, -s, -S, or -t. Due to the implementation of POSIX.2 (see [POSIX.2\(5\)](#)), the MAKEFLAGS values contains a leading - character. The make utility then reads the command line for additional options, which also take effect.

Next, make reads in a default makefile that typically contains predefined macro definitions, target entries for implicit rules, and additional rules, such as the rule for retrieving SCCS files. If present, make uses the file `make.rules` in the current directory; otherwise it reads the file `/usr/share/lib/make/make.rules`, which contains the standard definitions and rules. Use the directive:

```
include /usr/share/lib/make/make.rules
```

in your local `make.rules` file to include them.

Next, make imports variables from the environment (unless the -e option is in effect), and treats them as defined macros. Because make uses the most recent definition it encounters, a macro definition in the makefile normally overrides an environment variable of the same name. When -e is in effect, however, environment variables are read *after* all makefiles have been read. In that case, the environment variables take precedence over definitions in the makefile.

Next, make reads any makefiles you specify with -f, or one of `makefile` or `Makefile` as described above and then the state file, in the local directory if it exists. If the makefile contains a `.KEEP_STATE_FILE` target, then it reads the state file that follows the target. Refer to special target `.KEEP_STATE_FILE` for details.

Next (after reading the environment if -e is in effect), make reads in any macro definitions supplied as command line arguments. These override macro definitions in the makefile and the environment both, but only for the make command itself.

make exports environment variables, using the most recently defined value. Macro definitions supplied on the command line are not normally exported, unless the macro is also an environment variable.

make does not export macros defined in the makefile. If an environment variable is set, and a macro with the same name is defined on the command line, make exports its value as defined on the command line. Unless -e is in effect, macro definitions within the makefile take precedence over those imported from the environment.

The macros MAKEFLAGS, MAKE, SHELL, HOST_ARCH, HOST_MACH, and TARGET_MACH are special cases. See [Special-Purpose Macros](#) below for details.

Makefile Target Entries A target entry has the following format:

```
target [:::] [dependency] ... [; command] ... [command] ...
```

The first line contains the name of a target, or a space-separated list of target names, terminated with a colon or double colon. If a list of targets is given, this is equivalent to having a separate entry of the same form for each target. The colon(s) can be followed by a *dependency*, or a dependency list. make checks this list before building the target. The dependency list can be terminated with a semicolon (;), which in turn can be followed by a single Bourne shell command. Subsequent lines in the target entry begin with a TAB and contain Bourne shell commands. These commands comprise the rule for building the target.

Shell commands can be continued across input lines by escaping the NEWLINE with a backslash (\). The continuing line must also start with a TAB.

To rebuild a target, make expands macros, strips off initial TAB characters and either executes the command directly (if it contains no shell metacharacters), or passes each command line to a Bourne shell for execution.

The first *non-empty* line that does not begin with a TAB or # begins another target or macro definition.

Special Characters Special characters are defined below.

<i>Global</i>	#	Start a comment. The comment ends at the next NEWLINE. If the # follows the TAB in a command line, that line is passed to the shell (which also treats # as the start of a comment).
	include <i>filename</i>	If the word include appears as the first seven letters of a line and is followed by a SPACE or TAB, the string that follows is taken as a filename to interpolate at that line. include files can be nested to a depth of no more than about 16. If <i>filename</i> is a macro reference, it is expanded. If <i>filename</i> is surrounded by double quotes, make searches for a <i>filename</i> with relation to current

		makefile path. If not, <code>make</code> is supposed to find it with relation to <i>path</i> where <code>make</code> was launched.
<i>Targets and Dependencies</i>	:	Target list terminator. Words following the colon are added to the dependency list for the target or targets. If a target is named in more than one colon-terminated target entry, the dependencies for all its entries are added to form that target's complete dependency list.
	::	Target terminator for alternate dependencies. When used in place of a <code>:</code> the double-colon allows a target to be checked and updated with respect to alternate dependency lists. When the target is out-of-date with respect to dependencies listed in the first alternate, it is built according to the rule for that entry. When out-of-date with respect to dependencies in another alternate, it is built according the rule in that other entry. Implicit rules do not apply to double-colon targets; you must supply a rule for each entry. If no dependencies are specified, the rule is always performed.
	<i>target</i> [+ <i>target</i> . . .] :	Target group. The rule in the target entry builds all the indicated targets as a group. It is normally performed only once per <code>make</code> run, but is checked for command dependencies every time a target in the group is encountered in the dependency scan.
	%	Pattern matching wild card metacharacter. Like the <code>*</code> shell wild card, <code>%</code> matches any string of zero or more characters in a target name or dependency, in the target portion of a conditional macro definition, or within a pattern replacement macro reference. Notice that only one <code>%</code> can appear in a target, dependency-name, or pattern-replacement macro reference.
	<i>./pathname</i>	<code>make</code> ignores the leading <code>./</code> characters from targets with names given as pathnames relative to “dot,” the working directory.
<i>Macros</i>	=	Macro definition. The word to the left of this character is the macro name; words to the right comprise its value. Leading and trailing white space characters are stripped from the value. A word break following the <code>=</code> is implied.
	\$	Macro reference. The following character, or the parenthesized or bracketed string, is interpreted as a macro reference: <code>make</code> expands the reference (including the <code>\$</code>) by replacing it with the macro's value.
	()	
	{ }	Macro-reference name delimiters. A parenthesized or bracketed word appended to a <code>\$</code> is taken as the name of the macro being referred to. Without the delimiters, <code>make</code> recognizes only the first character as the macro name.
	\$\$	A reference to the dollar-sign macro, the value of which is the character <code>\$</code> . Used to pass variable expressions beginning with <code>\$</code> to the shell, to refer to environment

variables which are expanded by the shell, or to delay processing of dynamic macros within the dependency list of a target, until that target is actually processed.

- \\$ Escaped dollar-sign character. Interpreted as a literal dollar sign within a rule.
- += When used in place of =, appends a string to a macro definition (must be surrounded by white space, unlike =).
- := Conditional macro assignment. When preceded by a list of targets with explicit target entries, the macro definition that follows takes effect when processing only those targets, and their dependencies.
- :sh = Define the value of a macro to be the output of a command (see [Command Substitutions](#) below).
- :sh In a macro reference, execute the command stored in the macro, and replace the reference with the output of that command (see [Command Substitutions](#) below).

- Rules*
- + make always executes the commands preceded by a “+”, even when -n is specified.
 - make ignores any nonzero error code returned by a command line for which the first non-TAB character is a -. This character is not passed to the shell as part of the command line. make normally terminates when a command returns nonzero status, unless the -i or -k options, or the .IGNORE: special-function target is in effect.
 - @ If the first non-TAB character is a @, make does not print the command line before executing it. This character is not passed to the shell.
 - ? Escape command-dependency checking. Command lines starting with this character are not subject to command dependency checking.
 - ! Force command-dependency checking. Command-dependency checking is applied to command lines for which it would otherwise be suppressed. This checking is normally suppressed for lines that contain references to the ? dynamic macro (for example, \$?).
- When any combination of +, -, @, ?, or ! appear as the first characters after the TAB, all that are present apply. None are passed to the shell.

Special-Function Targets

When incorporated in a makefile, the following target names perform special-functions:

- .DEFAULT: If it has an entry in the makefile, the rule for this target is used to process a target when there is no other entry for it, no rule for building it, and no SCCS history file from which to retrieve a current version. make ignores any dependencies for this target.
- .DONE: If defined in the makefile, make processes this target and its dependencies after all other targets are built. This target is also performed when make halts with an error, unless the .FAILED target is defined.

<code>.FAILED:</code>	This target, along with its dependencies, is performed instead of <code>.DONE</code> when defined in the makefile and <code>make</code> halts with an error.
<code>.GET_POSIX:</code>	This target contains the rule for retrieving the current version of an SCCS file from its history file in the current working directory. <code>make</code> uses this rule when it is running in POSIX mode.
<code>.IGNORE:</code>	Ignore errors. When this target appears in the makefile, <code>make</code> ignores non-zero error codes returned from commands. When used in POSIX mode, <code>.IGNORE</code> could be followed by target names only, for which the errors is ignored.
<code>.INIT:</code>	If defined in the makefile, this target and its dependencies are built before any other targets are processed.
<code>.KEEP_STATE:</code>	If this target is in effect, <code>make</code> updates the state file, <code>.make.state</code> , in the current directory. This target also activates command dependencies, and hidden dependency checks. If either the <code>.KEEP_STATE:</code> target appears in the makefile, or the environment variable <code>KEEP_STATE</code> is set (<code>setenv KEEP_STATE</code>), <code>make</code> rebuilds everything in order to collect dependency information, even if all the targets were up to date due to previous <code>make</code> runs. See also the 环境变量 section. This target has no effect if used in POSIX mode.
<code>.KEEP_STATE_FILE:</code>	This target has no effect if used in POSIX mode. This target implies <code>.KEEP_STATE</code> . If the target is followed by a filename, <code>make</code> uses it as the state file. If the target is followed by a directory name, <code>make</code> looks for a <code>.make.state</code> file in that directory. If the target is not followed by any name, <code>make</code> looks for <code>.make.state</code> file in the current working directory.
<code>.MAKE_VERSION:</code>	A target-entry of the form: <code>.MAKE_VERSION: VERSION-number</code> enables version checking. If the version of <code>make</code> differs from the version indicated by a string like <code>VERSION-1.0</code> , <code>make</code> issues a warning message.
<code>.NO_PARALLEL:</code>	Currently, this target has no effect, it is, however, reserved for future use.
<code>.PARALLEL:</code>	Currently of no effect, but reserved for future use.
<code>.POSIX:</code>	This target enables POSIX mode.
<code>.PRECIOUS:</code>	List of files not to delete. <code>make</code> does not remove any of the files listed as dependencies for this target when interrupted. <code>make</code> normally

removes the current target when it receives an interrupt. When used in POSIX mode, if the target is not followed by a list of files, all the file are assumed precious.

- `.SCCS_GET`: This target contains the rule for retrieving the current version of an SCCS file from its history file. To suppress automatic retrieval, add an entry for this target with an empty rule to your makefile.
- `.SCCS_GET_POSIX`: This target contains the rule for retrieving the current version of an SCCS file from its history file. `make` uses this rule when it is running in POSIX mode.
- `.SILENT`: Run silently. When this target appears in the makefile, `make` does not echo commands before executing them. When used in POSIX mode, it could be followed by target names, and only those are executed silently.
- `.SUFFIXES`: The suffixes list for selecting implicit rules (see [The Suffixes List](#)).
- `.WAIT`: Currently of no effect, but reserved for future use.

Clearing Special Targets

In this version of `make`, you can clear the definition of the following special targets by supplying entries for them with no dependencies and no rule:

`.DEFAULT`, `.SCCS_GET`, and `.SUFFIXES`

Command Dependencies

When the `.KEEP_STATE`: target is effective, `make` checks the command for building a target against the state file. If the command has changed since the last `make` run, `make` rebuilds the target.

Hidden Dependencies

When the `.KEEP_STATE`: target is effective, `make` reads reports from [cpp\(1\)](#) and other compilation processors for any “hidden” files, such as `#include` files. If the target is out of date with respect to any of these files, `make` rebuilds it.

Macros

Entries of the form

macro=value

define macros. *macro* is the name of the macro, and *value*, which consists of all characters up to a comment character or unescaped `NEWLINE`, is the value. `make` strips both leading and trailing white space in accepting the value.

Subsequent references to the macro, of the forms: `$(name)` or `${name}` are replaced by *value*. The parentheses or brackets can be omitted in a reference to a macro with a single-character name.

Macro references can contain references to other macros, in which case nested references are expanded first.

*Suffix Replacement
Macro References*

Substitutions within macros can be made as follows:

```
$(name:string1=string2)
```

where *string1* is either a suffix, or a word to be replaced in the macro definition, and *string2* is the replacement suffix or word. Words in a macro value are separated by SPACE, TAB, and escaped NEWLINE characters.

*Pattern Replacement
Macro References*

Pattern matching replacements can also be applied to macros, with a reference of the form:

```
$(name: op%os= np%ns)
```

where *op* is the existing (old) prefix and *os* is the existing (old) suffix, *np* and *ns* are the new prefix and new suffix, respectively, and the pattern matched by % (a string of zero or more characters), is carried forward from the value being replaced. For example:

```
PROGRAM=fabricate  
DEBUG= $(PROGRAM:%=tmp/%-g)
```

sets the value of DEBUG to tmp/fabricate-g.

Notice that pattern replacement macro references cannot be used in the dependency list of a pattern matching rule; the % characters are not evaluated independently. Also, any number of % metacharacters can appear after the equal-sign.

Appending to a Macro

Words can be appended to macro values as follows:

```
macro += word . . .
```

*Special-Purpose
Macros*

When the MAKEFLAGS variable is present in the environment, make takes options from it, in combination with options entered on the command line. make retains this combined value as the MAKEFLAGS macro, and exports it automatically to each command or shell it invokes.

Notice that flags passed by way of MAKEFLAGS are only displayed when the -d, or -dd options are in effect.

The MAKE macro is another special case. It has the value make by default, and temporarily overrides the -n option for any line in which it is referred to. This allows nested invocations of make written as:

```
$(MAKE) . . .
```

to run recursively, with the -n flag in effect for all commands but make. This lets you use make -n to test an entire hierarchy of makefiles.

For compatibility with the 4.2 BSD make, the MFLAGS macro is set from the MAKEFLAGS variable by prepending a -. MFLAGS is not exported automatically.

The SHELL macro, when set to a single-word value such as `/usr/bin/csh`, indicates the name of an alternate shell to use. The default is `/bin/sh`. Notice that `make` executes commands that contain no shell metacharacters itself. Built-in commands, such as `dirs` in the C shell, are not recognized unless the command line includes a metacharacter (for instance, a semicolon). This macro is neither imported from, nor exported to the environment, regardless of `-e`. To be sure it is set properly, you must define this macro within every makefile that requires it.

The syntax of the VPATH macro is:

```
VPATH = [ pathname [ : pathname ] ... ]
```

VPATH specifies a list of directories to search for the files, which are targets or dependencies, when `make` is executed. VPATH is also used in order to search for the `include` files mentioned in the particular makefile.

When processing a target or a dependency or an include directive, `make` checks the existence of the file with the same name in the current directory. If the file is found to be missing, `make` searches for this file in the list of directories presented in VPATH (like the PATH variable in the shell). Unlike the PATH variable, VPATH is used in order to search for the files with relative pathnames. When `make` attempts to apply implicit rules to the target, it also searches for the dependency files using VPATH.

When the file is found using VPATH, internal macros `$(D)`, `$(F)`, `$(?)`, `$(*)`, and their alternative forms (with D or F appended) are set in accordance with the name derived from VPATH. For instance, if the target `subdir/foo.o` is found in the directory `/aaa/bbb` using VPATH, then the value of the internal macro `$(@)` for this target is `/aaa/bbb/subdir/foo.o`.

If a target or a dependency file is found using VPATH, then any occurrences of the word that is the same as the target name in the subsequent rules are replaced with the actual name of the target derived from VPATH.

For example:

```
VPATH=./subdir
file.o : file.c
        cc -c file.c -o file.o
```

If `file.c` is found in `./subdir`, then the command

```
cc -c ./subdir/file.c -o file.o
```

are executed.

The following macros are provided for use with cross-compilation:

HOST_ARCH	The processor type of the host system. By default, this is the output of the <code>mach(1)</code> command, prepended with <code>-</code> . Under normal circumstances, this value should never be altered by the user.
-----------	--

HOST_MACH	The machine architecture of the host system. By default, this is the output of the <code>arch(1)</code> command, prepended with <code>-</code> . Under normal circumstances, this value should never be altered by the user.
TARGET_ARCH	The processor type of the target system. By default, the output of <code>mach</code> , prepended with <code>-</code> .

Dynamic Macros

There are several dynamically maintained macros that are useful as abbreviations within rules. They are shown here as references; if you were to define them, `make` would simply override the definition.

<code>\$(*)</code>	The basename of the current target, derived as if selected for use with an implicit rule.
<code>\$(<)</code>	The name of a dependency file, derived as if selected for use with an implicit rule.
<code>\$(@)</code>	The name of the current target. This is the only dynamic macro whose value is strictly determined when used in a dependency list. (In which case it takes the form <code>\$\$@</code> .)
<code>\$(?)</code>	The list of dependencies that are newer than the target. Command-dependency checking is automatically suppressed for lines that contain this macro, just as if the command had been prefixed with a <code>?</code> . See the description of <code>?</code> , under Special Character Rules above. You can force this check with the <code>!</code> command-line prefix.
<code>\$(%)</code>	The name of the library member being processed. (See Library Maintenance below.)

To refer to the `$(@)` dynamic macro within a dependency list, precede the reference with an additional `$` character (as in, `$$@`). Because `make` assigns `$(<)` and `$(*)` as it would for implicit rules (according to the suffixes list and the directory contents), they can be unreliable when used within explicit target entries.

These macros can be modified to apply either to the filename part, or the directory part of the strings they stand for, by adding an upper case `F` or `D`, respectively (and enclosing the resulting name in parentheses or braces). Thus, `$(@D)` refers to the directory part of the string `$(@)`; if there is no directory part, `.` is assigned. `$(@F)` refers to the filename part.

Conditional Macro Definitions

A macro definition of the form:

```
target-list := macro = value
```

indicates that when processing any of the targets listed *and their dependencies*, *macro* is to be set to the *value* supplied. Notice that if a conditional macro is referred to in a dependency list, the `$` must be delayed (use `$$` instead). Also, *target-list* can contain a `%` pattern, in which case the macro is conditionally defined for all targets encountered that match the pattern. A pattern replacement reference can be used within the *value*.

You can temporarily append to a macros value with a conditional definition of the form:

```
target-list := macro += value
```

Predefined Macros

make supplies the macros shown in the table that follows for compilers and their options, host architectures, and other commands. Unless these macros are read in as environment variables, their values are not exported by make. If you run make with any of these set in the environment, it is a good idea to add commentary to the makefile to indicate what value each is expected to take. If `-r` is in effect, make does not read the default makefile (`./make.rules` or `/usr/share/lib/make/make.rules`) in which these macro definitions are supplied.

<i>Table of Predefined Macros</i>		
<i>Use</i>	<i>Macro</i>	<i>Default Value</i>
Library	AR	ar
Archives	ARFLAGS	rv
Assembler	AS	as
Commands	ASFLAGS	
	COMPILE.s	\$(AS) \$(ASFLAGS)
	COMPILE.S	\$(CC) \$(ASFLAGS) \$(CPPFLAGS) -c
C	CC	cc
Compiler	CFLAGS	
Commands	CPPFLAGS	
	COMPILE.c	\$(CC) \$(CFLAGS) \$(CPPFLAGS) -c
	LINK.c	\$(CC) \$(CFLAGS) \$(CPPFLAGS) \$(LDFLAGS)
C++	CCC	CC
Compiler	CCFLAGS	CFLAGS
Commands	CPPFLAGS	
	COMPILE.cc	\$(CCC) \$(CCFLAGS) \$(CPPFLAGS) -c
	LINK.cc	\$(CCC) \$(CCFLAGS) \$(CPPFLAGS) \$(LDFLAGS)
	COMPILE.C	\$(CCC) \$(CCFLAGS) \$(CPPFLAGS) -c
	LINK.C	\$(CCC) \$(CCFLAGS) \$(CPPFLAGS) \$(LDFLAGS)

<i>Table of Predefined Macros</i>		
<i>Use</i>	<i>Macro</i>	<i>Default Value</i>
FORTRAN 77 Compiler Commands	FC FFLAGS COMPILE.f LINK.f COMPILE.F LINK.F	f77 \$(FC) \$(FFLAGS) -c \$(FC) \$(FFLAGS) \$(LDFLAGS) \$(FC) \$(FFLAGS) \$(CPPFLAGS) -c \$(FC) \$(FFLAGS) \$(CPPFLAGS) \$(LDFLAGS)
FORTRAN 90 Compiler Commands	FC F90FLAGS COMPILE.f90 LINK.f90 COMPILE.ftn LINK.ftn	f90 \$(F90C) \$(F90FLAGS) -c \$(F90C) \$(F90FLAGS) \$(LDFLAGS) \$(F90C) \$(F90FLAGS) \$(CPPFLAGS) -c \$(F90C) \$(F90FLAGS) \$(CPPFLAGS) \$(LDFLAGS)
Link Editor Command	LD LDFLAGS	ld
lex Command	LEX LFLAGS LEX.l	lex \$(LEX) \$(LFLAGS) -t
lint Command	LINT LINTFLAGS LINT.c	lint \$(LINT) \$(LINTFLAGS) \$(CPPFLAGS)
Modula 2 Commands	M2C M2FLAGS	m2c

<i>Table of Predefined Macros</i>		
<i>Use</i>	<i>Macro</i>	<i>Default Value</i>
	MODFLAGS DEFFLAGS COMPILE.def COMPILE.mod	\$(M2C) \$(M2FLAGS) \$(DEFFLAGS) \$(M2C) \$(M2FLAGS) \$(MODFLAGS)
Pascal Compiler Commands	PC PFLAGS COMPILE.p LINK.p	pc \$(PC) \$(PFLAGS) \$(CPPFLAGS) -c \$(PC) \$(PFLAGS) \$(CPPFLAGS) \$(LDFLAGS)
Ratfor Compilation Commands	RFLAGS COMPILE.r LINK.r	\$(FC) \$(FFLAGS) \$(RFLAGS) -c \$(FC) \$(FFLAGS) \$(RFLAGS) \$(LDFLAGS)
rm Command	RM	rm -f
sccs Command	SCCSFLAGS SCCSGETFLAGS	-s
yacc Command	YACC YFLAGS YACC.y	yacc \$(YACC) \$(YFLAGS)
Suffixes List	SUFFIXES	.o .c .c~ .cc .cc~ .y .y~ .l .l~ .s .s~ .sh .sh~ .S .S~ .ln .h .h~ .f .f~ .F .F~ .mod .mod~ .sym .def .def~ .p .p~ .r .r~ .cps .cps~ .C .C~ .Y .Y~ .L .L .f90 .f90~ .ftn .ftn~

Implicit Rules

When a target has no entry in the makefile, `make` attempts to determine its class (if any) and apply the rule for that class. An implicit rule describes how to build any target of a given class, from an associated dependency file. The class of a target can be determined either by a pattern, or by a suffix; the corresponding dependency file (with the same basename) from which such a target might be built. In addition to a predefined set of implicit rules, `make` allows you to define your own, either by pattern, or by suffix.

Pattern Matching Rules

A target entry of the form:

```
tp%ts: dp%ds
      rule
```

is a pattern matching rule, in which *tp* is a target prefix, *ts* is a target suffix, *dp* is a dependency prefix, and *ds* is a dependency suffix (any of which can be null). The % stands for a basename of zero or more characters that is matched in the target, and is used to construct the name of a dependency. When `make` encounters a match in its search for an implicit rule, it uses the rule in that target entry to build the target from the dependency file. Pattern-matching implicit rules typically make use of the `$(` and `$(` dynamic macros as placeholders for the target and dependency names. Other, regular dependencies can occur in the dependency list; however, none of the regular dependencies can contain %. An entry of the form:

```
tp%ts: [dependency... ] dp%ds [dependency... ]
      rule
```

is a valid pattern matching rule.

Suffix Rules

When no pattern matching rule applies, `make` checks the target name to see if it ends with a suffix in the known suffixes list. If so, `make` checks for any suffix rules, as well as a dependency file with same root and another recognized suffix, from which to build it.

The target entry for a suffix rule takes the form:

```
DsTs: rule
```

where *Ts* is the suffix of the target, *Ds* is the suffix of the dependency file, and *rule* is the rule for building a target in the class. Both *Ds* and *Ts* must appear in the suffixes list. (A suffix need not begin with a . to be recognized.)

A suffix rule with only one suffix describes how to build a target having a null (or no) suffix from a dependency file with the indicated suffix. For instance, the `.c` rule could be used to build an executable program named `file` from a C source file named `file.c`. If a target with a null suffix has an explicit dependency, `make` omits the search for a suffix rule.

Table of Standard Implicit (Suffix) Rules for Assembly Files

Implicit Rule Name	Command Line
<code>.s.o</code>	<code>\$(COMPILE.s) -o \$@ \$<</code>

<i>Table of Standard Implicit (Suffix) Rules for Assembly Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.s.a</code>	\$(COMPILE.s) -o \$\$% \$< \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%
<code>.s~.o</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.s \$(COMPILE.s) -o \$@ \$*.s
<code>.S.o</code>	\$(COMPILE.S) -o \$@ \$<
<code>.S.a</code>	\$(COMPILE.S) -o \$\$% \$< \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%
<code>.S~.o</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.S \$(COMPILE.S) -o \$@ \$*.S
<code>.S~.a</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.S \$(COMPILE.S) -o \$\$% \$*.S \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%

<i>Table of Standard Implicit (Suffix) Rules for C Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.c</code>	\$(LINK.c) -o \$@ \$< \$(LDLIBS)
<code>.c.ln</code>	\$(LINT.c) \$(OUTPUT_OPTION) -i \$<

<i>Table of Standard Implicit (Suffix) Rules for C Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.c.o</code>	<code>\$(COMPILE.c) \$(OUTPUT_OPTION) \$<</code>
<code>.c.a</code>	<code>\$(COMPILE.c) -o \$% \$<</code> <code>\$(AR) \$(ARFLAGS) \$@ \$%</code> <code>\$(RM) \$%</code>
<code>.c~</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.c</code> <code>\$(CC) \$(CFLAGS) \$(LDFLAGS) -o \$@ \$*.c</code>
<code>.c~.o</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.c</code> <code>\$(CC) \$(CFLAGS) -c \$*.c</code>
<code>.c~.ln</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.c</code> <code>\$(LINT.c) \$(OUTPUT_OPTION) -c \$*.c</code>
<code>.c~.a</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.c</code> <code>\$(COMPILE.c) -o \$% \$*.c</code> <code>\$(AR) \$(ARFLAGS) \$@ \$%</code> <code>\$(RM) \$%</code>

<i>Table of Standard Implicit (Suffix) Rules for C++ Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.cc</code>	<code>\$(LINK.cc) -o \$@ \$< \$(LDLIBS)</code>
<code>.cc.o</code>	<code>\$(COMPILE.cc) \$(OUTPUT_OPTION) \$<</code>

<i>Table of Standard Implicit (Suffix) Rules for C++ Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.cc.a</code>	\$(COMPILE.cc) -o \$% \$< \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%
<code>.cc~</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.cc \$(LINK.cc) -o \$@ \$*.cc \$(LDLIBS)
<code>.cc.o</code>	\$(COMPILE.cc) \$(OUTPUT_OPTION) \$<
<code>.cc~.o</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.cc \$(COMPILE.cc) \$(OUTPUT_OPTION) \$*.cc
<code>.cc.a</code>	\$(COMPILE.cc) -o \$% \$< \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%
<code>.cc~.a</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.cc \$(COMPILE.cc) -o \$% \$*.cc \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%
<code>.C</code>	\$(LINK.C) -o \$@ \$< \$(LDLIBS)
<code>.C~</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.C \$(LINK.C) -o \$@ \$*.C \$(LDLIBS)
<code>.C.o</code>	\$(COMPILE.C) \$(OUTPUT_OPTION) \$<

<i>Table of Standard Implicit (Suffix) Rules for C++ Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.C~.o</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.C \$(COMPILE.C) \$(OUTPUT_OPTION) \$*.C
<code>.C.a</code>	\$(COMPILE.C) -o \$% \$< \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%
<code>.C~.a</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.C \$(COMPILE.C) -o \$% \$*.C \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%

<i>Table of Standard Implicit (Suffix) Rules for FORTRAN 77 Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.f</code>	\$(LINK.f) -o \$@ \$< \$(LDLIBS)
<code>.f.o</code>	\$(COMPILE.f) \$(OUTPUT_OPTION) \$<
<code>.f.a</code>	\$(COMPILE.f) -o \$% \$< \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%
<code>.f</code>	\$(LINK.f) -o \$@ \$< \$(LDLIBS)
<code>.f~</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.f \$(FC) \$(FFLAGS) \$(LDFFLAGS) -o \$@ \$*.f

<i>Table of Standard Implicit (Suffix) Rules for FORTRAN 77 Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
.f~.o	\$(GET) \$(GFLAGS) -p \$< > \$*.f \$(FC) \$(FFLAGS) -c \$*.f
.f~.a	\$(GET) \$(GFLAGS) -p \$< > \$*.f \$(COMPILE.f) -o %% \$*.f \$(AR) \$(ARFLAGS) @\$@ %% \$(RM) %%
.F	\$(LINK.F) -o @\$ \$< \$(LDLIBS)
.F.o	\$(COMPILE.F) \$(OUTPUT_OPTION) \$<
.F.a	\$(COMPILE.F) -o %% \$< \$(AR) \$(ARFLAGS) @\$@ %% \$(RM) %%
.F~	\$(GET) \$(GFLAGS) -p \$< > \$*.F \$(FC) \$(FFLAGS) \$(LD_FLAGS) -o @\$ \$*.F
.F~.o	\$(GET) \$(GFLAGS) -p \$< > \$*.F \$(FC) \$(FFLAGS) -c \$*.F
.F~.a	\$(GET) \$(GFLAGS) -p \$< > \$*.F \$(COMPILE.F) -o %% \$*.F \$(AR) \$(ARFLAGS) @\$@ %% \$(RM) %%

<i>Table of Standard Implicit (Suffix) Rules for FORTRAN 90 Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.f90</code>	<code>\$(LINK.f90) -o \$@ \$< \$(LDLIBS)</code>
<code>.f90~</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.f90</code> <code>\$(LINK.f90) -o \$@ \$*.f90 \$(LDLIBS)</code>
<code>.f90.o</code>	<code>\$(COMPILE.f90) \$(OUTPUT_OPTION) \$<</code>
<code>.f90~.o</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.f90</code> <code>\$(COMPILE.f90) \$(OUTPUT_OPTION) \$*.f90</code>
<code>.f90.a</code>	<code>\$(COMPILE.f90) -o \$% \$<</code> <code>\$(AR) \$(ARFLAGS) \$@ \$%</code> <code>\$(RM) \$%</code>
<code>.f90~.a</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.f90</code> <code>\$(COMPILE.f90) -o \$% \$*.f90</code> <code>\$(AR) \$(ARFLAGS) \$@ \$%</code> <code>\$(RM) \$%</code>
<code>.ftn</code>	<code>\$(LINK.ftn) -o \$@ \$< \$(LDLIBS)</code>
<code>.ftn~</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.ftn</code> <code>\$(LINK.ftn) -o \$@ \$*.ftn \$(LDLIBS)</code>
<code>.ftn.o</code>	<code>\$(COMPILE.ftn) \$(OUTPUT_OPTION) \$<</code>
<code>.ftn~.o</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.ftn</code>

<i>Table of Standard Implicit (Suffix) Rules for FORTRAN 90 Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
	<code>\$(COMPILE.ftn) \$(OUTPUT_OPTION) \$*.ftn</code>
<code>.ftn.a</code>	<code>\$(COMPILE.ftn) -o \$% \$<</code> <code>\$(AR) \$(ARFLAGS) \$@ \$%</code> <code>\$(RM) \$%</code>
<code>.ftn~.a</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.ftn</code> <code>\$(COMPILE.ftn) -o \$% \$*.ftn</code> <code>\$(AR) \$(ARFLAGS) \$@ \$%</code> <code>\$(RM) \$%</code>

<i>Table of Standard Implicit (Suffix) Rules for lex Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.l</code>	<code>\$(RM) \$*.c</code> <code>\$(LEX.l) \$< > \$*.c</code> <code>\$(LINK.c) -o \$@ \$*.c \$(LDLIBS)</code> <code>\$(RM) \$*.c</code>
<code>.l.c</code>	<code>\$(RM) \$@</code> <code>\$(LEX.l) \$< > \$@</code>
<code>.l.ln</code>	<code>\$(RM) \$*.c</code> <code>\$(LEX.l) \$< > \$*.c</code> <code>\$(LINT.c) -o \$@ -i \$*.c</code> <code>\$(RM) \$*.c</code>
<code>.l.o</code>	<code>\$(RM) \$*.c</code>

<i>Table of Standard Implicit (Suffix) Rules for lex Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
	\$(LEX.l) \$< > \$*.c \$(COMPILE.c) -o \$@\$*.c \$(RM) \$*.c
.l~	\$(GET) \$(GFLAGS) -p \$< > \$*.l \$(LEX) \$(LFLAGS) \$*.l \$(CC) \$(CFLAGS) -c lex.yy.c rm -f lex.yy.c mv lex.yy.c \$@
.l~.c	\$(GET) \$(GFLAGS) -p \$< > \$*.l \$(LEX) \$(LFLAGS) \$*.l mv lex.yy.c \$@
.l~.ln	\$(GET) \$(GFLAGS) -p \$< > \$*.l \$(RM) \$*.c \$(LEX.l) \$*.l > \$*.c \$(LINT.c) -o \$@ -i \$*.c \$(RM) \$*.c
.l~.o	\$(GET) \$(GFLAGS) -p \$< > \$*.l \$(LEX) \$(LFLAGS) \$*.l \$(CC) \$(CFLAGS) -c lex.yy.c rm -f lex.yy.c mv lex.yy.c \$@

<i>Table of Standard Implicit (Suffix) Rules for Modula 2 Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
.mod	\$(COMPILE.mod) -o \$@ -e \$@ \$<
.mod.o	\$(COMPILE.mod) -o \$@ \$<
.def.sym	\$(COMPILE.def) -o \$@ \$<
.def~.sym	\$(GET) \$(GFLAGS) -p \$< > \$*.def \$(COMPILE.def) -o\$@ \$*.def
.mod~	\$(GET) \$(GFLAGS) -p \$< > \$*.mod \$(COMPILE.mod) -o \$@ -e \$@ \$*.mod
.mod~.o	\$(GET) \$(GFLAGS) -p \$< > \$*.mod \$(COMPILE.mod) -o \$@ \$*.mod
.mod~.a	\$(GET) \$(GFLAGS) -p \$< > \$*.mod \$(COMPILE.mod) -o \$% \$*.mod \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%

<i>Table of Standard Implicit (Suffix) Rules for NeWS Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
.cps.h	cps \$*.cps
.cps~.h	\$(GET) \$(GFLAGS) -p \$< > \$*.cps \$(CPS) \$(CPSFLAGS) \$*.cps

<i>Table of Standard Implicit (Suffix) Rules for Pascal Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.p</code>	<code>\$(LINK.p) -o \$@ \$< \$(LDLIBS)</code>
<code>.p.o</code>	<code>\$(COMPILE.p) \$(OUTPUT_OPTION) \$<</code>
<code>.p~</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.p</code> <code>\$(LINK.p) -o \$@ \$*.p \$(LDLIBS)</code>
<code>.p~.o</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.p</code> <code>\$(COMPILE.p) \$(OUTPUT_OPTION) \$*.p</code>
<code>.p~.a</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.p</code> <code>\$(COMPILE.p) -o \$% \$*.p</code> <code>\$(AR) \$(ARFLAGS) \$@ \$%</code> <code>\$(RM) \$%</code>

<i>Table of Standard Implicit (Suffix) Rules for Ratfor Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.r</code>	<code>\$(LINK.r) -o \$@ \$< \$(LDLIBS)</code>
<code>.r.o</code>	<code>\$(COMPILE.r) \$(OUTPUT_OPTION) \$<</code>
<code>.r.a</code>	<code>\$(COMPILE.r) -o \$% \$<</code> <code>\$(AR) \$(ARFLAGS) \$@ \$%</code> <code>\$(RM) \$%</code>
<code>.r~</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.r</code> <code>\$(LINK.r) -o \$@ \$*.r \$(LDLIBS)</code>

<i>Table of Standard Implicit (Suffix) Rules for Ratfor Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.r~.o</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.r \$(COMPILE.r) \$(OUTPUT_OPTION) \$*.r
<code>.r~.a</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.r \$(COMPILE.r) -o \$% \$*.r \$(AR) \$(ARFLAGS) \$@ \$% \$(RM) \$%

<i>Table of Standard Implicit (Suffix) Rules for SCCS Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.SCCS_GET</code>	sccs \$(SCCSFLAGS) get \$(SCCSGETFLAGS) \$@ -G\$@
<code>.SCCS_GET_POSIX</code>	sccs \$(SCCSFLAGS) get \$(SCCSGETFLAGS) \$@
<code>.GET_POSIX</code>	\$(GET) \$(GFLAGS) s.\$@

<i>Table of Standard Implicit (Suffix) Rules for Shell Scripts</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.sh</code>	cat \$< >\$@ chmod +x \$@
<code>.sh~</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.sh cp \$*.sh \$@ chmod a+x \$@

<i>Table of Standard Implicit (Suffix) Rules for yacc Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
<code>.y</code>	\$(YACC.y) \$< \$(LINK.c) -o \$@ y.tab.c \$(LDLIBS) \$(RM) y.tab.c
<code>.y.c</code>	\$(YACC.y) \$< mv y.tab.c \$@
<code>.y.ln</code>	\$(YACC.y) \$< \$(LINT.c) -o \$@ -i y.tab.c \$(RM) y.tab.c
<code>.y.o</code>	\$(YACC.y) \$< \$(COMPILE.c) -o \$@ y.tab.c \$(RM) y.tab.c
<code>.y~</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.y \$(YACC) \$(YFLAGS) \$*.y \$(COMPILE.c) -o \$@ y.tab.c \$(RM) y.tab.c
<code>.y~.c</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.y \$(YACC) \$(YFLAGS) \$*.y mv y.tab.c \$@
<code>.y~.ln</code>	\$(GET) \$(GFLAGS) -p \$< > \$*.y \$(YACC.y) \$*.y \$(LINT.c) -o \$@ -i y.tab.c

<i>Table of Standard Implicit (Suffix) Rules for yacc Files</i>	
<i>Implicit Rule Name</i>	<i>Command Line</i>
	<code>\$(RM) y.tab.c</code>
<code>.y~.o</code>	<code>\$(GET) \$(GFLAGS) -p \$< > \$*.y</code> <code>\$(YACC) \$(YFLAGS) \$*.y</code> <code>\$(CC) \$(CFLAGS) -c y.tab.c</code> <code>rm -f y.tab.c</code> <code>mv y.tab.o \$@</code>

make reads in the standard set of implicit rules from the file `/usr/share/lib/make/make.rules`, unless `-r` is in effect, or there is a `make.rules` file in the local directory that does not include that file.

The Suffixes List

The suffixes list is given as the list of dependencies for the `.SUFFIXES:` special-function target. The default list is contained in the `SUFFIXES` macro (See *Table of Predefined Macros* for the standard list of suffixes). You can define additional `.SUFFIXES:` targets; a `.SUFFIXES` target with no dependencies clears the list of suffixes. Order is significant within the list; make selects a rule that corresponds to the target's suffix and the first dependency-file suffix found in the list. To place suffixes at the head of the list, clear the list and replace it with the new suffixes, followed by the default list:

```
.SUFFIXES:
.SUFFIXES: suffixes $(SUFFIXES)
```

A tilde (~) indicates that if a dependency file with the indicated suffix (minus the ~) is under SCCS its most recent version should be retrieved, if necessary, before the target is processed.

Library Maintenance

A target name of the form:

```
lib(member ...)
```

refers to a member, or a space-separated list of members, in an `ar(1)` library.

The dependency of the library member on the corresponding file must be given as an explicit entry in the makefile. This can be handled by a pattern matching rule of the form:

```
lib(%.s): %.s
```

where `.s` is the suffix of the member; this suffix is typically `.o` for object libraries.

A target name of the form:

```
lib( (symbol) )
```

refers to the member of a randomized object library that defines the entry point named *symbol*.

Command Execution Command lines are executed one at a time, *each by its own process or shell*. Shell commands, notably `cd`, are ineffectual across an unescaped `NEWLINE` in the makefile. A line is printed (after macro expansion) just before being executed. This is suppressed if it starts with a `@`, if there is a `.SILENT` entry in the makefile, or if `make` is run with the `-s` option. Although the `-n` option specifies printing without execution, lines containing the macro `$(MAKE)` are executed regardless, and lines containing the `@` special character are printed. The `-t` (`touch`) option updates the modification date of a file without executing any rules. This can be dangerous when sources are maintained by more than one person.

`make` invokes the shell with the `-e` (`exit-on-errors`) argument. Thus, with semicolon-separated command sequences, execution of the later commands depends on the success of the former. This behavior can be overridden by starting the command line with a `-`, or by writing a shell script that returns a non-zero status only as it finds appropriate.

Bourne Shell Constructs To use the Bourne shell `if` control structure for branching, use a command line of the form:

```
if expression ; \
then command ; \
    ... ; \
else command; \
    ... ; \
fi
```

Although composed of several input lines, the escaped `NEWLINE` characters insure that `make` treats them all as one (shell) command line.

To use the Bourne shell `for` control structure for loops, use a command line of the form:

```
for var in list ; \
do command; \
    ... ; \done
```

To refer to a shell variable, use a double-dollar-sign (`$$`). This prevents expansion of the dollar-sign by `make`.

Command Substitutions To incorporate the standard output of a shell command in a macro, use a definition of the form:

```
MACRO :sh =command
```

The command is executed only once, standard error output is discarded, and `NEWLINE` characters are replaced with `SPACES`. If the command has a non-zero exit status, `make` halts with an error.

To capture the output of a shell command in a macro reference, use a reference of the form:

```
$(MACRO :sh)
```

where *MACRO* is the name of a macro containing a valid Bourne shell command line. In this case, the command is executed whenever the reference is evaluated. As with shell command substitutions, the reference is replaced with the standard output of the command. If the command has a non-zero exit status, *make* halts with an error.

In contrast to commands in rules, the command is not subject for macro substitution; therefore, a dollar sign (\$) need not be replaced with a double dollar sign (\$\$).

Signals

INT, SIGTERM, and QUIT signals received from the keyboard halt *make* and remove the target file being processed unless that target is in the dependency list for `.PRECIOUS:`.

Compatibility with GNU *make*

The compatibility mode with GNU *make* changes Oracle Solaris *make*'s behavior with respect to the dynamic macro `$<`. By default the Oracle Solaris *make* treats this macro as the name of a dependency file, derived as if selected for use with an implicit rule.

GNU *make* treats this macro as the name of a dependency, even if it is not a file. If the `-x SUN_MAKE_COMPAT_MODE=GNU` option is passed to the Oracle Solaris *make*, it behaves as GNU *make* in this particular case.

示例

示例 1 Defining dependencies

This makefile says that `pgm` depends on two files `a.o` and `b.o`, and that they in turn depend on their corresponding source files (`a.c` and `b.c`) along with a common file `incl.h`:

```
pgm: a.o b.o
    $(LINK.c) -o $@a.o b.o
a.o: incl.h a.c
    cc -c a.c
b.o: incl.h b.c
    cc -c b.c
```

示例 2 Using implicit rules

The following makefile uses implicit rules to express the same dependencies:

```
pgm: a.o b.o
    cc a.o b.o -o pgm
a.o b.o: incl.h
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of *make*: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

KEEP_STATE This environment variable has the same effect as the `.KEEP_STATE:` special-function target. It enables command dependencies, hidden dependencies and writing of the state file.

USE_SVR4_MAKE This environment variable causes *make* to invoke the generic System V version of *make* (`/usr/lib/svr4.make`). See [sysV-make\(1\)](#).

MAKEFLAGS

This variable is interpreted as a character string representing a series of option characters to be used as the default options. The implementation accepts both of the following formats (but need not accept them when intermixed):

1. The characters are option letters without the leading hyphens or blank character separation used on a command line.
2. The characters are formatted in a manner similar to a portion of the `make` command line: options are preceded by hyphens and blank-character-separated. The `macro=name` macro definition operands can also be included. The difference between the contents of `MAKEFLAGS` and the command line is that the contents of the variable is not subjected to the word expansions associated with parsing the command line values. See [wordexp\(3C\)](#).

When the command-line options `-f` or `-p` are used, they take effect regardless of whether they also appear in `MAKEFLAGS`. If they otherwise appear in `MAKEFLAGS`, the result is undefined.

The `MAKEFLAGS` variable is accessed from the environment before the makefile is read. At that time, all of the options (except `-f` and `-p`) and command-line macros not already included in `MAKEFLAGS` are added to the `MAKEFLAGS` macro. The `MAKEFLAGS` macro is passed into the environment as an environment variable for all child processes. If the `MAKEFLAGS` macro is subsequently set by the makefile, it replaces the `MAKEFLAGS` variable currently found in the environment.

PROJECTDIR

Provides a directory to be used to search for SCCS files not found in the current directory. In all of the following cases, the search for SCCS files is made in the directory `SCCS` in the identified directory. If the value of `PROJECTDIR` begins with a slash, it shall be considered an absolute pathname. Otherwise, the value of `PROJECTDIR` is treated as a user name and that user's initial working directory shall be examined for a subdirectory `src` or `source`. If such a directory is found, it shall be used. Otherwise, the value is used as a relative pathname.

If `PROJECTDIR` is not set or has a null value, the search for SCCS files shall be made in the directory `SCCS` in the current directory. The setting of `PROJECTDIR` affects all files listed in the remainder of this utility description for files with a component named `SCCS`.

`SUN_MAKE_COMPAT_MODE` Causes make to change behavior according to the specified compatibility mode. Examples:

```
SUN_MAKE_COMPAT_MODE="POSIX"
  Support POSIX makefiles and compatibility with
  /usr/xpg4/bin/make

SUN_MAKE_COMPAT_MODE="SUN"
  Support Sun makefiles and compatibility with Oracle Solaris
  /usr/bin/make

SUN_MAKE_COMPAT_MODE="GNU"
  Support GNU makefiles and GNU make behavior (partially
  supported)

SUN_MAKE_COMPAT_MODE="SVR4"
  Support SVR4 makefiles and compatibility with
  /usr/lib/svr4.make
```

退出状态

When the `-q` option is specified, the make utility exits with one of the following values:

```
0    Successful completion.
1    The target was not up-to-date.
>1  An error occurred.
```

When the `-q` option is not specified, the make utility exits with one of the following values:

```
0    Successful completion
>0  An error occurred
```

文件

```
makefile          current version(s) of make description file
Makefile
s.makefile
s.Makefile        SCCS history files for the above makefile(s) in the
                  current directory

SCCS/s.makefile
SCCS/s.Makefile   SCCS history files for the above makefile(s)

make.rules        default file for user-defined targets, macros, and
                  implicit rules

/usr/share/lib/make/make.rules  makefile for standard implicit rules and macros (not
                                  read if make.rules is)

.make.state       state file in the local directory
```


属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/make

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

/usr/xpg4/bin/make

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[ar\(1\)](#), [arch\(1\)](#), [cd\(1\)](#), [cpp\(1\)](#), [lex\(1\)](#), [mach\(1\)](#), [sccs-get\(1\)](#), [sh\(1\)](#), [sysV-make\(1\)](#), [yacc\(1\)](#), [wordexp\(3C\)](#), [passwd\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [POSIX.2\(5\)](#), [standards\(5\)](#)

《Solaris Advanced User's Guide》

诊断

Don't know how to make target *target*

There is no makefile entry for *target*, and none of make's implicit rules apply (there is no dependency file with a suffix in the suffixes list, or the target's suffix is not in the list).

*** *target* removed.

make was interrupted while building *target*. Rather than leaving a partially-completed version that is newer than its dependencies, make removes the file named *target*.

*** *target* not removed.

make was interrupted while building *target* and *target* was not present in the directory.

*** *target* could not be removed, *reason*

make was interrupted while building *target*, which was not removed for the indicated reason.

Read of include file *file* failed

The makefile indicated in an include directive was not found, or was inaccessible.

Loop detected when expanding macro value *macro*'

A reference to the macro being defined was found in the definition.

Could not write state file *file*

You used the .KEEP_STATE: target, but do not have write permission on the state file.

***Error code *n*

The previous shell command returned a nonzero error code.

*** *signal message*

The previous shell command was aborted due to a signal. If `- core dumped` appears after the message, a core file was created.

Conditional macro conflict encountered

Displayed only when `-d` is in effect, this message indicates that two or more parallel targets currently being processed depend on a target which is built differently for each by virtue of conditional macros. Since the target cannot simultaneously satisfy both dependency relationships, it is conflicted.

已知问题

Some commands return nonzero status inappropriately; to overcome this difficulty, prefix the offending command line in the rule with a `-`.

Filenames with the characters `=`, `:`, or `@`, do not work.

You cannot build `file.o` from `lib(file.o)`.

Options supplied by `MAKEFLAGS` should be reported for nested make commands. Use the `-d` option to find out what options the nested command picks up from `MAKEFLAGS`.

This version of `make` is incompatible in certain respects with previous versions:

- The `-d` option output is much briefer in this version. `-dd` now produces the equivalent voluminous output.
- `make` attempts to derive values for the dynamic macros `$$`, `$$<`, and `$$?`, while processing explicit targets. It uses the same method as for implicit rules; in some cases this can lead either to unexpected values, or to an empty value being assigned. (Actually, this was true for earlier versions as well, even though the documentation stated otherwise.)
- `make` no longer searches for SCCS history (`s.`) files.
- Suffix replacement in macro references are now applied after the macro is expanded.

There is no guarantee that makefiles created for this version of `make` works with earlier versions.

If there is no `make.rules` file in the current directory, and the file `/usr/share/lib/make/make.rules` is missing, `make` stops before processing any targets. To force `make` to run anyway, create an empty `make.rules` file in the current directory.

Once a dependency is made, `make` assumes the dependency file is present for the remainder of the run. If a rule subsequently removes that file and future targets depend on its existence, unexpected errors can result.

When hidden dependency checking is in effect, the `$$?` macro's value includes the names of hidden dependencies. This can lead to improper filename arguments to commands when `$$?` is used in a rule.

Pattern replacement macro references cannot be used in the dependency list of a pattern matching rule.

Unlike previous versions, this version of `make` strips a leading `./` from the value of the `$$@` dynamic macro.

With automatic SCCS retrieval, this version of `make` does not support tilde suffix rules.

The only dynamic macro whose value is strictly determined when used in a dependency list is `$$` (takes the form `$$@`).

`make` invokes the shell with the `-e` argument. This cannot be inferred from the syntax of the rule alone.

引用名 `makekey` - 生成加密密钥

用法概要 `/usr/lib/makekey`

描述 `makekey` 通过增加搜索密钥空间所需的时间量来提高依赖于密钥的加密方案的可用性。它会尝试为其 *key* 读取 8 个字节（前八个输入字节），然后尝试为其 *salt* 读取 2 个字节（最后两个输入字节）。输出取决于采用一种目的在于难以计算（即需要大量的时间）的方式所进行的输入。

前八个输入字节 (*input key*) 可以是任意的 ASCII 字符。最后两个 (*salt*) 是从数字集、`.`、`/`、大写和小写字母当中最佳选择出的。*salt* 字符会作为输出的前两个字符重复。其余的 11 个输出字符是从与 *salt* 相同的集合中选择的，并且构成了 *output key*。

执行的转换基本上如下：*salt* 用于从全部基于美国国家标准局 DES 算法但分为 4096 种不同方式的 4096 台机密计算机中选择一台。使用 *input key* 作为密钥，将一个常量字符串馈送到该计算机中并且循环多次。得到的 64 位会在结果中分发到 66 个 *output key* 位中。

`makekey` 适用于执行加密的程序。它的输入和输出通常为管道。

另请参见 [ed\(1\)](#)、[vi\(1\)](#)、[passwd\(4\)](#)

附注 `makekey` 可根据输入是在终端键入还是从文件重定向而生成不同结果。

引用名	man – find and display reference manual pages
用法概要	<pre>man [-] [-adFlrt] [-M path] [-T macro-package] [-s section] name... man [-M path] [-s section] -k query... man [-M path] -f file... man [-M path] [-s section] -K query...</pre>
描述	<p>The <code>man</code> command displays information from the reference manuals. It displays complete manual pages that you select by <i>name</i>, or one-line summaries selected either by <i>query</i> (<code>-k</code> or <code>-K</code>), or by the name of an associated file (<code>-f</code>). If no manual page is located, <code>man</code> prints an error message.</p>
Source Format	<p>Reference Manual pages are marked up with either <code>nroff</code> (see nroff(1)) or SGML (Standard Generalized Markup Language) tags (see sgml(5)). The <code>man</code> command recognizes the type of markup and processes the file accordingly. The various source files are kept in separate directories depending on the type of markup.</p>
Location of Manual Pages	<p>The online Reference Manual page directories are conventionally located in <code>/usr/share/man</code>. The <code>nroff</code> sources are located in the <code>/usr/share/man/man*</code> directories. The SGML sources are located in the <code>/usr/share/man/sman*</code> directories. Each directory corresponds to a section of the manual. Since these directories are optionally installed, they might not reside on your host. You might have to mount <code>/usr/share/man</code> from a host on which they do reside.</p> <p>If there are preformatted, up-to-date versions in the corresponding <code>cat*</code> or <code>fmt*</code> directories, <code>man</code> simply displays or prints those versions. If the preformatted version of interest is out of date or missing, <code>man</code> reformats it prior to display and stores the preformatted version if <code>cat*</code> or <code>fmt*</code> is writable. The index files are not updated. See catman(1M). If directories for the preformatted versions are not provided, <code>man</code> reformats a page whenever it is requested. <code>man</code> uses a temporary file to store the formatted text during display.</p> <p>If the standard output is not a terminal, or if the <code>'-'</code> flag is given, <code>man</code> pipes its output through cat(1). Otherwise, <code>man</code> pipes its output through more(1) to handle paging and underlining on the screen.</p>
Query Strings	<p>Using <code>-k</code> or <code>-K</code> options, manual pages can be searched with <i>query</i>, one or more terms or phrases. It supports index-file-based, full text searching, stemming, and section matching. For information regarding how to generate the index files, refer to catman(1M) and man(5).</p> <p>Stemming for English, for example, identifies the string <code>cats</code>, <code>catlike</code>, <code>catty</code>, and so forth, based on the root <code>cat</code>. It identifies <code>stemmer</code>, <code>stemming</code>, and <code>stemmed</code> based on <code>stem</code>. A stemming algorithm reduces the words <code>fishing</code>, <code>fished</code>, <code>fish</code>, and <code>fisher</code> to the root word, <code>fish</code>.</p> <p>Matching is done in case-insensitive manner. Stemming is done for English manual pages only.</p>

Matched manual pages are sorted and presented based on the score of the query matches in ascending order.

Oracle Solaris manual pages are divided into sections such as NAME, SYNOPSIS, DESCRIPTION, and so forth. Users can specify the scope of search into a section as details described in the -K option.

选项

The following options are supported:

- a
Shows all manual pages matching *name* within the MANPATH search path. Manual pages are displayed in the order found.
- d
Debugs. Displays what a section-specifier evaluates to, method used for searching, and paths searched by man.
- f *file* ...
man attempts to locate manual pages related to any of the given *files*. It strips the leading path name components from each *file*, and then prints one-line summaries containing the resulting basename or names.

This option uses the index files. Refer to [catman\(1M\)](#) and [man\(5\)](#) for details on how index files are generated.

- F
Forces man to search all directories specified by MANPATH or the man.cf file, rather than using the index lookup files. This option is useful if the index files are not up to date and they have been made the default behavior of the man command. The option therefore does not have to be invoked and is documented here for reference only.
- k *query* ...
Prints out one-line summaries from the index files.

See the -K option for information regarding how the index files are generated. If there are no index files, manual page files are directly looked up, therefore yielding slower response time than cases where index files exist.

- K *query* ...
Search for the specified query from the index files. If there are no index files, search is directly done on the manual pages, which causes a much slower search.

If you supply a section name ending with a colon (:) at the query option argument as the first text from left, just as *section name: query*, the search for the query string is done on the specified section only. If the specified section name does not exist, it will list all the supported section name for users.

The index files in /usr/share/man and /usr/gnu/share/man used by -f, -k, and -K are automatically generated when man pages in those directories are installed or updated and

the packages delivering them have tagged the files with `restart_fmri=svc:/application/man-index:default` as specified in [pkg\(5\)](#). They may also be generated by running `svcadm restart application/man-index` manually, or running [catman\(1M\)](#) with the `-w`.

`-l`

Lists all manual pages found matching *name* within the search path.

`-M path`

Specifies an alternate search path for manual pages. *path* is a colon-separated list of directories that contain manual page directory subtrees. For example, if *path* is `/usr/share/man:/usr/local/man`, `man` searches for *name* in the standard location, and then `/usr/local/man`. When used with the `-f`, `-k` or `-K` options, the `-M` option must appear first. Each directory in the *path* is assumed to contain subdirectories of the form `man*` or `sman*`, one for each section. This option overrides the `MANPATH` environment variable.

`-r`

Reformats the manual page, but does not display it. This replaces the `man - -t name` combination.

`-s section ...`

Specifies sections of the manual for `man` to search. The directories searched for *name* are limited to those specified by *section*. *section* can be a numerical digit, perhaps followed by one or more letters to match the desired section of the manual, for example, “3lib”. Also, *section* can be a word, for example, `local`, `new`, `old`, `public`. *section* can also be a letter. To specify multiple sections, separate each section with a comma. This option overrides the `MANPATH` environment variable and the `man.cf` file. See Search Path below for an explanation of how `man` conducts its search.

`-t`

`man` arranges for the specified manual pages to be troffed to a suitable raster output device (see [troff\(1\)](#)). If both the `-` and `-t` flags are given, `man` updates the troffed versions of each named *name* (if necessary), but does not display them.

`-T macro-package`

Formats manual pages using *macro-package* rather than the standard `-man` macros defined in `/usr/share/lib/tmac/an`. See Search Path under USAGE for a complete explanation of the default search path order.

操作数

The following operand is supported:

name

The name of a standard utility or a keyword.

用法

The usage of `man` is described below:

Manual Page Sections Entries in the reference manuals are organized into *sections*. A section name consists of a major section name, typically a single digit, optionally followed by a subsection name, typically one or more letters. An unadorned major section name, for example, “9”, does not act as an abbreviation for the subsections of that name, such as “9e”, “9f”, or “9s”. That is, each subsection must be searched separately by `man -s`. Each section contains descriptions apropos to a particular reference category, with subsections refining these distinctions. See the `intro` manual pages for an explanation of the classification used in this release.

The following contains a brief description of each manual page section and the information it references:

- Section 1 describes, in alphabetical order, commands available with the operating system.
- Section 1M describes, in alphabetical order, commands that are used chiefly for system maintenance and administration purposes.
- Section 2 describes all of the system calls. Most of these calls have one or more error returns. An error condition is indicated by an otherwise impossible returned value.
- Section 3 describes functions found in various libraries, other than those functions that directly invoke UNIX system primitives, which are described in Section 2.
- Section 4 outlines the formats of various files. The C structure declarations for the file formats are given where applicable.
- Section 5 contains miscellaneous documentation such as character-set tables.
- Section 7 describes various special files that refer to specific hardware peripherals and device drivers. STREAMS software drivers, modules and the STREAMS-generic set of system calls are also described.
- Section 9E describes the DDI (Device Driver Interface)/DKI (Driver/Kernel Interface), DDI-only, and DKI-only entry-point routines a developer can include in a device driver.
- Section 9F describes the kernel functions available for use by device drivers.
- Section 9S describes the data structures used by drivers to share information between the driver and the kernel.

Search Path Before searching for a given *name*, `man` constructs a list of candidate directories and sections. `man` searches for *name* in the directories specified by the `MANPATH` environment variable.

In the absence of `MANPATH`, `man` constructs its search path based upon the `PATH` environment variable, primarily by substituting `man` for the last component of the `PATH` element. Special provisions are added to account for unique characteristics of directories such as `/sbin`, `/usr/ucb`, `/usr/xpg4/bin`, and others. If the file argument contains a `/` character, the *dirname* portion of the argument is used in place of `PATH` elements to construct the search path.

Within the manual page directories, `man` confines its search to the sections specified in the following order:

- *sections* specified on the command line with the `-s` option

- *sections* embedded in the MANPATH environment variable
- *sections* specified in the man .cf file for each directory specified in the MANPATH environment variable

If none of the above exist, man searches each directory in the manual page path, and displays the first matching manual page found.

The man .cf file has the following format:

```
MANSECTS=section[ ,section] . . .
```

Lines beginning with '#' and blank lines are considered comments, and are ignored. Each directory specified in MANPATH can contain a manual page configuration file, specifying the default search order for that directory.

Formatting Manual Pages

Manual pages are marked up in `nroff(1)` or `sgml(5)`. `nroff` manual pages are processed by `nroff(1)` or `troff(1)` with the `-man` macro package. Please refer to `man(5)` for information on macro usage. SGML—tagged manual pages are processed by an SGML parser and passed to the formatter.

Preprocessing nroff Manual Pages

When formatting an `nroff` manual page, man examines the first line to determine whether it requires special processing. If the first line is a string of the form:

```
'\" X
```

where *X* is separated from the "" by a single SPACE and consists of any combination of characters in the following list, man pipes its input to `troff(1)` or `nroff(1)` through the corresponding preprocessors.

e
 `eqn(1)`, or `neqn` for `nroff`

r
 `refer(1)`

t
 `tbl(1)`

v
 `vgrind(1)`

If `eqn` or `neqn` is invoked, it automatically reads the file `/usr/pub/eqnchar` (see `eqnchar(5)`). If `nroff(1)` is invoked, `col(1)` is automatically used.

Referring to Other nroff Manual Pages

If the first line of the `nroff` manual page is a reference to another manual page entry fitting the pattern:

```
.so man*/sourcefile
```

man processes the indicated file in place of the current one. The reference must be expressed as a path name relative to the root of the manual page directory subtree.

When the second or any subsequent line starts with `.so`, `man` ignores it; `troff(1)` or `nroff(1)` processes the request in the usual manner.

Processing SGML Manual Pages

Manual pages are identified as being marked up in SGML by the presence of the string `<!DOCTYPE`. If the file also contains the string `SHADOW_PAGE`, the file refers to another manual page for the content. The reference is made with a file entity reference to the manual page that contains the text. This is similar to the `.so` mechanism used in the `nroff` formatted manual pages.

环境变量

See `environ(5)` for descriptions of the following environment variables that affect the execution of `man`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

MANPATH

A colon-separated list of directories; each directory can be followed by a comma-separated list of sections. If set, its value overrides `/usr/share/man` as the default directory search path, and the `man.cf` file as the default section search path. The `-M` and `-s` flags, in turn, override these values.)

PAGER

A program to use for interactively delivering `man`'s output to the screen. If not set, `'more -s'` is used. See `more(1)`.

TCAT

The name of the program to use to display `troffed` manual pages.

TROFF

The name of the formatter to use when the `-t` flag is given. If not set, `troff(1)` is used.

示例

示例 1 Creating a Text Version of a Manual Page

The following example creates the `pipe(2)` manual page in `ascii` text:

```
% man pipe.2 | col -x -b > pipe.text
```

This is an alternative to using `man -t`, which sends the manual page to the default printer, if the user wants a text file version of the manual page.

示例 2 Getting a List of Manual Pages that Match One or More Terms

The following example gets a list of manual pages that match for the term `zfs` or `create`:

```
% man -K zfs create
```

示例 3 Getting a List of Manual Pages that Match One or More Phrases

The following example gets a list of manual pages that match for the quote-enclosed phrases, `"zfs create"` or `"storage pool"`.

```
% man -K 'zfs create' "storage pool"
```

示例 4 Getting a List of Manual Pages that Match Terms or Phrases in a Section

The following example gets a list of manual pages that have the term `zfs` in the SEE ALSO section:

```
% man -K see also: zfs
```

The following example gets a list of manual pages that have the phrase “zfs create” in the Examples section:

```
% man -K examples: "zfs create"
```

退出状态

The following exit values are returned:

```
0
    Successful completion.
```

```
>0
    An error occurred.
```

文件

```
/usr/share/man
    Root of the standard manual page directory subtree
```

```
/usr/share/man/man?/*
    Unformatted nroff manual entries
```

```
/usr/share/man/man_index/*
    Table of Contents and keyword database.
```

Generated files include:

- /usr/share/man/man-index/man.idx
- /usr/share/man/man-index/man.dic
- /usr/share/man/man-index/man.req
- /usr/share/man/man-index/man.pos
- /usr/share/man/man-index/man.doc

```
/usr/share/man/sman?/*
    Unformatted SGML manual entries
```

```
/usr/share/man/cat?/*
    nroffed manual entries
```

```
/usr/share/man/fmt?/*
    troffed manual entries
```

```
/usr/share/lib/tmac/an
    Standard -man macro package
```

```
/usr/share/lib/sgml/locale/C/dtd/*
    SGML document type definition files
```

`/usr/share/lib/sgml/locale/C/solbook/*`
 SGML style sheet and entity definitions directories

`/usr/share/lib/pub/eqnchar`
 Standard definitions for eqn and neqn

`man.cf`
 Default search order by section

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	text/doctools
CSI	Enabled, see NOTES.
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[apropos\(1\)](#), [cat\(1\)](#), [col\(1\)](#), [eqn\(1\)](#), [more\(1\)](#), [nroff\(1\)](#), [refer\(1\)](#), [tbl\(1\)](#), [ttr\(1\)](#), [vgrind\(1\)](#), [whatis\(1\)](#), [catman\(1M\)](#), [attributes\(5\)](#), [environ\(5\)](#), [eqnchar\(5\)](#), [man\(5\)](#), [sgml\(5\)](#), [standards\(5\)](#)

附注

The `-f`, `-k`, and `-K` options use the index files which are created by the SMF service as specified in [man\(5\)](#), or by manually using [catman\(1M\)](#) with the `-w` option.

The `windex` database file is no longer used. The `windex` database file has replaced with the new index files.

The `man` command is CSI-capable. However, some utilities invoked by the `man` command, namely, `ttr`, `eqn`, `neqn`, `refer`, `tbl`, and `vgrind`, are not verified to be CSI-capable. Because of this, the `man` command with the `-t` option can not handle non-EUC data. Also, using the `man` command to display manual pages that require special processing through `eqn`, `neqn`, `refer`, `tbl`, or `vgrind` can not be CSI-capable.

已知问题

The manual is supposed to be reproducible either on a phototypesetter or on an ASCII terminal. However, on a terminal some information (indicated by font changes, for instance) is lost.

Some dumb terminals cannot process the vertical motions produced by the `e` (see [eqn\(1\)](#)) preprocessing flag. To prevent garbled output on these terminals, when you use `e`, also use `t`, to invoke [col\(1\)](#) implicitly. This workaround has the disadvantage of eliminating superscripts and subscripts, even on those terminals that can display them. Control-q clears a terminal that gets confused by [eqn\(1\)](#) output.

引用名	mconnect – 连接到 SMTP 邮件服务器套接字
用法概要	mconnect [-p <i>port</i>] [-r] [<i>hostname</i>]
描述	mconnect 实用程序可打开与给定主机上的邮件服务器的连接，所以可以独立于所有其他邮件软件对其进行测试。如果没有给定主机，会建立与本地主机的连接。服务器可能会在此连接上使用简单邮件传输协议 (Simple Mail Transfer Protocol, SMTP)。通过键入 <code>quit</code> 命令退出。键入 <code>EOF</code> 可将文件结束标记发送至服务器。中断会立即关闭连接并退出。
选项	支持以下选项： <p><code>-p <i>port</i></code> 指定端口号作为下一个参数，而不是缺省 SMTP 端口（端口号 25）。</p> <p><code>-r</code> Raw 模式：禁用缺省线路缓冲和输入处理。这会对端口号 25 产生类似于 telnet(1) 的效果。</p>
操作数	支持下列操作数： <p><i>hostname</i> 给定主机的名称。</p>
用法	mconnect 命令启用 IPv6。请参见 ip6(7P) 。
文件	<code>/etc/mail/sendmail.hf</code> SMTP 命令的帮助文件。
属性	有关下列属性的说明，请参见 attributes(5) ：

属性类型	属性值
可用性	service/network/smtp/sendmail

另请参见 [telnet\(1\)](#)、[sendmail\(1M\)](#)、[attributes\(5\)](#)、[ip6\(7P\)](#)

由 Postel, Jonathan B. 编著的《Simple Mail Transfer Protocol》，RFC821，南加州大学信息科学研究所出版，1982 年 8 月。

引用名 mcs – manipulate the comment section of an object file

用法概要 mcs [-cdHpVz] [-A *file*] [-a *string*][-n *name*] *file*...

描述 The `mcs` command is used to display, compress, or append content to comment sections in an ELF object file. It can also be used to delete comment or non-comment sections, and to zero `SHT_PROGBITS` sections. Unless otherwise specified, the section named `.comment` is manipulated. `mcs` cannot modify or delete a section that is contained within a segment.

If the input file is an archive (see [ar.h\(3HEAD\)](#)), the archive is treated as a set of individual files. For example, if the `-a` option is specified, the string is appended to the comment section of each object file in the archive; if the archive member is not an object file, then it is left unchanged.

`mcs` must be given one or more of the options described below. It applies each option, in the order given, to each file.

For append operations, if the object does not already contain a section with the specified name, `mcs` will create a new empty section with that name before performing the append operation.

选项 The following options are supported:

- A *file* Appends the contents of the given file to the comment section of the object files.
- a *string* Appends *string* to the comment section of the object files.
- c Compresses the contents of the comment section of the object files. All duplicate entries are removed. The ordering of the remaining entries is not disturbed.
- d Deletes the specified section from the object files.
- H When `-p` is used, suppress the output of the name of the file, as well as any blank lines normally inserted between the output for each file.
- n *name* Specifies the name of the section to access if other than `.comment`. By default, `mcs` deals with the section named `.comment`. `mcs` can take multiple `-n` options to allow for specification of multiple sections. `mcs` always manipulations sections assuming that they have the format of a comment section.
- p Prints the contents of the comment section on the standard output. Unless used with `-H`, each section printed is prefixed with the name of the file from which it was extracted, using the format *file*[*member_name*] : for archive files and *file*: for other files.
- V Prints a message on standard error giving information about the version of `mcs`.
- z Replaces any `SHT_PROGBITS` sections with zeros while retaining the original attributes of the sections.

示例

示例 1 Printing a file's comment section

The following entry

```
example% mcs -p elf.file
```

prints the comment section of the file `elf.file`.

示例 2 Appending a string to a comment section

The following entry

```
example% mcs -a xyz elf.file
```

appends string `xyz` to `elf.file`'s comment section.

示例 3 Stripping a specified non-allocable section

Although used primarily with comment sections, `mcs` can operate on any non-allocable section. In contrast to the `strip` command, which removes a predefined selection of non-allocable sections, `mcs` can be used to delete a specific section. The following entry

```
example% mcs -d -n .annotate elf.file
```

removes the section named `.annotate` from the file `elf.file`.

文件

`/tmp/mcs*` temporary files

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities
Interface Stability	Committed

另请参见

[ar\(1\)](#), [as\(1\)](#), [ld\(1\)](#), [strip\(1\)](#), [ar.h\(3HEAD\)](#), [elf\(3ELF\)](#), [a.out\(4\)](#), [attributes\(5\)](#)

附注

When `mcs` deletes a section using the `-d` option, it tries to bind together sections of type `SHT_REL` and target sections pointed to by the `sh_info` section header field. If one is to be deleted, `mcs` attempts to delete the other of the pair.

The `-z` option removes the contents of `SHT_PROGBITS` sections while retaining the original ELF structure of the object. The need for use of the `-z` option is limited. However, the option can be used to deliver an object file when the contents of `SHT_PROGBITS` sections are not relevant.

引用名 mdb – 模块调试器

用法概要 mdb [-fkmuwyaFKMSUW] [\pm o option] [-p pid] [-s distance]
 [-I path] [-L path] [-P prompt] [-R root]
 [-V dis-version] [object [core] | core | suffix]

描述

介绍 mdb 实用程序是针对实时操作系统、操作系统故障转储、用户进程、用户进程核心转储以及目标文件执行底层调试和编辑的可扩展实用程序。有关 mdb 功能的更为详细的说明，请参见手册《[Oracle Solaris Modular Debugger Guide](#)》。

调试是分析软件程序的执行和状态以便消除缺陷的过程。传统的调试工具提供了用于执行控制的工具，因此，程序员可以在受控环境中重新执行程序并显示程序数据的当前状态，或者借助开发程序时使用的源语言对表达式进行求值。

不幸的是，这些技术通常不适用于以下情形：调试复杂的软件系统（例如操作系统，其中出现的错误可能无法再现，并且程序状态非常多且分布在各处）、经过高度优化的程序（这些程序的调试信息已删除，或者自身即为底层调试工具）、开发人员只能访问事后分析信息的客户情况。

mdb 提供了用于调试这些程序和方案的可完全定制的环境，包括动态模块工具，程序员可使用该工具实现自己的调试命令，以便执行特定于程序的分析。每个 mdb 模块可用于在多种不同上下文中检查程序，包括实时和事后分析。

定义 **目标**是调试器所检查的程序。mdb 当前提供了对以下类型目标的支持：用户进程、用户进程核心文件、实时操作系统（通过 /dev/kmem 和 /dev/ksyms）、操作系统故障转储、记录在操作系统故障转储内部的用户进程映像、ELF 目标文件以及原始二进制文件。每个目标导出一组标准的属性，其中包括一个或多个地址空间、一个或多个符号表、一组装入目标文件和一组线程，这些内容均可以使用下述调试器命令检查。

调试器命令在 mdb 术语中称为 *dcmd*（发音为 dee-command），是调试器中可以访问当前目标的任意属性的例程。mdb 解析标准输入中的命令，然后执行对应的 *dcmd*。每个 *dcmd* 也可以接受字符串或数字参数列表，如下语法说明中所示。mdb 包含一组始终可供使用的内置 *dcmd*，如下所示。您也可以通过编写自己的 *dcmd* 来扩展 mdb 自身的功能，如《[Oracle Solaris Modular Debugger Guide](#)》中所述。

遍历器是一组例程，用来描述如何在特定程序数据结构的元素中进行遍历或迭代。遍历器封装 *dcmd* 以及 mdb 自身中的数据结构实现。您可以按交互方式使用遍历器，也可将它们用作基元来构建其他 *dcmd* 或遍历器。在使用 *dcmd* 时，您可以将自己的遍历器作为调试器模块的一部分实现，从而扩展 mdb。

调试器模块（即 *dmod*，发音为 dee-mod）是动态载入的库，包含一组 *dcmd* 和遍历器。在初始化期间，mdb 尝试装入与目标中存在的装入目标文件相对应的 *dmod*。接下来，在运行 mdb 时，您可以随时装入或卸载 *dmod*。随 mdb 提供了一组标准 *dmod*，它们可用于调试 Solaris 内核。《[Oracle Solaris Modular Debugger Guide](#)》中包含有关开发自己的调试器模块的更多信息。

宏文件是包含一组要执行的命令的文本文件。宏文件通常用于自动执行显示简单数据结构的进程。mdb 为执行针对 [adb\(1\)](#) 编写的宏文件提供了完整的向后兼容性，而 Solaris 安装包括一组用于调试 Solaris 内核的宏文件，它们可以在任意一种工具中使用。

语法

调试器处理标准输入中的命令。如果标准输入为终端，则 mdb 提供终端编辑功能。mdb 也可以处理来自宏文件以及来自 `dcmd` 流水线的命令，如下所述。语言语法的设计以下面的概念为中心：计算表达式的值（通常为目标中的内存地址），然后对该地址应用 `dcmd`。当前地址位置称为**点**，使用“.”来引用相关值。

元字符为以下字符之一：

```
[ ] | ! / \ ? = > $ : ;
          NEWLINE SPACE TAB
```

空白为**制表符**或**空格**。**词**是由一个或多个不括起的元字符分隔的字符序列。一些元字符仅在特定上下文中起到分隔符的作用，如下文所述。**标识符**是字母、数字、下划线、句点或反引号组成的序列，以字母、下划线或句点开头。标识符用作符号、变量、`dcmd` 和遍历器的名称。使用**换行**或分号(;)来分隔各个命令。

使用以下词或元字符之一表示 `dcmd`：

```
/ \ ? = > $character :character ::identifier
```

使用元字符命名或者使用单个 \$ 或 : 作为前缀的 `dcmd` 作为内置运算符提供，并且与旧的 [adb\(1\)](#) 实用程序的命令集实现完全兼容。`dcmd` 在解析之后，/、\、?、=、>、\$ 和 : 字符在参数列表终止之前不再识别为元字符。

简单命令是一个 `dcmd`，后面跟随由零个或多个以空格分隔的词组成的序列。除非在下面的**引用**和**算术扩展**中指定，否则词将作为参数传递到调用的 `dcmd`。每个 `dcmd` 返回一个退出状态，指示成功、失败或者在调用时使用了无效参数。

流水线是由 | 分隔的一个或多个简单命令。与 shell 不同，mdb 流水线中的 `dcmd` 不作为单独进程执行。对流水线进行解析之后，按照从左到右的顺序依次调用各个 `dcmd`。对每个 `dcmd` 的输出进行处理并存储，如下面的 `dcmd` **流水线**中所述。左侧的 `dcmd` 完成之后，经过处理的输出将用作流水线中下一个 `dcmd` 的输入。如果任何 `dcmd` 都未返回成功的退出状态，流水线将异常中止。

表达式是一个词序列，对该序列进行求值以计算得出 64 位的无符号整数值。词使用下面的**算术扩展**中所述的规则求值。

命令

命令可以是以下内容之一：

```
pipeline [! word...][;]
```

简单命令或流水线可以选择使用 ! 字符作为后缀，指示调试器应打开 [pipe\(2\)](#) 并将 mdb 流水线中上一个 `dcmd` 的标准输出发送到外部进程。该外部进程是使用以下方法创建的：执行 `$SHELL -c`，后面跟随通过在 ! 字符后串联词所构成的字符串。有关详细信息，请参见下文的 **Shell 转义**。

expression pipeline [! *word ...*] [;]

可以使用一个表达式作为简单命令或流水线的前缀。在执行流水线之前，将点的值（使用“.”表示的变量）设置为表达式的值。

expression , expression pipeline [! *word ...*] [;]

可以使用两个表达式作为简单命令或流水线的前缀。对第一个表达式求值，确定点的新值；对第二个表达式求值，确定流水线中第一个 *dcmd* 的重复计数。此 *dcmd* 将执行 *count* 次数，然后再执行流水线中的下一个 *dcmd*。重复计数仅应用于流水线中的第一个 *dcmd*。

, *expression pipeline* [! *word ...*] [;]

如果省略了第一个表达式，则不修改点，但是将根据表达式的值重复流水线中的第一个 *dcmd*。

expression [! *word ...*] [;]

一个命令只能由一个算术表达式组成。对表达式求值并将点变量设置为表达式值，然后使用点的新值来执行上一个 *dcmd* 和参数。

expression , expression [! *word ...*] [;]

一个命令只能由一个点表达式和重复计数表达式组成。将点设置为第一个表达式的值之后，上一个 *dcmd* 和参数将重复执行由第二个表达式的值指定的次数。

, *expression* [! *word ...*] [;]

如果省略第一个表达式，则不修改点，但上一个 *dcmd* 和参数将重复执行由计数表达式的值指定的次数。

! *word ...* [;]

如果命令以 ! 字符开头，则不执行任何 *dcmd*，而调试器仅执行 *\$SHELL -c*，后面跟随通过在 ! 字符后串联词所构成的字符串。

注释

词以 // 开头时，将忽略该词以及后面直至换行之前的所有字符。

算术扩展

当 *mdb* 命令前面为表示起始地址的可选表达式时，或者为起始地址和重复计数时，将执行算术扩展。还可以执行算术扩展来计算 *dcmd* 的数字参数。在参数列表中可以出现算术表达式，该表达式必须括在方括号中，前面使用美元符号 (*\$[expression]*)。该表达式将由表达式的值取代。

表达式可以包含以下任意特殊词：

整数

指定的整数值。整数值可以使用以下前缀：*0i* 或 *0I* 表示二进制值；*0o* 或 *0O* 表示八进制值；*0t* 或 *0T* 表示十进制值；*0x* 或 *0X* 表示十六进制值（缺省值）。

0[tT][0-9]+.[0-9]+

转换为 IEEE 双精度浮点值表示法的指定十进制浮点值。

'*ccccccc*'

通过将每个字符转换为等于 ASCII 值的字节计算得到的整数值。在字符常量中最多可以指定八个字符。字符从最低有效字节开始，按逆序（从右到左）压缩为整数。

<标识符

以 *identifier* 命名的变量的值。

标识符

以 *identifier* 命名的符号的值。

(表达式)

expression 的值。

.

点的值。

&

最近执行 `dcmd` 所用的点的值。

+

使用当前增量递增的点的值。

^

使用当前增量递减的点的值。

增量为全局变量，用于存储上一个格式化 `dcmd` 读取的字节总数。有关增量的详细信息，请参见下文的**格式化 dcmd** 中讨论的内容。

一元运算符从右向左执行运算，优先级高于二元运算符。一元运算符包括：

#表达式

逻辑否定。

~表达式

按位取反。

-表达式

整数否定。

%表达式

目标文件位置的指针大小数量的值，该位置对应于目标虚拟地址空间中的虚拟地址 *expression*。

%/[csil]/expression

目标文件位置的字符、短整型、整型或长整型大小数量的值，该位置对应于目标虚拟地址空间中的虚拟地址 *expression*。

%/[1248]/expression

目标文件位置的单字节、双字节、四个字节的或八个字节数量的值，该位置对应于目标虚拟地址空间中的虚拟地址 *expression*。

***表达式**

目标虚拟地址空间中的虚拟地址 *expression* 处的指针大小数量的值。

***[csil]/expression**

目标虚拟地址空间中的虚拟地址 *expression* 处的字符、短整型、长整型大小数量的值。

***[1248]/expression**

目标虚拟地址空间中的虚拟地址 *expression* 处的单字节、双字节、四字节或八字节数量的值。

二元运算符从左向右执行运算，优先级低于一元运算符。二元运算符按照优先级从高到低的顺序为：

*

整数相乘。

%

整数相除。

#

将左侧向上取整到右侧的下一个倍数。

+

整数相加。

-

整数相减。

<<

左移位。

>>

右移位。

==

逻辑等。

!=

逻辑不等。

&

按位与。

^

按位异或。

|

按位同或。

引用 除非括起来，否则上述各个元字符（请参见**语法**）将终止词。字符可以括起来（强制 mdb 按原义解释各个字符，而不解释为任何特殊含义），方法是使用一对单引号（' '）或双引号（" "）将这些字符括起来。单引号内部不能出现单引号。在双引号内部，mdb 可以识别 C 编程语言字符转义序列。

Shell 转义 ! 字符可用于在 `mdb` 命令和用户 `shell` 之间创建流水线。如果 `$SHELL` 环境变量已设置，`mdb` 将为 `shell` 转义派生并执行此程序；否则将使用 `/bin/sh`。使用 `-c` 选项，后面跟随通过在 `!` 字符后串联词所构成的字符串，以此调用 `shell`。`!` 字符优先级高于除分号 (`;`) 和换行之外的所有其他元字符。检测到 `shell` 转义时，后面直到下一个分号或换行的其余字符将按原义传递到 `shell`。`shell` 命令的输出不能通过流水线输出到 `mdb dcmd`。由 `shell` 转义执行的命令将命令输出直接发送到终端，而不是发送到 `mdb`。

变量 `variable` 是变量名称、对应的整数值以及一组属性。变量名是由字符、数字、下划线或句点组成的序列。可以使用 `> dcmd` 或 `::typeset dcmd` 为变量赋值，变量的属性可以使用 `::typeset dcmd` 来处理。每个变量的值用一个 64 位无符号整数表示。变量可以具有以下一个或多个属性：只读（不能由用户修改）、持久（不能由用户取消设置）和已标记（用户定义的指示符）。

以下变量定义为持久变量：

0

使用 `/`、`\`、`?` 或 `= dcmd` 列显的最近值。

9

用于 `$< dcmd` 的最近计数。

b

数据区的根基的虚拟地址。

d

数据区大小（以字节为单位）。

e

入口点的虚拟地址。

m

目标的主目标文件的初始字节数（幻数），如果尚未读取目标文件，则为零。

t

文本区大小（以字节为单位）。

hits

与匹配软件事件说明符匹配的次数。请参见下文的事件回调。

thread

当前代表线程的线程标识符。标识符的值取决于当前目标使用的线程模型。请参见下文的线程支持。

此外，`mdb` 内核和进程目标将代表线程寄存器集的当前值导出为命名变量。这些变量的名称取决于目标的平台和指令集体系结构。

符号名称解析

按照上面的语法说明中的解释，表达式上下文中存在的符号标识符的计算结果为此符号的值。该值通常表示与目标虚拟地址空间中的符号关联的存储的虚拟地址。目标可以支持多个符号表，包括但不限于针对多个装入目标文件中每个目标文件（例如用户进程中的共享库或者 Solaris 内核中的内核模块）的主可执行符号表、主动态符号

表、运行时链接编辑器符号表以及标准和动态符号表。目标通常先搜索主可执行符号表，然后搜索一个或多个其他符号表。请注意，ELF 符号表只包含外部、全局和静态符号的条目；自动符号不显示在由 `mdb` 处理的符号表中。

此外，`mdb` 提供了私有的用户定义符号表，该表将在搜索任何目标符号表之前进行搜索。私有符号表最初是空的，可以使用 `::nmadd` 和 `::nmdel dcmd` 处理。`::nm -P` 选项可用于显示私有符号表的内容。用户可以使用私有符号表为原始程序中缺少的或已删除的程序函数或数据创建符号定义。然后，只要 `mdb` 将符号名称转换为地址，或者将地址转换为最接近的符号，就会使用这些定义。

由于目标包含多个符号表，并且每个符号表可以包含来自多个目标文件的符号，所以可以存在同名的不同符号。`mdb` 使用反引号 (') 字符作为符号名称作用范围运算符，使程序员能够在这种情况下获取所需符号的值。程序员可以将用于解析符号名称的作用范围指定为以下值之一：`object' name`；`file' name` 或 `object' file' name`。目标文件标识符指装入目标文件的名称。文件标识符指在指定目标文件的符号表中具有 `STT_FILE` 类型的符号的源文件的基名。目标文件标识符的解释取决于目标类型。

`mdb` 内核目标需要 *object* 来指定已装入内核模块的基名。例如，符号名称

```
specfs'_init
```

计算结果为 `specfs` 内核模块中 `_init` 符号的值。

`mdb` 进程目标需要 *object* 来指定可执行文件的名称或者已装入共享库的名称。可以采用以下形式之一：

1. 完全匹配（即完整路径名）：`/usr/lib/libc.so.1`
2. 精确基名匹配：`libc.so.1`
3. 初始基名，一直匹配到“.”后缀之前的内容：`libc.so` 或 `libc`
4. 文本字符串 `a.out` 可接受作为可执行文件的别名。

进程目标也可以接受在上述四种形式中的任意一种前面添加可选的链接图 ID (`lmid`)。可以在初始的“LM”后面跟随十六进制的链接图 ID，然后再跟随一个反引号，以此指定 `lmid` 前缀。例如，符号名称

```
LM0'libc.so.1'_init
```

计算结果为 `_init` 符号的值，该符号位于在链接图 0 (`LM_ID_BASE`) 上装入的 `libc.so.1` 库中。如果相同库在多个链接图上装入，则链接图说明符对于解决符号命名冲突是必需的。有关链接图的更多信息，请参见《[链接程序和库指南](#)》和 `dlopen(3C)`。在根据 `showlmid` 选项的设置列显符号时，将显示链接图标识符，如“选项”部分所述。

如果符号与十六进制整数值之间出现命名冲突，`mdb` 在将不确定的标记求值为整数值之前，将先尝试将该标记求值为符号。例如，标记 `f` 可以引用在十六进制（缺省基数）中指定的十进制整数值 `15`，也可以引用目标符号表中名为 `f` 的全局变量。如果存在具有不确定名称的符号，则可以使用显式 `0x` 或 `0X` 前缀来指定整数值。

dcmd 和遍历器名称解析

如前面所述，每个 `mdb dmod` 提供了一组 `dcmd` 和遍历器。`dcmd` 和遍历器是在两个不同的全局名称空间中跟踪的。`mdb` 还会跟踪与每个 `dmod` 关联的 `dcmd` 和遍历器名称空间。在一个指定的 `dmod` 中不允许具有同名的 `dcmd` 或遍历器：存在此类命名冲突的 `dmod` 无法装入。在全局名称空间中，允许来自不同 `dmod` 的 `dcmd` 或遍历器之间的名称冲突。出现冲突时，要装入的第一个具有该特定名称的 `dcmd` 或遍历器将在全局名称空间中优先。备用定义按照装入顺序保存在列表中。反引号字符 (‘) 可以在 `dcmd` 或遍历器名称中用作作用范围运算符来选择备用定义。例如，如果 `dmod m1` 和 `m2` 均提供了一个 `dcmd d`，并且 `m1` 在 `m2` 之前装入，则：

```
::d
    执行 m1 的 d 定义。

::m1'd
    执行 m1 的 d 定义。

::m2'd
    执行 m2 的 d 定义。
```

如果模块 `m1` 现在已卸载，则全局定义列表中的下一个 `dcmd (m2'd)` 将提升为具有全局可见性。使用下面所述的 `::which dcmd` 可以确定 `dcmd` 或遍历器的当前定义。使用 `::which -v` 选项可以显示全局定义列表。

dcmd 流水线

使用 `|` 运算符可以将 `dcmd` 组合成为流水线。流水线的作用是从一个 `dcmd` 或遍历器向另一个 `dcmd` 或遍历器传递值列表，通常为虚拟地址。流水线阶段可用于将指针从一种类型的数据结构映射到对应数据结构的指针，以便对地址列表进行排序或者选择具有特定属性的结构地址。

`mdb` 按照从左到右的顺序执行流水线中的各个 `dcmd`。最左侧的 `dcmd` 使用点的当前值执行，或者使用在命令开头由显式表达式指定的值执行。遇到 `|` 运算符时，`mdb` 在左侧的 `dcmd` 输出与 `mdb` 解析器之间创建流水线（共享缓冲区），并创建空的值列表。当 `dcmd` 执行时，相应的标准输出将放在流水线中，然后由解析器使用和求值，等同于 `mdb` 从标准输入读取了此数据。每一行必须包含使用**换行**或分号 (;) 终止的算术表达式。表达式的值将附加到与流水线关联的值列表中。如果检测到语法错误，流水线将异常中止。

当位于 `|` 运算符左侧的 `dcmd` 完成时，将使用与流水线关联的值列表来调用位于 `|` 运算符右侧的 `dcmd`。对于列表中的每个值，点将设置为该值，并且执行右侧的 `dcmd`。只有流水线中最右侧的 `dcmd` 会将输出列显到标准输出中。如果流水线中的任何 `dcmd` 向标准错误生成了输出，这些消息将直接列显到标准错误，而不作为流水线的一部分进行处理。

信号处理

调试器忽略 `PIPE` 和 `QUIT` 信号。`INT` 信号用于异常中止当前正在执行的命令。调试器拦截 `ILL`、`TRAP`、`EMT`、`FPE`、`BUS` 和 `SEGV` 信号并提供对这些信号的特殊处理。如果异步生成了这些信号中的任意一个（即从其他使用 `kill(2)` 的进程传送的信号），`mdb` 会将信号恢复到缺省处理和转储核心。但是，如果这些信号中的任意一个由调试器进程本身同步生成，并且来自外部装入 `dmod` 的 `dcmd` 当前正在执行，并且标准输入为终端，则 `mdb` 提供了一个选项菜单，允许用户选择强制执行核心转储、退出而不生成核

心转储、停止以便由调试器附加或者尝试恢复。恢复选项将异常中止所有活动命令并卸载出现故障时相应的 dcmd 处于活动状态的 dmod。随后，它可由用户重新装入。恢复选项可以为错误频发的 dcmd 提供有限的保护。有关与恢复选项相关的风险的信息，请参见下文的“警告”部分，使用错误恢复机制。

命令重新输入 从终端设备输入的最后一个 HISTSIZE（缺省值为 128）命令的文本保存在内存中。接下来介绍的内嵌编辑工具提供了用于从历史记录列表中搜索和提取元素的键映射。

内嵌编辑 如果标准输入为终端设备，则 mdb 提供了一些简单的 emacs 风格的工具，这些工具可用于编辑命令行。编辑模式下的 search、previous 和 next 命令提供了对历史记录列表的访问。在搜索时，只匹配字符串，不匹配模式。在下表中，控制字符的表示法为在插入记号 (^) 后面跟随以大写形式显示的字符。转义序列的表示法为后跟有字符的 M-。例如，M-f（发音为 meta-eff）的输入方法为：按下 ESC，然后按下 'f'；在支持 Meta 键的键盘上则按下 Meta，然后按下 'f'。使用回车或换行来提交和执行命令行。编辑命令包括：

^F

将光标向前（向右）移动一个字符。

M-f

将光标向前移动一个词。

^B

将光标向后（向左）移动一个字符。

M-b

将光标向后移动一个词。

^A

将光标移到行的开头。

^M

将光标移到行的结尾。

^D

如果当前行不为空，则删除当前字符。如果当前行为空，则 ^D 将指示 EOF，并且调试器将退出。

M-^H

（Meta 退格）删除上一个词。

^K

删除光标至行尾之间的内容。

^L

清除屏幕并重新列显当前行。

^T

将当前字符与下一个字符换位。

^N

从历史记录中提取下一个命令。每次输入 **^N** 时，将按时间检索下一个命令。

^P

从历史记录中提取上一个命令。每次输入 **^P** 时，将按时间检索上一个命令。

^R[*string*]

在历史记录中反向搜索包含 *string* 的上一个命令行。字符串应使用回车或换行终止。如果省略 *string*，则检索包含最近使用的字符串的上一个历史记录元素。

编辑模式还会将以下用户定义的序列解释为编辑命令。用户定义的序列可以使用 [stty\(1\)](#) 命令读取或修改。

erase

用户定义的删除字符（通常为 **^H** 或 **^?**）。删除上一字符。

intr

用户定义的中断字符（通常为 **^C**）。异常中止当前命令并列显新的提示符。

kill

用户定义的中止字符（通常为 **^U**）。中止当前整个命令行。

quit

用户定义的退出字符（通常为 **^**）。退出调试器。

suspend

用户定义的暂停字符（通常为 **^Z**）。暂停调试器。

werase

用户定义的词删除字符（通常为 **^W**）。删除前面的词。

如果键盘支持带有方向键的小键盘，**mdb** 会将以下击键解释为编辑命令：

向上箭头

从历史记录中提取上一个命令（与 **^P** 相同）。

向下箭头

从历史记录中提取下一个命令（与 **^N** 相同）。

左箭头

将光标向后移动一个字符（与 **^B** 相同）。

右箭头

将光标向前移动一个字符（与 **^F** 相同）。

输出页面调度程序

mdb 提供了内置的输出页面调度程序。如果调试器的标准输出是终端设备，将启用输出页面调度程序。每次执行命令时，**mdb** 在生成了满屏输出后将暂停，并显示页面调度程序提示符：

```
>> More [<space>, <cr>, q, n, c, a] ?
```

页面调度程序识别以下键序：

空格

显示下一个满屏输出。

a, A

中止当前顶级命令并返回到提示符。

c, C

继续显示输出，不在每次满屏时停止，直至当前顶级命令完成。

n, N, **换行, 回车**

显示下一行输出。

q, Q, ^C, ^\

仅退出（异常中止）当前 `dcmd`。

格式化 dcmd

`/`、`\`、`?`和`=`元字符用于表示特殊输出格式的 `dcmd`。这些 `dcmd` 中的每一个可接受由一个或多个格式字符、重复计数或括起字符串组成的参数列表。格式字符是下表中显示的 ASCII 字符之一。格式字符用于从目标中读取并格式化数据。重复计数是在格式字符之前的正整数，始终按照 Base 10（十进制）来解释。重复计数也可以指定为使用方括号括起的表达式，前面带有美元符号（`$[]`）。字符串参数必须使用双引号括起来（`" "`）。格式参数之间不必留有空格。

格式化 `dcmd` 包括：

`/`

从由点指定的虚拟地址开始，显示来自目标的虚拟地址空间的数据。

`\`

从由点指定的物理地址开始，显示来自目标的物理地址空间的数据。

`?`

从与由点指定的虚拟地址对应的目标文件位置开始，显示来自目标的主目标文件的数据。

`=`

以各个指定的数据格式显示点本身的值。因此，`= dcmd` 在基数之间转换和执行算术运算时非常有用。

除了点之外，`mdb` 还可以跟踪另一种称为**增量**的全局值。增量表示点与上一个格式化 `dcmd` 读取的所有数据后面的地址之间的距离。例如，如果执行了点等于地址 A 的格式化 `dcmd`，并且显示 4 个字节的整数，则在此 `dcmd` 完成后，点仍旧为 A，但增量设置为 4。`+` 字符（在上面的**算术扩展**中介绍）现在的计算结果为值 `A + 4`，可用于将点重置为后续 `dcmd` 的下一个数据目标文件的地址。

大部分格式字符将增大增量的值，增加的大小为与数据格式大小对应的字节数，如表中所示。格式字符的表可以使用 `::formats dcmd` 在 `mdb` 中显示。格式字符包括：

`+` 将点增加计数值（可变大小）

-	将点减少计数值（可变大小）
B	十六进制整数（1个字节）
C	使用C字符表示法的字符（1个字节）
D	十进制有符号整数（4个字节）
E	十进制无符号长整数（8个字节）
F	双精度数（8个字节）
G	八进制无符号长整数（8个字节）
H	交换字节，短整数（4个字节）
I	地址和反汇编指令（可变大小）
J	十六进制无符号长整数（8个字节）
K	十六进制 <code>uintptr_t</code> （4个或8个字节）
N	换行
O	八进制无符号整数（4个字节）
P	符号（4个或8个字节）
Q	八进制有符号整数（4个字节）
R	二进制整数（8个字节）
S	使用C字符串表示法的字符串（可变大小）
T	水平制表符
U	十进制无符号整数（4个字节）
V	十进制无符号整数（1个字节）
W	缺省基数无符号整数（4个字节）
X	十六进制整数（4个字节）
Y	已解码的 <code>time32_t</code> （4个字节）
Z	十六进制无符号长整数（8个字节）
^	将点减少增量*计数值（可变大小）
a	将点作为符号+偏移
b	八进制无符号整数（1个字节）
c	字符（1个字节）
d	十进制有符号短整数（2个字节）

e	十进制有符号长整数（8 个字节）
f	浮点数（4 个字节）
g	八进制有符号长整数（8 个字节）
h	交换字节（2 个字节）
i	反汇编指令（可变大小）
n	新行
o	八进制无符号短整数（2 个字节）
p	符号（4 个或 8 个字节）
q	八进制有符号短整数（2 个字节）
r	空格
s	原始字符串（可变大小）
t	水平制表符
u	十进制无符号短整数（2 个字节）
v	十进制有符号整数（1 个字节）
w	缺省基数无符号短整数（2 个字节）
x	十六进制短整数（2 个字节）
y	已解码的 <code>time64_t</code> （8 个字节）

`/`、`\` 和 `?` 格式化 `dcmd` 也可用于写入目标的虚拟地址空间、物理地址空间或目标文件，方法是将以下修饰符之一指定为第一个格式字符，然后指定词列表，该词列表可以直接为值，也可以为用方括号括起的表达式，前面带有美元符号（`$[1]`）。

写入修饰符包括：

- v** 将每个表达式值的最低字节写入从点所指定的位置开始的目标。
- w** 将每个表达式值的最低 2 个字节写入从点所指定的位置开始的目标。
- W** 将每个表达式值的最低 4 个字节写入从点所指定的位置开始的目标。
- Z** 将每个表达式值的全部 8 个字节写入从点所指定的位置开始的目标。

`/`、`\` 和 `?` 格式化 `dcmd` 也可分别用于搜索目标虚拟地址空间、物理地址空间和目标文件中的特定整数值，方法是将以下修饰符之一指定为第一个格式字符，然后指定值和

可选的掩码。值和掩码均可以直接指定为值，也可以指定为使用方括号括起的表达式，在前面使用美元符号。如果只指定了值，mdb 将读取合适大小的整数，并在包含匹配值的地址处停止。如果指定了值 *V* 和掩码 *M*，mdb 将读取合适大小的整数，并在包含值 *X*（其中 $(X \& M) == V$ ）的地址处停止。dcmd 完成之后，点将更新为包含匹配项的地址。如果找不到匹配项，点将保留为最后一个读取的地址。

搜索修饰符包括：

- I 搜索指定的 2 字节值。
- L 搜索指定的 4 字节值。
- M 搜索指定的 8 字节值。

请注意，对于用户和内核目标而言，地址空间通常由一组不连续的段组成。从没有对对应段的地址进行读取是非法的。如果搜索到达段边界但没有找到匹配项，将在由于读取超出段边界结尾而失败时异常中止。

执行控制

mdb 提供了用于控制和跟踪实时运行程序的执行的工具。目前，只有用户进程目标提供对执行控制的支持。mdb 提供了简单的执行控制模型：目标进程可以使用 `::run` 从调试器内部启动，mdb 也可以使用 `:A`、`::attach` 或 `-p` 命令行选项附加到现有进程，如下文所述。用户可以指定跟踪的软件事件列表。每次目标进程中发生跟踪的事件时，目标中的所有线程将停止，触发事件的线程将被选定作为代表线程，同时控制权将返回给调试器。将目标程序设置为运行之后，可以通过键入用户定义的中断字符（通常为 `^c`）将控制权异步返回给调试器。

软件事件是目标程序中由调试器观察的状态转换。例如，调试器可以观察程序计数器寄存器到关注值（断点）的转换或发出特殊信号。

软件事件说明符是对软件事件类的说明，由调试器用于对目标程序进行检测，以便观察这些事件。`::events dcmd` 用于列出软件事件说明符。每个事件说明符有一组标准属性与之关联，如下文 `::events` 中所述。

调试器可以观察大量不同的软件事件，包括断点、监视点、信号、计算机故障和系统调用。可以使用 `::bp`、`::fltp`、`::sigbp`、`::sysbp` 或 `::wp` 创建新说明符。每个说明符都具有关联的回调（要执行的 mdb 命令字符串，就像在命令提示符中键入了这些内容）和属性集合，如下文所述。可以为同一个事件创建任意数量的说明符，每个说明符可以具有不同的回调和属性。可以使用 `::events dcmd` 显示跟踪事件和对应事件说明符属性的当前列表。事件说明符属性在下面的 `::events` 和 `::evset dcmd` 说明中定义。

执行控制内置 `dcmd`（在下面介绍）始终可用，但当应用于不支持执行控制的目标时，将发出一条错误消息，指示不受支持。有关执行、附加、释放及作业控制与调试器执行控制进行交互的详细信息，请参见下文的“附注”部分。

事件回调

使用 `::evset dcmd` 和事件跟踪 `dcmd` 可以将事件回调（使用 `-c` 选项）与各个事件说明符关联。事件回调是表示 `mdb` 命令的字符串，在目标中出现对应事件时执行。这些命令的执行与在命令提示符中键入这些内容时相同。在执行各个回调之前，点变量设置为代表线程的程序计数器的值，而 `"hits"` 变量设置为此说明符匹配的次数（包括当前匹配）。

如果事件回调本身包含一个或多个继续在目标中执行的命令（例如，`::cont` 或 `::step`），则这些命令不立即在目标中执行，而是等待再次停止。相反，在事件回调内部，继续 `dcmd` 将注意到继续操作当前处于暂挂状态，然后立即返回。因此，如果一个事件回调中包含了多个 `dcmd`，则步骤或继续 `dcmd` 应为最后一个指定的命令。在执行所有事件回调之后，如果所有匹配事件回调请求了继续，目标将立即恢复执行。如果请求了冲突的继续操作，具有最高优先级的操作将确定要发生的继续操作类型。优先级的顺序从高到低依次为：单步执行，步过（下一步），步出，继续。

线程支持

`mdb` 提供了用于检查与目标关联的各个线程的栈和寄存器的工具。持久性 `"thread"` 变量包含当前代表线程标识符。线程标识符的格式取决于目标。`::regs` 和 `::fpregs dcmd` 可用于检查代表线程的寄存器集，如果其他线程的寄存器集当前可用，也可进行检查。此外，代表线程的寄存器集将导出为一组命名变量。用户可以将 `> dcmd` 应用于对应的命名变量，从而修改一个或多个寄存器的值。

`mdb` 内核目标将导出对应内部线程结构的虚拟地址，作为指定线程的标识符。《Oracle Solaris Modular Debugger Guide》提供了有关 Solaris 内核中线程调试支持的更多信息。`mdb` 进程目标提供了相应的支持，可以检查使用本机 `lwp_*` 接口、`/usr/lib/libthread.so` 或 `/usr/lib/lwp/libthread.so` 的多线程用户进程。调试实时用户进程时，`mdb` 将检测是单线程进程 `dlopen` 还是关闭 `libthread`，并自动即时调整线程模型的视图。根据应用程序使用的线程模型，进程目标线程标识符分别对应于代表线程的 `lwpid_t`、`thread_t` 或 `pthread_t`。

如果 `mdb` 在调试用户进程目标并且目标使用了编译器支持的线程局部存储，则 `mdb` 自动将引用线程局部存储的符号名称求值为与当前代表线程对应的存储的地址。`::tls` 内置 `dcmd` 可用于显示除代表线程之外的其他线程的符号值。

内置 dcmd

`mdb` 提供了一组已定义的内置 `dcmd`。其中一些 `dcmd` 仅适用于特定目标：如果 `dcmd` 不适用于当前目标，则将失败并列显消息，指示 `"command is not supported by current target"`。在许多情况下，`mdb` 为旧的 `adb(1)` `dcmd` 名称提供了等效助记符 (`::identifier`)。例如，提供的 `::quit` 与 `$q` 等同。有经验的 `adb(1)` 程序员，或者喜欢简洁或崇尚高深莫测编程的程序员，会偏好使用内置的 `$` 或 `:` 形式。而刚开始接触 `mdb` 的程序员可能会偏好使用更详细的 `::` 形式。内置形式按字母顺序排列。如果 `$` 或 `:` 形式具有等效的 `::identifier`，则将在 `::identifier` 形式下方显示。内置 `dcmd` 包括：

`> variable-name`

`>/modifier/variable-name`

将点值赋值给指定的命名变量。一些变量为只读，无法修改。如果 `>` 后面跟随前后带有 `//` 的修饰符字符，则该值将作为赋值的一部分进行修改。修饰符字符包括：

- c 无符号字符值（1 个字节）
- s 无符号短整数值（2 个字节）
- i 无符号整数值（4 个字节）
- l 无符号长整数值（32 位系统下为 4 个字节，64 位系统下为 8 个字节）

请注意，这些运算符不执行类型转换，而是提取指定数量的低位字节（在小尾数法体系结构上）或高位字节（在大尾数法体系结构上）。提供修饰符是为了实现向后兼容性；应改为使用 `mdb */modifier/` 和 `%/modifier/` 语法。

`$< macro-name`

从指定的宏文件读取和执行命令。可以按绝对路径或相对路径提供文件名。如果文件名是简单名称（即，其中不包含 '/'），`mdb` 将在宏文件头文件路径中搜索文件。如果当前在处理另一个宏文件，则此文件将关闭，并使用新文件替换。

`$<< macro-name`

从指定的宏文件读取和执行命令（与使用 `$<` 一样），但不关闭当前打开的宏文件。

`$?`

如果是用户进程或核心文件，则列显进程 ID 和当前信号，然后列显代表线程的常规寄存器集。

`[address] $C [count]`

列显 C 栈回溯，包括栈帧指针信息。如果 `dcmd` 前面是显式地址，将显示在此虚拟内存地址开始的回溯。否则，将显示代表线程的栈。如果将可选计数值指定为参数，则在输出中为每个栈帧显示不超过 `count` 个参数。

`[base] $d`

获取或设置缺省输出基数。如果 `dcmd` 的前面为显式表达式，则缺省输出基数将设置为指定的 `base`；否则当前基数将按照 Base 10（十进制）列显。缺省基数为 Base 16（十六进制）。

`$e`

列显所有类型为 `target` 文件或函数的已知外部（全局）符号、符号的值以及存储在目标虚拟地址空间中此位置的前 4 个（32 位 `mdb`）或 8 个（64 位 `mdb`）字节。`::nm dcmd` 提供了更多灵活选项来显示符号表。

`$P prompt-string`

将提示符设置为指定的 `prompt-string`。缺省提示符为 `'>'`。提示符也可以使用 `::set -P` 或 `-P` 命令行选项来设置。

distance \$s

获取或设置与地址到符号名称转换的 *distance* 匹配的符号。与距离匹配的符号模式将在“选项”部分随 `-s` 命令行选项一起介绍。与距离匹配的符号也可以使用 `::set -s` 选项修改。如果未指定距离，则显示当前设置。

\$v

列显具有非零值的命名变量的列表。`::vars dcmd` 提供了用于列出变量的其他选项。

width \$w

将输出页的 *width* 设置为指定值。通常，此命令并不是必需的，因为 `mdb` 会向终端查询其宽度并处理大小调整事件。

\$W

重新打开目标以便执行写入，等同于在命令行上使用 `-w` 选项执行 `mdb`。写入模式也可以使用 `::set -w` 选项启用。

`[pid]::attach [core | pid]`

`[pid]:A [core | pid]`

如果用户进程目标处于活动状态，则附加到指定的进程 ID 或核心文件并进行调试。核心文件路径名应以字符串参数形式指定。进程 ID 可以按字符串参数形式指定，也可以指定为 `dcmd` 前面的表达式的值。请记住，缺省基数为十六进制，因此将使用 `pgrep(1)` 或 `ps(1)` 获取的十进制 PID 指定为表达式时，应该在前面加上 `"0t"`。

`[address]::bp [-/-dDesT] [-c cmd] [-n count] sym ...`

`address:b [cmd ...]`

在指定位置设置断点。`::bp dcmd` 在每个指定的地址或符号处设置断点，包括 `dcmd` 前某个显式表达式指定的可选位置，以及 `dcmd` 之后的每个字符串或即时值。该参数可以为符号名称，也可以是表示所需特定虚拟地址的直接值。如果指定了符号名称，则它可以引用目标进程中目前无法求值的符号。即，它可以由尚未打开的装入目标文件的目标文件名和其中的函数名组成。在这种情况下，断点将延迟，在目标中不活动，直至装入了与指定名称匹配的目标文件。打开装入目标文件时会自动启用断点。在共享库中定义的符号上的断点应始终使用符号名称设置，而不能使用地址表达式，因为地址可以引用对应的过程链接表 (PLT) 条目而非实际符号定义。如果 PLT 条目随后解析为实际符号定义，则可以使用运行时链接编辑器覆盖在 PLT 条目上设置的断点。`-d`、`-D`、`-e`、`-s`、`-t`、`-T`、`-c` 和 `-n` 选项的含义与在 `::evset dcmd` 中相同，如下文所述。如果使用了 `dcmd` 的 `:b` 形式，则仅在由 `dcmd` 前面的表达式指定的虚拟地址上设置断点。`:b dcmd` 之后的参数会串联在一起形成回调字符串。如果此字符串包含元字符，必须引用该字符串。

`::cat filename ...`

串联并显示文件。可以按相对路径名或绝对路径名指定各个文件名。文件内容将列显到标准输出，但无法传递到输出页面调度程序。此 `dcmd` 旨在随 `|` 运算符一起使用；程序员可以使用存储在外部文件中的地址列表来启动流水线。

`::cont [SIG]`

`:c [SIG]`

暂停调试器，继续执行目标程序，并等待直至出现所关注的软件事件之后终止或停止。如果由于调试器已附加到正在运行的程序并且启用了 `-o nostop` 选项，从而导致目标已在运行，此 `dcmd` 将等待目标在出现所关注事件后终止或停止。如果将可选的信号名称或编号（请参见 `signal.h(3HEAD)`）指定为参数，则该信号将在恢复执行过程中立即传输到目标。如果跟踪了 `SIGINT` 信号，可以键入用户定义的中断字符（通常为 `^C`），将控制权异步返回给调试器。此 `SIGINT` 信号将自动清除，下次继续时目标将观察不到该信号。如果当前没有目标程序在运行，`::cont` 将启动新的程序开始运行，如同使用 `::run` 一样。

`address ::context`

`address $p`

到指定进程的上下文切换。上下文切换操作仅在使用内核目标时有效。进程上下文使用它在内核虚拟地址空间中的 `proc` 结构的 `address` 来指定。使用特殊上下文地址 `"0"` 来表示内核本身的上下文。在检查故障转储期间，只有当转储中包含指定用户进程的物理内存页（与只有内核页相对）时，`mdb` 才能执行上下文切换。可以使用 `dumpadm(1M)` 配置内核故障转储工具，以便转储所有页或当前用户进程的页。`::status dcmd` 可用于显示当前故障转储的内容。

用户请求从内核目标进行上下文切换时，`mdb` 构建表示指定用户进程的新目标。进行切换之后，新目标在全局级别插入 `dcmd`：因此，`/ dcmd` 现在将格式化并显示来自用户进程虚拟地址空间的数据，`::mappings dcmd` 将显示用户进程地址空间中的映射等等。内核目标可以通过执行 `0::context` 来还原。

`::dcmds`

列出可用 `dcmd` 并列显每个 `dcmd` 的简要说明。

`[address] ::delete [id | all]`

`[address] :d [id | all]`

删除具有指定 ID 号的事件说明符。ID 号参数在缺省情况下按照十进制解释。如果在 `dcmd` 前面指定了可选的地址，将删除与指定虚拟地址关联的所有事件说明符（例如，所有影响该地址的断点或监视点）。如果指定了特殊参数 `"all"`，将删除所有事件说明符，但标记为粘滞（`T` 标记）的说明符除外。`::events dcmd` 显示当前事件说明符列表。

`[address] ::dis [-fw] [-n count] [address]`

在由最后一个参数指定的 `address` 处或周围，或者在点的当前值处开始反汇编。如果地址与已知函数的开头匹配，则将反汇编整个函数。否则，将列显指定地址前后的指令“窗口”以便提供上下文。缺省情况下，从目标的虚拟地址空间读取指令。如果存在 `-f` 选项，则改为从目标的目标文件读取指令。如果调试器当前未附加到实时进程、核心文件或故障转储，缺省情况下将启用 `-f` 选项。`-w` 选项可用于强制“窗口”模式，即使地址是已知函数的开头也是如此。窗口大小的缺省值为十个指令，可以使用 `-n` 选项显式指定指令数量。

`::disasms`

列出可用的反汇编程序模式。初始化目标时，`mdb` 尝试选择合适的反汇编程序模式。用户可以使用 `::dismode dcmd` 将模式更改为列出的任意模式。

```
::dismode [ mode ]
```

```
$V [ mode ]
```

获取或设置反汇编程序模式。如果未指定参数，将列显当前反汇编程序模式。如果指定了 *mode* 参数，会将反汇编程序切换到指定的模式。可以使用 `::disasms dcmd` 显示可用反汇编程序的列表。

```
::dmods [ -l ] [ module-name ]
```

列出已装入的调试器模块。如果指定了 `-l` 选项，将在各个 `dmod` 名称下列显相关联的 `dcmd` 和遍历器的列表。可以通过将特定 `dmod` 名称指定为附加参数，将输出限制为该 `dmod`。

```
[ address ] ::dump [ -eqrstu ] [ -f|-p ]
```

```
[ -g bytes ] [ -w paragraphs ]
```

对于包含由点指定的地址的内存，列显内存的 16 字节对齐区域的十六进制和 ASCII 内存转储。如果为 `::dump` 指定了重复计数，这将解释为要转储的字节数而非重复次数。`::dump dcmd` 还可以识别以下选项：

```
-e
```

字节存储顺序的调整。`-e` 选项采用 4 字节词。`-g` 选项可用于更改缺省词大小。

```
-f
```

从与指定虚拟地址对应的目标文件位置读取数据，而不是从目标的虚拟地址空间读取。如果调试器当前未附加到实时进程、核心文件或故障转储，缺省情况下将启用 `-f` 选项。

```
-g bytes
```

显示 *bytes* 组中的字节。缺省组大小为 4 个字节。组大小必须为除以行宽的 2 的幂。

```
-p
```

将 *address* 解释为目标地址空间中的物理地址位置而非虚拟地址。

```
-q
```

不列显数据的 ASCII 解码。

```
-r
```

相对于开始地址对行进行编号，而不是使用每一行的显式地址。此选项暗含 `-u` 选项。

```
-s
```

省略重复行。

```
-t
```

只读取并显示指定地址的内容，而不是读取和列显整行。

```
-u
```

取消对齐输出而不是将输出在段落边界对齐。

```
-w paragraphs
```

按每行 16 字节的段落来显示段落。*paragraphs* 的缺省数量为 1。`-w` 可接受的最大值为 16。

`::echo [string | value ...]`

将参数列显到标准输出，以空格分隔，并且以**换行**终止。用 `$()` 括起的表达式将计算得出某个值，并按照缺省基数列显。

`::eval command`

对指定的字符串求值并作为命令执行。如果命令包含元字符或空格，则应括在双引号或单引号中。

`::events [-av]`

`$b [-av]`

显示软件事件说明符的列表。每个事件说明符都分配有一个唯一的 ID 号，可用于在以后删除或修改事件说明符。调试器也可以针对自己的内部事件启用跟踪。这些事件只有在存在 `-a` 选项时才显示。如果存在 `-v` 选项，将显示包括任何说明符不活动原因在内的更为详细的内容。以下为示例输出：

```
> ::events
  ID S TA HT LM Description                Action
-----
[ 1 ] - T  1  0 stop on SIGINT                  -
[ 2 ] - T  0  0 stop on SIGQUIT                -
[ 3 ] - T  0  0 stop on SIGILL                  -
...
[ 11] - T  0  0 stop on SIGXCPU                 -
[ 12] - T  0  0 stop on SIGXFSZ                 -
[ 13] -   2  0 stop at libc'printf             ::echo printf
>
```

下表说明了每一列的含义。使用 `::help events` 可以获取有关此信息的汇总。

ID

事件说明符标识符。如果启用了说明符，标识符将显示在方括号 `[]` 中；如果禁用了说明符，标识符将显示在圆括号 `()` 中；如果目标程序当前已由于与指定说明符匹配的事件而停止，标识符将显示在尖括号 `<>` 中。

S

事件说明符状态。状态为以下符号之一：

-

事件说明符处于空闲状态。未运行任何目标程序时，所有说明符都处于空闲状态。目标程序正在运行时，如果无法对某个说明符求值，则该说明符处于空闲状态（例如，尚未装入的共享目标文件中的延迟断点）。

+

事件说明符处于活动状态。目标继续执行时，调试器将检测到此类型的事件。

*

事件说明符已设置。此状态表示目标当前正在启用了检测该事件类型的状态下运行。只有在调试器使用 `-o nostop` 选项附加到正在运行的程序时，此状态才可见。

! 由于操作系统错误，事件说明符未设置。::events -v 选项可用于显示有关检测失败原因的更多信息。

TA

临时、粘滞和自动事件说明符属性。可以显示以下一个或多个符号：

t

事件说明符为临时，下次目标停止时将被删除（无论是否匹配）。

T

事件说明符为粘滞状态，不会由::delete all 或 :z 删除。该说明符可以通过在::delete 中显式指定 ID 号来删除。

d

命中计数等于命中限制时，将自动禁用事件说明符。

D

命中计数等于命中限制时，将自动删除事件说明符。

s

命中计数等于命中限制时，将自动停止目标。

HT

当前命中计数。此列显示自该事件说明符创建以来，对应的软件事件在目标中出现的次数。

LM

当前命中限制。此列显示使自动禁用、自动删除或自动停止行为生效的命中计数限制。这些行为可以使用::evset dcmd 配置，如下文所述。

描述

与指定说明符匹配的软件事件类型的说明。

操作

要在发生对应的软件事件时执行的回调字符串。此回调的执行与在命令提示符中键入该内容相同。

`[id] ::evset [-/-dDestT] [-c cmd] [-n count] id ...`

修改一个或多个软件事件说明符的属性。对于通过在 dcmd 前面的可选表达式以及在 dcmd 后面的可选参数列表进行标识的说明符，每个说明符都设置了属性。除非指定了显式基数，否则参数列表将解释为十进制整数列表。::evset dcmd 可以识别以下选项：

-d

命中计数达到命中限制时，禁用事件说明符。如果指定了 -d 形式的选项，将禁用此行为。禁用某个事件说明符之后，调试器将删除任何对应的检测，并忽略对应的软件事件，直至随后重新启用了该说明符。如果没有 -n 选项，将立即禁用说明符。

- D
命中计数达到命中限制时，删除事件说明符。如果指定了 -D 形式的选项，将禁用此行为。-D 选项优先级高于 -d 选项。命中限制可以使用 -n 选项配置。
- e
启用事件说明符。如果指定了 -e 形式的选项，将禁用该说明符。
- s
命中计数达到命中限制时，停止目标程序。如果指定了 -s 形式的选项，将禁用此行为。-s 行为可以通知调试器：其操作与每次执行说明符回调后发布 ::cont 相同，但第 N 次执行除外，N 是说明符命中限制的当前值。-s 选项优先级高于 -D 选项和 -d 选项。
- t
将事件说明符标记为临时。临时标识符在目标下次停止时将自动删除，不论是否是由于与指定说明符对应的软件事件而造成目标停止。如果指定了 -t 形式的选项，将删除临时标记。-t 选项优先级高于 -T 选项。
- T
将事件说明符标记为粘滞。粘滞说明符不能由 ::delete all 或 :z 删除。可以通过将相应说明符 ID 指定为 ::delete 的显式参数来删除它们。如果指定了 -T 形式的选项，将删除粘滞属性。缺省事件说明符集在最初均标记为粘滞。
- c
每次目标程序中发生对应的软件事件时，执行指定的 *cmd* 字符串。当前回调字符串可以使用 ::events 显示。
- n
将当前的命中限制值设置为 *count*。如果当前未设置命中限制，并且 -n 选项没有使用 -s 或 D，则将命中限制设置为 1。

使用 ::help evset 可以获取有关此信息的汇总。

```
:: files
```

```
$f
```

列显已知源文件的列表（类型为 STT_FILE 的符号显示在各种目标符号表中）。

```
[flt] :: fltbp [-/-dDestT] [-c cmd] [-n count] flt ...
```

跟踪指定的计算机故障。可以在 dcmd 前面使用可选的故障号来标识故障，也可以在 dcmd 后面使用故障名称或编号的列表来标识（请参见

<sys/fault.h>）。-d、-D、-e、-s、-t、-T、-c 和 -n 选项具有与用于 ::evset dcmd 时的相同意义。

```
[thread] :: fpregs
```

```
[thread] $x, $X, $y, $Y
```

列显代表线程的浮点寄存器集。如果指定了线程，将显示该线程的浮点寄存器。线程表达式应为上面的**线程支持**中所述的线程标识符之一。

::formats

列出可用于 /、\、? 和 = 格式化 dcmd 的输出格式字符。格式及这些字符的用法在上面的 **格式化 dcmd** 中介绍。

::grep command

对指定的命令字符串求值，如果点的新值非零，将列显点的旧值。如果 *command* 包含空格或元字符，则必须括起来。**::grep dcmd** 可在流水线中用于过滤地址列表。

::help [dcmd-name]

不带参数时，**::help dcmd** 将列显 mdb 中可用的帮助工具的简短说明。如果指定了 *dcmd-name*，mdb 将列显该 dcmd 的用法汇总。

signal :i

如果目标是实时用户进程，将忽略指定的信号并允许将它透明地传送到目标。从跟踪事件列表中删除跟踪指定信号传送的所有事件说明符。缺省情况下，忽略的信号集初始化为补充的信号集合，这些信号集合在缺省情况下导致进程转储核心（请参见 [signal.h\(3HEAD\)](#)），但在缺省情况下跟踪的 SIGINT 除外。

\$i

显示调试器忽略并由目标直接处理的信号的列表。可以使用 **::events dcmd** 获取跟踪的信号的信息。

[address] ::if [-p] {type member tests | [type] [at off] test}

对测试进行评估，如果测试为 true，则输出点的旧值。

-p

使用物理地址而非虚拟地址。

有两种类型的测试。第一种允许您测试结构或联合的成员的值得。第二种允许您测试相对于点的偏移。随后可以使用 AND 或 OR 联接这些测试以生成更复杂的测试。

下面从测试结构成员的示例开始介绍：

```
::if "struct foo" namep <> 0
```

此语句将在 foo 结构中的 namep 元素不为 0 时输出点的旧值。

```
::if "struct foo" namep <> 0 AND namep->name <> 0 AND
namep->name streq "bar"
```

如果由 foo 中的 namep 指向的结构中的 name 元素不为 0 且指向包含 bar 字符串的字符串，则以上语句将输出点的旧值。

```
::if "struct foo" name <> 0 AND name streq "bar" and value = 0x123
```

仅当名称为 bar 且 value 为 0x123 时，以上语句才会显示输出。

第二种形式的测试用于查看给定偏移处的类型是否具有给定值。以下是一个示例：

```
::if uint_t at 0x34 = 0x123
```

如果偏移 0x34 处的 `uint_t` 值是 0x123，以上语句会显示输出。同样，也可以使用 AND 或 OR 将该测试与其他测试联接起来。对于此种形式，`::if` 具有一些内部使用的内置类型，即使在没有任何符号类型信息时，也可以使用这些类型。这些类型包括：

```
char, uchar, short, ushort, int, uint, long, ulong,
longlong, ulonglong, pointer, addr
```

如果类型为 `addr`，则会测试实际地址，而不是该地址处的值。因此，语句：

```
::if addr at 0x34 = 0x100034
```

...仅在点+0x34 为 0x100034 时才报告点的值，而语句：

```
::if pointer at 0x34 = 0x100034
```

...仅在值处于点+0x34 时才输出。

更为有用的是，语句：

```
::if "char *" at 0x34 streq "foo"
```

...将在偏移 0x34 处所指向的字符串为 `foo` 时才输出点的旧值。

可能的测试包括：

```
=          - True if the values are equal.
<>        - True if the values are not equal.
<         - True if the value is less than the right hand value.
<=        - True if the value is less than or equal to the right
           hand value.
>         - True if the value is greater than the right hand value.
>=        - True if the value is greater than or equal to the right
           hand value.
&         - True if the values ANDed together are non zero.
^         - True if the values XORed together are non zero.
streq     - True if the strings exactly match.
strneq    - True if the strings don't match.
strcaseeq - True if the strings match case insensitively.
strncaseq - True if the strings do not match case insensitively.
strleneq  - True if the string is this length.
strlneq   - True if the string is not this length.
strlengt  - True if the string is longer than this.
strlenlt  - True if the string is shorter than this.
strleng  - True if the string is this long or longer.
strlenle  - True if the string is this long or shorter
```

测试的右侧可以是以下任意一项：

- 结构或联合中的元素。
- 符号的值。
- 绝对值。

- `<var`—变量 `var` 的值。
- `$expr`—对 `mdb` 表达式 `expr` 求值后点的值。该值必须使用双引号 (“”) 括起来。例如：

```
"${<var}=J"
```

...将获取变量 `var` 的值。

```
::kill
```

```
:k
```

如果目标为实时用户进程，则强制终止目标。存在调试器时，如果目标是由调试器使用 `::run` 创建的，也将强制终止该目标。

```
$l
```

如果目标是用户进程，将列显代表线程的 LWPID。

```
$L
```

如果目标是用户进程，将列显目标中各个 LWP 的 LWPID。

```
[ address ] ::list [-b back_member] [-p] [-L] [type] member [ variable-name ]
```

遍历链接列表数据结构中的元素，并列显列表中每个元素的地址。可以使用可选的地址指定列表中第一个元素的地址。否则，列表将采用点的当前值作为开头。类型参数必须指定 C 结构或联合类型，并且该参数用于说明列表元素的类型，以便 `mdb` 可以在合适大小的目标文件中读取。如果 `::list` 可以确定类型，则可以省略 `type`。`member` 参数用于命名 `type` 类型的 `member`，其中包含指向下一个列表元素的指针。`::list dcmd` 将持续重复执行，直到发生以下情况：遇到 NULL 指针、再次达到第一个元素（循环列表）、启用了循环检测 (-L) 且检测到循环、启用了反向成员检查 (-b) 且检测到错误的反向指针或在读取元素时发生错误。如果指定了可选的 `variable-name`，将为指定的变量分配在 `mdb` 调用流水线的下一阶段时遍历的每一步骤所返回的值。

```
-b back_member
```

验证 `back_member` 是否指向列表的上一成员。允许列表的第一个成员具有 NULL 反向指针。

```
-L
```

检查列表中的循环。将针对列表中的每个元素输出一次。

```
-p
```

从物理地址而非虚拟地址读取。

`::list dcmd` 只能用于特定的目标文件，即包含供 `mdb` 使用的符号调试信息的目标文件。有关详细信息，请参见下文“附注”部分中的“符号调试信息”。

```
::load [-s] module-name
```

装入指定的 `dmod`。可以按绝对路径或相对路径提供模块名称。如果 `module-name` 是简单名称（即不包含 '/'），`mdb` 将在模块库路径中搜索该名称。不能装入存在名称冲突的模块，必须先卸载现有模块。如果存在 `-s` 选项，`mdb` 将保持无提示运行，在找不到模块或无法装入模块时不发出任何错误消息。


```
::log [ -d | [ -e ] filename ]
```

```
$> [ filename ]
```

启用或禁用输出记录。mdb 提供了交互式日志工具，可以将输入命令和标准输出记录到文件，同时仍与用户交互。-e 选项允许记录到指定文件，如果未指定文件名，将重新允许记录到以前的日志文件。-d 选项可以禁用日志。如果使用 \$> dcmd，则在指定了文件名参数时启用日志；否则将禁用日志。如果指定的日志文件已存在，mdb 会将任意新日志输出附加到文件。

```
::map command
```

使用指定为字符串参数的 *command*，将点的值映射到对应值，然后列显点的新值。如果 *command* 包含空格或元字符，则必须括起来。::map dcmd 可在流水线中用于将地址列表转换为新的地址列表。

```
[ address ] ::mappings [ name ]
```

```
[ address ] $m [ name ]
```

在目标的虚拟地址空间中列显各个映射的列表，包括各个映射的地址、大小和说明。如果 dcmd 的前面为 *address*，mdb 将只显示包含指定地址的映射。如果指定了字符串 *name* 参数，mdb 将只显示与该说明匹配的映射。

```
::next [ SIG ]
```

```
:e [ SIG ]
```

单步执行目标程序的一个指令，但步过子例程调用。如果将可选的信号名称或编号（请参见 [signal.h\(3HEAD\)](#)）指定为参数，则该信号将在恢复执行过程中立即传输到目标。如果当前没有目标程序在运行，::next 将启动新的程序开始运行，如同使用 ::run 一样，并在第一个指令处停止。

```
[ address ] ::nm [ -DPdghnopuvx ] [ -t types ]
```

```
[ -f format ] [ object ]
```

列显与当前目标关联的符号表。如果在 dcmd 前面指定了可选的地址，将只显示与 *address* 对应的符号的符号表条目。如果指定了 *object*，将只显示此装入目标文件的符号表。::nm dcmd 还可以识别以下选项：

```
-D
```

列显 *.dynsym*（动态符号表）而非 *.symtab*。

```
-P
```

列显私有符号表而非 *.symtab*。

```
-d
```

以十进制列显值和大小字段。

```
-g
```

仅列显全局符号。

```
-h
```

隐藏标题行。

```
-n
```

按名称对符号进行排序。

- o
以八进制列显值和大小字段。
- p
以一系列 `::nmadd` 命令形式列显符号。此操作可以与 `-P` 一起使用以便生成宏文件，该文件随后可以使用 `$<` 读入调试器。
- u
仅列显未定义的符号。
- v
按值对符号进行排序。
- x
以十六进制列显值和大小字段。
- t *type*[,*type* ...]
仅列显指定类型的符号。有效的 *type* 参数字符串为：
 - noty
STT_NOTYPE
 - objt
STT_OBJECT
 - func
STT_FUNC
 - sect
STT_SECTION
 - file
STT_FILE
 - comm
STT_COMMON
 - tls
STT_TLS
 - regi
STT_SPARC_REGISTER
- f *format*[,*format* ...]
仅列显指定符号信息。有效 *format* 参数字符串为：
 - ndx
符号表索引
 - val
符号值

size
大小，以字节为单位

type
符号类型

bind
移动绑定

oth
其它

shndx
区段索引

name
符号名称

ctype
符号的 C 类型（如果已知）

obj
定义符号的目标文件

value :: **nmadd** [-fo] [-e end] [-s size] *name*

将指定的符号 *name* 添加到私有符号表。mdb 提供了私有的可配置符号表，可用于在目标的符号表上插入，如上面的**符号名称解析**中所述。::nmadd dcmd 还可以识别以下选项：

-e
将符号的大小设置为 *end - value*。

-f
将符号的类型设置为 STT_FUNC。

-o
将符号的类型设置为 STT_OBJECT。

-s
将符号的大小设置为 *size*。

::**nmde1** *name*
从私有符号表中删除指定的符号 *name*。

::**objects** [-v]
列显目标的虚拟地址空间的映射，只显示与每个已知装入目标文件的主映射（通常为文本区段）相对应的那些映射。-v 选项显示各个装入目标文件的版本。版本信息并非对所有装入目标文件都可用。对于没有版本信息的装入目标文件，在 -v 信息的输出中，将作为 "Unknown" 版本列出。

::offsetof member

输出指定类型的指定成员的偏移。类型应为 C 结构的名称。如果未指定任何成员，则报告此类型的所有成员。

偏移按字节列显，除非 *member* 是位字段，在这种情况下将按位列显偏移。为了清楚起见，输出始终使用合适的单位作为后缀。类型名称可以使用反引号 (‘) 作用范围运算符，如上面的“符号名称解析”中所述。**::offsetof dcmd** 只能用于特定的目标文件，即包含供 **mdb** 使用的符号调试信息的目标文件。有关详细信息，请参见下文“附注”部分中的**符号调试信息**。

```
address ::print [-aCdiLptx] [-c lim]
[-l lim][type[member ...]]
```

使用指定的 *type* 信息列显位于指定虚拟 *address* 的数据结构。*type* 参数可以命名 C 结构、联合、枚举、基本整数类型或者指向任意这些类型的指针。如果类型名称包含空格（例如，"struct foo"），则必须使用单引号或双引号括起来。类型名称可以使用反引号 (‘) 作用范围运算符，如上面的**符号名称解析**中所述。如果类型为结构化类型，**::print dcmd** 将递归列显结构或联合的每个成员。如果不存在 *type* 参数，并且静态或全局 **STT_OBJECT** 符号与该地址匹配，**::print** 将自动推理合适的类型。如果指定了 *type* 参数，则后面可以跟随可选的 *member* 表达式列表，在这种情况下只显示指定 *type* 的成员和子成员。如果 *type* 包含其他结构化类型，每个成员字符串可以通过生成使用句点 (‘.’) 分隔符分隔的成员名称列表来引用子结构元素。**::print dcmd** 只能用于特定的目标文件，即包含供 **mdb** 使用的符号调试信息的目标文件。有关详细信息，请参见下文“附注”部分中的**符号调试信息**。显示数据结构之后，**::print** 会将点增加 *type* 的大小（以字节为单位）。

如果存在 **-a** 选项，将显示每个成员的地址。如果存在 **-p** 选项，**::print** 会将 *address* 解释为物理内存地址而非虚拟内存地址。如果存在 **-t** 选项，将显示每个成员的类型。如果存在 **-d** 或 **-x** 选项，将以十进制 (**-d**) 或十六进制 (**-x**) 显示所有整数。缺省情况下，使用试探式方法来确定应以十进制还是十六进制显示值。可以使用 **-c** 选项来限制在字符数组中读取并显示为字符串的字符数。如果存在 **-C** 选项，将不强制任何限制。可以使用 **-l** 选项来限制在标准数组中读取并显示的元素数。如果存在 **-L** 选项，将不强制任何限制，并且显示所有数组元素。**-c** 和 **-l** 的缺省值可以使用 **::set** 或 **-o** 命令行选项修改，如下文的“选项”部分所述。

如果指定了 **-i** 选项，地址值将解释为要列显的直接值。必须指定用于解释值的类型。如果类型小于 64 位，该直接值将作为类型的大小进行解释。**-i** 选项不能与 **-p** 选项一起使用。如果指定了 **-a** 选项，则显示的地址为从零开始的字节偏移。

```
::quit
```

```
$q
```

退出调试器。

```
[thread] ::regs
```

```
[thread] $r
```

列显代表线程的通用寄存器集。如果指定了线程，将显示该线程的通用寄存器集合。线程表达式应为上面的**线程支持**中所述的线程标识符之一。

```
::release [-a]
```

```
:R [-a]
```

释放以前附加的进程或核心文件。如果存在 `-a` 选项，则将释放进程，并且该进程将保留停止状态且被放弃。然后，可以通过 `prun(1)`（请参见 `proc(1)`）继续执行，也可以通过应用 `mdb` 或其他调试器来恢复。缺省情况下，如果释放的进程是由 `mdb` 使用 `::run` 创建的，将强制终止该进程；如果该进程是由 `mdb` 使用 `-p` 选项、使用 `::attach` 或 `:A dcmd` 附加的，将释放该进程并设置为正在运行。

```
::run [args...]
```

```
:r [args...]
```

启动新的目标程序，使用指定的参数运行并附加到该程序。参数不由 `shell` 解释。如果调试器已在检查实时运行的程序，它会先与此程序分离，如同使用 `::release` 一样。

```
::set [-wF] [-/ -o option] [-s distance] [-I path]
```

```
[-L path] [-P prompt]
```

获取或设置其他调试器属性。如果未指定任何选项，将显示当前调试器属性集合。`::set dcmd` 可以识别以下选项：

`-F`

强制接管 `::attach` 应用到的下一个用户进程，等同于在命令行上使用 `-F` 选项执行 `mdb`。

`-I`

设置用于定位宏文件的缺省路径。路径参数可以包含任意特殊标记，这些标记在“选项”部分针对 `-I` 命令行选项进行介绍。

`-L`

设置用于定位调试器模块的缺省路径。路径参数可以包含任意特殊标记，这些标记在“选项”部分针对 `-I` 命令行选项进行介绍。

`-o`

启用指定的调试器选项。如果使用了 `-o` 形式，将禁用选项。选项字符串在“选项”部分随 `-o` 命令行选项一起介绍。

`-P`

将命令提示符设置为指定的提示字符串。

`-s`

将符号匹配距离设置为指定的距离。有关详细信息，请参见“选项”部分的 `-s` 命令行选项的说明。

`-w`

重新打开目标以便执行写入，等同于在命令行上使用 `-w` 选项执行 `mdb`。

```
::showrev [-pv]
```

显示硬件和软件的修订信息。未指定任何选项时，将显示一般系统信息。`-v` 选项显示所有装入目标文件的版本信息，而 `-p` 选项仅显示已作为修补程序的一部分安

装在系统上的装入目标文件的版本信息。版本信息并非对所有装入目标文件都可用。没有版本信息的装入目标文件将在 `-p` 选项的输出中省略，而在 `-v` 选项的输出中作为 "Unknown" 版本列出。

`[signal] ::sigbp [-/-dDestT] [-c cmd] [-n count] SIG ...`

`[signal] :t [-/-dDestT] [-c cmd] [-n count] SIG ...`

跟踪指定信号的传送。可以在 `dcmd` 前面使用可选的信号编号来标识信号，也可以在 `dcmd` 后面使用信号名称或编号的列表来标识（请参见 [signal.h\(3HEAD\)](#)）。`-d`、`-D`、`-e`、`-s`、`-t`、`-T`、`-c` 和 `-n` 选项具有与用于 `::evset dcmd` 时的相同意义。最初，对缺省情况下导致进程转储核心的信号集（请参见 [signal.h\(3HEAD\)](#)）和 `SIGINT` 进行跟踪。

`::sizeof [-s size] [-r min max] [type [member ...]]`

选项如下：

`-s size`

显示此大小的条目的类型。

`-r min max`

显示此大小范围内的条目的类型。

列显指定 `type` 的大小（以字节为单位）。`type` 参数可以命名 C 结构、联合、枚举、基本整数类型或者指向任意这些类型的指针。类型名称可以使用反引号 (') 作用范围运算符，如上面的 [符号名称解析](#) 中所述。`member` 可以通过标准 C 语法使用数组索引运算符 "[`index`]"、结构成员运算符 "." 来指定。

`::sizeof dcmd` 只能用于特定的目标文件，即包含供 `mdb` 使用的符号调试信息的目标文件。有关详细信息，请参见下文“附注”部分中的 [符号调试信息](#)。

`[address] ::stack [count]`

`[address] $c [count]`

列显 C 栈回溯。如果 `dcmd` 前面是显式地址，将显示在此虚拟内存地址开始的回溯。否则，将显示代表线程的栈。如果将可选计数值指定为参数，则在输出中为每个栈帧显示不超过 `count` 个参数。

`::status`

列显与当前目标相关的信息汇总。

`::step [over | out] [SIG]`

`:s [SIG]`

`:u [SIG]`

单步执行目标程序的一个指令。如果将可选的信号名称或编号（请参见 [signal.h\(3HEAD\)](#)）指定为参数，则该信号将在恢复执行过程中立即传输到目标。如果指定了可选的 "over" 参数，`::step` 将步出子例程调用。`::step over` 参数与 `::next dcmd` 相同。如果指定了可选的 "out" 参数，目标程序将继续执行，直至代表线程从当前函数返回。如果当前没有目标程序在运行，`::step out` 将启动新的程序开始运行，如同使用 `::run` 一样，并在第一个指令处停止。`:s dcmd` 与 `::step` 相同。`:u dcmd` 与 `::step out` 相同。

`[syscall] :: sysbp [-/ -dDestT] [-io] [-c cmd]`

`[-n count] syscall...`

跟踪指定系统调用的进入或退出。可以在 `dcmd` 前面使用可选的系统调用号来标识系统调用，也可以在 `dcmd` 后面使用系统调用名称或编号的列表来标识（请参见 `<sys/syscall.h>`）。如果指定了 `-i` 选项（缺省值），事件说明符将在进入每个系统调用的内核时触发。如果指定了 `-o` 选项，事件说明符将在从内核中退出时触发。`-d`、`-D`、`-e`、`-s`、`-t`、`-T`、`-c` 和 `-n` 选项具有与用于 `::evset dcmd` 时的相同意义。

`thread :: tls symbol`

列显指定线程局部存储 (TLS) 符号在指定线程上下文中的存储地址。线程表达式应为上面的**线程支持**中所述的线程标识符之一。符号名称可使用上面的**符号名称解析**中所述的任何作用范围运算符。

`::typeset [-/ -t] variable-name ...`

设置命名变量的属性。如果指定了一个或多个变量名，则定义这些变量并且将它们设置为点的值。如果存在 `-t` 选项，将设置与各个变量关联的用户定义标记。如果存在 `-t` 选项，将清除标记。如果未指定变量名，将列显变量列表及其值。

`::unload module-name`

卸载指定的 `dmod`。可以使用 `::dmods dcmd` 列显活动 `dmod` 的列表。不能卸载内置模块。不能卸载繁忙的模块（即 `dcmd` 正在执行）。

`::unset variable-name ...`

从已定义变量的列表中取消设置（删除）指定的变量。`mdb` 导出的一些变量已标记为持久变量，不能由用户取消设置。

`::vars [-npt]`

列显命名变量的列表。如果存在 `-n` 选项，输出将限制为当前具有非零值的变量。如果存在 `-p` 选项，将按照适合于调试器使用 `$(dcmd)` 重新处理的形式列显变量。此选项可用于将变量记录到宏文件中，然后在以后恢复这些值。如果存在 `-t` 选项，将只列显标记的变量。可以使用 `::typeset dcmd` 的 `-t` 选项来标记变量。

`::version`

列显调试器版本号。

`address :: vtop [-a as]`

列显指定虚拟地址的物理地址映射（如果可行）。`::vtop dcmd` 仅在检查内核目标时可用，或者在检查内核故障转储内部的用户进程时可用（在发出 `::context dcmd` 之后）。

从内核上下文检查内核目标时，`-a` 选项可用于指定备用地址空间结构的地址 (`as`)，该地址空间结构应该用于虚拟地址到物理地址的转换。缺省情况下，使用内核的地址空间进行转换。即使转储内容只包含内核页，此选项也可用于活动地址空间。

`[address] :: walk walker-name [variable-name]`

使用指定的遍历器遍历数据结构的元素。可以使用 `::walkers dcmd` 列出可用遍历器。一些遍历器在全局数据结构上操作，不需要起始地址。例如，遍历内核中的

proc 结构列表。其他遍历器对必须显式指定地址的特定数据结构进行操作。例如，在给定向地址空间的指针的情况下，遍历区段的列表。在交互使用时，`::walk dcmd` 以缺省基数显示数据结构中各个元素的地址。`dcmd` 也可用于为流水线提供地址列表。遍历器名称可以使用反引号 (') 作用范围运算符，如上面的 `dcmd` 和 [遍历器名称解析](#) 中所述。如果指定了可选的 *variable-name*，将为指定的变量分配在 `mdb` 调用流水线的下一阶段时遍历的每一步骤所返回的值。

`::walkers`

列出可用遍历器并列显示每个遍历器的简要说明。

`::whence [-v] name...`

`::which [-v] name...`

列显导出指定的 `dcmd` 和遍历器的 `dmod`。这些 `dcmd` 可用于确定哪个 `dmod` 当前正在提供指定的 `dcmd` 或遍历器的全局定义。有关全局名称解析的详细信息，请参见上面的有关 `dcmd` 和 [遍历器名称解析](#) 的部分。`-v` 选项使 `dcmd` 按照优先级顺序列显各个 `dcmd` 和遍历器的备用定义。

`addr [,len]::wp [-/-dDestT] [-rwx] [-c cmd]`
`[-n count]`

`addr [,len]:a [cmd...]`

`addr [,len]:p [cmd...]`

`addr [,len]:w [cmd...]`

在指定地址设置监视点。可通过在 `dcmd` 之前指定可选的重复计数来设置被监视区域的长度（以字节为单位）。如果没有显式设置长度，缺省值为一个字节。`::wp dcmd` 允许将监视点配置为在存在任何读取（`-r` 选项）、写入（`-w` 选项）或执行（`-x` 选项）访问的组合时触发。`-d`、`-D`、`-e`、`-s`、`-t`、`-T`、`-c` 和 `-n` 选项具有与用于 `::evset dcmd` 时的相同意义。`:a dcmd` 可在指定地址处设置读取访问监视点。`:p dcmd` 可在指定地址处设置执行访问监视点。`:w dcmd` 可在指定地址处设置写入访问监视点。`:a`、`:p` 和 `:w dcmd` 之后的参数会串联在一起形成回调字符串。如果此字符串包含元字符，必须引用该字符串。

`::xdata`

列出由当前目标导出的外部数据缓冲区。外部数据缓冲区表示与无法通过标准目标工具访问的目标相关联的信息（即地址空间、符号表或寄存器集合）。这些缓冲区可以由 `dcmd` 使用；有关详细信息，请参见 [《Oracle Solaris Modular Debugger Guide》](#)。

`::z`

从跟踪的软件事件列表中删除所有事件说明符。事件说明符也可以使用 `::delete` 删除。

选项

支持以下选项：

`-A`

禁止自动装入 `mdb` 模块。缺省情况下，`mdb` 尝试装入与用户进程或核心文件中的活动共享库对应的调试器模块，或者尝试装入与实时操作系统或操作系统故障转储中的已装入内核模块对应的调试器模块。

-f

强制原始文件调试模式。缺省情况下，mdb 尝试推断目标文件和核心文件操作数是引用了用户可执行文件和核心转储，还是引用了成对的操作系统故障转储文件。如果无法推断文件类型，则缺省情况下，调试器将文件作为纯二进制数据进行检查。
-f 选项强制 mdb 将参数作为一组原始文件解释以进行检查。

-F

如果需要，强制接管指定的用户进程。缺省情况下，mdb 拒绝附加到已处于其他调试工具（例如 `truss(1)`）控制下的用户进程。使用 -F 选项，mdb 会强制附加到这些进程。这可能会在 mdb 和其他尝试控制进程的工具之间造成意外的交互。

-I *path*

设置用于定位宏文件的缺省路径。宏文件使用 `$<` 或 `$<< dcmd` 读取。路径是使用冒号 (:) 字符分界的目录名称序列。-I `include` 路径和 -L `library` 路径（请参见下文）也可以包含以下任意标记：

%i

展开到当前的指令集体系结构 (ISA) 名称 ('sparc'、'sparcv9' 或 'i386')。

%o

展开到要修改的路径的旧值。要将目录附加到现有路径之前或之后时，这非常有用。

%p

展开到当前平台字符串（可以为 `uname -i`，或者为存储在进程核心文件或故障转储中的平台字符串）。

%r

展开到根目录的路径名。可以使用 -R 选项指定备用根目录。如果不存在 -R 选项，将从指向 mdb 可执行文件自身的路径动态派生根目录。例如，如果执行 `/bin/mdb`，根目录将为 `/`。如果执行 `/net/hostname/bin/mdb`，根目录将派生为 `/net/hostname`。

%t

展开到当前目标的名称。这可以为文本字符串 'proc'（用户进程或用户进程核心文件）、'kvm'（内核故障转储或实时操作系统）或 'raw'（原始文件）。

32 位 mdb 的缺省包含文件路径为：

```
%r/usr/platform/%p/lib/adb:%r/usr/lib/adb
```

64 位 mdb 的缺省包含文件路径为：

```
%r/usr/platform/%p/lib/adb:%i:%r/usr/lib/adb/%i
```

-k

强制内核调试模式。缺省情况下，mdb 尝试推断目标文件和核心文件操作数是引用了用户可执行文件和核心转储，还是引用了成对的操作系统故障转储文件。-k 选项强制 mdb 假定这些文件为操作系统故障转储文件。如果未指定目标文件或核心操

作数，但指定了 `-k` 选项，`mdb` 将缺省使用目标文件 `/dev/ksyms` 和核心文件 `/dev/kmem`。限制只有组 `sys` 可以对 `/dev/kmem` 执行读取访问。写入访问需要 `ALL` 权限。

`-K`

装入 `kldb`，停止实时运行的操作系统内核，并继续到 `kldb` 调试器提示符。此选项只应在系统控制台上使用，因为后续的 `kldb` 提示符将显示在系统控制台上。

`-L path`

设置用于定位调试器模块的缺省路径。模块在启动时自动装入，或者使用 `::load dcmd` 装入。路径是使用冒号 (`:`) 字符分界的目录名称序列。`-L` 库路径也可以包含为上面的 `-I` 显示的任何标记。

`-m`

禁止按需装入内核模块符号。缺省情况下，`mdb` 处理已装入内核模块的列表并执行每模块符号表的按需装入。如果指定了 `-m` 选项，`mdb` 将不尝试处理内核模块列表或提供每模块符号表。因此，在启动时不装入与活动内核模块对应的 `mdb` 模块。

`-M`

预装所有内核模块符号。缺省情况下，`mdb` 执行内核模块符号的按需装入：当地址为该模块的文本或者引用了数据区时，将读取模块的完整符号表。使用 `-M` 选项时，`mdb` 将在启动期间装入所有内核模块的完整符号表。

`-o option`

启用指定的调试器选项。如果使用了 `-o` 形式的选项，将禁用指定的 `option`。除非在下面另有说明，否则缺省情况下将禁用所有选项。`mdb` 可以识别以下 `option` 参数：

`adb`

启用更严格的 `adb(1)` 兼容性。提示符设置为空字符串，并且禁用诸如输出页面调度程序等多种 `mdb` 功能。

`array_mem_limit=limit`

对 `::print` 显示的数组成员数设置缺省限制。如果 `limit` 是特殊标记 `none`，缺省情况下将显示所有数组成员。

`array_str_limit=limit`

对 `::print` 在列显字符数组时尝试作为 ASCII 字符串显示的字符数设置缺省限制。如果 `limit` 是特殊标记 `none`，缺省情况下将整个字符数组显示为字符串。

`follow_exec_mode=mode`

设置调试器跟踪 `exec(2)` 系统调用的行为。`mode` 应为以下命令常量之一：

`ask`

如果 `stdout` 是终端设备，调试器将在 `exec(2)` 系统调用返回后停止，然后提示用户决定跟踪 `exec` 还是停止。如果 `stdout` 不是终端设备，`ask` 模式将缺省为 `stop`。

follow

调试器将通过自动继续执行目标进程来跟踪 `exec`，并根据新的可执行文件重置所有映射和符号表。`follow` 行为在下文“附注”部分的与 `Exec 交互` 中进行了详细介绍。

stop

调试器将停止从 `exec` 系统调用返回的内容。`stop` 行为在下文“附注”部分的与 `Exec 交互` 中进行了详细介绍。

follow_fork_mode=mode

设置调试器跟踪 `fork(2)`、`fork1(2)` 或 `vfork(2)` 系统调用的行为。`mode` 应为以下命令常量之一：

ask

如果 `stdout` 是终端设备，调试器将在 `fork(2)` 系统调用返回后停止，然后提示用户决定跟踪父进程还是子进程。如果 `stdout` 不是终端设备，`ask` 模式将缺省为 `parent`。

parent

调试器跟踪父进程，与子进程分离并将它设置为正在运行。

child

调试器跟踪子进程，与父进程分离并将它设置为正在运行。

ignoreeof

终端中输入 EOF 序列 (^D) 时，调试器不退出。必须使用 `::quit dcmd` 退出。

nostop

指定了 `-p` 选项或者应用了 `::attach` 或 `:A dcmd` 时，在附加到用户进程时不停止该进程。`nostop` 行为在下文“附注”部分的 `进程附加和释放` 中进行详细介绍。

pager

启用输出页面调度程序（缺省）。

repeatlast

如果在终端输入了 **换行** 作为完整命令时，`mdb` 将使用点的当前值重复上一个命令。`-o adb` 暗含此选项。

showlmid

对于使用链接图而非 `LM_ID_BASE` 和 `LM_ID_LDSO` 的用户应用程序，`mdb` 为用户应用程序中的符号命名和标识提供了支持，如上面的 `符号名称解析` 中所述。链接图上除 `LM_ID_BASE` 或 `LM_ID_LDSO` 之外的符号显示为 `LMlmid'library'symbol`，其中 `lmid` 是以缺省输出基数 (16) 显示的链接图 ID。用户可以选择通过启用 `showlmid` 选项来配置 `mdb`，以便显示所有符号和目标文件的链接图 ID 范围，包括与 `LM_ID_BASE` 和 `LM_ID_LDSO` 关联的那些内容。处理目标文件名的内置 `dcmd` 根据上面的 `showlmid` 值显示链接图 ID，包括 `::nm`、`::mappings`、`$m` 和 `::objects`。

- p *pid*
附加到指定的进程 ID 并停止该进程。mdb 使用 `/proc/pid/object/a.out` 文件作为可执行文件路径名。
- P *prompt*
设置命令提示符。缺省提示符为 `'>'`。
- R *root*
设置路径名扩展的根目录。缺省情况下，从 mdb 可执行文件自身的路径名派生根目录。根目录在路径名扩展期间替换 `%r` 标记。
- s *distance*
设置地址到符号名称的转换到指定 *distance* 的符号匹配距离。缺省情况下，mdb 将距离设置为零，这将启用智能匹配模式。每个 ELF 符号表条目包括值 *V* 和大小 *S*，表示函数或数据目标文件的大小（以字节为单位）。在智能模式中，如果 *A* 在范围 `[V, V + S)` 中，mdb 会将地址 *A* 与指定符号匹配。如果指定了任意非零距离，将使用相同的算法，但上面的表达式中的 *S* 始终为指定的绝对距离，并且忽略符号大小。
- S
抑制用户的 `~/.mdbrc` 文件的处理。缺省情况下，如果由 `$HOME` 定义的用户起始目录中存在宏文件 `.mdbrc`，mdb 将读取和处理宏文件。如果存在 `-S` 选项，将不读取此文件。
- u
强制用户调试模式。缺省情况下，mdb 尝试推断目标文件和核心文件操作数是引用了用户可执行文件和核心转储，还是引用了成对的操作系统故障转储文件。`-u` 选项强制 mdb 假定这些文件不是操作系统故障转储文件。
- U
如果 `kmdb` 已装入，则将它卸载。如果不在使用 `kmdb`，则应将它卸载，以便将内核调试器所使用的内存释放回可供操作系统使用的空闲内存。
- v *version*
设置反汇编程序版本。缺省情况下，mdb 将尝试推断调试目标的适当反汇编程序版本。反汇编程序可以使用 `-V` 选项显式设置。`::disasms dcmd` 列出可用的反汇编程序版本。
- w
打开指定的目标文件和核心文件进行写入。
- W
允许对映射到 I/O 设备的内存地址的访问。缺省情况下，mdb 不允许此类访问，因为许多设备不会针对无效的软件处理提供硬件保护。请仅在调试设备驱动程序时谨慎使用此选项。
- y
发送用于 `tty` 模式的显式终端初始化序列。某些终端（例如 `cmdtool(1)`）需要显式初始化序列以便切换到 `tty` 模式。如果没有此初始化序列，诸如 `standout` 模式这样的终端功能将无法供 mdb 使用。

操作数	<p>支持下列操作数：</p> <p><i>object</i> 指定要检查的 ELF 格式目标文件。mdb 提供了检查和编辑 ELF 格式可执行文件 (ET_EXEC)、ELF 动态库文件 (ET_DYN)、ELF 可重定位目标文件 (ET_REL) 和操作系统 unix.X 符号表文件的功能。</p> <p><i>core</i> 指定 ELF 进程核心文件 (ET_CORE) 或操作系统故障转储 vmcore.X 文件。如果提供的 ELF 核心文件操作数没有对应的目标文件，mdb 将尝试使用不同算法推断生成核心的可执行文件的名称。如果找不到可执行文件，mdb 仍将执行，但某些符号信息可能会不可用。</p> <p><i>suffix</i> 指定表示一对操作系统故障转储文件的数字后缀。例如，如果后缀为 '3'，mdb 推断应检查文件 'unix.3' 和 'vmcore.3'。如果这些文件不存在，但 'vmdump.3' 存在，将列显一条消息，指示必须先运行 <code>savecore -f vmdump.3</code> 以便解压缩转储文件。如果当前目录中存在具有相同名称的实际文件，数字字符串将不解释为后缀。</p>
用法	<p>mdb 以大文件感知方式处理所有输入文件（包括脚本、目标文件、核心文件和原始数据文件）。有关处理大于等于 2 GB (2^{31} 字节) 的大文件的详细信息，请参见 largefile(5)。</p>
退出状态	<p>将返回以下退出值：</p> <p>0 调试器成功完成执行。</p> <p>1 发生了致命错误。</p> <p>2 指定的命令行选项无效。</p>
环境变量	<p>HISTSIZE 此变量用于确定命令历史记录列表的最大长度。如果不存在此变量，缺省长度将为 128。</p> <p>HOME 此变量用于确定 .mdbrc 文件可以驻留在其中的用户起始目录的路径名。如果不存在此变量，不会发生 .mdbrc 处理。</p> <p>SHELL 此变量用于确定用来处理通过 ! 元字符请求的 shell 转义的 shell 路径名。如果不存在此变量，将使用 /bin/sh。</p>
文件	<p><code>\$HOME/.mdbrc</code> 用户 mdb 初始化文件。在初始化调试目标之后、但在执行模块自动装入或从标准输入中读取任何命令之前处理 .mdbrc 文件（如果存在）。</p>

`/dev/kmem`

内核虚拟内存映像设备。此设备特殊文件在检查实时操作系统时用作核心文件。

`/dev/ksyms`

内核符号表设备。此设备特殊文件在检查实时操作系统时用作目标文件。

`/proc/pid/*`

在检查和控制用户进程时读取的进程信息文件。

`/usr/lib/adb`

`/usr/platform/platform-name/lib/adb`

使用 `$<` 和 `$<< dcmd` 读取的宏文件的缺省目录。`platform-name` 是平台的名称，该名称从核心文件或故障转储中的信息派生，也可从当前计算机派生，就像由 `uname -i` 派生一样（请参见 [uname\(1\)](#)）。

`/usr/lib/mdb`

`/usr/platform/platform-name/lib/mdb`

使用 `::load dcmd` 装入的调试器模块的缺省目录。`platform-name` 是平台的名称，该名称从核心文件或故障转储中的信息派生，也可从当前计算机派生，就像由 `uname -i` 派生一样（请参见 [uname\(1\)](#)）。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	developer/debug/mdb
接口稳定性	Committed（已确定）

另请参见

[adb\(1\)](#)、[cmdtool\(1\)](#)、[gcore\(1\)](#)、[proc\(1\)](#)、[pgrep\(1\)](#)、[ps\(1\)](#)、[stty\(1\)](#)、[truss\(1\)](#)、[uname\(1\)](#)、[coreadm\(1\)](#)

《链接程序和库指南》

《Oracle Solaris Modular Debugger Guide》

警告

使用错误恢复机制

调试器及其 `dmod` 在相同地址空间中执行，因此有错误的 `dmod` 很可能导致 `mdb` 转储核心或者发生错误的行为。`mdb` 恢复功能（如上面的[信号处理](#)中所述）为这些情况提供了有限的恢复机制。但是，`mdb` 不可能明确了解有问题的 `dmod` 是仅破坏了自身的状态，还是破坏了调试器的全局状态。因此，无法确保恢复操作安全，也无法保证调试器接下来不会出现故障。在恢复之后，最安全的操作过程是保存所有重要调试信息，然后退出并重新启动调试器。

使用调试器修改实时操作系统

使用调试器修改（即写入到）实时运行操作系统的地址空间中是极度危险的，在用户损坏了核心数据结构的情况下，可能会导致系统故障。

附注

检查进程核心文件的限制

mdb 不支持对 Solaris 2.6 之前的 Solaris 发行版生成的进程核心文件进行检查。调试 Solaris 9 发行版或更早发行版生成的核心文件时，符号信息可能不可用。由于这些核心文件中不存在文本区段和只读数据，符号信息可能与转储核心时进程中存在的数据不匹配。在早于 Solaris 9 的发行版中，文本区段和只读数据在缺省情况下包括在核心文件中。用户可以使用 `coreadm(1M)` 配置用户进程，从核心文件中排除该信息。因此，mdb 为这些核心文件提供的信息无法与进程转储核心时存在的数据匹配。Solaris x86 系统中的核心文件无法在 Solaris SPARC 系统上进行检查，反之亦然。

检查故障转储文件的限制

Solaris 7 和更早发行版的故障转储只能在使用与操作系统发行版相对应的 `libkvm` 时才能进行检查。如果使用不同操作系统发行版的 `dmod` 检查某个操作系统发行版的故障转储，内核实现中的更改可能会导致某些 `dcmd` 或遍历器无法正常工作。mdb 在检测到这种情况时将发出警告消息。Solaris x86 系统中的故障转储无法在 Solaris SPARC 系统上进行检查，反之亦然。

32 位和 64 位调试器之间的关系

mdb 支持对 32 位和 64 位程序进行调试。在检查目标并确定数据模型之后，mdb 在必要时会自动重新执行与目标具有相同数据模型的 mdb 二进制文件。这种方法简化了编写调试器模块的任务，因为装入的模块使用与主要目标相同的数据模型。只能将 64 位调试器用于调试 64 位目标程序。64 位调试器只能在运行 64 位操作环境的系统中使用。

执行 64 位进程的调试器在调试 32 位进程时，可能需要重新执行自身，反之亦然。这种情况的处理在下面的与 `Exec` 交互中进行详细介绍。

与 Exec 交互

受控进程执行成功的 `exec(2)` 时，调试器的行为由 `::set -o follow_exec_mode` 选项控制，如上文所述。如果调试器和被调试的进程具有相同的数据模型，`"stop"` 和 `"follow"` 模式将确定 mdb 是自动继续执行目标还是在 `exec` 之后返回调试器提示符。如果调试器和被调试的进程具有不同的数据模型，`"follow"` 行为将导致 mdb 使用合适的数据模型自动重新执行 mdb 二进制文件并重新附加到进程，仍然在从 `exec` 返回时停止。并非所有调试器状态在此重新执行过程中均会保留。

如果 32 位被调试进程执行了 64 位程序，`"stop"` 将返回到命令提示符，但调试器不再能够检查进程，因为它现在使用 64 位数据模型。要恢复调试，请执行 `::release -a dcmd`，退出 mdb，然后执行 `mdb -p pid` 以便将 64 位调试器重新附加到进程。

如果 64 位被调试进程执行 32 位程序，`"stop"` 将返回到命令提示符，但调试器只提供有限的功能用于检查新进程。所有内置 `dcmd` 均按预期工作，但可以装入的 `dcmd` 并非如此，因为它们不执行数据模型结构转换。用户应按上文所述，将调试器释放并重新附加到进程，以便恢复完整的调试功能。

与作业控制交互

如果调试器附加到由作业控制停止的进程（即，为了响应 `SIGTSTP`、`SIGTTIN` 或 `SIGTTOU` 而停止），则在由继续 `dcmd` 继续执行时，进程无法重新设置为正在运行。如果被调试进程是相同会话的成员（即与 mdb 共享相同的控制终端），mdb 会尝试将关联的进程组提升到前台，并使用 `SIGCONT` 继续执行进程，以便将进程从作业控制停止中恢复。将 mdb 与此类进程分离时，它在退出前会将进程组还原到后台。如果

被调试进程不是相同会话的成员，mdb 将无法安全地将进程组提升到前台，因此它会针对调试器继续执行进程，但进程仍由作业控制停止。mdb 在这种情况下将列显警告，用户必须从合适的 shell 发出 "fg" 命令以便恢复进程。

进程附加和释放

mdb 附加到正在运行的进程时，进程将停止并保持停止状态，直至应用了继续 dcmd 之一或者调试器退出。如果在使用 `-p` 将调试器附加到进程之前，或者在发出 `::attach` 或 `:A` 命令之前，`-o nostop` 选项已启用，则 mdb 将附加到进程，但不停止进程。虽然进程仍在运行，它可以检测为正常（尽管带有不一致的结果），并且可能会启用断点或其他跟踪标志。如果 `:c` 或 `::cont` dcmd 在进程运行的时候执行，调试器将等待进程停止。如果没有发生跟踪的软件事件，用户可以在 `:c` 或 `::cont` 之后发送中断 (^c)，以便强制进程停止并将控制权返回给调试器。

在执行 `:R`、`::release`、`:r`、`::run`、`$q` 或 `::quit` dcmd 时，或者调试器由于 EOF 或信号而终止时，mdb 将释放当前正在运行的进程（如果有）。如果进程最初是由调试器使用 `:r` 或 `::run` 创建的，则会在释放时强制终止，就如使用 SIGKILL 一样。如果在将 mdb 附加到进程之前进程已在运行，该进程在释放时将再次设置为正在运行。进程可以使用 `::release -a` 选项释放、保持停止或者被放弃。

符号调试信息

`::list`、`::offsetof`、`::print` 和 `::sizeof` dcmd 要求一个或多个装入目标文件包含适用于 mdb 的压缩符号调试信息。此信息当前只能用于某些 Solaris 内核模块。

开发者信息

《Oracle Solaris Modular Debugger Guide》提供了有关 mdb 功能的更为详细的说明，以及面向调试器模块开发者的信息。

头文件 `<sys/mdb_modapi.h>` 包含 MDB 模块 API 中函数的原型，而 `/source/demo/mdb-examples` 软件包在目录 `/usr/demo/mdb` 中提供了示例模块的源代码。

引用名	mesg – permit or deny messages								
用法概要	mesg [-n -y n y]								
描述	The mesg utility will control whether other users are allowed to send messages via write(1) , talk(1) , or other utilities to a terminal device. The terminal device affected is determined by searching for the first terminal in the sequence of devices associated with standard input, standard output, and standard error, respectively. With no arguments, mesg reports the current state without changing it. Processes with appropriate privileges may be able to send messages to the terminal independent of the current state.								
选项	The following options are supported: <ul style="list-style-type: none"> -n n Denies permission to other users to send message to the terminal. See write(1). -y y Grants permission to other users to send messages to the terminal. 								
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of mesg: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.								
退出状态	The following exit values are returned: <ul style="list-style-type: none"> 0 if messages are receivable. 1 if messages are not receivable. 2 on error. 								
文件	/dev/tty* terminal devices /dev/pts/* terminal devices								
属性	See attributes(5) for descriptions of the following attributes:								
	<table border="1"> <thead> <tr> <th>ATTRIBUTE TYPE</th> <th>ATTRIBUTE VALUE</th> </tr> </thead> <tbody> <tr> <td>Availability</td> <td>system/core-os</td> </tr> <tr> <td>Interface Stability</td> <td>Committed</td> </tr> <tr> <td>Standard</td> <td>See standards(5).</td> </tr> </tbody> </table>	ATTRIBUTE TYPE	ATTRIBUTE VALUE	Availability	system/core-os	Interface Stability	Committed	Standard	See standards(5) .
ATTRIBUTE TYPE	ATTRIBUTE VALUE								
Availability	system/core-os								
Interface Stability	Committed								
Standard	See standards(5) .								
另请参见	talk(1) , write(1) , attributes(5) , environ(5) , standards(5)								

引用名	mkdir – make directories
用法概要	<code>/usr/bin/mkdir [-m <i>mode</i>] [-p] <i>dir</i>...</code>
描述	<p>The <code>mkdir</code> command creates the named directories in mode 777 (possibly altered by the file mode creation mask <code>umask(1)</code>).</p> <p>Standard entries in a directory (for instance, the files “.”, for the directory itself, and “..”, for its parent) are made automatically. <code>mkdir</code> cannot create these entries by name. Creation of a directory requires write permission in the parent directory.</p> <p>The owner-ID and group-ID of the new directories are set to the process's effective user-ID and group-ID, respectively. <code>mkdir</code> calls the <code>mkdir(2)</code> system call.</p>
setgid and mkdir	<p>To change the setgid bit on a newly created directory, you must use <code>chmod g+s</code> or <code>chmod g-s</code> after executing <code>mkdir</code>.</p> <p>The setgid bit setting is inherited from the parent directory.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"><code>-m <i>mode</i></code> This option allows users to specify the mode to be used for new directories. Choices for modes can be found in <code>chmod(1)</code>.<code>-p</code> With this option, <code>mkdir</code> creates <i>dir</i> by creating all the non-existing parent directories first. The mode given to intermediate directories is the difference between 777 and the bits set in the file mode creation mask. The difference, however, must be at least 300 (write and execute permission for the user).
操作数	<p>The following operand is supported:</p> <p><i>dir</i> A path name of a directory to be created.</p>
用法	See <code>largefile(5)</code> for the description of the behavior of <code>mkdir</code> when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).
示例	<p>示例 1 Using <code>mkdir</code></p> <p>The following example:</p> <pre>example% mkdir -p ltr/jd/jan</pre> <p>creates the subdirectory structure <code>ltr/jd/jan</code>.</p>
环境变量	See <code>environ(5)</code> for descriptions of the following environment variables that affect the execution of <code>mkdir</code> : <code>LANG</code> , <code>LC_ALL</code> , <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>NLSPATH</code> .
退出状态	<p>The following exit values are returned:</p> <ul style="list-style-type: none">0 All the specified directories were created successfully or the <code>-p</code> option was specified and all the specified directories now exist.

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[chmod\(1\)](#), [rm\(1\)](#), [sh\(1\)](#), [umask\(1\)](#), [Intro\(2\)](#), [mkdir\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

引用名 mkmsgs – create message files for use by gettext

用法概要 mkmsgs [-o] [-i *locale*] *inputstrings* *msgfile*

描述 The mkmsgs utility is used to create a file of text strings that can be accessed using the text retrieval tools (see [gettext\(1\)](#), [srchtxt\(1\)](#), [exstr\(1\)](#), and [gettext\(3C\)](#)). It will take as input a file of text strings for a particular geographic locale (see [setlocale\(3C\)](#)) and create a file of text strings in a format that can be retrieved by both [gettext\(1\)](#) and [gettext\(3C\)](#). By using the `-i` option, you can install the created file under the `/usr/lib/locale/locale/LC_MESSAGES` directory (`locale` corresponds to the language in which the text strings are written).

inputstrings is the name of the file that contains the original text strings. *msgfile* is the name of the output file where mkmsgs writes the strings in a format that is readable by [gettext\(1\)](#) and [gettext\(3C\)](#). The name of *msgfile* can be up to 14 characters in length, but may not contain either `\0` (null) or the ASCII code for `/` (slash) or `:` (colon).

The input file contains a set of text strings for the particular geographic locale. Text strings are separated by a newline character. Nongraphic characters must be represented as alphabetic escape sequences. Messages are transformed and copied sequentially from *inputstrings* to *msgfile*. To generate an empty message in *msgfile*, leave an empty line at the correct place in *inputstrings*.

Strings can be changed simply by editing the file *inputstrings*. New strings must be added only at the end of the file; then a new *msgfile* file must be created and installed in the correct place. If this procedure is not followed, the retrieval function will retrieve the wrong string and software compatibility will be broken.

选项 The following options are supported:

- `-o` Overwrite *msgfile*, if it exists.
- `-i locale` Install *msgfile* in the `/usr/lib/locale/locale/LC_MESSAGES` directory. Only someone who is super user or a member of group `bin` can create or overwrite files in this directory. Directories under `/usr/lib/locale` will be created if they do not exist.

示例 示例 1 Using the mkmsgs command.

The following example shows an input message source file `C.str`:

```
File %s:\t cannot be opened\n
%s: Bad directory\n
.
.
.
write error\n
.
.
```

示例 2 Using Input Strings From C.str to Create Text Strings in a File

The following command uses the input strings from C.str to create text strings in the appropriate format in the file UX in the current directory:

```
example% mkmsgs C.str UX
```

示例 3 Using Input Strings From FR.str to Create Text Strings in a File

The following command uses the input strings from FR.str to create text strings in the appropriate format in the file UX in the directory /usr/lib/locale/fr/LC_MESSAGES:

```
example% mkmsgs -i fr FR.str UX
```

These text strings would be accessed if you had set the environment variable LC_MESSAGES=fr and then invoked one of the text retrieval tools listed at the beginning of the DESCRIPTION section.

文件

/usr/lib/locale/locale/LC_MESSAGES/* message files created by mkmsgs

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale

另请参见

[exstr\(1\)](#), [gettxt\(1\)](#), [srchtxt\(1\)](#), [gettxt\(3C\)](#), [setlocale\(3C\)](#), [attributes\(5\)](#)

引用名 mkstr – create an error message file by massaging C source files

用法概要 /usr/ucb/mkstr [-] *messagefile prefix filename...*

描述 The `mkstr` utility creates files of error messages. You can use `mkstr` to make programs with large numbers of error diagnostics much smaller, and to reduce system overhead in running the program — as the error messages do not have to be constantly swapped in and out.

`mkstr` processes each of the specified *filenames*, placing a massaged version of the input file in a file with a name consisting of the specified *prefix* and the original source file name. A typical example of using `mkstr` would be:

```
mkstr pistrings processed *.c
```

This command would cause all the error messages from the C source files in the current directory to be placed in the file `pistrings` and processed copies of the source for these files to be placed in files whose names are prefixed with *processed*.

To process the error messages in the source to the message file, `mkstr` keys on the string `'error('` in the input stream. Each time it occurs, the C string starting at the `'` is placed in the message file followed by a null character and a NEWLINE character; the null character terminates the message so it can be easily used when retrieved, the NEWLINE character makes it possible to sensibly `cat` the error message file to see its contents. The massaged copy of the input file then contains a `lseek` pointer into the file which can be used to retrieve the message, that is:

```
char efilename[ ] = "/usr/lib/pi_strings";
int efil = -1;

error(a1, a2, a3, a4)
{
    char
    buf[256];
    if (efil < 0) {

        efil = open(efilename, 0);
        if (efil < 0) {
oops:
            perror (efilename);
            exit (1);
        }
    }
    if (lseek(efil, (long) a1, 0) || read(efil, buf, 256) <= 0)
        goto oops;
    printf(buf, a2, a3, a4);
}
```

选项 – Place error messages at the end of the specified message file for recompiling part of a large `mkstred` program.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [xstr\(1\)](#), [attributes\(5\)](#)

引用名 mktemp – make temporary filename

用法概要 mktemp [-dtqu] [--directory] [--quiet] [--dry-run] [-p *directory*]
 [--suffix=*suff*] [--tmpdir[=*dir*]] [*template*]

描述 The `mktemp` utility makes a temporary filename. To do this, `mktemp` takes the specified filename template and overwrites a portion of it to create a unique filename. See `OPERANDS`.

The template is converted to a path name using the `mktemp(3C)`, `mkdtemp(3C)`, `mkstemp(3C)`, and `mkstemp(3C)` library functions.

If `mktemp` can successfully generate a unique filename, the file (or directory) is created with file permissions such that it is only readable and writable by its owner (unless the `-u` flag is given) and the filename is printed to standard output.

`mktemp` allows shell scripts to safely use temporary files. Traditionally, many shell scripts take the name of the program with the PID as a suffix and used that as a temporary filename. This kind of naming scheme is predictable and the race condition it creates is easy for an attacker to win. A safer, though still inferior approach is to make a temporary directory using the same naming scheme. While this guarantees that a temporary file is not subverted, it still allows a simple denial of service attack. Use `mktemp` instead.

选项 The following options are supported:

`-d, --directory`
Make a directory instead of a file.

`-p directory`
Use the specified directory as a prefix when generating the temporary filename. The directory is overridden by the user's `TMPDIR` environment variable if it is set. This option implies the `-t` flag.

`-q, --quiet`
Fail silently if an error occurs. This is useful if a script does not want error output to go to standard error.

`--suffix=suff`
Append *suff* to the template. *suff* must not contain a slash (/). This option is implied when the template ends in characters other than replaceable X characters, and does not remove the requirement to include such X characters.

`--tmpdir[=dir]`
'*template*' is relative to *dir*. If *dir* is not specified, the user's environment variable `TMPDIR` is used if set, else `/tmp` is used. This option does not support a template with an absolute name and unlike with the use of `-t`, a template may contain slashes.

`-t`
Generate a path rooted in a temporary directory. This directory is chosen as follows: If the user's `TMPDIR` environment variable is set, the directory contained therein is used.

Otherwise, if the `-p` flag was given the specified directory is used. If none of the above apply, `/tmp` is used. In this mode, the template (if specified) should be a directory component (as opposed to a full path) and thus should not contain any forward slashes.

`-u, --dry-run`

Operate in unsafe mode. The temp file is unlinked before `mktemp` exits. This is slightly better than `mktemp(3C)`, but still introduces a race condition. Use of this option is discouraged.

操作数

The following operands are supported:

template *template* can be any filename with three to six Xs included in it, for example `/tmp/tfile.XXXXXX`.

If *template* is not specified, a default of `tmp.XXXXXX` is used and the `-t` flag is implied.

If *template* has multiple series of Xs, the final series is used for the replacement text, unless `--suffix` specifies otherwise.

If *template* has characters after the final set of Xs, then the `--suffix` option is implied to be all such characters, unless explicitly specified.

示例

示例 1 Using `mktemp`

The following example illustrates a simple use of `mktemp` in a `sh(1)` script. In this example, the script quits if it cannot get a safe temporary file.

```
TMPFILE='mktemp /tmp/example.XXXXXX'
if [ -z "$TMPFILE" ]; then exit 1; fi
echo "program output" >> $TMPFILE
```

示例 2 Using `mktemp` to Support `TMPDIR`

The following example uses `mktemp` to support for a user's `TMPDIR` environment variable:

```
TMPFILE='mktemp -t example.XXXXXX'
if [ -z "$TMPFILE" ]; then exit 1; fi
echo "program output" >> $TMPFILE
```

示例 3 Using `mktemp` Without Specifying the Name of the Temporary File

The following example uses `mktemp` without specifying the name of the temporary file. In this case the `-t` flag is implied.

```
TMPFILE='mktemp'
if [ -z "$TMPFILE" ]; then exit 1; fi
echo "program output" >> $TMPFILE
```

示例 4 Using mktemp with a Default Temporary Directory Other than /tmp

The following example creates the temporary file in /extra/tmp unless the user's TMPDIR environment variable specifies otherwise:

```
TMPFILE='mktemp -p /extra/tmp example.XXXXXX'
if [ -z "$TMPFILE" ]; then exit 1; fi
echo "program output" >> $TMPFILE
```

示例 5 Using mktemp to Remove a File

The following example attempts to create two temporary files. If creation of the second temporary file fails, mktemp removes the first file before exiting:

```
TMP1='mktemp -t example.1.XXXXXX'
if [ -z "$TMP1" ]; then exit 1; fi
TMP2='mktemp -t example.2.XXXXXX'
if [ -z "$TMP2" ]; then
    rm -f $TMP1
    exit 1
fi
```

示例 6 Using mktemp

The following example does not exit if mktemp is unable to create the file. That part of the script has been protected.

```
TMPFILE='mktemp -q -t example.XXXXXX'
if [ ! -z "$TMPFILE" ]
then
    # Safe to use $TMPFILE in this block
    echo data > $TMPFILE
    ...
    rm -f $TMPFILE
fi
```

示例 7 Using mktemp with Suffix Option

The following command illustrates the use of the suffix option. The effect of this command is to create the temporary file ex.q5N.SUFF.

```
# mktemp --suffix=.SUFF ex.XXXXXX
ex.q5Ngid.SUFF
```

示例 8 Using Suffix and Tmpdir Options

The following command illustrates the use of the suffix and tmpdir options.

```
# mktemp --tmpdir=$HOME --suffix=.bar foo.XXXXXX
/root/foo.7Za0_N.bar
```

示例 9 Using Directory and Suffix Options

The following command uses both the directory and suffix options.

```
# mktemp --directory --suffix=.bar foo.XXXXXX
foo.GSa03d.bar
# ls -l
drwx----- 2 root    staff      512 Mar 19 2012 foo.GSa0.bar
```

示例 10 Supporting a Template with Non-Trailing Xs

The following command shows the use of the directory option with non-trailing X characters. In this command, the `--suffix=suff` option is implied, where `bar` is used as the suffix.

```
# mktemp XXfooXXXXXXbar
XXfooaFY0N6bar
```

示例 11 Using the Quiet and Tmpdir Options

The following command illustrates the use of the quiet and tmpdir options.

```
# mktemp --quiet --tmpdir=/tmp foo
[No diagnostic message is returned]
```

示例 12 Using mktemp with Multiple Options

The following command combines the use of the dry-run, tmpdir, and suffix options.

```
# mktemp --dry-run --tmpdir=$HOME --suffix=SUFF
/root/tmp.qdaGc0SUFF
# ls -l /root/tmp.qdaGc0SUFF
/root/tmp.qdaGc0SUFF: No such file or directory
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `mktemp` with the `-t` option: `TMPDIR`.

TMPDIR Name a directory used for creating temporary files to override system default; used by `mktemp`.

退出状态

The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed

另请参见

[sh\(1\)](#), [mkdtemp\(3C\)](#), [mkstemp\(3C\)](#), [mkstemp\(3C\)](#), [mktemp\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注

The `mktemp` utility appeared in OpenBSD 2.1. The Solaris implementation uses only as many 'Xs' as are significant for [mktemp\(3C\)](#), [mkstemp\(3C\)](#), and [mkstemp\(3C\)](#).

引用名	moe - 显示路径名的最佳扩展
用法概要	moe [-c] [-32 -64] [-s -v] <i>path</i>
描述	<p>moe 实用程序显示包含保留的运行时链接程序标记的路径名的最佳扩展。这些标记可用于定义动态目标文件中的依赖项、<code>filtee</code> 和运行路径。这些运行时标记的扩展为选择能够在该计算机上以最佳性能运行的目标文件和搜索路径提供了灵活的机制。请参见 ld.so.1(1)。</p> <p>例如，标记 <code>\$HWCAP</code> 可用于表示过滤器和依赖项。此标记的运行时解释使所分析的一系列目标文件能够确定它们是否适合随某个进程装入。这些目标文件根据每个目标文件需要执行的硬件功能排序。moe 返回最适合在当前平台上执行的目标文件的名称。</p> <p>moe 通过将所提供的 <i>path</i> 传递到 dlopen(3C)，同时传递 <code>RTLD_FIRST</code> 标志来分析路径名称。因此，保留标记的扩展是由 <code>ld.so.1</code> 执行的，就像在正在执行的进程中发生的扩展一样。尽管使用 <code>dlopen()</code> 调用可对多个目标文件进行分析，但是 <code>RTLD_FIRST</code> 标志可确保只处理最佳目标文件。</p> <p>缺省情况下，moe 对指定的 <i>path</i> 分析两次。第一次分析会查找 32 位目标文件。第二次分析（如果适用）会查找 64 位目标文件。通常，32 位目标文件和 64 位目标文件分别被置于不同的目录中。这些目录的名称通常能够反映目录所包含的目标文件类。如果 32 位目标文件和 64 位目标文件占用同一目录，可通过多次传递 moe 捕获所有实例。如果用户无法从所指定的路径名推断该目录可能包含的目标文件类，也可以灵活地使用多次传递。</p> <p>有关运行时链接程序执行的保留标记扩展的完整说明，请参见 《链接程序和库指南》。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -32 仅分析 32 位目标文件。 -64 仅分析 64 位目标文件。 -c 在每个路径名之前加上目标文件类作为前缀。 -s 无提示。不显示任何最佳名称或错误诊断消息。仅提供一个错误返回值。此选项只有与 -32 和 -64 选项一起使用时才有意义。-s 选项不可与 -v 选项一起使用。 -v 详细模式。如果无法确定最佳扩展名，则会向标准错误写入一条错误诊断消息。-v 选项不可与 -s 选项一起使用。
操作数	<p>支持下列操作数：</p> <p><i>path</i> 要扩展的路径名。</p>

示例 以下示例使用 `moe` 显示了 `/usr/lib/libc` 目录中的目标文件的最佳扩展。此目录包含一系列 Intel 目标文件，构建这些目标文件是为了使用各种硬件功能。

```
% moe '/usr/lib/libc/$HWCAP'
/usr/lib/libc/libc_hwcap.so.1
```

`-c` 选项可用于明确最佳目标文件的类。

```
% moe -c '/usr/lib/libc/$HWCAP'
32-bit: /usr/lib/libc/libc_hwcap.so.1
```

以下示例使用 `moe` 显示了 `/opt/ISV/cpu` 目录分层结构下的目标文件的最佳扩展。这些目录包含一系列针对各种平台构建的 SPARC 目标文件。

```
% moe -c -64 '/opt/ISV/$ISALIST/isa.so.1'
64-bit: /opt/ISV/sparcv9/isa.so.1
```

`-v` 可用于诊断未返回最佳名称的实例。如果尝试将前面的路径名作为 32 位目标文件进行检查，将会产生下面的诊断消息。

```
% moe -c -v -32 '/opt/ISV/$ISALIST/isa.so.1'
32-bit: /opt/ISV/sparcv9/isa.so.1: wrong ELF class: ELFCLASS64
```

退出状态 当使用 `-32` 或 `-64` 选项时，成功的最佳扩展将返回 `0`，其他情况下将返回非零值。未使用 `-32` 或 `-64` 选项时，返回值始终为 `0`。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/linker
接口稳定性	Committed (已确定)

另请参见 [ld.so.1\(1\)](#)、[optisa\(1\)](#)、[isalist\(1\)](#)、[dlmopen\(3C\)](#)、[attributes\(5\)](#)

《链接程序和库指南》

引用名	more, page – browse or page through a text file
用法概要	<pre> /usr/bin/more [-cdfllrsuw] [-lines] [+ <i>linenumber</i>] [+/ <i>pattern</i>] [<i>file</i>]... /usr/bin/page [-cdfllrsuw] [-lines] [+ <i>linenumber</i>] [+/ <i>pattern</i>] [<i>file</i>]... /usr/xpg4/bin/more [-cdeisu] [-n <i>number</i>] [-p <i>command</i>] [-t <i>tagstring</i>] [<i>file</i>]... /usr/xpg4/bin/more [-cdeisu] [-n <i>number</i>] [+ <i>command</i>] [-t <i>tagstring</i>] [<i>file</i>]... </pre>
描述	<p>The more utility is a filter that displays the contents of a text file on the terminal, one screenful at a time. It normally pauses after each screenful. <code>/usr/bin/more</code> then prints <code>--More--</code> and <code>/usr/xpg4/bin/more</code> then prints <i>file</i> at the bottom of the screen. If more is reading from a file rather than a pipe, the percentage of characters displayed so far is also shown.</p> <p>The more utility scrolls up to display one more line in response to a RETURN character. more displays another screenful in response to a SPACE character. Other commands are listed below.</p> <p>The page utility clears the screen before displaying the next screenful of text. page only provides a one-line overlap between screens.</p> <p>The more utility sets the terminal to NOECHO mode, so that the output can be continuous. Commands that you type do not normally show up on your terminal, except for the / and ! commands.</p> <p>The <code>/usr/bin/more</code> utility exits after displaying the last specified file. <code>/usr/xpg4/bin/more</code> prompts for a command at the last line of the last specified file.</p> <p>If the standard output is not a terminal, more acts just like cat(1), except that a header is printed before each file in a series.</p>
选项	<p>The following options are supported for both <code>/usr/bin/more</code> and <code>/usr/xpg4/bin/more</code>:</p> <ul style="list-style-type: none"> -c Clears before displaying. Redraws the screen instead of scrolling for faster displays. This option is ignored if the terminal does not have the ability to clear to the end of a line. -d Displays error messages rather than ringing the terminal bell if an unrecognized command is used. This is helpful for inexperienced users. -s Squeeze. Replaces multiple blank lines with a single blank line. This is helpful when viewing nroff(1) output on the screen. <p><code>/usr/bin/more</code></p> <p>The following options are supported for <code>/usr/bin/more</code> only:</p> <ul style="list-style-type: none"> -f Does not fold long lines. This is useful when lines contain nonprinting characters or escape sequences, such as those generated when nroff(1) output is piped through ul(1).

- l Does not treat FORMFEED characters (Control-l) as page breaks. If -l is not used, more pauses to accept commands after any line containing a ^L character (Control-l). Also, if a file begins with a FORMFEED, the screen is cleared before the file is printed.
- r Normally, more ignores control characters that it does not interpret in some way. The -r option causes these to be displayed as ^C where C stands for any such control character.
- u Suppresses generation of underlining escape sequences. Normally, more handles underlining, such as that produced by `nroff(1)`, in a manner appropriate to the terminal. If the terminal can perform underlining or has a stand-out mode, more supplies appropriate escape sequences as called for in the text file.
- w Normally, more exits when it comes to the end of its input. With -w, however, more prompts and waits for any key to be struck before exiting.
- lines Displays the indicated number of *lines* in each screenful, rather than the default (the number of lines in the terminal screen less two).
- +linenumber Start up at *linenumber*.
- +/*pattern* Start up two lines above the line containing the regular expression *pattern*.
Note: Unlike editors, this construct should *not* end with a '/'. If it does, then the trailing slash is taken as a character in the search pattern.

/usr/xpg4/bin/more

The following options are supported for /usr/xpg4/bin/more only:

- e Exits immediately after writing the last line of the last file in the argument list.
- i Performs pattern matching in searches without regard to case.
- n *number* Specifies the number of lines per screenful. The *number* argument is a positive decimal integer. The -n option overrides any values obtained from the environment.
- p *command*
+*command* For each file examined, initially executes the more command in the *command* argument. If the command is a positioning command, such as a line number or a regular expression search, set the current position to represent the final results of the command, without writing any intermediate lines of the file. For example, the two commands:


```
more -p 1000j file
more -p 1000G file
```

are equivalent and start the display with the current position at line 1000, bypassing the lines that j would write and scroll off the screen if it had been

issued during the file examination. If the positioning command is unsuccessful, the first line in the file will be the current position.

- t *tagstring* Writes the screenful of the file containing the tag named by the *tagstring* argument. See the [ctags\(1\)](#) utility.
- u Treats a backspace character as a printable control character, displayed as a ^H (Control-h), suppressing backspacing and the special handling that produces underlined or standout-mode text on some terminal types. Also, does not ignore a carriage-return character at the end of a line.

If both the -t *tagstring* and -p *command* (or the obsolescent +*command*) options are given, the -t *tagstring* is processed first.

用法

Environment `more` uses the terminal's [terminfo\(4\)](#) entry to determine its display characteristics.

`more` looks in the environment variable `MORE` for any preset options. For instance, to page through files using the -c mode by default, set the value of this variable to -c. (Normally, the command sequence to set up this environment variable is placed in the `.login` or `.profile` file).

Commands The commands take effect immediately. It is not necessary to type a carriage return unless the command requires a *file*, *command*, *tagstring*, or *pattern*. Up to the time when the command character itself is given, the user may type the line kill character to cancel the numerical argument being formed. In addition, the user may type the erase character to redisplay the `'-More--(xx%)'` or *file* message.

In the following commands, *i* is a numerical argument (1 by default).

*i*SPACE Display another screenful, or *i* more lines if *i* is specified.

*i*RETURN Display another line, or *i* more lines, if specified.

ib

i^B (Control-b) Skip back *i* screenfuls and then print a screenful.

id

i^D (Control-d) Scroll forward one half screenful or *i* more lines. If *i* is specified, the count becomes the default for subsequent d and u commands.

if Skip *i* screens full and then print a screenful.

h Help. Give a description of all the `more` commands.

^L (Control-l) Refresh.

in Search for the *i* th occurrence of the last *pattern* entered.

q	
Q	Exit from more.
is	Skip <i>i</i> lines and then print a screenful.
v	Drop into the vi editor at the current line of the current file.
iz	Same as SPACE, except that <i>i</i> , if present, becomes the new default number of lines per screenful.
=	Display the current line number.
<i>i/pattern</i>	Search forward for the <i>i</i> th occurrence of the regular expression <i>pattern</i> . Display the screenful starting two lines before the line that contains the <i>i</i> th match for the regular expression <i>pattern</i> , or the end of a pipe, whichever comes first. If more is displaying a file and there is no match, its position in the file remains unchanged. Regular expressions can be edited using erase and kill characters. Erasing back past the first column cancels the search command.
! <i>command</i>	Invoke a shell to execute <i>command</i> . The characters % and !, when used within <i>command</i> are replaced with the current filename and the previous shell command, respectively. If there is no current filename, % is not expanded. Prepend a backslash to these characters to escape expansion.
:f	Display the current filename and line number.
:n	Skip to the <i>i</i> th next filename given in the command line, or to the last filename in the list if <i>i</i> is out of range.
:p	Skip to the <i>i</i> th previous filename given in the command line, or to the first filename if <i>i</i> is out of range. If given while more is positioned within a file, go to the beginning of the file. If more is reading from a pipe, more simply rings the terminal bell.
:q	
:Q	Exit from more (same as q or Q).

/usr/bin/more The following commands are available only in */usr/bin/more*:

'	Single quote. Go to the point from which the last search started. If no search has been performed in the current file, go to the beginning of the file.
.	Dot. Repeat the previous command.
^ \	Halt a partial display of text. more stops sending output, and displays the usual - -More - - prompt. Some output is lost as a result.

/usr/xpg4/bin/more The following commands are available only in */usr/xpg4/bin/more*:

<i>i</i> ^F	(Control-f) Skip <i>i</i> screens full and print a screenful. (Same as <i>if</i> .)
-------------	---

<code>^G</code>	(Control-g) Display the current line number (same as =).
<code>ig</code>	Go to line number <i>i</i> with the default of the first line in the file.
<code>iG</code>	Go to line number <i>i</i> with the default of the Last line in the file.
<code>ij</code>	Display another line, or <i>i</i> more lines, if specified. (Same as <i>i</i> RETURN.)
<code>ik</code>	Scroll backwards one or <i>i</i> lines, if specified.
<code>mletter</code>	Mark the current position with the name <i>letter</i> .
<code>N</code>	Reverse direction of search.
<code>r</code>	Refresh the screen.
<code>R</code>	Refresh the screen, discarding any buffered input.
<code>iu</code>	
<code>i^U</code>	(Control-u) Scroll backwards one half a screen of <i>i</i> lines, if specified. If <i>i</i> is specified, the count becomes the new default for subsequent <code>d</code> and <code>u</code> commands.
<code>ZZ</code>	Exit from <code>more</code> (same as <code>q</code>).
<code>:e file</code>	Examine (display) a new file. If no <i>file</i> is specified, the current file is redisplayed.
<code>:t tagstring</code>	Go to the tag named by the <i>tagstring</i> argument and scroll/rewrite the screen with the tagged line in the current position. See the <code>ctags</code> utility.
<code>'letter</code>	Return to the position that was previously marked with the name <i>letter</i> .
<code>''</code>	Return to the position from which the last move of more than a screenful was made. Defaults to the beginning of the file.
<code>i?[!]pattern</code>	Search backward in the file for the <i>i</i> th line containing the <i>pattern</i> . The <i>!</i> specifies to search backward for the <i>i</i> th line that does not contain the <i>pattern</i> .
<code>i/!pattern</code>	Search forward in the file for the <i>i</i> th line that does not contain the <i>pattern</i> .
<code>![command]</code>	Invoke a shell or the specified command.

Large File Behavior See [largefile\(5\)](#) for the description of the behavior of `more` and `page` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `more`: `LANG`, `LC_ALL`, `LC_COLLATE` (`/usr/xpg4/bin/more` only), `LC_CTYPE`, `LC_MESSAGES`, `NLSPATH`, and `TERM`.

`/usr/xpg4/bin/more` The following environment variables also affect the execution of `/usr/xpg4/bin/more`:
`COLUMNS` Overrides the system selected horizontal screen size.

EDITOR Used by the `v` command to select an editor.

LINES Overrides the system selected vertical screen size. The `-n` option has precedence over **LINES** in determining the number of lines in a screen.

MORE A string specifying options as described in the **OPTIONS** section, above. As in a command line, The options must be separated by blank characters and each option specification must start with a `-`. Any command line options are processed after those specified in **MORE** as though the command line were: `more $MORE options operands`

退出状态

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

文件

`/usr/lib/more.help` help file for `/usr/bin/more` and `/usr/bin/page` only.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/more`
`/usr/bin/page`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Not enabled

`/usr/xpg4/bin/more`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[cat\(1\)](#), [csh\(1\)](#), [ctags\(1\)](#), [man\(1\)](#), [nroff\(1\)](#), [script\(1\)](#), [sh\(1\)](#), [ul\(1\)](#), [terminfo\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

`/usr/bin/more`
`/usr/bin/page`

[regcomp\(3C\)](#)

`/usr/xpg4/bin/more`

[regex\(5\)](#)

附注

`/usr/bin/more`

Skipping backwards is too slow on large files.

/usr/xpg4/bin/more This utility will not behave correctly if the terminal is not set up properly.

引用名 mp – PDL (Page Description Language, 页面描述语言) 优质打印过滤器文本

用法概要

```
mp [-A4] [-C] [-D target_printer_name] [-F] [-L localname]
  [-P target_spool_printer] [-PS] [-US] [-a] [-c chars]
  [-d] [-e] [-ff] [-fp] [-l] [-ll] [-m] [-M] [-n] [-o]
  [-p prologue] [-s subject] [-tm] [-ts]
  [-u config_file_path] [-v] [-w words] [-z point_size]
  [-?] [filename]...
```

描述

mp 程序在没有 -D 或 -P 选项的情况下进行调用时，将按顺序读取每个 *filename* 并生成可发送到标准输出的、PostScript(tm) 格式的内容的美化版本。如果不提供 *filename* 参数，mp 将读取标准输入。如果标准输入是一个终端，输入将由 EOF 信号（通常为 Control-d）终止。

-D 和 -P 选项需要目标打印机名称作为参数并生成目标打印机的页面描述语言 (Page Description Language, PDL)。-D 选项使 PDL 输出到 stdout，而 -P 选项使 PDL 直接假脱机到打印机。缺少这些选项时，mp 将生成缺省的 PostScript 输出。

mp 程序接受各种 Solaris 语言环境的国际文本文件，并生成适合于指定语言环境的输出。输出还将包含适当的文本布局。例如，输出将包含双向文本着色和成形，因为 mp 中支持复杂文本布局 (complex text layout, CTL)。

邮件项、新闻文章、普通 ASCII 文件、完整邮件文件夹及摘要均属 mp 可接受的输入格式。输出格式包括灰度级菱形或与菱形相同的维的轮廓，在每页的顶部和底部包含标题信息。

选项

支持以下选项：

- a 将文件格式化为新闻文章。顶部标题包含以下文本：“来自**新闻组**的文章”，其中，“**新闻组**”是在“**新闻组**：”行上发现的第一个新闻组。
- A4 使用 A4 纸张大小（8.26 x 11.69 英寸）。
- c *chars* 可从用户的 /etc/passwd 条目的 *gecos* 字段中提取的最大字符数。缺省值为 18。
- C mp 不使用 “\nFrom” 来表示新邮件的开始，而是会查找（并使用）Content-Length: 邮件头的值。如果 Content-Length 没有将您引导至下一个 “\nFrom”，则该邮件头是错误的邮件头，此时 mp 将返回去查找邮件文件夹中的下一个 “\nFrom”。
- d 将文件格式化为摘要。
- D *target_printer_name* 生成目标打印机的 PDL。需要 X 打印服务器连接。*target_printer_name* 可以是 *printer_name@machine[:display_number]* 形式，也可以仅为 *printer_name*。在第一种形式中，mp 尝试连接到 X 打印服务器的显示器 *machine[:display_number]*，其中的目标打印机为 *printer_name*。

- e 采用 ELM 邮件前端中间文件格式。从 ELM 内部打印消息（使用 "p" 命令）时使用，尤其用于打印标记的消息。此选项必须在 ELM 选项设置中指定。
- ff 对文件进行格式化，以用于 Filofax 个人记事本。
- fp 对文件进行格式化，以用于 Franklin Planner 个人记事本。
- F 顶部标题将包含邮寄文章的**发件人**，而不是打印邮件文件的**收件人**。对于拥有自己的个人打印机的人员来说，这是一个有用的选项。
- l 在横向模式下对输出进行格式化。每张纸将打印两页文本。
- ll 在横向模式下对输出进行格式化。每张纸将打印一页文本。这对于打印行长度超过标准行的文件非常有用。
- L *localename* 提供要打印文件的语言环境。如果该命令行选项不存在，mp 将查找 MP_LANG 环境变量。如果该环境变量不存在，则使用 LANG 环境变量。如果上述选项均不存在，mp 将尝试确定其运行所在的语言环境。如果无法确定语言环境，mp 会假定自己是在 C 语言环境中运行。
- m 将文件格式化为邮件文件夹，打印多条消息。
- M 强制 mp 使用 mp.conf 文件来打印输出，即使存在该语言环境 prolog.ps 文件。打印到非本机 PostScript 打印机时非常有用。
- n 关闭页眉和页脚中的灰色条和关联的信息。用于获取与 "*lp filename*" 的输出相类似的输出。
- o 将文件格式化为普通 ASCII 文件。
- p *prologue* 将 *prologue* 文件用作 PostScript/Xprt 的序言文件，覆盖此前定义的任何文件名。该文件指定打印输出的格式。对于 PostScript 输出，*prologue?* 文件将具有 .ps 扩展名。对于 Xprt 客户机（指定 -D 选项时），该文件将具有 .xpr 扩展名。下文的“提供的序言文件”部分对这些文件进行了定义。
- P *target_spool_printer* 将 PDL 假脱机到目标打印机。不向 stdout 发送任何输出。需要 X 打印服务器连接。*target_spool_printer* 可以是 *printer_name@machine[:display_number]* 形式，也可以仅为 *printer_name*。在第一种形式中，mp 尝试连接到显示器 *machine[:display_number]*，其中的目标打印机为 *printer_name*。
- PS 如果邮件消息或摘要消息仅具有 PostScript 作为消息的文本，通常情况下将直接通过这一步。指定此选项时，PostScript 将打印为文本。

<code>-s subject</code>	使用 <i>subject</i> 作为打印输出的新主题。如果打印的是已在命令行上指定的普通 ASCII 文件，主题将缺省为这些文件中每个文件的名称。
<code>-tm</code>	对文件进行格式化，以用于 Time Manager 个人记事本。
<code>-ts</code>	对文件进行格式化，以用于 Time/System International 个人记事本。
<code>-US</code>	使用 US 纸张大小（8.5 x 11 英寸）。这是缺省纸张大小。
<code>-u config_file_path</code>	指定缺省文件 <code>/usr/lib/lp/locale/locale_name/mp/mp.conf</code> 的备用配置文件。必须使用绝对文件路径名。
<code>-v</code>	打印此版本 mp 的版本号。
<code>-w words</code>	可从用户的 <code>/etc/passwd</code> 条目的 <code>gecos</code> 字段中提取的最大字数。缺省值为 3。
<code>-z point_size</code>	以 <i>point_size</i> 所指定的点尺寸打印输出文本。纵向打印的内部缺省值为 12 个点，横向打印则为 9 个点。
<code>-?</code>	打印 mp 的使用状态行。请注意，如果使用的是 <code>cs(1)</code> ，则 <code>?</code> 字符必须进行转义。

操作数

支持下列操作数：

filename 要读取的文件的名称。

示例

mp 打印过滤器可用于打印安装在用户计算机中的任何语言环境中的文件。

示例 1 打印日文文本文件

可以通过输入以下内容在任何非日文 PostScript 打印机中打印以 euc 代码集编码的日文文本文件：

```
example% mp -L ja_JP.eucJP -M ja_JP_eucJP.txt | lp
```

这里，`-L` 选项指定语言环境，`-M` 选项调用 `mp.conf` 配置文件而非缺省的 `prolog.ps` 文件。如果使用的是 `ja_JP.eucJP`，则存在

`/usr/lib/lp/locale/ja_JP.eucJP/mp/mp.conf` 和

`/usr/openwin/lib/locale/ja_JP.eucJP/print/prolog.ps` 文件。因此，使用 `-M` 选项覆盖缺省 `prolog.ps` 文件的优先级。使用 `mp.conf` 作为配置文件即可打印到任何 PostScript 打印机。

通过 `-L` 选项指定的语言环境的编码必须与要打印的文本文件的编码相同。在上述日文文件示例中，如果文本文件是以 Shift-JIS 编码的，请使用以下命令，因为语言环境 `ja_JP.PCK` 是以 SJIS 编码的：

```
example% mp -L ja_JP.PCK -M SJIS.txt | lp
```


示例2 在 Xprt 模式下运行

如果 X 打印服务器守护进程 (`/usr/openwin/bin/Xprt`) 正在网络的任意系统中运行, 则可按以下方式调用 `mp`, 使之以 `Xprt` 所支持的任意页面描述语言进行输出 (`display_number` 的缺省值为 `2100`):

```
example% setenv XPSERVERLIST "machine1[:display_number1] \  
machine2[:display_number2] machine3[:display_number3]"
```

或

```
example% setenv XPDISPLAY machine_name[:display_number]
```

使用选项 `-D printer_name[@machine[:display_number]]` 或 `-P printer_name[@machine[:display_number]]` 可以获得最大的优先级, `mp` 会尝试连接到 `machine[:display_number]` 上运行的且具有 `printer_name` 的 `Xprt`。未指定情况下, 缺省的 `display_number` 值为 `2100`。如果此方法失败, 会尝试 `printer_name`, 并通过下述逻辑获取 `Xprt` 显示器。如果在命令行上仅输入 `-D printer_name` 或 `-P printer_name`, 则以下内容有效。

`mp` 将检查 `XPSERVERLIST` 中是否存在空格分隔的 `Xprt` 服务器的列表, 直至找到一个支持 `printer_name` 参数的服务器。如果找不到, `mp` 将检查 `XPDISPLAY` 环境变量, 该变量的格式为 `machine[:display_number]`。如果该变量尚未设置或无效, `mp` 将尝试连接到缺省显示器 `:2100`。如果仍然无法成功, `mp` 将退出并显示错误消息。

在设置 `XPSERVERLIST` 或 `XPDISPLAY` 的情况下, 要将数据输出到目标打印机, 请输入以下内容:

```
example% mp -D printer_name -L ja_JP.eucJP \  
-M ja_JP_eucJP.txt | lp -d printer_name
```

在 `Xprt` 客户机模式下工作时, 要实现直接假脱机, 请使用 `-P` 选项:

```
example% mp -P printer_name -L ja_JP.eucJP -M ja_JP_eucJP.txt
```

示例3 关闭页眉和页脚

使用 `-n` 选项关闭 `mp` 页眉和页脚:

```
example% mp -n mytext.txt | lp
```

示例4 打印长文本行

使用 `-ll` 选项可在横向模式下打印长度超过 80 列行的文本文件:

```
example% mp -ll mytext.txt | lp
```

示例5 指定打印点尺寸

使用 `-z` 选项指定任意点尺寸。在本例中, 点尺寸为 20 个点:

```
example% mp -z 20 mytext.txt | lp
```

环境变量	XPSERVERLIST	<p>如果 <code>-D</code> 或 <code>-P</code> 的参数采用 <code>printer_name@machine[:display_number]</code> 形式，则仅在 <code>machine[:display_number]</code> 不支持 <code>printer_name</code> 的情况下，才会使用 XPSERVERLIST。</p> <p>XPSERVERLIST 包含要将打印机连接到的 Xprt 显示器的空格分隔列表。mp 将按顺序查看整个列表以获取可以支持给定打印机的 Xprt 服务器，mp 会在其中找到将要连接到的显示器的第一个实例处退出。如果未设置该选项，将使用环境变量 XPDISPLAY。</p>
	XPDISPLAY	<p>如果在命令行中仅通过 <code>printer_name</code> 参数指定 <code>-D</code> 或 <code>-P</code> 选项，环境中没有设置 XPSERVERLIST 变量，将使用 XPDISPLAY 变量来确定运行 X 打印服务器的 <code>machine[:display_number]</code> 来连接客户机。如果 XPDISPLAY 也没有设置，打印服务器启动脚本将在运行客户机的计算机的 2100 端口上启动 Xprt 服务器。作业完成后，脚本即终止打印服务器。如果 XPDISPLAY 已设置，mp 客户机将尝试联系在 XPDISPLAY 上运行的打印服务器。在这种情况下，将不会尝试启动尚未运行的服务器。</p>
	MP_PROLOGUE	<p>用于确定保存页面格式文件（.xpr 或 .ps）的目录。这些文件确定页面装饰、每个物理页的逻辑页数、横向格式还是纵向格式等。缺少 MP_PROLOGUE 时，目录的缺省位置为 <code>/usr/lib/lp/locale/C/mp</code>。</p>
	MP_LANG LANG	<p>如果 <code>-D</code> 选项或 <code>-P</code> 选项均未指定，会将一个序言文件前置到要打印的输出。序言文件称为</p> <p><code>/usr/openwin/lib/locale/localename/print/prolog.ps</code> 或 <code>/usr/lib/lp/locale/localename/mp/prolog.ps</code>，其中 <code>localename</code> 为 MP_LANG 或 LANG 环境变量（如果存在）的值。考虑到向后兼容性的原因，如果这两个变量都存在，将首选文件 <code>/usr/openwin/lib/locale/localename/print/prolog.ps</code>。如果这些文件都不存在，且未指定 <code>-D</code> 选项，将使用名为 <code>/usr/lib/lp/locale/localename/mp/mp.conf</code> 的语言环境配置文件作为配置信息的源，以替代要打印的序言信息。存在 <code>prolog.ps</code> 时，将禁用 <code>mp.conf</code> 以实现向后兼容性。</p>
退出状态		<p>将返回以下退出值：</p> <p>0 成功完成。</p> <p>1 出现错误。</p>
提供的序言文件		<p>提供了以下序言文件。具有 .ps 扩展名的文件用于 PostScript 输出。具有 .xpr 扩展名的文件用于打印服务器客户机。.xpr 文件是针对 300dpi 打印机而创建的，并将扩展到其他分辨率值。</p> <p><code>mp.common.ps</code> 此目录中针对所有其他 .ps 文件的通用序言文件。</p>

mp.pro.ps mp.pro.xpr	缺省情况下使用。
mp.pro.ff.ps mp.pro.ff.xpr	在 -ff 选项有效的情况下使用。
mp.pro.fp.ps mp.pro.fp.xpr	在 -fp 选项有效的情况下使用。
mp.pro.tm.ps mp.pro.tm.xpr	在 -tm 选项有效的情况下使用。
mp.pro.ts.ps mp.pro.ts.xpr	在 -ts 选项有效的情况下使用。
mp.pro.alt.ps mp.pro.alt.xpr	对缺省序言文件的其他方式的修改，将在部标题的右角输出页号。
mp.pro.l.ps mp.pro.l.xpr	用于横向输出的序言文件。
mp.pro.ll.ps mp.pro.ll.xpr	用于横向输出的序言文件，打印行长度超过标准行的文件时。
mp.pro.altl.ps mp.pro.altl.xpr	用于横向输出的备用序言文件。
.cshrc	<code>csh(1)</code> 的初始化文件。
.mailrc	<code>mail(1)</code> 的初始化文件。
/usr/bin/mp	可执行文件。
/usr/lib/lp/locale/C/mp/mp.conf	缺省配置文件。
/usr/lib/lp/locale/C/mp/mp.common.ps	此目录中针对所有其他 .ps 文件的通用序言文件。不适用于 .xpr 文件。
/usr/lib/lp/locale/C/mp/mp.pro.ps /usr/lib/lp/locale/C/mp/mp.pro.xpr	用于邮件打印的缺省序言文件。
/usr/lib/lp/locale/C/mp/mp.pro.l.ps /usr/lib/lp/locale/C/mp/mp.pro.l.xpr	横向格式的缺省序言文件。

文件

`/usr/lib/lp/locale/C/mp/mp.pro.ll.ps`
`/usr/lib/lp/locale/C/mp/mp.pro.ll.xpr`
 横向格式的缺省序言文件，每页一列。打印具有长行的文件时非常有用。

`/usr/lib/lp/locale/C/mp/mp.pro.altl.ps`
`/usr/lib/lp/locale/C/mp/mp.pro.altl.xpr`
 横向格式的备用序言文件。

`/usr/lib/lp/locale/C/mp/mp.pro.alt.ps`
`/usr/lib/lp/locale/C/mp/mp.pro.alt.xpr`
 备选“缺省”序言文件。将页号插入每页的右下角。

`/usr/lib/lp/locale/C/mp/mp.pro.ff.ps`
`/usr/lib/lp/locale/C/mp/mp.pro.ff.xpr`
 Filofax 格式的缺省序言文件。

`/usr/lib/lp/locale/C/mp/mp.pro.fp.ps`
`/usr/lib/lp/locale/C/mp/mp.pro.fp.xpr`
 Franklin Planner 格式的缺省序言文件。

`/usr/lib/lp/locale/C/mp/mp.pro.tm.ps`
`/usr/lib/lp/locale/C/mp/mp.pro.tm.xpr`
 Time Manager 格式的缺省序言文件。

`/usr/lib/lp/locale/C/mp/mp.pro.ts.ps`
`/usr/lib/lp/locale/C/mp/mp.pro.ts.xpr`
 Time/System International 格式的缺省序言文件。

`/usr/openwin/lib/locale/localename/print/prolog.ps`
`/usr/lib/lp/locale/localename/mp/prolog.ps`
 特定于语言环境的缺省序言文件，可用作 `mp.conf` 文件的替代。有关关系的更多详细信息，请参见“环境变量”部分。

`mp.conf` 和 `.xpr` 文件的结构和格式在《国际语言环境指南》中进行了说明。如果需要使用替换字体（包括打印机常驻字体）或对输出格式进行更改，请参见本文档。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	print/mp

另请参见

[csh\(1\)](#)、[mail\(1\)](#)、[mailtool\(1\)](#)、[attributes\(5\)](#)

《国际语言环境指南》

引用名	mpss.so.1 – 用于设置首选页面大小的共享目标文件
用法概要	mpss.so.1
描述	<p>mpss.so.1 是一个共享目标文件，可以通过它来选择性地为已启动的进程及其子孙进程配置首选栈和/或堆页面大小。要启用 mpss.so.1，需要在环境中提供以下字符串（请参见 ld.so.1(1)）以及一个或多个 MPSS（Multiple Page Size Support，多页面大小支持）环境变量：</p> <p>LD_PRELOAD=\$LD_PRELOAD:mpss.so.1</p>
环境变量	<p>在预装入后，mpss.so.1 共享目标文件会读取以下环境变量，以确定任何首选页面大小要求以及这些页面大小可能特定于的任何进程。</p> <p>MPS SHEAP=<i>size</i> MPS SSTACK=<i>size</i></p> <p>MPS SHEAP 和 MPS SSTACK 分别指定了堆和栈的首选页面大小。指定的页面大小将应用于所创建的所有进程。</p> <p><i>size</i> 必须是受支持的页面大小（请参见 pagesize(1)）或 0，在后一种情况下，将由系统来选择合适的页面大小？（请参见 memcntl(2)）。</p> <p><i>size</i> 可使用 K、M、G 或 T 加以限定，以分别指定千字节、兆字节、千兆字节或兆兆字节。</p> <p>MPSSCFGFILE=<i>config-file</i></p> <p><i>config-file</i> 是一个文本文件，其中包含一个或多个 mpss 配置条目，配置条目的格式如下：</p> <p><i>exec-name exec-args:heap-size:stack-size</i></p> <p><i>exec-name</i> 指定应用程序或可执行文件的名称。相应的首选页面大小是为与在文件中找到的第一个 <i>exec-name</i> 匹配的新创建进程（请参见 getexecname(3C)）设置的。</p> <p><i>exec-name</i> 可以是完整的路径名、基名或模式字符串。有关模式匹配的讨论，请参见 sh(1) 中的“生成文件名”。</p> <p><i>exec-args</i> 是与参数相匹配的选择性指定模式字符串。只有未指定 <i>exec-args</i> 时或者 <i>exec-args</i> 出现在 <i>exec-name</i> 的参数内时，才会设置首选页面大小。</p> <p>如果未指定 <i>heap-size</i> 和/或 <i>stack-size</i>，则不会设置相应的首选页面大小。</p> <p>MPSSCFGFILE 优先于 MPS SHEAP 和 MPS SSTACK。未设置 MPSSCFGFILE 时，将从文件 /etc/mpss.conf（若存在该文件）中获取首选页面大小设置。</p>

MPSSERRFILE=*pathname* 缺省情况下，使用级别 LOG_ERR 和工具 LOG_USER 通过 `syslog(3C)` 记录错误消息。如果 **MPSSERRFILE** 包含有效的 *pathname*（例如 `/dev/stderr`），将在其中记录错误消息。

示例

示例 1 使用 **MPSSCFGFILE** 配置首选页面大小

以下 Bourne shell 命令（请参见 [sh\(1\)](#)）使用 **MPSSCFGFILE** 环境变量为其 `exec` 名称以 `foo` 开头的一组选定应用程序配置首选页面大小。假设之前已使用某个文本编辑器（如 [vi\(1\)](#)）创建了 **MPSS** 配置文件 `mpsscfcg`。`cat(1)` 命令仅转储内容。

```
example$ LD_PRELOAD=$LD_PRELOAD:mpss.so.1
example$ MPSSCFGFILE=mpsscfcg
example$ export LD_PRELOAD MPSSCFGFILE
example$ cat $MPSSCFGFILE
foo*:512K:64K
```

在应用程序启动后，可使用 `pmap`（请参见 [proc\(1\)](#)）来查看已配置的实际页面大小。

```
example$ foobar &
example$ pmap -s 'pgrep foobar'
```

如果没有配置所需的页面大小（显示在 `pmap` 输出中），可能是因为在 **MPSS** 配置文件或环境变量存在错误。请检查错误日志（缺省为：`/var/adm/messages`）中的错误。

如果未发现任何错误，则可能归因于资源限制或对齐限制。请参见“附注”部分。

示例 2 使用 **MPS SHEAP** 和 **MPS STACK** 配置首选页面大小

以下 Bourne shell 命令使用 **MPS SHEAP** 和 **MPS STACK** 环境变量为所有应用程序配置 512K 的堆首选页面大小和 64K 的栈首选页面大小。

```
example$ LD_PRELOAD=$LD_PRELOAD:mpss.so.1
example$ MPS SHEAP=512K
example$ MPS STACK=64K
example$ export LD_PRELOAD MPS SHEAP MPS STACK
```

示例 3 优先级规则（“示例 2”的续）

MPSSCFGFILE 中的首选页面大小配置优先于 **MPS SHEAP** 和 **MPS STACK**。将以下命令附加到 **Example 2** 的命令后，这表示将在配置文件中为所有应用程序配置 512K 的堆首选页面大小和 64K 的栈首选页面大小，除了那些应用程序、`ls` 命令以及所有以 `ora1` 作为参数并以 `ora` 开头的应用程序。

```
example$ MPSSCFGFILE=mpsscfcg2
example$ export MPSSCFGFILE
example$ cat $MPSSCFGFILE
ls::
ora* ora1:4m:4m
```

文件 `/usr/lib/ld/map.bssalign` 用于对齐 `bss` 的模板链接编辑器 `mapfile`（请参见“附注”部分）。

`/etc/mpss.conf` 配置文件

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/extended-system-utilities
接口稳定性	Committed（已确定）

另请参见 [cat\(1\)](#)、[ld\(1\)](#)、[ld.so.1\(1\)](#)、[pagesize\(1\)](#)、[ppgsz\(1\)](#)、[proc\(1\)](#)、[sh\(1\)](#)、[vi\(1\)](#)、[exec\(2\)](#)、[fork\(2\)](#)

附注 堆和栈首选页面大小是继承的。子进程的首选页面大小与其父进程的相同。在执行 [exec\(2\)](#) 时，除非已通过 `mpss` 共享目标文件配置了首选页面大小，否则首选页面大小将重新设置为缺省的系统页面大小。

也可使用 [ppgsz\(1\)](#)（一个 `proc` 工具）来设置首选栈和/或堆页面大小。它不能基于名称匹配选择性地为后代配置页面大小。

另请参见 [ppgsz\(1\)](#) 中的“附注”部分。

引用名 msgcc – C language message catalog compiler

用法概要 msgcc [-M-*option*] [cc-*optionsoption*] *file*...

描述 msgcc is a C language message catalog compiler. It accepts cc style options and arguments.

A [msgcpp\(1\)](#) .msg file is generated for each input .c file. If the -c option is not specified then a [gencat\(1\)](#) format .msg file is generated from the input .mso and .msg files. If -c is not specified then a .msg suffix is appended to the -o file if it doesn't already have a suffix. The default output is a.out.msg if -c and -o are not specified.

If -M-new is not specified then messages are merged with those in the pre-existing -o file.

选项 The following options are supported:

cc-options Specify cc style options and arguments.

-M-*option* Set a msgcc option.

Specify option as one of the following:

mkmsgs The -o file is assumed to be in [mkmsgs\(1\)](#) format.

new Create a new -o file.

preserve Messages in the -o file that are not in new .msg file arguments are preserved. The default is to either reuse the message numbers with new message text that is similar to the old or to delete the message text, leaving an unused message number.

set=number Set the message set number to *number*. The default is 1.

similar=number The message text similarity message threshold. The similarity measure between old and new message text is:

$$100 * (2 * \text{gzip}(\text{old} + \text{new}) \backslash \\ / (\text{gzip}(\text{old}) + \text{gzip}(\text{new})) - 1)$$

where $\text{gzip}(x)$ is the size of text x when compressed by gzip. The default threshold is `$_similar_`\$. A threshold of 0 turns off message replacement, but unused old messages are still deleted. Use -M-preserve to preserve all old messages.

verbose Trace similar message replacements on the standard error.

操作数 The following operands are supported:

file Specifies the name of the file on which msgcc operates.

退出状态

0 Successful completion.

>0 An error occurred.

示例

示例1 Usingmsgcc

The following example uses msgcc to extract localizable strings from the file `hello.c`, marked using `ERROR_dictionary()`, writes them to the file `hello.mso`, and creates a gencat format `xxx.msg` file:

```
example% cat hello.c

#include <stdio.h>
#include <stdlib.h>

/*
 * dummy macro to avoid including
 * libast headers
 */
#define ERROR_dictionary(x) x

int main(int ac, char *av[])
{
    puts( ERROR_dictionary("hello world") );
    return( EXIT_SUCCESS );
}

example% msgcc -o xxx -D__STDC__ -D__i386 hello.c

example% cat hello.mso
str "hello world"

example% cat xxx.msg
$ xxx message catalog
$translation msgcc 2007-09-25
$set 1
$quote "
1 "hello world"
```

Authors

Glenn Fowler, gsf@research.att.com

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/astdev
Interface Stability	Volatile

另请参见

[cpp\(1\)](#), [gencat\(1\)](#), [mkmsgs\(1\)](#), [msggen\(1\)](#), [msgcpp\(1\)](#), [msgcvt\(1\)](#), [attributes\(5\)](#)

引用名	msgcpp – C language message catalog preprocessor
用法概要	msgcpp [-ACEHMPVX] [-D <i>name</i> [= <i>value</i>]] [-I <i>directory</i>] [-U <i>name</i>] [-T[<i>length</i>]] [-Y <i>directory</i>] [<i>input</i> [<i>output</i>]]
描述	<p>msgcpp is a C language message catalog preprocessor. It accepts cpp(1) style options and arguments. msgcpp preprocesses an input C source file and emits keyed lines to the output, usually for further processing by msgcc(1). msgcc output is in the gencat(1) syntax. Candidate message text is determined by arguments to the last <code><error.h></code> and <code><option.h></code> functions. The msgcpp keyed output lines are:</p> <p><i>cmd command</i> <i>command</i> is a candidate for --??keys option string generation. This is triggered by <code>b_command(int argc, in the input.</code></p> <p><i>def name string</i> <i>name</i> is a candidate variable with <i>string</i> value string.</p> <p><i>str string</i> <i>string</i> should be entered into the catalog.</p> <p><i>var name</i> If <i>def name</i> occurs then its string value should be entered into the catalog.</p> <p>The input source file is preprocessed with the <code>pp:allpossible</code> option on. This enables non-C semantics. All source should first be compiled error-free with a real compiler before running msgcpp. The following changes are enabled for the top level files. Included file behavior is not affected.</p> <ol style="list-style-type: none"> 1. All <code>#if</code>, <code>#ifdef</code> and <code>#ifndef</code> branches are enabled. 2. The first definition for a macro is retained, even when subsequent <code>#define</code> statements would normally redefine the macro. <code>#undef</code> must be used to redefine a macro. 3. Macro calls with an improper number of arguments are silently ignored. 4. <code>#include</code> on non-existent headers are silently ignored. 5. Invalid C source characters are silently ignored. <p><code>msgcat.h</code> is included if it exists. This file may contain macro definitions for functions that translate string arguments. If <code>foo</code> is a function that translates its string arguments then include the line <code>#define foo _TRANSLATE_</code> in <code>msgcat.h</code>, or specify the option <code>-Dfoo=_TRANSLATE_</code>. If <code>bar</code> is a function that translates string arguments if the first argument is <code>stderr</code>, then use either <code>#define bar _STDIO_</code> or <code>-Dbar=_STDIO_</code>.</p> <p>The macro <code>_BLD_msgcat</code> is defined to be 1. As an alternative to <code>msgcat.h</code>, <code>_TRANSLATE_</code> definitions could be placed inside <code>#ifdef _BLD_msgcat ... #endif</code>.</p>
选项	<p>The following options are supported:</p> <p>-A</p> <p>--assert=<i>assertion</i> Enter the assertion using <code>#assert</code> for system V compatibility.</p> <p>-C</p> <p>--comments Pass comments to the output.</p>

Comments are omitted by default.

-D

--define=*name*[=*value*]

Define the macro *name* to have *value*. This is the only portable way to pass options through cc to [cpp\(1\)](#).

- If *=value* is omitted, *value* is assumed to be 1.
- If *name* begins with :, then it is interpreted as a libpp #pragma pp: statement.
- If *name* begins with %, it is interpreted as a libpp # directive statement.
- If *name* begins with a - or a +, it is interpreted as a libpp option.

- turns the option on, + turns it off.

- Most options have a #pragma counterpart that is listed with the option definition.

-D-C

pp:compatibility

Preprocess for K&R C compatibility.

-D-Dlevel

pp:debug level *level*

Set the debug trace level.

Specify *level* as a number greater than or equal to 0. Higher levels produce more output. Levels higher than 3 can only be enabled in the -g compiled versions.

-D-Fname

Set the main input file name to *name*. This only affects the error messages and the line sync output.

-D-H

pp:hosted

All directories are hosted. Compatibility warning messages from the hosted directory headers are suppressed.

-D-I

pp:cdir

All directories contain C headers. This option is only used only with -D-+.

-D-K

pp:keyargs

- Enable the non-standard *name=value* macro argument mode.
- D-L[id]**
pp:lineid [id]
Set the line sync directive id to *id*. If *id* is not specified, set to null.
- D-M**
pp:nomultiple
Disable multiple include detection.
- D-P**
pp:passthrough
Enable the non-standard passthrough mode. This can be useful for processing non-C input.
- D-Q**
pp:dump
Dump macro definitions to the output so that the output may be passed through cpp again. This is used for generating precompiled headers.
- D-R**
pp:transition
Enable the transition preprocessing mode. This is used for compilers that cannot make up their semantics between K&R and ISO C.
- D-S**
pp:strict
Enable strict preprocessing semantics and warnings. This works with any mode (compatibility, transition, or the default ISO).
- D-Ttest**
pp:test test
Enable implementation specific test code according to *test*.
- D-W**
pp:warn
Enable pedantic warnings in non-hosted files.
- D-X[cc]**
Preprocess for the *cc* compiler, which must be an executable path or an executable on \$PATH.
- D-Z**
pp:pool
Enable pool mode.

	-D-d	List canonicalized <code>#define</code> statements for non-predefined macros in the output.
	-D-m	List canonicalized <code>#define</code> statements for all macros. All other output is disabled.
	-D+	
	pp:plusplus	Preprocess for the C++ dialect.
-E		
--preprocess		Ignored; for compatibility with very old compilers.
-H		
--include-reference		Emit <code>#include</code> file paths on the standard error, one per line, indented to show nesting.
-I		
--include[=<i>directory</i>]		Append <i>directory</i> to the list of directories searched for <code>#include</code> files.
		If <i>directory</i> is <code>-</code> :
		1. <code>-I</code> directories before <code>-I-</code> are searched only for <code>"..."</code> include files
		2. <code>-I</code> directories after <code>-I-</code> are searched for <code>"..."</code> and <code><"..."></code> include files
		3. the <code>directory .</code> is searched only if it is explicitly specified by an <code>-I</code> option
	-I-C<i>directory</i>	
	pp:cdir <i>directory</i>	Mark <i>directory</i> as a C header directory. This option is used with <code>pp:plusplus</code> .
	-I-D[<i>file</i>]	
		Read the default probe definitions from <i>file</i> , or ignore the default definitions if <i>file</i> is omitted.
	-I-H<i>directory</i>	
	pp:hostdir <i>directory</i>	Mark <i>directory</i> as a hosted directory. Headers from hosted directories have compatibility warnings disabled.
	-I-I<i>header</i>	
	pp:ignore <i>header</i>	Add <i>header</i> to the list of ignored headers.

- I-*Mfile*
file contains a sequence of header [= "map"] lines, where header is either <name> or "name", and "map" is an explicit binding for header. header is ignored if "map" is omitted.
- I-*Rfile*
 Include *file* but do not emit text or line syncs.
- I-*Sdirectory*
 Add *directory* to the default standard include directory list.
- I-*Tfile*
 Include *file* and emit text to the output file. The option value can be omitted.
- M
 --dependencies
 Generate [make\(1S\)](#) dependencies. This option is not needed with `nmake`.
- The -M option can be followed by optional flags to change the dependency output styles.
- The following optional flags are supported:
- D Generate dependencies in a separate .d file. Preprocessed output is still written to output, or the standard output if output is omitted.
 - G Also generate missing dependencies.
 - M Only generate local header dependencies. Hosted headers are omitted. Hosted headers are determined by the -I-H option and the --pp:hosted and pp:hostdir pragmas. No special distinction is made between the "" and <> include styles.
- P
 --sync
 Emit line syncs.
- Line sync is turned on by default. -P means --nosync.
- T[*length*]
 If not gcc, truncate identifiers to *length* characters for compatibility with old AT&T compilers.
- U
 --undefine=*name*
 Remove the definition for the macro *name*.
- V
 --version
 Emit the libpp version.

-X
--argmode Enable *name=value* macro arguments for ease1 compatibility.
-Y
--standard=*directory* Add *directory* to the list searched for #include <...> files.

操作数

The following operands are supported:

input Specifies C source file to preprocess.

output Specifies output file.

退出状态

0 Successful completion.

>0 An error occurred.

示例

示例 1 Using msgcpp to Extract Localizable Strings

The following example uses msgcpp to extract localizable strings from the file `hello.c`, marked using the `ERROR_dictionary()`, and writes them to the file `hello.mso`:

```
example% cat hello.c
```

```
#include <stdio.h>
#include <stdlib.h>

/*
 * dummy macro to avoid including
 * libast headers
 */
#define ERROR_dictionary(x) x

int main(int ac, char *av[])
{
    puts( ERROR_dictionary("hello world") );
    puts( ERROR_dictionary("hello all") );
    return( EXIT_SUCCESS );
}
```

```
example% msgcpp -D__STDC__ -D__i386 hello.c hello.mso
```

```
example% cat hello.mso
str "hello world"
str "hello all"
```

Authors

Glenn Fowler, gsf@research.att.com

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/astdev
Interface Stability	Volatile

另请参见

[cpp\(1\)](#), [gencat\(1\)](#), [msgcc\(1\)](#), [msgcvt\(1\)](#), [msggen\(1\)](#), [make\(1S\)](#), [attributes\(5\)](#)

Kernighan, Brian W. and Ritchie, Dennis M., The C Programming Language, Prentice Hall, 1988.

引用名 msgcvt – convert message file to and from HTML

用法概要 msgcvt [-hmr]

描述 msgcvt reads a [gencat\(1\)](#) format file on the standard input and converts it to HTML on the standard output. The input file must contain the control statement `"` and use the `"` character to quote message text. The output is in a form suitable for automatic translation by web sites such as <http://babelfish.yahoo.com>.

选项 The following options are supported:

-h

--html Generate HTML from [gencat\(1\)](#) input.

This is the default.

-m

--msg Generate a [gencat\(1\)](#) message file from (presumably translated) HTML. Wide characters are UTF-8 encoded.

-r

--raw The message file is raw message text, one message per line, with no quoting or line numbering.

退出状态 0 Successful completion.

>0 One or more specified jobs does not exist.

示例 示例 1 Generating a gencat Message Catalog File

The following example generates a [gencat\(1\)](#) message catalog file from an HTML file:

```
example% cat example.html | msgcvt -m > examplecat
```

Authors Glenn Fowler, gsf@research.att.com

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/astdev
Interface Stability	Volatile

另请参见 [gencat\(1\)](#), [msgcc\(1\)](#), [msggen\(1\)](#), [attributes\(5\)](#)

引用名	msgfmt – create a message object from a message file																						
用法概要	<pre>msgfmt [-D <i>dir</i> --directory=<i>dir</i>] [-f --use-fuzzy] [-g] [-o <i>output-file</i> --output-file=<i>output-file</i>] [-s] [--strict] [-v] [--verbose] <i>filename.po...</i></pre>																						
描述	<p>The msgfmt utility creates message object files from portable object files (<i>filename.po</i>), without changing the portable object files.</p> <p>The .po file contains messages displayed to users by system commands or by application programs. .po files can be edited. The messages in these files can be rewritten in any language supported by the system.</p> <p>The xgettext(1) command can be used to create .po files from script or programs.</p> <p>msgfmt interprets data as characters according to the current setting of the LC_CTYPE locale category or according to the codeset specified in the .po file.</p>																						
选项	<p>The following options are supported:</p> <table border="0"> <tr> <td style="padding-right: 1em;">-D <i>dir</i></td> <td></td> </tr> <tr> <td style="padding-right: 1em;">--directory=<i>dir</i></td> <td>Adds <i>dir</i> to the list for input files search.</td> </tr> <tr> <td style="padding-right: 1em;">-f</td> <td></td> </tr> <tr> <td style="padding-right: 1em;">--use-fuzzy</td> <td>Uses fuzzy entries in output. If this option is not specified, fuzzy entries are not included into the output. These options are ignored if Solaris message catalogs are processed.</td> </tr> <tr> <td style="padding-right: 1em;">-g</td> <td>Directs the utility to generate the GNU-compatible message catalog file. This option cannot be specified with the -s option.</td> </tr> <tr> <td style="padding-right: 1em;">-o <i>output-file</i></td> <td></td> </tr> <tr> <td style="padding-right: 1em;">--output=<i>output-file</i></td> <td>Specifies the output file name as <i>output-file</i>. All domain directives and duplicate msgids in the .po file are ignored.</td> </tr> <tr> <td style="padding-right: 1em;">-s</td> <td>Directs the utility to generate the Solaris message catalog file. This option cannot be specified with the -g option.</td> </tr> <tr> <td style="padding-right: 1em;">--strict</td> <td>Directs the utility to append the suffix .mo to the generating message object file name if it doesn't have this suffix. This option is ignored if Solaris message catalogs are processed.</td> </tr> <tr> <td style="padding-right: 1em;">-v</td> <td></td> </tr> <tr> <td style="padding-right: 1em;">--verbose</td> <td>Verbose. Lists duplicate message identifiers if Solaris message catalog files are processed. Message strings are not redefined.</td> </tr> </table> <p>If GNU-compatible message files are processed, this option detects and diagnoses input file anomalies which might represent translation errors. The msgid and msgstr strings are studied and</p>	-D <i>dir</i>		--directory= <i>dir</i>	Adds <i>dir</i> to the list for input files search.	-f		--use-fuzzy	Uses fuzzy entries in output. If this option is not specified, fuzzy entries are not included into the output. These options are ignored if Solaris message catalogs are processed.	-g	Directs the utility to generate the GNU-compatible message catalog file. This option cannot be specified with the -s option.	-o <i>output-file</i>		--output= <i>output-file</i>	Specifies the output file name as <i>output-file</i> . All domain directives and duplicate msgids in the .po file are ignored.	-s	Directs the utility to generate the Solaris message catalog file. This option cannot be specified with the -g option.	--strict	Directs the utility to append the suffix .mo to the generating message object file name if it doesn't have this suffix. This option is ignored if Solaris message catalogs are processed.	-v		--verbose	Verbose. Lists duplicate message identifiers if Solaris message catalog files are processed. Message strings are not redefined.
-D <i>dir</i>																							
--directory= <i>dir</i>	Adds <i>dir</i> to the list for input files search.																						
-f																							
--use-fuzzy	Uses fuzzy entries in output. If this option is not specified, fuzzy entries are not included into the output. These options are ignored if Solaris message catalogs are processed.																						
-g	Directs the utility to generate the GNU-compatible message catalog file. This option cannot be specified with the -s option.																						
-o <i>output-file</i>																							
--output= <i>output-file</i>	Specifies the output file name as <i>output-file</i> . All domain directives and duplicate msgids in the .po file are ignored.																						
-s	Directs the utility to generate the Solaris message catalog file. This option cannot be specified with the -g option.																						
--strict	Directs the utility to append the suffix .mo to the generating message object file name if it doesn't have this suffix. This option is ignored if Solaris message catalogs are processed.																						
-v																							
--verbose	Verbose. Lists duplicate message identifiers if Solaris message catalog files are processed. Message strings are not redefined.																						

compared. It is considered abnormal if one string starts or ends with a newline while the other does not. Also, if the string represents a format string used in a printf-like function, both strings should have the same number of % format specifiers, with matching types. If the flag `c` - format appears in the special comment '#' for this entry, a check is performed.

用法

The format of portable object files (.po files) is defined as follows. Each .po file contains one or more lines, with each line containing either a comment or a statement. Comments start the line with a pound sign (#) and end with the newline character. All comments (except special comments described later) and empty lines are ignored. The format of a statement is:

directive *value*

Each *directive* starts at the beginning of the line and is separated from *value* by white space (such as one or more space or tab characters). *value* consists of one or more quoted strings separated by white space. Use any of the following types of directives for the Solaris message file:

```
domain domainname
msgid message_identifier
msgstr message_string
```

For a GNU-compatible message file, use any of the following types of directives:

```
domain domainname
msgid message_identifier
msgid_plural untranslated_string_plural
msgstr message_string
msgstr[n] message_string
```

The behavior of the `domain` directive is affected by the options used. See `OPTIONS` for the behavior when the `-o` or `--output-file` options are specified. If the `-o` or `--output-file` options are not specified, the behavior of the `domain` directive is as follows:

- All msgids from the beginning of each .po file to the first `domain` directive are put into a default message object file. The default message object file is named `messages.mo`, if the Solaris message catalog file format is used to generate the message object file or if the `--strict` option is specified. Otherwise, the default message object file is named `messages`.
- When `msgfmt` encounters a `domain domainname` directive in the .po file, all following msgids until the next `domain` directive are put into the message object file, named `domainname.mo`, if the Solaris message catalog file format is used to generate the message object file or if the `--strict` option is specified. Otherwise, the msgids are put into the message object file named `domainname`.
- Duplicate msgids are defined in the scope of each domain. That is, a msgid is considered a duplicate only if the identical msgid exists in the same domain.

- All duplicate msgids are ignored.

The `msgid` directive specifies the value of a message identifier associated with the directive that follows it. The `msgid_plural` directive specifies the plural form message specified to the plural message handling functions `ngettext()`, `dngettext()`, or `dcngettext()`. The *message_identifier* string identifies a target string to be used at retrieval time. Each statement containing a `msgid` directive must be followed by a statement containing a `msgstr` directive or `msgstr[n]` directives.

The `msgstr` directive specifies the target string associated with the *message_identifier* string declared in the immediately preceding `msgid` directive.

The directive `msgstr[n]` (where $n = 0, 1, 2, \dots$) specifies the target string to be used with plural form handling functions `ngettext()`, `dngettext()`, and `dcngettext()`.

Message strings can contain the escape sequences `\n` for newline, `\t` for tab, `\v` for vertical tab, `\b` for backspace, `\r` for carriage return, `\f` for formfeed, `\\` for backslash, `\"` for double quote, `\so\`; a for alarm, `\ddd` for octal bit pattern, and `\xDD` for hexadecimal bit pattern.

Comments for a GNU-compatible message file should be in one of the following formats (the `msgfmt` utility will ignore these comments when processing Solaris message files):

```
# translator-comments
# . automatic-comments
#: reference..
#, flag
```

The `'#:'` comments indicate the location of the `msgid` string in the source files in *filename:line* format. The `'#'`, `'#.'`, and `'#:'` comments are informative only and are silently ignored by the `msgfmt` utility. The `'#,'` comments require one or more flags separated by the comma character. The following *flags* can be specified:

`fuzzy` This flag can be inserted by the translator. It shows that the `msgstr` string might not be a correct translation (anymore). Only the translator can judge if the translation requires further modification or is acceptable as is. Once satisfied with the translation, the translator removes this `fuzzy` flag. If this flag is specified, the `msgfmt` utility will not generate the entry for the immediately following `msgid` in the output message catalog.

`c-format`

`no-c-format` The `c-format` flag indicates that the `msgid` string is used as a format string by `printf`-like functions. In case the `c-format` flag is given for a string, the `msgfmt` utility does some more tests to check the validity of the translation.

In the GNU-compatible message file, the `msgid` entry with empty string (`""`) is called the header entry and treated specially. If the message string for the header entry contains `np\lurals=value`, the value indicates the number of plural forms. For example, if `np\lurals=4`,

there are four plural forms. If `nplurals` is defined, the same line should contain `plural=expression`, separated by a semicolon character. The *expression* is a C language expression to determine which version of `msgstr[n]` is to be used based on the value of *n*, the last argument of `ngettext()`, `dngettext()`, or `dcngettext()`. For example,

```
nplurals=2; plural= n == 1 ? 0 : 1
```

indicates that there are two plural forms in the language. `msgstr[0]` is used if `n == 1`, otherwise `msgstr[1]` is used. For another example:

```
nplurals=3; plural= n == 1 ? 0 : n == 2 ? 1 : 2
```

indicates that there are three plural forms in the language. `msgstr[0]` is used if `n == 1`, `msgstr[1]` is used if `n == 2`, otherwise `msgstr[2]` is used.

If the header entry contains a `charset=codeset` string, the *codeset* is used to indicate the codeset to be used to encode the message strings. If the output string's codeset is different from the message string's codeset, codeset conversion from the message string's codeset to the output string's codeset will be performed upon the call of `gettext()`, `dgettext()`, `dcgettext()`, `ngettext()`, `dngettext()`, and `dcngettext()` for the GNU-compatible message catalogs. The output string's codeset is determined by the current locale's codeset (the return value of `nL_langinfo(CODESET)`) by default, and can be changed by the call of `bind_textdomain_codeset()`.

Message catalog file format

The `msgfmt` utility can generate the message object both in Solaris message catalog file format and in GNU-compatible message catalog file format. If the `-s` option is specified and the input file is a Solaris `.po` file, the `msgfmt` utility generates the message object in Solaris message catalog file format. If the `-g` option is specified and the input file is a GNU `.po` file, the `msgfmt` utility generates the message object in GNU-compatible message catalog file format. If neither the `-s` nor `-g` option is specified, the `msgfmt` utility determines the message catalog file format as follows:

- If the `.po` file contains a valid GNU header entry (having an empty string for `msgid`), the `msgfmt` utility uses the GNU-compatible message catalog file format.
- Otherwise, the `msgfmt` utility uses the Solaris message catalog file format.

If the `msgfmt` utility determined that the Solaris message catalog file format is used, as above, but found the `.po` file contains directives that are specific to the GNU-compatible message catalog file format, such as `msgid_plural` and `msgstr[n]`, the `msgfmt` utility handles those directives as invalid specifications.

示例

示例 1 Creating message objects from message files

In this example, `module1.po` and `module2.po` are portable message objects files.

```
example% cat module1.po
# default domain "messages.mo"
```

示例 1 Creating message objects from message files (续)

```

msgid "msg 1"
msgstr "msg 1 translation"
#
domain "help_domain"
msgid "help 2"
msgstr "help 2 translation"
#
domain "error_domain"
msgid "error 3"
msgstr "error 3 translation"
example% cat module2.po
# default domain "messages.mo"
msgid "mesg 4"
msgstr "mesg 4 translation"
#
domain "error_domain"
msgid "error 5"
msgstr "error 5 translation"
#
domain "window_domain"
msgid "window 6"
msgstr "window 6 translation"

```

The following command will produce the output files `messages.mo`, `help_domain.mo`, and `error_domain.mo` in Solaris message catalog file format:

```
example% msgfmt module1.po
```

The following command will produce the output files `messages.mo`, `help_domain.mo`, `error_domain.mo`, and `window_domain.mo` in Solaris message catalog file format:

```
example% msgfmt module1.po module2.po
```

The following command will produce the output file `hello.mo` in Solaris message catalog file format:

```
example% msgfmt -o hello.mo module1.po module2.po
```

环境变量

See [environ\(5\)](#) for descriptions of the following environmental variables that affect the execution of `msgfmt`: `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Enabled

另请参见

[xgettext\(1\)](#), [gettext\(3C\)](#), [setlocale\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注

Installing message catalogs under the C locale is pointless, since they are ignored for the sake of efficiency.

引用名	msggen – generate a machine independent formatted message catalog																										
用法概要	msggen [-fls] <i>catfile</i> [<i>msgfile</i>]																										
描述	<p>msggen merges the message text source file <i>msgfile</i> into a machine independent formatted message catalog <i>catfile</i>. The file <i>catfile</i> is created if it does not already exist. If <i>catfile</i> does exist, its messages are included in the new <i>catfile</i>. If set and message numbers collide, the new message text defined in <i>msgfile</i> replaces the old message text currently contained in <i>catfile</i>.</p> <p>Non-ASCII characters must be UTF-8 encoded. iconv(1) can be used to convert to/from UTF-8.</p>																										
选项	<p>The following options are supported:</p> <p>-f --format List the printf(3C) format signature for each message in <i>catfile</i>. A format signature is one line containing one character for each format specification:</p> <table> <tr><td>c</td><td>char</td></tr> <tr><td>d</td><td>double</td></tr> <tr><td>D</td><td>long double</td></tr> <tr><td>f</td><td>float</td></tr> <tr><td>h</td><td>short</td></tr> <tr><td>i</td><td>int</td></tr> <tr><td>j</td><td>long long</td></tr> <tr><td>l</td><td>long</td></tr> <tr><td>p</td><td>void*</td></tr> <tr><td>s</td><td>string</td></tr> <tr><td>t</td><td>ptrdiff_t</td></tr> <tr><td>z</td><td>size_t</td></tr> <tr><td>?</td><td>unknown</td></tr> </table> <p>-l --list List <i>catfile</i> in UTF-8 msgfile form.</p> <p>-s --set Convert the <i>catfile</i> to a message set number and print the number on the standard output.</p>	c	char	d	double	D	long double	f	float	h	short	i	int	j	long long	l	long	p	void*	s	string	t	ptrdiff_t	z	size_t	?	unknown
c	char																										
d	double																										
D	long double																										
f	float																										
h	short																										
i	int																										
j	long long																										
l	long																										
p	void*																										
s	string																										
t	ptrdiff_t																										
z	size_t																										
?	unknown																										

操作数

The following operands are supported:

catfile Machine independent formatted message catalog file.

msgfile Message text source file.

用法

Message text source files are in [gencat\(1\)](#) format, defined as follows. The fields of a message text source line are separated by a single blank character. Any other blank characters are considered to be part of the subsequent field. The `NL_*` constants are defined in one or both of `<limits.h>` and `<nل_types.h>`.

\$comment

A line beginning with a `$` followed by a blank character is treated as a comment.

\$delset n comment

This line deletes message set *n* from an existing message catalog. *n* denotes the set number [1, `NL_SETMAX`]. Any text following the set number is treated as a comment.

\$quote c

This line specifies an optional quote character *c*, which can be used to surround message-text so that trailing spaces or empty messages are visible in a message source line. By default, or if an empty `$quote` directive is supplied, no quoting of message-text is recognized.

\$set n comment

This line specifies the set identifier of the following messages until the next `$set` or end-of-file (EOF) appears. *n* denotes the set identifier, which is defined as a number in the range [1, `NL_SETMAX`]. Set numbers need not be contiguous. Any text following the set identifier is treated as a comment. If no `$set` directive is specified in a message text source file, all messages are located in message set 1.

\$translation identification YYYY-MM-DD[, . . .]

Append translation information to the message catalog header. Only the newest date for a given identification is retained in the catalog. Multiple translation lines are combined into a single, comma-separated list.

m message-text

m denotes the message identifier, which is defined as a number in the range [1, `NL_MSGMAX`]. The message-text is stored in the message catalogue with the set identifier specified by the last `$set` directive, and with message identifier *m*. If the message-text is empty, and a blank character field separator is present, an empty string is stored in the message catalogue. If a message source line has a message number, but neither a field separator nor message-text, the existing message with that number (if any) is deleted from the catalogue. Message identifiers need not be contiguous. There are no *message-text* length restrictions.

退出状态

0 Successful completion.

>0 One or more specified jobs does not exist.

示例

示例 1 Using msggen

The following example generates a message catalog xxx from the message file xxx.msg:

```
example% msggen xxx xxx.msg
```

Authors Glenn Fowler, gsf@research.att.com

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	developer/astdev
Interface Stability	Volatile

另请参见 [gencat\(1\)](#), [iconv\(1\)](#), [msgcc\(1\)](#), [printf\(3C\)](#), [attributes\(5\)](#)

引用名	msgget – get a message from a message catalog
用法概要	msgget <i>locale</i> [<i>command</i> :] <i>catalog</i> [<i>set</i> .] <i>number</i> [<i>text</i>]
描述	msgget gets the message corresponding to the parameters. See OPERANDS.
操作数	The following operands are supported: <i>catalog</i> Specifies the message catalog name. <i>command</i> Specifies command-specific message. <i>locale</i> Specifies the locale. If <i>locale</i> is - then the current locale is used. [<i>set</i>] . <i>number</i> Identifies the message by message number and an optional message set. If specified as - , the message set and number are determined by looking up text in the corresponding C locale message catalog. <i>text</i> Specifies the text of the message to be output upon error.
退出状态	0 Successful completion. >0 An error occurred.

示例

示例 1 Getting a Message in the Current Locale

The following example gets msg 1 in the current locale from message catalog hello:

```
example% msgget - hello 1
hello world
```

Authors Glenn Fowler, gs f@research.att.com

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/astdev
Interface Stability	Volatile

另请参见 [iconv\(1\)](#), [msgcc\(1\)](#), [msggen\(1\)](#), [attributes\(5\)](#)

引用名	mt – magnetic tape control
用法概要	mt [-f <i>tapename</i>] <i>command</i> ... [<i>count</i>]
描述	The <i>mt</i> utility sends commands to a magnetic tape drive. If -f <i>tapename</i> is not specified, the environment variable TAPE is used. If TAPE does not exist, <i>mt</i> uses the device /dev/rmt/0n.
选项	The following options are supported: -f <i>tapename</i> Specifies the raw tape device.
操作数	The following operands are supported: <i>count</i> The number of times that the requested operation is to be performed. By default, <i>mt</i> performs <i>command</i> once. Multiple operations of <i>command</i> can be performed by specifying <i>count</i> . <i>command</i> The following available commands that can be sent to a magnetic tape drive are supported. Only as many characters as are required to uniquely identify a <i>command</i> need be specified. asf Specifies absolute space to <i>count</i> file number. This is equivalent to a rewind followed by a fsf <i>count</i> . bsf Back spaces over <i>count</i> EOF marks. The tape is positioned on the beginning-of-tape side of the EOF mark. bsr Back spaces <i>count</i> records. bssf Back spaces over the requested number of sequential file marks. Sequential file marks are where the file marks are one right after the other with no other blocks of any kind between the file marks. The number argument specifies how many sequential file marks to which to space. For example, bssf 4 searches backwards to the first place where there are 4 sequential file marks and positions to the BOP side of the 4th file mark. This command is not supported by all drives. eof weof Writes <i>count</i> EOF marks at the current position on the tape. fsf Forward spaces over <i>count</i> EOF marks. The tape is positioned on the first block of the file. fsr Forward spaces <i>count</i> records. fssf Forward spaces the over requested number of sequential file marks. Sequential file marks are where the file marks are one right after the other with no other blocks of any kind between the file marks. The number argument specifies how many sequential file marks to

which to space. For example, `fsf 4` searches forwards to the first place where there are 4 sequential file marks and positions after the 4th file mark.

This command is not supported by all drives.

<code>load</code>	Requests drive load and thread current media. Not supported by all drives.
<code>lock</code>	Prevents media removal.
<code>nbsf</code>	Back spaces <i>count</i> files. The tape is positioned on the first block of the file. This is equivalent to <i>count+1</i> <code>bsfs</code> followed by one <code>fsf</code> .
<code>seek</code>	Positions to requested logical tape position.
<code>tell</code>	Gets and prints current logical tape position.
<code>unlock</code>	Allows media removal.

If *count* is specified with any of the following commands, the *count* is ignored and the command is performed only once.

<code>config</code>	Reads the drives current configuration from the driver and displays it in <code>st.conf</code> format. See st(7D) for definition of fields and there meanings.
<code>eom</code>	Spaces to the end of recorded media on the tape. This is useful for appending files onto previously written tapes.
<code>erase</code>	Erases the entire tape. Some tape drives have option settings where only portions of the tape can be erased. Be sure to select the correct setting to erase the whole tape. Erasing a tape can take a long time depending on the device and/or tape. Refer to the device specific manual for time details.
<code>forcereserve</code>	Attempts to break a SCSI II reserve issued by another initiator. When this command completes, the drive is not reserved for the current initiator, but is available for use. This command can be only be executed by those with super-user privileges.
<code>offline</code> <code>rewoffl</code>	Rewinds the tape and, if appropriate, takes the drive unit off-line by unloading the tape.
<code>release</code>	Re-establishes the default behavior of releasing at close.

reserve	Allows the tape drive to remain reserved after closing the device. The drive must then be explicitly released.
retension	Rewinds the cartridge tape completely, then winds it forward to the end of the reel and back to beginning-of-tape to smooth out tape tension.
rewind	Rewinds the tape.
status	Prints status information about the tape unit.

Status information can include the sense key reported by the drive, the residual and retries for the last operation, the current tape position reported in file number, and the number of blocks from the beginning of that file. It might also report that WORM media is loaded in that drive.

退出状态	0	All operations were successful.
	1	Command was unrecognized or mt was unable to open the specified tape drive.
	2	An operation failed.

文件 /dev/rmt/* magnetic tape interface

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	SUNWcsu

另请参见 [tar\(1\)](#), [tcopy\(1\)](#), [ar.h\(3HEAD\)](#), [attributes\(5\)](#), [mtio\(7I\)](#), [st\(7D\)](#)

已知问题 Not all devices support all options. Some options are hardware-dependent. Refer to the corresponding device manual page.

mt is architecture sensitive. Heterogeneous operation (that is, SPARC to x86 or the reverse) is not supported.

引用名

mv – move files

用法概要

```
/usr/bin/mv [-fi] source target_file
```

```
/usr/bin/mv [-fi] source... target_dir
```

```
/usr/xpg4/bin/mv [-fi] source target_file
```

```
/usr/xpg4/bin/mv [-fi] source... target_dir
```

描述

In the first synopsis form, the mv utility moves the file named by the *source* operand to the destination specified by the *target_file*. *source* and *target_file* can not have the same name. If *target_file* does not exist, mv creates a file named *target_file*. If *target_file* exists, its contents are overwritten. This first synopsis form is assumed when the final operand does not name an existing directory.

In the second synopsis form, mv moves each file named by a *source* operand to a destination file in the existing directory named by the *target_dir* operand. The destination path for each *source* is the concatenation of the target directory, a single slash character (/), and the last path name component of the *source*. This second form is assumed when the final operand names an existing directory.

If mv determines that the mode of *target_file* forbids writing, it prints the mode (see [chmod\(2\)](#)), ask for a response, and read the standard input for one line. If the response is affirmative, the mv occurs, if permissible; otherwise, the command exits. Notice that the mode displayed can not fully represent the access permission if *target* is associated with an ACL. When the parent directory of *source* is writable and has the sticky bit set, one or more of the following conditions must be true:

- the user must own the file
- the user must own the directory
- the file must be writable by the user
- the user must be a privileged user

If *source* is a file and *target_file* is a link to another file with links, the other links remain and *target_file* becomes a new file.

If *source* and *target_file/target_dir* are on different file systems, mv copies the source and deletes the original. Any hard links to other files are lost. mv attempts to duplicate the source file characteristics to the target, that is, the owner and group id, permission modes, modification and access times, ACLs, and extended attributes, if applicable. For symbolic links, mv preserves only the owner and group of the link itself.

If unable to preserve owner and group id, mv clears S_ISUID and S_ISGID bits in the target. mv prints a diagnostic message to stderr if unable to clear these bits, though the exit code is not affected. mv might be unable to preserve extended attributes if the target file system does not have extended attribute support. /usr/xpg4/bin/mv prints a diagnostic message to stderr for all other failed attempts to duplicate file characteristics. The exit code is not affected.

In order to preserve the source file characteristics, users must have the appropriate file access permissions. This includes being super-user or having the same owner id as the destination file.

选项

The following options are supported:

- f `mv` moves the file(s) without prompting even if it is writing over an existing *target*. Note that this is the default if the standard input is not a terminal.
- i `mv` prompts for confirmation whenever the move would overwrite an existing target. This is done regardless of whether the input is coming from a terminal. If the prompt for confirmation fails, this is equivalent to the user answering in the negative. An affirmative answer means that the move should proceed. Any other answer prevents `mv` from overwriting *target*.

`/usr/bin/mv`

Specifying both the -f and the -i options is not considered an error. The -f option overrides the -i option.

`/usr/xpg4/bin/mv`

Specifying both the -f and the -i options is not considered an error. The last option specified determines the behavior of `mv`.

操作数

The following operands are supported:

- source* A path name of a file or directory to be moved.
- target_file* A new path name for the file or directory being moved.
- target_dir* A path name of an existing directory into which to move the input files.

用法

See [largefile\(5\)](#) for the description of the behavior of `mv` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `mv`: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

Affirmative responses are processed using the extended regular expression defined for the `yesexpr` keyword in the LC_MESSAGES category of the user's locale. The locale specified in the LC_COLLATE category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for `yesexpr`. The locale specified in LC_CTYPE determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the `yesexpr`. See [locale\(5\)](#).

退出状态

The following exit values are returned:

- 0 All input files were moved successfully.
- >0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/mv

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed

/usr/xpg4/bin/mv

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Standard

另请参见 [cp\(1\)](#), [cpio\(1\)](#), [ln\(1\)](#), [rm\(1\)](#), [setfacl\(1\)](#), [chmod\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

附注 A - - permits the user to mark explicitly the end of any command line options, allowing mv to recognize filename arguments that begin with a -. As an aid to BSD migration, mv accepts - as a synonym for - -. This migration aid might disappear in a future release.

引用名	nawk – pattern scanning and processing language
用法概要	<pre> /usr/bin/nawk [-F ERE] [-v assignment] 'program' -f progfile... [argument]... /usr/xpg4/bin/awk [-F ERE] [-v assignment]... 'program' -f progfile... [argument]... </pre>
描述	<p>The <code>/usr/bin/nawk</code> and <code>/usr/xpg4/bin/awk</code> utilities execute <i>programs</i> written in the <code>nawk</code> programming language, which is specialized for textual data manipulation. A <code>nawk program</code> is a sequence of patterns and corresponding actions. The string specifying <i>program</i> must be enclosed in single quotes (<code>'</code>) to protect it from interpretation by the shell. The sequence of pattern - action statements can be specified in the command line as <i>program</i> or in one, or more, file(s) specified by the <code>-f progfile</code> option. When input is read that matches a pattern, the action associated with the pattern is performed.</p> <p>Input is interpreted as a sequence of records. By default, a record is a line, but this can be changed by using the <code>RS</code> built-in variable. Each record of input is matched to each pattern in the <i>program</i>. For each pattern matched, the associated action is executed.</p> <p>The <code>nawk</code> utility interprets each input record as a sequence of fields where, by default, a field is a string of non-blank characters. This default white-space field delimiter (blanks and/or tabs) can be changed by using the <code>FS</code> built-in variable or the <code>-F ERE</code> option. The <code>nawk</code> utility denotes the first field in a record <code>\$1</code>, the second <code>\$2</code>, and so forth. The symbol <code>\$0</code> refers to the entire record; setting any other field causes the reevaluation of <code>\$0</code>. Assigning to <code>\$0</code> resets the values of all fields and the <code>NF</code> built-in variable.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> <code>-F ERE</code> Define the input field separator to be the extended regular expression <i>ERE</i>, before any input is read (can be a character). <code>-f progfile</code> Specifies the pathname of the file <i>progfile</i> containing a <code>nawk</code> program. If multiple instances of this option are specified, the concatenation of the files specified as <i>progfile</i> in the order specified is the <code>nawk</code> program. The <code>nawk</code> program can alternatively be specified in the command line as a single argument. <code>-v assignment</code> The <i>assignment</i> argument must be in the same form as an <i>assignment</i> operand. The assignment is of the form <i>var=value</i>, where <i>var</i> is the name of one of the variables described below. The specified assignment occurs before executing the <code>nawk</code> program, including the actions associated with <code>BEGIN</code> patterns (if any). Multiple occurrences of this option can be specified.
操作数	<p>The following operands are supported:</p>

- program* If no `-f` option is specified, the first operand to `nawk` is the text of the `nawk` program. The application supplies the *program* operand as a single argument to `nawk`. If the text does not end in a newline character, `nawk` interprets the text as if it did.
- argument* Either of the following two types of *argument* can be intermixed:
- file* A pathname of a file that contains the input to be read, which is matched against the set of patterns in the program. If no *file* operands are specified, or if a *file* operand is `-`, the standard input is used.
- assignment* An operand that begins with an underscore or alphabetic character from the portable character set, followed by a sequence of underscores, digits and alphabetic characters from the portable character set, followed by the `=` character specifies a variable assignment rather than a pathname. The characters before the `=` represent the name of a `nawk` variable. If that name is a `nawk` reserved word, the behavior is undefined. The characters following the equal sign is interpreted as if they appeared in the `nawk` program preceded and followed by a double-quote (`"`) character, as a `STRING` token, except that if the last character is an unescaped backslash, it is interpreted as a literal backslash rather than as the first character of the sequence `\.` The variable is assigned the value of that `STRING` token. If the value is considered a *numericstring*, the variable is assigned its numeric value. Each such variable assignment is performed just before the processing of the following *file*, if any. Thus, an assignment before the first *file* argument is executed after the `BEGIN` actions (if any), while an assignment after the last *file* argument is executed before the `END` actions (if any). If there are no *file* arguments, assignments are executed before processing the standard input.

Input Files

Input files to the `nawk` program from any of the following sources:

- any *file* operands or their equivalents, achieved by modifying the `nawk` variables `ARGV` and `ARGC`
- standard input in the absence of any *file* operands
- arguments to the `getline` function

must be text files. Whether the variable `RS` is set to a value other than a newline character or not, for these files, implementations support records terminated with the specified separator up to `{LINE_MAX}` bytes and can support longer records.

If `-f progfile` is specified, the files named by each of the *progfile* option-arguments must be text files containing an nawk program.

The standard input are used only if no *file* operands are specified, or if a *file* operand is `-`.

扩展描述

A nawk program is composed of pairs of the form:

```
pattern { action }
```

Either the pattern or the action (including the enclosing brace characters) can be omitted. Pattern-action statements are separated by a semicolon or by a newline.

A missing pattern matches any record of input, and a missing action is equivalent to an action that writes the matched record of input to standard output.

Execution of the nawk program starts by first executing the actions associated with all BEGIN patterns in the order they occur in the program. Then each *file* operand (or standard input if no files were specified) is processed by reading data from the file until a record separator is seen (a newline character by default), splitting the current record into fields using the current value of FS, evaluating each pattern in the program in the order of occurrence, and executing the action associated with each pattern that matches the current record. The action for a matching pattern is executed before evaluating subsequent patterns. Last, the actions associated with all END patterns is executed in the order they occur in the program.

Expressions in nawk

Expressions describe computations used in *patterns* and *actions*. In the following table, valid expression operations are given in groups from highest precedence first to lowest precedence last, with equal-precedence operators grouped between horizontal lines. In expression evaluation, where the grammar is formally ambiguous, higher precedence operators are evaluated before lower precedence operators. In this table *expr*, *expr1*, *expr2*, and *expr3* represent any expression, while *lvalue* represents any entity that can be assigned to (that is, on the left side of an assignment operator).

Syntax	Name	Type of Result	Associativity
<code>(expr)</code>	Grouping	type of <i>expr</i>	n/a
<code>\$expr</code>	Field reference	string	n/a
<code>++ lvalue</code>	Pre-increment	numeric	n/a
<code>-- lvalue</code>	Pre-decrement	numeric	n/a
<code>lvalue ++</code>	Post-increment	numeric	n/a
<code>lvalue --</code>	Post-decrement	numeric	n/a
<code>expr ^ expr</code>	Exponentiation	numeric	right
<code>! expr</code>	Logical not	numeric	n/a

Syntax	Name	Type of Result	Associativity
$+ \text{expr}$	Unary plus	numeric	n/a
$- \text{expr}$	Unary minus	numeric	n/a
$\text{expr} * \text{expr}$	Multiplication	numeric	left
$\text{expr} / \text{expr}$	Division	numeric	left
$\text{expr} \% \text{expr}$	Modulus	numeric	left
$\text{expr} + \text{expr}$	Addition	numeric	left
$\text{expr} - \text{expr}$	Subtraction	numeric	left
$\text{expr} \text{expr}$	String concatenation	string	left
$\text{expr} < \text{expr}$	Less than	numeric	none
$\text{expr} <= \text{expr}$	Less than or equal to	numeric	none
$\text{expr} != \text{expr}$	Not equal to	numeric	none
$\text{expr} == \text{expr}$	Equal to	numeric	none
$\text{expr} > \text{expr}$	Greater than	numeric	none
$\text{expr} >= \text{expr}$	Greater than or equal to	numeric	none
$\text{expr} \sim \text{expr}$	ERE match	numeric	none
$\text{expr} !\sim \text{expr}$	ERE non-match	numeric	none
$\text{expr} \text{ in array}$	Array membership	numeric	left
$(\text{index}) \text{ in array}$	Multi-dimension array membership	numeric	left
$\text{expr} \&\& \text{expr}$	Logical AND	numeric	left
$\text{expr} \text{expr}$	Logical OR	numeric	left
$\text{expr}1 ? \text{expr}2 : \text{expr}3$	Conditional expression	type of selected <i>expr2</i> or <i>expr3</i>	right
$\text{lvalue} \wedge= \text{expr}$	Exponentiation assignment	numeric	right
$\text{lvalue} \% = \text{expr}$	Modulus assignment	numeric	right
$\text{lvalue} * = \text{expr}$	Multiplication assignment	numeric	right

Syntax	Name	Type of Result	Associativity
<i>lvalue</i> /= <i>expr</i>	Division assignment	numeric	right
<i>lvalue</i> += <i>expr</i>	Addition assignment	numeric	right
<i>lvalue</i> -= <i>expr</i>	Subtraction assignment	numeric	right
<i>lvalue</i> = <i>expr</i>	Assignment	type of <i>expr</i>	right

Each expression has either a string value, a numeric value or both. Except as stated for specific contexts, the value of an expression is implicitly converted to the type needed for the context in which it is used. A string value is converted to a numeric value by the equivalent of the following calls:

```
setlocale(LC_NUMERIC, "");
numeric_value = atof(string_value);
```

A numeric value that is exactly equal to the value of an integer is converted to a string by the equivalent of a call to the `sprintf` function with the string `%d` as the `fmt` argument and the numeric value being converted as the first and only *expr* argument. Any other numeric value is converted to a string by the equivalent of a call to the `sprintf` function with the value of the variable `CONVFMT` as the `fmt` argument and the numeric value being converted as the first and only *expr* argument.

A string value is considered to be a *numeric string* in the following case:

1. Any leading and trailing blank characters is ignored.
2. If the first unignored character is a + or -, it is ignored.
3. If the remaining unignored characters would be lexically recognized as a `NUMBER` token, the string is considered a *numeric string*.

If a - character is ignored in the above steps, the numeric value of the *numeric string* is the negation of the numeric value of the recognized `NUMBER` token. Otherwise the numeric value of the *numeric string* is the numeric value of the recognized `NUMBER` token. Whether or not a string is a *numeric string* is relevant only in contexts where that term is used in this section.

When an expression is used in a Boolean context, if it has a numeric value, a value of zero is treated as false and any other value is treated as true. Otherwise, a string value of the null string is treated as false and any other value is treated as true. A Boolean context is one of the following:

- the first subexpression of a conditional expression.
- an expression operated on by logical NOT, logical AND, or logical OR.
- the second expression of a `for` statement.
- the expression of an `if` statement.
- the expression of the `while` clause in either a `while` or `do . . . while` statement.

- an expression used as a pattern (as in Overall Program Structure).

The `nawk` language supplies arrays that are used for storing numbers or strings. Arrays need not be declared. They are initially empty, and their sizes changes dynamically. The subscripts, or element identifiers, are strings, providing a type of associative array capability. An array name followed by a subscript within square brackets can be used as an *lvalue* and as an expression, as described in the grammar. Unsubscripted array names are used in only the following contexts:

- a parameter in a function definition or function call.
- the `NAME` token following any use of the keyword `in`.

A valid array *index* consists of one or more comma-separated expressions, similar to the way in which multi-dimensional arrays are indexed in some programming languages. Because `nawk` arrays are really one-dimensional, such a comma-separated list is converted to a single string by concatenating the string values of the separate expressions, each separated from the other by the value of the `SUBSEP` variable.

Thus, the following two index operations are equivalent:

```
var[expr1, expr2, ... exprn]
var[expr1 SUBSEP expr2 SUBSEP ... SUBSEP exprn]
```

A multi-dimensional *index* used with the `in` operator must be put in parentheses. The `in` operator, which tests for the existence of a particular array element, does not create the element if it does not exist. Any other reference to a non-existent array element automatically creates it.

Variables and Special Variables

Variables can be used in an `nawk` program by referencing them. With the exception of function parameters, they are not explicitly declared. Uninitialized scalar variables and array elements have both a numeric value of zero and a string value of the empty string.

Field variables are designated by a `$` followed by a number or numerical expression. The effect of the field number *expression* evaluating to anything other than a non-negative integer is unspecified. Uninitialized variables or string values need not be converted to numeric values in this context. New field variables are created by assigning a value to them. References to non-existent fields (that is, fields after `$NF`) produce the null string. However, assigning to a non-existent field (for example, `$(NF+2) = 5`) increases the value of `NF`, create any intervening fields with the null string as their values and cause the value of `$0` to be recomputed, with the fields being separated by the value of `OFS`. Each field variable has a string value when created. If the string, with any occurrence of the decimal-point character from the current locale changed to a period character, is considered a *numeric string* (see Expressions in `nawk` above), the field variable also has the numeric value of the *numeric string*.

`/usr/bin/nawk,`
`/usr/xpg4/bin/awk`

`nawk` sets the following special variables that are supported by both `/usr/bin/nawk` and `/usr/xpg4/bin/awk`:

`ARGC` The number of elements in the `ARGV` array.

ARGV	<p>An array of command line arguments, excluding options and the <i>program</i> argument, numbered from zero to ARGV-1.</p> <p>The arguments in ARGV can be modified or added to; ARGV can be altered. As each input file ends, nawk treats the next non-null element of ARGV, up to the current value of ARGV-1, inclusive, as the name of the next input file. Setting an element of ARGV to null means that it is not treated as an input file. The name - indicates the standard input. If an argument matches the format of an <i>assignment</i> operand, this argument is treated as an assignment rather than a <i>file</i> argument.</p>
ENVIRON	<p>The variable ENVIRON is an array representing the value of the environment. The indices of the array are strings consisting of the names of the environment variables, and the value of each array element is a string consisting of the value of that variable. If the value of an environment variable is considered a <i>numeric string</i>, the array element also has its numeric value.</p> <p>In all cases where nawk behavior is affected by environment variables (including the environment of any commands that nawk executes via the <code>system</code> function or via pipeline redirections with the <code>print</code> statement, the <code>printf</code> statement, or the <code>getline</code> function), the environment used is the environment at the time nawk began executing.</p>
FILENAME	<p>A pathname of the current input file. Inside a BEGIN action the value is undefined. Inside an END action the value is the name of the last input file processed.</p>
FNR	<p>The ordinal number of the current record in the current file. Inside a BEGIN action the value is zero. Inside an END action the value is the number of the last record processed in the last file processed.</p>
FS	<p>Input field separator regular expression; a space character by default.</p>
NF	<p>The number of fields in the current record. Inside a BEGIN action, the use of NF is undefined unless a <code>getline</code> function without a <i>var</i> argument is executed previously. Inside an END action, NF retains the value it had for the last record read, unless a subsequent, redirected, <code>getline</code> function without a <i>var</i> argument is performed prior to entering the END action.</p>
NR	<p>The ordinal number of the current record from the start of input. Inside a BEGIN action the value is zero. Inside an END action the value is the number of the last record processed.</p>
OFMT	<p>The <code>printf</code> format for converting numbers to strings in output statements " <code>%.6g</code>" by default. The result of the conversion is unspecified if the value of OFMT is not a floating-point format specification.</p>
OFS	<p>The <code>print</code> statement output field separator; a space character by default.</p>

ORS	The print output record separator; a newline character by default.
LENGTH	The length of the string matched by the <code>match</code> function.
RS	The first character of the string value of RS is the input record separator; a newline character by default. If RS contains more than one character, the results are unspecified. If RS is null, then records are separated by sequences of one or more blank lines. Leading or trailing blank lines do not produce empty records at the beginning or end of input, and the field separator is always newline, no matter what the value of FS.
RSTART	The starting position of the string matched by the <code>match</code> function, numbering from 1. This is always equivalent to the return value of the <code>match</code> function.
SUBSEP	The subscript separator string for multi-dimensional arrays. The default value is <code>\034</code> .

`/usr/xpg4/bin/awk`

The following variable is supported for `/usr/xpg4/bin/awk` only:

CONVFMT	The <code>printf</code> format for converting numbers to strings (except for output statements, where OFMT is used). The default is <code>%.6g</code> .
---------	---

Regular Expressions

The `/usr/xpg4/bin/nawk` utility makes use of the extended regular expression notation (see [regex\(5\)](#)) except that it allows the use of C-language conventions to escape special characters within the EREs, namely `\\`, `\a`, `\b`, `\f`, `\n`, `\r`, `\t`, `\v`, and those specified in the following table. These escape sequences are recognized both inside and outside bracket expressions. Records need not be separated by newline characters and string constants can contain newline characters, so even the `\n` sequence is valid in `nawk` EREs. Using a slash character within the regular expression requires escaping as shown in the table below:

Escape Sequence	Description	Meaning
<code>\"</code>	Backslash quotation-mark	Quotation-mark character
<code>\/</code>	Backslash slash	Slash character
<code>\ddd</code>	A backslash character followed by the longest sequence of one, two, or three octal-digit characters (01234567). If all of the digits are 0, (that is, representation of the NULL character), the behavior is undefined.	The character encoded by the one-, two- or three-digit octal integer. Multi-byte characters require multiple, concatenated escape sequences, including the leading <code>\</code> for each byte.
<code>\c</code>	A backslash character followed by any character not described in this table or special characters (<code>\\</code> , <code>\a</code> , <code>\b</code> , <code>\f</code> , <code>\n</code> , <code>\r</code> , <code>\t</code> , <code>\v</code>).	Undefined

A regular expression can be matched against a specific field or string by using one of the two regular expression matching operators, `~` and `! ~`. These operators interpret their right-hand operand as a regular expression and their left-hand operand as a string. If the regular expression matches the string, the `~` expression evaluates to the value 1, and the `! ~` expression evaluates to the value 0. If the regular expression does not match the string, the `~` expression evaluates to the value 0, and the `! ~` expression evaluates to the value 1. If the right-hand operand is any expression other than the lexical token `ERE`, the string value of the expression is interpreted as an extended regular expression, including the escape conventions described above. Notice that these same escape conventions also are applied in the determining the value of a string literal (the lexical token `STRING`), and is applied a second time when a string literal is used in this context.

When an `ERE` token appears as an expression in any context other than as the right-hand of the `~` or `! ~` operator or as one of the built-in function arguments described below, the value of the resulting expression is the equivalent of:

```
$0 ~ /ere/
```

The `ere` argument to the `gsub`, `match`, `sub` functions, and the `fs` argument to the `split` function (see `String Functions`) is interpreted as extended regular expressions. These can be either `ERE` tokens or arbitrary expressions, and are interpreted in the same manner as the right-hand side of the `~` or `! ~` operator.

An extended regular expression can be used to separate fields by using the `-F ERE` option or by assigning a string containing the expression to the built-in variable `FS`. The default value of the `FS` variable is a single space character. The following describes `FS` behavior:

1. If `FS` is a single character:
 - If `FS` is the space character, skip leading and trailing blank characters; fields are delimited by sets of one or more blank characters.
 - Otherwise, if `FS` is any other character `c`, fields are delimited by each single occurrence of `c`.
2. Otherwise, the string value of `FS` is considered to be an extended regular expression. Each occurrence of a sequence matching the extended regular expression delimits fields.

Except in the `gsub`, `match`, `split`, and `sub` built-in functions, regular expression matching is based on input records. That is, record separator characters (the first character of the value of the variable `RS`, a newline character by default) cannot be embedded in the expression, and no expression matches the record separator character. If the record separator is not a newline character, newline characters embedded in the expression can be matched. In those four built-in functions, regular expression matching are based on text strings. So, any character (including the newline character and the record separator) can be embedded in the pattern and an appropriate pattern matches any character. However, in all `nawk` regular expression matching, the use of one or more `NULL` characters in the pattern, input record or text string produces undefined results.

Patterns	<p>A <i>pattern</i> is any valid <i>expression</i>, a range specified by two expressions separated by comma, or one of the two special patterns BEGIN or END.</p>
Special Patterns	<p>The nawk utility recognizes two special patterns, BEGIN and END. Each BEGIN pattern is matched once and its associated action executed before the first record of input is read (except possibly by use of the <code>getline</code> function in a prior BEGIN action) and before command line assignment is done. Each END pattern is matched once and its associated action executed after the last record of input has been read. These two patterns have associated actions.</p> <p>BEGIN and END do not combine with other patterns. Multiple BEGIN and END patterns are allowed. The actions associated with the BEGIN patterns are executed in the order specified in the program, as are the END actions. An END pattern can precede a BEGIN pattern in a program.</p> <p>If an nawk program consists of only actions with the pattern BEGIN, and the BEGIN action contains no <code>getline</code> function, nawk exits without reading its input when the last statement in the last BEGIN action is executed. If an nawk program consists of only actions with the pattern END or only actions with the patterns BEGIN and END, the input is read before the statements in the END actions are executed.</p>
Expression Patterns	<p>An expression pattern is evaluated as if it were an expression in a Boolean context. If the result is true, the pattern is considered to match, and the associated action (if any) is executed. If the result is false, the action is not executed.</p>
Pattern Ranges	<p>A pattern range consists of two expressions separated by a comma. In this case, the action is performed for all records between a match of the first expression and the following match of the second expression, inclusive. At this point, the pattern range can be repeated starting at input records subsequent to the end of the matched range.</p>
Actions	<p>An action is a sequence of statements. A statement can be one of the following:</p> <pre>if (expression) statement [else statement] while (expression) statement do statement while (expression) for (expression ; expression ; expression) statement for (var in array) statement delete array[subscript] #delete an array element break continue { [statement] . . . } expression # commonly variable = expression print [expression-list] [>expression] printf format [,expression-list] [>expression] next # skip remaining patterns on this input line exit [expr] # skip the rest of the input; exit status is expr return [expr]</pre> <p>Any single statement can be replaced by a statement list enclosed in braces. The statements are terminated by newline characters or semicolons, and are executed sequentially in the order that they appear.</p>

The `next` statement causes all further processing of the current input record to be abandoned. The behavior is undefined if a `next` statement appears or is invoked in a `BEGIN` or `END` action.

The `exit` statement invokes all `END` actions in the order in which they occur in the program source and then terminate the program without reading further input. An `exit` statement inside an `END` action terminates the program without further execution of `END` actions. If an expression is specified in an `exit` statement, its numeric value is the exit status of `nawk`, unless subsequent errors are encountered or a subsequent `exit` statement with an expression is executed.

Output Statements

Both `print` and `printf` statements write to standard output by default. The output is written to the location specified by `output_redirection` if one is supplied, as follows:

```
> expression>> expression | expression
```

In all cases, the *expression* is evaluated to produce a string that is used as a full pathname to write into (for `>` or `>>`) or as a command to be executed (for `|`). Using the first two forms, if the file of that name is not currently open, it is opened, creating it if necessary and using the first form, truncating the file. The output then is appended to the file. As long as the file remains open, subsequent calls in which *expression* evaluates to the same string value simply appends output to the file. The file remains open until the `close` function, which is called with an expression that evaluates to the same string value.

The third form writes output onto a stream piped to the input of a command. The stream is created if no stream is currently open with the value of *expression* as its command name. The stream created is equivalent to one created by a call to the `popen(3C)` function with the value of *expression* as the *command* argument and a value of `w` as the *mode* argument. As long as the stream remains open, subsequent calls in which *expression* evaluates to the same string value writes output to the existing stream. The stream remains open until the `close` function is called with an expression that evaluates to the same string value. At that time, the stream is closed as if by a call to the `pclose` function.

These output statements take a comma-separated list of *expression s* referred in the grammar by the non-terminal symbols `expr_list`, `print_expr_list` or `print_expr_list_opt`. This list is referred to here as the *expression list*, and each member is referred to as an *expression argument*.

The `print` statement writes the value of each expression argument onto the indicated output stream separated by the current output field separator (see variable `OFS` above), and terminated by the output record separator (see variable `ORS` above). All expression arguments is taken as strings, being converted if necessary; with the exception that the `printf` format in `OFMT` is used instead of the value in `CONVFMT`. An empty expression list stands for the whole input record (`$0`).

The `printf` statement produces output based on a notation similar to the File Format Notation used to describe file formats in this document Output is produced as specified with the first expression argument as the string `format` and subsequent expression arguments as the strings `arg1` to `argn`, inclusive, with the following exceptions:

1. The *format* is an actual character string rather than a graphical representation. Therefore, it cannot contain empty character positions. The space character in the *format* string, in any context other than a *flag* of a conversion specification, is treated as an ordinary character that is copied to the output.
2. If the character set contains a Delta character and that character appears in the *format* string, it is treated as an ordinary character that is copied to the output.
3. The *escape sequences* beginning with a backslash character is treated as sequences of ordinary characters that are copied to the output. Note that these same sequences is interpreted lexically by `nawk` when they appear in literal strings, but they is not treated specially by the `printf` statement.
4. A *field width* or *precision* can be specified as the `*` character instead of a digit string. In this case the next argument from the expression list is fetched and its numeric value taken as the field width or precision.
5. The implementation does not precede or follow output from the `d` or `u` conversion specifications with blank characters not specified by the *format* string.
6. The implementation does not precede output from the `o` conversion specification with leading zeros not specified by the *format* string.
7. For the `c` conversion specification: if the argument has a numeric value, the character whose encoding is that value is output. If the value is zero or is not the encoding of any character in the character set, the behavior is undefined. If the argument does not have a numeric value, the first character of the string value is output; if the string does not contain any characters the behavior is undefined.
8. For each conversion specification that consumes an argument, the next expression argument is evaluated. With the exception of the `c` conversion, the value is converted to the appropriate type for the conversion specification.
9. If there are insufficient expression arguments to satisfy all the conversion specifications in the *format* string, the behavior is undefined.
10. If any character sequence in the *format* string begins with a `%` character, but does not form a valid conversion specification, the behavior is unspecified.

Both `print` and `printf` can output at least `{LINE_MAX}` bytes.

Functions The `nawk` language has a variety of built-in functions: arithmetic, string, input/output and general.

Arithmetic Functions The arithmetic functions, except for `int`, are based on the ISO C standard. The behavior is undefined in cases where the ISO C standard specifies that an error be returned or that the behavior is undefined. Although the grammar permits built-in functions to appear with no arguments or parentheses, unless the argument or parentheses are indicated as optional in the following list (by displaying them within the `[]` brackets), such use is undefined.

`atan2(y,x)` Return arctangent of y/x .

<code>cos(x)</code>	Return cosine of x , where x is in radians.
<code>sin(x)</code>	Return sine of x , where x is in radians.
<code>exp(x)</code>	Return the exponential function of x .
<code>log(x)</code>	Return the natural logarithm of x .
<code>sqrt(x)</code>	Return the square root of x .
<code>int(x)</code>	Truncate its argument to an integer. It is truncated toward 0 when $x > 0$.
<code>rand()</code>	Return a random number n , such that $0 \leq n < 1$.
<code>srand([expr])</code>	Set the seed value for <code>rand</code> to <i>expr</i> or use the time of day if <i>expr</i> is omitted. The previous seed value is returned.

String Functions

The string functions in the following list shall be supported. Although the grammar permits built-in functions to appear with no arguments or parentheses, unless the argument or parentheses are indicated as optional in the following list (by displaying them within the [] brackets), such use is undefined.

<code>gsub(ere,repl[, in])</code>	Behave like <code>sub</code> (see below), except that it replaces all occurrences of the regular expression (like the <code>ed</code> utility global substitute) in <code>\$0</code> or in the <i>in</i> argument, when specified.
<code>index(s,t)</code>	Return the position, in characters, numbering from 1, in string <i>s</i> where string <i>t</i> first occurs, or zero if it does not occur at all.
<code>length([s])</code>	Return the length, in characters, of its argument taken as a string, or of the whole record, <code>\$0</code> , if there is no argument.
<code>match(s,ere)</code>	Return the position, in characters, numbering from 1, in string <i>s</i> where the extended regular expression <i>ere</i> occurs, or zero if it does not occur at all. <code>RSTART</code> is set to the starting position (which is the same as the returned value), zero if no match is found; <code>RLENGTH</code> is set to the length of the matched string, <code>-1</code> if no match is found.
<code>split(s,a[,fs])</code>	Split the string <i>s</i> into array elements $a[1], a[2], \dots, a[n]$, and return n . The separation is done with the extended regular expression <i>fs</i> or with the field separator <code>FS</code> if <i>fs</i> is not given. Each array element has a string value when created. If the string assigned to any array element, with any occurrence of the decimal-point character from the current locale changed to a period character, would be considered a <i>numeric string</i> ; the array element also has the numeric value of the <i>numeric string</i> . The effect of a null string as the value of <i>fs</i> is unspecified.

<code>sprintf(fmt,expr,expr,...)</code>	Format the expressions according to the <code>printf</code> format given by <i>fmt</i> and return the resulting string.
<code>sub(ere,repl[, in])</code>	Substitute the string <i>repl</i> in place of the first instance of the extended regular expression <i>ERE</i> in string <i>in</i> and return the number of substitutions. An ampersand (&) appearing in the string <i>repl</i> is replaced by the string from <i>in</i> that matches the regular expression. An ampersand preceded with a backslash (\) is interpreted as the literal ampersand character. An occurrence of two consecutive backslashes is interpreted as just a single literal backslash character. Any other occurrence of a backslash (for example, preceding any other character) is treated as a literal backslash character. If <i>repl</i> is a string literal, the handling of the ampersand character occurs after any lexical processing, including any lexical backslash escape sequence processing. If <i>in</i> is specified and it is not an <code>lvalue</code> , the behavior is undefined. If <i>in</i> is omitted, <code>nawk</code> uses the current record (<code>\$0</code>) in its place.
<code>substr(s,m[, n])</code>	Return the at most <i>n</i> -character substring of <i>s</i> that begins at position <i>m</i> , numbering from 1. If <i>n</i> is missing, the length of the substring is limited by the length of the string <i>s</i> .
<code>tolower(s)</code>	Return a string based on the string <i>s</i> . Each character in <i>s</i> that is an upper-case letter specified to have a <code>tolower</code> mapping by the <code>LC_CTYPE</code> category of the current locale is replaced in the returned string by the lower-case letter specified by the mapping. Other characters in <i>s</i> are unchanged in the returned string.
<code>toupper(s)</code>	Return a string based on the string <i>s</i> . Each character in <i>s</i> that is a lower-case letter specified to have a <code>toupper</code> mapping by the <code>LC_CTYPE</code> category of the current locale is replaced in the returned string by the upper-case letter specified by the mapping. Other characters in <i>s</i> are unchanged in the returned string.

All of the preceding functions that take *ERE* as a parameter expect a pattern or a string valued expression that is a regular expression as defined below.

Input/Output and General Functions

The input/output and general functions are:

<code>close(expression)</code>	Close the file or pipe opened by a <code>print</code> or <code>printf</code> statement or a call to <code>getline</code> with the same string-valued <i>expression</i> . If the <code>close</code> was successful, the function returns <code>0</code> ; otherwise, it returns non-zero.
--------------------------------	--

expression|getLine[*var*] Read a record of input from a stream piped from the output of a command. The stream is created if no stream is currently open with the value of *expression* as its command name. The stream created is equivalent to one created by a call to the `popen` function with the value of *expression* as the *command* argument and a value of *r* as the *mode* argument. As long as the stream remains open, subsequent calls in which *expression* evaluates to the same string value reads subsequent records from the file. The stream remains open until the `close` function is called with an expression that evaluates to the same string value. At that time, the stream is closed as if by a call to the `pclose` function. If *var* is missing, `$0` and `NF` is set. Otherwise, *var* is set.

The `getLine` operator can form ambiguous constructs when there are operators that are not in parentheses (including concatenate) to the left of the `|` (to the beginning of the expression containing `getLine`). In the context of the `$` operator, `|` behaves as if it had a lower precedence than `$`. The result of evaluating other operators is unspecified, and all such uses of portable applications must be put in parentheses properly.

`getLine` Set `$0` to the next input record from the current input file. This form of `getLine` sets the `NF`, `NR`, and `FNR` variables.

`getLine var` Set variable *var* to the next input record from the current input file. This form of `getLine` sets the `FNR` and `NR` variables.

`getLine [var] < expression` Read the next record of input from a named file. The *expression* is evaluated to produce a string that is used as a full pathname. If the file of that name is not currently open, it is opened. As long as the stream remains open, subsequent calls in which *expression* evaluates to the same string value reads subsequent records from the file. The file remains open until the `close` function is called with an expression that evaluates to the same string value. If *var* is missing, `$0` and `NF` is set. Otherwise, *var* is set.

The `getLine` operator can form ambiguous constructs when there are binary operators that are not in parentheses (including concatenate) to the right of the `<` (up to the end of the expression containing the `getLine`). The result of evaluating such a construct is unspecified, and all such uses of portable applications must be put in parentheses properly.

`system(expression)` Execute the command given by *expression* in a manner equivalent to the `system(3C)` function and return the exit

status of the command.

All forms of `getline` return 1 for successful input, 0 for end of file, and -1 for an error.

Where strings are used as the name of a file or pipeline, the strings must be textually identical. The terminology “same string value” implies that “equivalent strings”, even those that differ only by space characters, represent different files.

User-defined Functions The nawk language also provides user-defined functions. Such functions can be defined as:

```
function name(args, . . .) { statements }
```

A function can be referred to anywhere in an nawk program; in particular, its use can precede its definition. The scope of a function is global.

Function arguments can be either scalars or arrays; the behavior is undefined if an array name is passed as an argument that the function uses as a scalar, or if a scalar expression is passed as an argument that the function uses as an array. Function arguments are passed by value if scalar and by reference if array name. Argument names are local to the function; all other variable names are global. The same name is not used as both an argument name and as the name of a function or a special nawk variable. The same name must not be used both as a variable name with global scope and as the name of a function. The same name must not be used within the same scope both as a scalar variable and as an array.

The number of parameters in the function definition need not match the number of parameters in the function call. Excess formal parameters can be used as local variables. If fewer arguments are supplied in a function call than are in the function definition, the extra parameters that are used in the function body as scalars are initialized with a string value of the null string and a numeric value of zero, and the extra parameters that are used in the function body as arrays are initialized as empty arrays. If more arguments are supplied in a function call than are in the function definition, the behavior is undefined.

When invoking a function, no white space can be placed between the function name and the opening parenthesis. Function calls can be nested and recursive calls can be made upon functions. Upon return from any nested or recursive function call, the values of all of the calling function's parameters are unchanged, except for array parameters passed by reference. The `return` statement can be used to return a value. If a `return` statement appears outside of a function definition, the behavior is undefined.

In the function definition, newline characters are optional before the opening brace and after the closing brace. Function definitions can appear anywhere in the program where a *pattern-action* pair is allowed.

用法

The `index`, `length`, `match`, and `substr` functions should not be confused with similar functions in the ISO C standard; the nawk versions deal with characters, while the ISO C standard deals with bytes.

Because the concatenation operation is represented by adjacent expressions rather than an explicit operator, it is often necessary to use parentheses to enforce the proper evaluation precedence.

See [largefile\(5\)](#) for the description of the behavior of `nawk` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

The `nawk` program specified in the command line is most easily specified within single-quotes (for example, `'program'`) for applications using `sh`, because `nawk` programs commonly contain characters that are special to the shell, including double-quotes. In the cases where a `nawk` program contains single-quote characters, it is usually easiest to specify most of the program as strings within single-quotes concatenated by the shell with quoted single-quote characters. For example:

```
nawk '/'\''/ { print "quote:", $0 }'
```

prints all lines from the standard input containing a single-quote character, prefixed with `quote:.`

The following are examples of simple `nawk` programs:

示例 1 Write to the standard output all input lines for which field 3 is greater than 5:

```
$3 > 5
```

示例 2 Write every tenth line:

```
(NR % 10) == 0
```

示例 3 Write any line with a substring matching the regular expression:

```
/(G|D)(2[0-9][[:alpha:]]*)/
```

示例 4 Print any line with a substring containing a G or D, followed by a sequence of digits and characters:

This example uses character classes `digit` and `alpha` to match language-independent digit and alphabetic characters, respectively.

```
/(G|D)([[:digit:]][[:alpha:]]*)/
```

示例 5 Write any line in which the second field matches the regular expression and the fourth field does not:

```
$2 ~ /xyz/ && $4 !~ /xyz/
```

示例 6 Write any line in which the second field contains a backslash:

```
$2 ~ /\
```

示例 7 Write any line in which the second field contains a backslash (alternate method):

Notice that backslash escapes are interpreted twice, once in lexical processing of the string and once in processing the regular expression.

```
$2 ~ "\\\""
```

示例 8 Write the second to the last and the last field in each line, separating the fields by a colon:

```
{OFS=":";print $(NF-1), $NF}
```

示例 9 Write the line number and number of fields in each line:

The three strings representing the line number, the colon and the number of fields are concatenated and that string is written to standard output.

```
{print NR ":" NF}
```

示例 10 Write lines longer than 72 characters:

```
{length($0) > 72}
```

示例 11 Write first two fields in opposite order separated by the OFS:

```
{ print $2, $1 }
```

示例 12 Same, with input fields separated by comma or space and tab characters, or both:

```
BEGIN { FS = ", [\t]*|[\t]+" }  
      { print $2, $1 }
```

示例 13 Add up first column, print sum and average:

```
{s += $1 }  
END {print "sum is ", s, " average is", s/NR}
```

示例 14 Write fields in reverse order, one per line (many lines out for each line in):

```
{ for (i = NF; i > 0; --i) print $i }
```

示例 15 Write all lines between occurrences of the strings “start” and “stop”:

```
/start/, /stop/
```

示例 16 Write all lines whose first field is different from the previous one:

```
$1 != prev { print; prev = $1 }
```

示例 17 Simulate the echo command:

```
BEGIN {  
      for (i = 1; i < ARGV; ++i)  
          printf "%s%s", ARGV[i], i==ARGV-1?"\n":""
```

示例 17 Simulate the echo command: (续)

```
}
```

示例 18 Write the path prefixes contained in the PATH environment variable, one per line:

```
BEGIN {
    n = split (ENVIRON["PATH"], path, ":")
    for (i = 1; i <= n; ++i)
        print path[i]
}
```

示例 19 Print the file “input”, filling in page numbers starting at 5:

If there is a file named input containing page headers of the form

Page#

and a file named program that contains

```
/Page/{ $2 = n++; }
{ print }
```

then the command line

```
nawk -f program n=5 input
```

prints the file input, filling in page numbers starting at 5.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect execution: LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

LC_NUMERIC Determine the radix character used when interpreting numeric input, performing conversions between numeric and string values and formatting numeric output. Regardless of locale, the period character (the decimal-point character of the POSIX locale) is the decimal-point character recognized in processing awk programs (including assignments in command-line arguments).

退出状态

The following exit values are returned:

0 All input files were processed successfully.

>0 An error occurred.

The exit status can be altered within the program by using an `exit` expression.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/nawk

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

/usr/xpg4/bin/awk

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4

另请参见

[awk\(1\)](#), [ed\(1\)](#), [egrep\(1\)](#), [grep\(1\)](#), [lex\(1\)](#), [sed\(1\)](#), [popen\(3C\)](#), [printf\(3C\)](#), [system\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#), [XPG4\(5\)](#)

Aho, A. V., B. W. Kernighan, and P. J. Weinberger, *The AWK Programming Language*, Addison-Wesley, 1988.

诊断

If any *file* operand is specified and the named file cannot be accessed, nawk writes a diagnostic message to standard error and terminate without any further action.

If the program specified by either the *program* operand or a *progfile* operand is not a valid nawk program (as specified in EXTENDED DESCRIPTION), the behavior is undefined.

附注

Input white space is not preserved on output if fields are involved.

There are no explicit conversions between numbers and strings. To force an expression to be treated as a number add 0 to it; to force it to be treated as a string concatenate the null string ("") to it.

引用名 nc, netcat – 任意 TCP 与 UDP 连接和侦听

用法概要

```
nc -h

nc [-46dnrtuvz] [-i interval] [-P proxy_username] [-p port]
  [-s source_ip_address] [-T dspc] [-w timeout]
  [-X proxy_protocol] [-x proxy_address[:port]][-L timeout]
  [-e program] [-b bufsize] [-q timeout] [-m bytes]
  [-I bufsize][-O bufsize] hostname port_list

nc -l [-46DdEFnrtuvzZ] [-i interval] [-T dspc] [-e program]
  [-b bufsize] [-q timeout] [-R address/port[/proto]] [-m bytes]
  [-L timeout] [-I bufsize] [-O bufsize] [hostname] port

nc -l [-46DEFdnrtuvzZ] [-i interval] [-T dspc] [-e program]
  [-b bufsize] [-q timeout] [-R address/port[/proto]] [-m bytes]
  [-L timeout] [-I bufsize] [-O bufsize]
  -p port [hostname]

nc -U [-Ddtvz] [-i interval] [-w timeout] [-e program]
  [-b bufsize] [-q timeout] [-m bytes] path

nc -Ul [-46DdktvZ] [-i interval] [-e program] [-b bufsize]
  [-q timeout] [-R address/port[/proto]] [-m bytes] path
```

描述 nc (或 netcat) 实用程序可用于与 TCP 或 UDP 相关的各种任务。nc 可以打开 TCP 连接, 发送 UDP 数据包, 侦听任意 TCP 和 UDP 端口, 执行端口扫描, 以及处理 IPv4 和 IPv6。与 [telnet\(1\)](#) 不同, nc 精细地编写脚本, 并将错误消息分隔到标准错误中, 而不是将错误消息发送到标准输出。

nc 命令通常用于以下任务:

- 简单 TCP 代理
- 基于 HTTP 客户机和服务器的 shell 脚本
- 网络守护进程测试
- 适用于 [ssh\(1\)](#) 的 SOCKS 或 HTTP ProxyCommand

nc 命令还可以作为 netcat 运行 (使用相同选项)。

选项 支持以下选项:

- 4
强制 nc 仅使用 IPv4 地址。
- 6
强制 nc 仅使用 IPv6 地址。
- b *bufsize*
为读取操作指定缓冲区大小。
缺省值为 1024 字节。

- D
启用对套接字的调试。
- d
不尝试从 `stdin` 进行读取。
- E
使用独占绑定来侦听 TCP 或 UDP 套接字。
在没有 `-l` 选项的情况下单独使用此选项是错误的。
此选项与 `-u` 选项结合使用时，不会产生任何影响。
- e *program*
接受连接或建立连接之后执行外部程序。在执行之前，`stdin`, `stdout`, `stderr` 会重定向到网络描述符。仅有一个端口可供该选项使用。
此选项与 `-R`、`-k` 或 `-i` 选项结合使用是错误的。
- F
在 `stdin` 上看到 EOF 后，不要关闭网络套接字以进行写入。
- h
输出 nc 帮助。
- I *bufsize*
设置接收（输入）套接字缓冲区大小。
此选项与 `-u` 选项结合使用时，不会产生任何影响。
- i *interval*
指定发送和接收的文本行之间的延迟时间 *interval*。
以秒为单位指定时间间隔，可能包含小数。
此选项还会导致与多个端口的连接之间产生延迟时间，因而也会影响端口扫描模式。
- k
强制 nc 在其当前连接关闭后侦听另一连接。
在没有 `-l` 选项的情况下单独使用此选项是错误的。
此选项与 `-e` 选项结合使用是错误的。
- L *timeout*
在关闭时逗留 (Linger on close) — 在网络描述符关闭后直到指定的超时时间（以秒为单位），等待消息发送。
- l
侦听传入连接，而不是启动到远程主机的连接。
此选项与 `-s` 或 `-z` 选项结合使用是错误的。

如果 `-l` 选项与通配符套接字（未指定任何 IP 地址或主机名）一起使用但不与 `-4` / `-6` 选项一起使用，则既可接受 IPv4 连接也可接受 IPv6 连接。

-m *byte_count*

接收至少 *byte_count* 字节后退出。当与 `-l` 选项结合使用时，*byte_count* 会与从客户机接收的字节数进行比较。

byte_count 必须大于 0，而小于 `INT_MAX`。

-N *file*

在 UDP 端口扫描模式下指定文件。此文件的内容用作每个发出的 UDP 包的有效载荷。

在没有 `-u` 和 `-z` 选项的情况下单独使用此选项是错误的。

-n

不对任何地址、主机名或端口执行任何命名或服务查找操作。

使用此选项意味着 *hostname* 和 *port* 参数被限制为数字值。

除了对参数施加限制外，与 `-v` 选项一起使用时，所有地址和端口都将以数字形式输出。此选项与 `-u` 选项结合使用时，不会产生任何影响。

-O *bufsize*

设置发送（输出）套接字缓冲区大小。

此选项与 `-u` 选项结合使用时，不会产生任何影响。

-P *proxy_username*

指定提供给要求验证的代理服务器的一个用户名 (*proxy_username*)。如果未指定 *proxy_username*，则不会尝试进行验证。目前仅 HTTP CONNECT 代理支持代理验证。

此选项与 `-l` 选项结合使用是错误的。

-p *port*

未与 `-l` 选项结合使用时，根据特权限制和可用性指定 nc 应使用的源端口。与 `-l` 选项结合使用时，设置侦听端口。

仅当未指定全局端口参数时，此选项可与 `-l` 选项结合使用。

-q *timeout*

在 `stdin` 上接收到 EOF 后，等待指定的秒数，然后退出。

-R *addr/port[/proto]*

对指定的 *host* 和 *port* 执行端口重定向。

接受连接后，nc 会连接到远程 *host/port*，并在客户机与远程主机之间传递所有数据。重定向规范的 *proto*（协议）部分可以是 `tcp` 或 `udp`。如果未指定 *proto*，`redirector` 将使用与服务器相同的协议。

此选项与 `-z` 选项结合使用是错误的。

- r
在由 *port_list* 参数指定的所有端口中随机（而非按顺序）选择目标端口。

此选项与 -l 选项结合使用是错误的。
- s *source_ip_address*
指定用于发送数据包的接口的 IP。

此选项与 -l 选项结合使用是错误的。
- T *dscp*
为连接指定区分服务代码点。

对于 IPv4，此选项指定 IP 服务类型 (Type of Service, ToS) IP 标题字段，参数的有效值为字符串标记 *lowdelay*、*throughput*、*reliability* 或前面带有 0x 的 8 位十六进制值。

对于 IPv6（通信流量类），只能使用十六进制值。
- t
使 nc 将 RFC 854 DON'T 和 WON'T 响应发送到 RFC 854 DO 及 WILL 请求。这样就可以使用 nc 编写 telnet 会话脚本。
- U
指定使用 Unix 域套接字。如果不与 -l、nc 一起指定此选项，则它将变成 AF_UNIX 客户机。如果与 -l 选项一起指定此选项，则会创建 AF_UNIX 服务器。

使用此选项要求必须向 nc 提供单个有效的 Unix 域路径参数，而不是提供主机名或端口。
- u
使用 UDP，而不是缺省选项 TCP。
- v
指定详细输出。
- w *timeout*
如果连接和 stdin 空闲超过了 *timeout* 秒，则无提示地关闭连接。

缺省设置是没有超时。

此选项对客户机模式下的连接建立阶段或服务器模式下的等待连接过程没有任何影响。
- X *proxy_protocol*
与代理服务器通信时，使用该指定协议。受支持的协议为 4 (SOCKS v.4)、5 (SOCKS v.5) 和 connect (HTTP 代理)。如果未指定协议，则使用 SOCKS v. 5。

此选项与 -l 选项结合使用是错误的。

`-x proxy_address[:port]`

使用 *proxy_address* 和 *port* 上的代理请求到 *hostname* 的连接。如果未指定 *port*，则使用代理协议的已知端口（SOCKS 为 1080，HTTP 为 3128）。

此选项与 `-l` 选项结合使用是错误的。

此选项不适用于 IPv6 地址的数字表示形式。

`-Z`

在侦听模式下，使用 `SO_ALLZONES` 套接字选项绑定到所有区域中的地址/端口。

此选项需要 `SYS_NET_CONFIG` 特权。

`-z`

执行端口扫描。对于 TCP 端口（缺省），尝试在不发送数据的情况下执行连接扫描（完整三路信号握手）。对于 UDP (`-u`)，缺省情况下会发送空 UDP 包。要指定 UDP 有效载荷，可以使用 `-N` 选项。

UDP 扫描模式具有估计能力，如果它没有接收到否定响应（"ICMP Destination Port Unreachable"（无法访问 ICMP 目标端口）消息），它会考虑打开一个端口。对于这种模式，使用 `-w` 选项设置的超时时间将用来等待来自远程节点的 ICMP 消息或数据。通过 `-v`，接收到的任何数据都会作为十六进制字节转储到 `stderr`。

由于大多数操作系统会限制发送 ICMP 消息（以响应输入包）的速率，所以有必要在执行 UDP 扫描时使用 `-i`，否则结果会不可靠。

此选项与 `-l` 选项结合使用是错误的。

操作数

支持下列操作数：

hostname 指定主机名。

hostname 可以是数字 IP 地址或者符号主机名（除非已指定 `-n` 选项）。

通常，除非已指定 `-l` 选项或者使用了 `-u`（在此情况下，参数是一个路径），否则必须指定 *hostname*。如果随 `-l` 选项指定了 *hostname* 参数，则还必须给定 *port* 参数，并且 `nc` 会尝试绑定到该地址和端口。如果没有随 `-l` 选项指定 *hostname* 参数，则 `nc` 会尝试在给定的 *port* 的通配符套接字上侦听。

path 指定路径名。

port

port_list 指定端口。

port_list 可以指定为单个整数、范围或两者的组合。请以 *nn-mm* 形式指定范围。*port_list* 至少必须有一个成员，但可以有多个以逗号分隔的端口/范围。

通常，除非已指定 `-u` 选项（在此情况下，必须指定 Unix 域套接字路径，而不指定 `hostname`），否则必须指定目标端口。

将包含多个端口的端口列表与 `-e` 选项结合使用是错误的。

用法

客户机/服务器模型 使用 `nc` 构建最基本的客户机/服务器模型非常简单。在一个控制台上，启动在特定端口上侦听连接的 `nc`。例如，命令：

```
$ nc -l 1234
```

在端口 1234 上侦听连接。在另一个控制台上（或另一台计算机上），连接到 `nc` 正在侦听的计算机和端口：

```
$ nc 127.0.0.1 1234
```

现在端口之间应当有一个连接。在第二个控制台上键入的任何内容都将串联到第一个控制台，反之亦然。在连接建立后，`nc` 不会真正关心哪一端用作**服务器**，哪一端用作**客户机**。可使用 EOF (Ctrl/d) 终止连接。

数据传输

可以对上一部分中的示例进行扩展，以构建基本的数据传送模型。在连接的一端输入的任何信息都将输出到连接的另一端，并且可以轻松捕获输入和输出，以便模仿文件传送。

通过使用 `nc` 启动在特定端口上的侦听，并将输出捕获到一个文件中：

```
$ nc -l 1234 > filename.out
```

使用另一台计算机，连接到正在侦听的 `nc` 进程，向其馈送要传送的文件：

```
$ nc host.example.com 1234 < filename.in
```

完成文件传送后，连接将自动关闭。

与服务器通信

有时，通过**手工**（而不是通过用户界面）与服务器进行通信非常有用。当可能需要验证服务器正在发送什么数据来响应客户机发出的命令时，它可以帮助排除故障。

例如，要检索某个 Web 站点的主页：

```
$ echo -n "GET / HTTP/1.0\r\n\r\n" | nc host.example.com 80
```

这也将显示 Web 服务器发送的标头。如果需要，可以使用 `sed(1)` 等工具过滤这些标头。

如果用户了解服务器要求的请求格式，可以构造更为复杂的示例。再如，可使用以下方法将电子邮件提交到 SMTP 服务器：

```
$ nc localhost 25 << EOF
HELO host.example.com
```

```
MAIL FROM: <user@host.example.com>
RCTP TO: <user2@host.example.com>
DATA
Body of email.
.
QUIT
EOF
```

端口扫描

知道目标计算机上哪些端口是打开的并正在运行服务可能非常有用。可以使用 `-z` 标志来指示 `nc` 报告打开的端口，而不是启动连接。

在此示例中：

```
$ nc -z host.example.com 20-30
Connection to host.example.com 22 port [tcp/ssh] succeeded!
Connection to host.example.com 25 port [tcp/smtp] succeeded!
```

指定了端口范围以将搜索限制在端口 20 至 30 之间。

此外，了解正在运行的服务器软件及版本可能非常有用。该信息通常包含在问候标题内。要检索这些问候标题，首先需要建立连接，然后检索标题，之后断开连接。此操作可通过使用 `-w` 标志指定较小的超时，或者通过向服务器发出 `QUIT` 命令来实现。

```
$ echo "QUIT" | nc host.example.com 20-30
SSH-2.0-Sun_SSH_1.1
Protocol mismatch.
220 host.example.com IMS SMTP Receiver Version 0.84 Ready
```

inetd 功能

可能的用途之一是使用 `inetd(1M)` 创建简单的服务。

以下示例创建了一个从主机 `realwww` 上的 TCP 端口 8080 到端口 80 的重定向：

```
# cat << EOF >> /etc/services
wwwredir 8080/tcp # WWW redirect
EOF
# cat << EOF > /tmp/wwwredir.conf
wwwredir stream tcp nowait nobody /usr/bin/nc /usr/bin/nc -w 3 realwww 80
EOF
# inetconv -i /tmp/wwwredir.conf
wwwredir -> /var/svc/manifest/network/wwwredir-tcp.xml
Importing wwwredir-tcp.xml ...Done
# inetadm -l wwwredir/tcp
SCOPE NAME=VALUE
name="wwwredir"
endpoint_type="stream"
proto="tcp"
isrpc=FALSE
wait=FALSE
exec="/usr/bin/nc -w 3 realwww 80"
```

```

arg0="/usr/bin/nc"
user="nobody"
default bind_addr=""
default bind_fail_max=-1
default bind_fail_interval=-1
default max_con_rate=-1
default max_copies=-1
default con_rate_offline=-1
default failrate_cnt=40
default failrate_interval=60
default inherit_env=TRUE
default tcp_trace=TRUE
default tcp_wrappers=FALSE

```

特权 要绑定到特权端口号，需要向 nc 授予 `net_privaddr` 特权。如果配置了 Solaris Trusted Extensions，并且 nc 应侦听的端口被配置为多级别端口，则 nc 还需要具有 `net_bindmlp` 特权。

通过在 `user_attr(4)` 中在帐户的缺省特权集中指定这些特权，可以直接将它们分配给用户或角色。但是，这意味着该用户或角色启动的所有应用程序都拥有这些附加特权。要仅在调用 nc 时授予 `privileges(5)`，建议创建并分配一个 `rbac(5)` 权限配置文件。有关其他信息，请参见“示例”部分。

示例

示例1 使用 nc

打开到 `host.example.com` 的端口 42 的 TCP 连接，使用端口 3141 作为源端口，超时为 5 秒：

```
$ nc -p 3141 -w 5 host.example.com 42
```

打开到 `host.example.com` 的端口 53 的 UDP 连接：

```
$ nc -u host.example.com 53
```

打开到 `host.example.com` 的端口 42 的 TCP 连接，使用 `10.1.2.3` 作为连接的本地端的 IP：

```
$ nc -s 10.1.2.3 host.example.com 42
```

将一个包含端口和端口范围的列表用于针对各种端口的端口扫描：

```
$ nc -z host.example.com 21-25,53,80,110-120,443
```

在某个 Unix 域套接字上创建连接并侦听：

```
$ nc -lU /var/tmp/dsocket
```

在关联端口为 8888 的 UDP 套接字上创建连接并侦听：

```
$ nc -u -l -p 8888
```

示例1 使用 nc (续)

这等效于：

```
$ nc -u -l 8888
```

在关联端口为 2222 的 TCP 套接字上创建连接并侦听，并且只绑定到地址 127.0.0.1：

```
$ nc -l 127.0.0.1 2222
```

通过将逗留 (linger) 选项和超时时间设置为 0，连接到 TCP 端口、发送一些数据然后终止与 TCP RST 段的连接（而不是传统的 TCP 关闭握手）：

```
$ echo "foo" | nc -L 0 host.example.com 22
```

从本地端口 4545 对主机 host.example.com 上的端口 22 执行端口重定向：

```
$ nc -R host.example.com/22 -l 4545
```

在这之后，应该可以运行 [ssh\(1\)](#) 客户机并连接到 host.example.com（使用运行上述命令的 host redir.example.com）：

```
$ ssh -oStrictHostKeyChecking=no -p 4545 redir.example.com
```

还可以让 nc 侦听 TCP 端口并将 TCP 数据流转换为 UDP（反之亦然）：

```
$ nc -R host.example.com/53/udp -l 4666
```

使用 10.2.3.4 的端口 8080 上的 HTTP 代理连接到 host.example.com 的端口 42。[ssh\(1\)](#) 也可使用此示例。有关更多信息，请参见 [ssh_config\(4\)](#) 中的 ProxyCommand 指令。

```
$ nc -x10.2.3.4:8080 -Xconnect host.example.com 42
```

还是同一示例，这一次如果代理要求验证，则使用用户名 ruser 来支持代理验证：

```
$ nc -x10.2.3.4:8080 -Xconnect -Pruser host.example.com 42
```

可以按类似如下方式有效地完成基本的 UDP 端口扫描：

```
$ nc -z -w 3 -u -i 0.5 host.example.com 11-100
```

在每 2 个端口之间，将暂停 0.5 秒（从而规避 ICMP 消息速率限制）并最多等待 3 秒以接收回复。如果没有接收到回复，端口可能会打开。

要作为具有附加特权的用户或角色（例如缺省的 root 帐户）使用最可能小的特权集运行 nc，还可以使用 [ppriv\(1\)](#) 来调用它。例如，将其限制为仅以绑定到某个特权端口的特权运行：

```
$ ppriv -e -sA=basic,!file_link_any,!proc_exec,!proc_fork,\
!proc_info,!proc_session,net_privaddr nc -l 42
```

示例1 使用nc (续)

要允许用户或角色仅以 `net_privaddr` 特权使用 `nc`，则需要创建一个权限配置文件。

```
/etc/security/exec_attr
Netcat privileged:solaris:cmd:::/usr/bin/nc:prvs=net_privaddr
```

```
/etc/security/prof_attr
Netcat privileged:::Allow nc to bind to privileged ports:help=None.html
```

使用 `user_attr(4)` 分配该权限配置文件以允许用户或角色运行 `nc`，从而允许其在任何端口上侦听。要允许用户或角色使用 `nc` 仅在特定端口上侦听，则应在权限配置文件中指定一个包装脚本：

```
/etc/security/exec_attr
Netcat restricted:solaris:cmd:::/usr/bin/nc-restricted:prvs=net_privaddr
```

```
/etc/security/prof_attr
Netcat restricted:::Allow nc to bind to privileged ports:help=None.html
```

并且编写一个用以限制许可选项的 `shell` 脚本，例如，编写一个只允许在 42 和 64 之间的端口（不含两者）上绑定的脚本：

```
/usr/bin/nc-restricted:

#!/bin/sh
[ $# -eq 1 ] && [ $1 -gt 42 -a $1 -lt 64 ] && /usr/bin/nc -l -p "$1"
```

当用户或角色通过配置文件 `shell` 使用该包装脚本调用 `nc` 时，这将授予额外的特权。请参见 `pfsh(1)`、`pfksh(1)`、`pfcsh(1)` 和 `pfexec(1)`。

直接调用 `nc` 时不会以附加特权运行它，在不使用 `pfexec` 或配置文件 `shell` 的情况下调用该脚本时也是如此。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	network/netcat
接口稳定性	请参见下文。

数据包名称是 "Committed"（已确定）。`-4`、`-6`、`-l`、`-n`、`-p`、`-u` 和 `-w` 选项及其参数（如果有），命令行语法是 "Committed"（已确定）。`name` 和 `port` 列表参数是 "Committed"（已确定）。端口范围语法是 "Uncommitted"（未确定）。所有其他命令行选项及其参数的接口稳定性级别是 "Uncommitted"（未确定）。

另请参见

[cat\(1\)](#)、[pfcsh\(1\)](#)、[pfexec\(1\)](#)、[pfksh\(1\)](#)、[pfsh\(1\)](#)、[ppriv\(1\)](#)、[sed\(1\)](#)、[ssh\(1\)](#)、[telnet\(1\)](#)、[inetd\(8\)](#)

作者

nc 的原始实现的作者是 Hobbit (hobbit@avian.org)。

Eric Jackson (ericj@monkey.org) 重新编写了 nc，增加了对 IPv6 的支持。

附注

如果 nc 的实例正在侦听通配符套接字（无论指定的地址族如何），仍可以将其他 nc 进程绑定到具体 IP 地址并接受与该地址的连接。例如，通过运行以下进程：

```
$ nc -4 -l 5656
```

可以运行另一个 nc 进程，对特定 IP 地址和同一端口进行侦听：

```
$ nc -4 -l 10.20.30.40 5656
```

后一个进程接受与地址 `10.20.30.40` 以及端口 `5656` 的 TCP 连接，而前一个进程接受与端口 `5656` 以及不同地址的所有 TCP 连接。

此外，还可以通过绑定到 IPv4 通配符套接字从侦听通配符套接字（不指定地址族）的进程窃取 IPv4 连接。要禁止出现这种情况以及上述行为，可以使用 `-E` 选项。

引用名	ncab2clf – 将二进制日志文件转换为通用日志文件格式
用法概要	<code>/usr/bin/ncab2clf [-Dhv] [-i <i>input-file</i>] [-o <i>output-file</i>] [-b <i>size</i>] [-n <i>number</i>] [-s <i>datetime</i>]</code>
描述	ncab2clf 命令用于将 Solaris 网络高速缓存和加速器 (Network Cache and Accelerator, NCA) 生成的日志文件从二进制格式转换为通用日志文件 (Common Log File, CLF) 格式。如果未指定 <i>input-file</i> , 则 b2clf 将使用 <code>stdin</code> 。如果未指定 <i>output-file</i> , 则输出将写入到 <code>stdout</code> 。
选项	<ul style="list-style-type: none"> -b 指定 <code>binary-log-file</code> 分块的大小 (以千字节为单位), 缺省值为 64 千字节。 -D 指定禁用直接 I/O。 -h 显示用法消息。 -i <i>input-file</i> 指定输入文件。 -n <i>number</i> 输出 <i>number</i> 个 CLF 记录。 -o <i>output-file</i> 指定输出文件。 -s <i>datetime</i> 跳过 <i>datetime</i> 指定的日期和时间之前的所有记录。可以按 CLF 格式或 <code>touch(1)</code> 实用程序指定的格式来指定日期和时间。CLF 格式是主流格式, 因此 b2clf 首先分析使用 CLF 的 <i>datetime</i>。 -v 提供详细的输出。

示例

示例1 将二进制文件转换为通用日志文件格式

以下示例将二进制文件 `/var//logs/.blf` 转换为通用日志文件格式的文件 `/var//logs/.clf`。

```
example% ncab2clf -D -i /var/nca/logs/nca.blf -o /var/nca/logs/nca.clf
```

示例2 转换多个日志文件

以下脚本可用于转换多个日志文件。由 "*" 指定的目录必须仅包含日志文件。

```
#!/bin/ksh
for filename in *
do
    ncab2clf -D < $filename > $filename.clf
done
```

示例3 对原始设备使用 -s 和 -n

以下示例显示了如何对原始设备使用 ncab2clf。如果不使用 -n 选项, 则缺省情况下会从开头到尾转换文件中的所有记录。下面所示的用 -s 指定的日期和时间是 CLF 格式的。

```
example% ncab2clf -s '10/Apr/2001:09:23:13' -n 100 < /dev/dsk/c2t1d0s6
```

退出状态 将返回以下退出值：

0 文件转换成功

>0 出现错误。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/network/http-cache-accelerator
接口稳定性	Committed（已确定）

另请参见 [ncakmod\(1\)](#)、[nca.if\(4\)](#)、[ncakmod.conf\(4\)](#)、[ncalogd.conf\(4\)](#)、[attributes\(5\)](#)、[nca\(7d\)](#)

《系统管理指南：IP 服务》

附注 NCA 生成的二进制日志文件可能会变得非常大。转换这些大型二进制文件时，请将 `-b` 选项用于 `ncab2clf` 命令，以帮助提高性能。

如果写入的数据没有作为大型块出现，则直接 I/O 对用户比较有利。不过，如果用户希望使用 `-b` 选项来转换大型块中的日志文件，则应通过使用 `-d` 选项禁用直接 I/O。

- 引用名** ncakmod – 启动或停止 NCA 内核模块
- 用法概要** `/etc/init.d/ncakmod start | stop`
- 描述** ncakmod 用于启动或停止 Solaris 网络高速缓存和加速器 (Network Cache and Accelerator, NCA) 内核模块。
- 当在命令行中指定 `start` 选项时，将为 `nca.if` 文件中列出的所有物理接口激活 NCA 内核模块。使用 `stop` 选项调用 `ncakmod` 命令时，NCA 内核模块将输出以下消息：
- ```
To stop NCA, please set the status configuration parameter
to disable in ncakmod.conf and then reboot your system. See
the ncakmod.conf(4) manual page for more information.
```
- 要在系统上正确停止 NCA，必须先编辑 `ncakmod.conf(4)` 文件，将 `status` 字段设置为 "disable"，然后重新引导系统。
- 选项**
- `start` 启动 NCA 内核模块。
- `stop` 描述用于停止 NCA 功能的当前方法。
- 示例**
- 示例 1** 启动和停止 NCA 功能
- 以下命令用于启动 NCA 功能：
- ```
example% /etc/init.d/ncakmod start
```
- 文件**
- `/etc/init.d/ncakmod` NCA 内核模块启动脚本。
- `/etc/nca/ncakmod.conf` 指定 NCA 内核模块的配置选项。
- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/network/http-cache-accelerator
接口稳定性	Committed (已确定)

另请参见 [ncab2clf\(1\)](#)、[ncad_addr\(4\)](#)、[nca.if\(4\)](#)、[ncakmod.conf\(4\)](#)、[ncalogd.conf\(4\)](#)、[attributes\(5\)](#)、[nca\(7\)](#)

引用名	newform – change the format of a text file
用法概要	newform [-s] [-i <i>tabspec</i>] [-o <i>tabspec</i>] [-bn] [-en] [-pn] [-an] [-f] [-cchar] [-ln] [<i>filename</i>]. . .
描述	<p>newform reads lines from the named <i>filenames</i>, or the standard input if no input file is named, and reproduces the lines on the standard output. Lines are reformatted in accordance with command line options in effect.</p> <p>Except for -s, command line options may appear in any order, may be repeated, and may be intermingled with the optional <i>filenames</i>. Command line options are processed in the order specified. This means that option sequences like “-e15 -l60” will yield results different from “-l60 -e15”. Options are applied to all <i>filenames</i> on the command line.</p>
选项	<p>The following options are supported:</p> <p>-s Shears off leading characters on each line up to the first tab and places up to 8 of the sheared characters at the end of the line. If more than 8 characters (not counting the first tab) are sheared, the eighth character is replaced by a * and any characters to the right of it are discarded. The first tab is always discarded.</p> <p> An error message and program exit will occur if this option is used on a file without a tab on each line. The characters sheared off are saved internally until all other options specified are applied to that line. The characters are then added at the end of the processed line.</p> <p> For example, to convert a file with leading digits, one or more tabs, and text on each line, to a file beginning with the text, all tabs after the first expanded to spaces, padded with spaces out to column 72 (or truncated to column 72), and the leading digits placed starting at column 73, the command would be:</p> <pre>newform -s -i -l -a -e filename</pre> <p>-i<i>tabspec</i> Input tab specification: expands tabs to spaces, according to the tab specifications given. <i>Tabspec</i> recognizes all tab specification forms described in tabs(1). In addition, <i>tabspec</i> may be -, in which newform assumes that the tab specification is to be found in the first line read from the standard input (see fspec(4)). If no <i>tabspec</i> is given, <i>tabspec</i> defaults to -8. A <i>tabspec</i> of -0 expects no tabs; if any are found, they are treated as -1.</p> <p>-o<i>tabspec</i> Output tab specification: replaces spaces by tabs, according to the tab specifications given. The tab specifications are the same as for -i<i>tabspec</i>. If no <i>tabspec</i> is given, <i>tabspec</i> defaults to -8. A <i>tabspec</i> of -0 means that no spaces will be converted to tabs on output.</p> <p>-bn Truncate <i>n</i> characters from the beginning of the line when the line length is greater than the effective line length (see -ln). Default is to truncate the number of characters necessary to obtain the effective line length. The default</p>

value is used when `-b` with no `n` is used. This option can be used to delete the sequence numbers from a COBOL program as follows:

```
newform -l1 -b7 filename
```

- `-en` Same as `-bn` except that characters are truncated from the end of the line.
- `-pn` Prefix `n` characters (see `-cchar`) to the beginning of a line when the line length is less than the effective line length. Default is to prefix the number of characters necessary to obtain the effective line length.
- `-an` Same as `-pn` except characters are appended to the end of a line.
- `-f` Write the tab specification format line on the standard output before any other lines are output. The tab specification format line which is printed will correspond to the format specified in the `last -o` option. If no `-o` option is specified, the line which is printed will contain the default specification of `-8`.
- `-cchar` Change the prefix/append character to `char`. Default character for `char` is a space.
- `-ln` Set the effective line length to `n` characters. If `n` is not entered, `-l` defaults to 72. The default line length without the `-l` option is 80 characters. Note: Tabs and backspaces are considered to be one character (use `-i` to expand tabs to spaces).

The `-l1` must be used to set the effective line length shorter than any existing line in the file so that the `-b` option is activated.

操作数

The following operand is supported:

filename Input file

退出状态

The following exit values are returned:

- 0 Successful operation.
- 1 Operation failed.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[csplit\(1\)](#), [tabs\(1\)](#), [fspec\(4\)](#), [attributes\(5\)](#)

诊断

All diagnostics are fatal.

```
usage: . . . newform was called with a bad option.
"not -s format" There was no tab on one line.
```

"can't open file"	Self-explanatory.
"internal line too long"	A line exceeds 512 characters after being expanded in the internal work buffer.
"tabspec in error"	A tab specification is incorrectly formatted, or specified tab stops are not ascending.
"tabspec indirection illegal"	A <i>tabspec</i> read from a file (or standard input) may not contain a <i>tabspec</i> referencing another file (or standard input).

附注

`newform` normally only keeps track of physical characters; however, for the `-i` and `-o` options, `newform` will keep track of backspaces in order to line up tabs in the appropriate logical columns.

`newform` will not prompt the user if a *tabspec* is to be read from the standard input (by use of `-i-` or `-o-`).

If the `-f` option is used, and the last `-o` option specified was `-o-`, and was preceded by either a `-o-` or a `-i-`, the tab specification format line will be incorrect.

引用名 newgrp – log in to a new group

用法概要

Command /usr/bin/newgrp [-| -l] [group]

sh Built-in newgrp [argument]

ksh88 Built-in *newgrp [argument]

ksh Built-in +newgrp [argument]

描述

Command The newgrp command logs a user into a new group by changing a user's real and effective group ID. The user remains logged in and the current directory is unchanged. The execution of newgrp always replaces the current shell with a new shell, even if the command terminates with an error (unknown group).

Any variable that is not exported is reset to null or its default value. Exported variables retain their values. System variables (such as PS1, PS2, PATH, MAIL, and HOME), are reset to default values unless they have been exported by the system or the user. For example, when a user has a primary prompt string (PS1) other than \$ (default) and has not exported PS1, the user's PS1 is set to the default prompt string \$, even if newgrp terminates with an error. Note that the shell command `export` (see [sh\(1\)](#) and [set\(1\)](#)) is the method to export variables so that they retain their assigned value when invoking new shells.

With no operands and options, newgrp changes the user's group IDs (real and effective) back to the group specified in the user's password file entry. This is a way to exit the effect of an earlier newgrp command.

A password is demanded if the group has a password and the user is not listed in /etc/group as being a member of that group. The only way to create a password for a group is to use [passwd\(1\)](#), then cut and paste the password from /etc/shadow to /etc/group. Group passwords are antiquated and not often used.

sh Built-in Equivalent to `exec newgrp argument` where *argument* represents the options and/or operand of the newgrp command.

ksh88 Built-in Equivalent to `exec to/bin/newgrp argument` where *argument* represents the options and/or operand of the newgrp command.

On this man page, [ksh88\(1\)](#) commands that are preceded by one or two * (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.

3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by ****** that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.

ksh Built-in

Equivalent to `exec to/bin/newgrp argument` where *argument* represents the options and/or operand of the `newgrp` command.

On this man page, [ksh\(1\)](#) commands that are preceded by one or two + (plus signs) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. They are not valid function names.
5. Words, following a command preceded by **++** that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and field splitting and file name generation are not performed.

选项

The following option is supported:

- l | - Change the environment to what would be expected if the user actually logged in again as a member of the new group.

操作数

The following operands are supported:

group A group name from the group database or a non-negative numeric group ID. Specifies the group ID to which the real and effective group IDs is set. If *group* is a non-negative numeric string and exists in the group database as a group name (see [getgrnam\(3C\)](#)), the numeric group ID associated with that group name is used as the group ID.

argument sh and ksh88 only. Options and/or operand of the `newgrp` command.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `newgrp`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

If `newgrp` succeeds in creating a new shell execution environment, whether or not the group identification was changed successfully, the exit status is the exit status of the shell. Otherwise, the following exit value is returned:

- >0 An error occurred.

文件 /etc/group System group file
 /etc/passwd System password file

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/newgrp,
ksh88, sh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

ksh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

另请参见

[login\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [set\(1\)](#), [sh\(1\)](#), [Intro\(3\)](#), [getgrnam\(3C\)](#), [group\(4\)](#), [passwd\(4\)](#),
[attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名	newtask – 创建新任务，还可以更改项目
用法概要	newtask [-p <i>project</i>] [-v] [-c <i>pid</i> [-F _l] [<i>command...</i>]]
描述	<p>newtask 命令执行用户的缺省 shell 或某个指定命令，从而将所执行的命令置于指定项目所拥有的某个新任务中。用户的缺省 shell 是在 passwd 数据库中指定的，并使用 getpwnam() 来确定。</p> <p>此外，newtask 可用来使已在运行的进程进入新创建的任务。还可以此命令形式为新任务指定项目。对于任务关键型且不能重新启动的进程，要将它们放置到新项目中，可能需要以上述方式进行指定。</p> <p>在扩展记帐处于活动状态时，newtask 命令还会创建一条任务记帐记录来标记前面的系统任务的完成。</p>
选项	<p>支持以下选项：</p> <p>-c <i>pid</i> 使某个正在运行的进程进入新创建的任务。还可以使用 -p 选项为新任务指定项目。调用方用户必须拥有进程或者具有超级用户特权。</p> <p>如果是要更改项目，则进程所有者必须是指定的项目的成员，或者调用方用户必须具有超级用户特权。当为某个正在运行的进程更改项目时，该项目的池绑定以及资源控制都将修改，以便与新项目的配置匹配。没有在项目条目中显式指定的控制将被保留。</p> <p>此选项与 -F 和 -l 选项不兼容。</p> <p>-F 创建一个最终任务，在最终任务中，进一步的 newtask 或 settaskid(2) 调用将失败。在某些站点上，最终任务有助于简化查明资源消耗原因的步骤。</p> <p>-l 将环境更改为用户作为新项目的成员实际重新登录时预期使用的环境。</p> <p>-p 更改新任务的项目 ID，使其与给定项目名称相关联。调用方用户必须是所请求项目的有效成员或者必须具有超级用户特权，该命令才能成功。如果未指定项目名称，则新任务将在调用方用户的当前项目中启动。</p> <p>-v 详细：在新的系统任务开始时显示系统任务 ID。</p>
操作数	<p>支持下列操作数：</p> <p><i>project</i> 应当对所创建的任务使用的资源负责的项目。必须已在 nsswitch.conf(4) 中定义的项目数据库中定义了所请求的项目。</p> <p><i>command</i> 要作为新任务执行的命令。如果未指定命令，则调用用户的登录 shell。（如果登录 shell 不可用，则调用 /bin/sh）。</p>
示例	<p>示例 1 创建新 Shell</p> <p>以下示例在 canada 项目中创建一个新 shell，并显示任务 ID：</p>

示例1 创建新 Shell (续)

```
example$ id -p
uid=565(gh) gid=10(staff) projid=10(default)
example$ newtask -v -p canada
38
example$ id -p
uid=565(gh) gid=10(staff) projid=82(canada)
```

示例2 运行 date 命令

以下示例在 `russia` 项目中运行 `date` 命令：

```
example$ newtask -p russia date
Tue Aug 31 11:12:10 PDT 1999
```

示例3 更改现有进程的项目

以下示例将 `pid` 为 9999 的现有进程的项目更改为 `russia`：

```
example$ newtask -c 9999 -p russia
```

退出状态

将返回以下退出值：

- 0 成功执行。
- 1 执行过程中发生致命错误。
- 2 指定的命令行选项无效。

文件

`/etc/project` 包含此计算机的有效项目定义的本地数据库。
`/proc/pid/*` 进程信息和控制文件。

属性

有关以下属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[proc\(1\)](#)、[id\(1M\)](#)、[poolbind\(1M\)](#)、[execvp\(2\)](#)、[setrctl\(2\)](#)、[settaskid\(2\)](#)、[setproject\(3PROJECT\)](#)、

引用名	nice – invoke a command with an altered scheduling priority
用法概要	<pre>/usr/bin/nice [-increment -n increment] command [argument]...</pre> <pre>/usr/xpg4/bin/nice [-increment -n increment] command [argument]...</pre>
csh Builtin	nice [-increment +increment] [command]
描述	<p>The nice utility invokes <code>command</code>, requesting that it be run with a different system scheduling priority. The <code>prctl(1)</code> command is a more general interface to scheduler functions.</p> <p>The invoking process (generally the user's shell) must be in a scheduling class that supports nice.</p> <p>If the C shell (see csh(1)) is used, the full path of the command must be specified. Otherwise, the csh built-in version of nice will be invoked. See <code>csh Builtin</code> below.</p>
/usr/bin/nice	If nice executes commands with arguments, it uses the default shell <code>/usr/bin/sh</code> (see sh(1)).
/usr/xpg4/bin/nice	If nice executes commands with arguments, it uses <code>/usr/xpg4/bin/sh</code> (see ksh88(1)).
csh Builtin	nice is also a csh built-in command with behavior different from the utility versions. See csh(1) for description.
选项	<p>The following options are supported:</p> <p><code>-increment -n increment</code> <i>increment</i> is a positive or negative decimal integer that has the same effect on the execution of the utility as if the utility had called the <code>nice()</code> function with the numeric value of the <i>increment</i> option-argument. See nice(2). <code>nice()</code> errors, other than <code>EINVAL</code>, are ignored. If not specified, an increment of 10 is assumed.</p> <p>The super-user may run commands with priority higher than normal by using a negative increment such as <code>-10</code>. A negative <i>increment</i> assigned by an unprivileged user is ignored.</p>
操作数	<p>The following operands are supported:</p> <p><i>command</i> The name of a command that is to be invoked. If <i>command</i> names any of the special built-in utilities (see shell_builtins(1)), the results are undefined.</p> <p><i>argument</i> Any string to be supplied as an argument when invoking <i>command</i>.</p>
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of nice: <code>LANG</code> , <code>LC_ALL</code> , <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , <code>PATH</code> , and <code>NLSPATH</code> .

退出状态

If *command* is invoked, the exit status of *nice* will be the exit status of *command*. Otherwise, *nice* will exit with one of the following values:

- 1 - 125 An error occurred.
- 126 *command* was found but could not be invoked.
- 127 *command* could not be found.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/nice

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

/usr/xpg4/bin/nice

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[csh\(1\)](#), [ksh88\(1\)](#), [nohup\(1\)](#), [priocntl\(1\)](#), [sh\(1\)](#), [shell_builtins\(1\)](#), [nice\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名

nl – line numbering filter

用法概要

```
/usr/bin/nl [-p] [-b type] [-d delim] [-f type]
           [-h type] [-i incr] [-l num] [-n format]
           [-s sep] [-w width] [-v startnum] [file]

/usr/xpg4/bin/nl [-p] [-b type] [-d delim] [-f type]
                [-h type] [-i incr] [-l num] [-n format] [-s sep]
                [-w width] [-v startnum] [file]
```

描述

The `nl` utility reads lines from the named *file*, or the standard input if no *file* is named, and reproduces the lines on the standard output. Lines are numbered on the left in accordance with the command options in effect.

`nl` views the text it reads in terms of logical pages. Line numbering is reset at the start of each logical page. A logical page consists of a header, a body, and a footer section. Empty sections are valid. Different line numbering options are independently available for header, body, and footer. For example, `-bt` (the default) numbers non-blank lines in the body section and does not number any lines in the header and footer sections.

The start of logical page sections are signaled by input lines containing nothing but the following delimiter character(s):

Line contents	Start Of
\:\::	header
\::	body
\:	footer

Unless optioned otherwise, `nl` assumes the text being read is in a single logical page body.

选项

Command options may appear in any order and may be intermingled with an optional file name. Only one file may be named. The specified default is used when the option is not entered on the command line. `/usr/xpg4/bin/nl` options require option arguments. A SPACE character *may* separate options from option arguments. `/usr/bin/nl` options *may* have option arguments. If option-arguments of `/usr/bin/nl` options are not specified, these options result in the default. The supported options are:

`-btype` Specifies which logical page body lines are to be numbered. Recognized *types* and their meanings are:

- a number all lines
- t number all non-empty lines.
- n no line numbering

pexp number only lines that contain the regular expression specified in *exp*. See NOTES below.

Default *type* for logical page body is *t* (text lines numbered).

-ftype Same as *-btype* except for footer. Default *type* for logical page footer is *n* (no lines numbered).

-ddelim The two delimiter characters specifying the start of a logical page section may be changed from the default characters (*\ :*) to two user-specified characters. If only one character is entered, the second character remains the default character (*:*). No space should appear between the *-d* and the delimiter characters. To enter a backslash, use two backslashes.

-htype Same as *-btype* except for header. Default *type* for logical page header is *n* (no lines numbered).

-iincr *incr* is the increment value used to number logical page lines. Default *incr* is 1.

-lnum *num* is the number of blank lines to be considered as one. For example, *-l2* results in only the second adjacent blank being numbered (if the appropriate *-ha*, *-ba*, and/or *-fa* option is set). Default *num* is 1.

-nformat *format* is the line numbering format. Recognized values are:

ln left justified, leading zeroes suppressed

rn right justified, leading zeroes suppressed

rz right justified, leading zeroes kept

Default *format* is *rn* (right justified).

-p Do not restart numbering at logical page delimiters.

-ssep *sep* is the character(s) used in separating the line number and the corresponding text line. Default *sep* is a TAB.

-vstartnum *startnum* is the initial value used to number logical page lines. Default *startnum* is 1.

-wwidth *width* is the number of characters to be used for the line number. Default *width* is 6.

操作数

The following operand is supported:

file A path name of a text file to be line-numbered.

示例 示例 1 An example of the nl command

The command:

```
example% nl -v10 -i10 -d!+ filename1
```

will cause the first line of the page body to be numbered 10, the second line of the page body to be numbered 20, the third 30, and so forth. The logical page delimiters are !+.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of nl: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态 The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

文件

<code>/usr/lib/locale/locale/LC_COLLATE/CollTable</code>	Collation table generated by localedef
<code>/usr/lib/locale/locale/LC_COLLATE/coll.so</code>	Shared object containing string transformation library routines

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [pr\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [regex\(5\)](#), [regexp\(5\)](#), [standards\(5\)](#)

附注 Internationalized Regular Expressions are used in the POSIX and "C" locales. In other locales, Internationalized Regular Expressions are used if the following two conditions are met:

- `/usr/lib/locale/locale/LC_COLLATE/CollTable` is present.
- `/usr/lib/locale/locale/LC_COLLATE/coll.so` is not present.

Otherwise, Simple Regular Expressions are used.

Internationalized Regular Expressions are explained on [regex\(5\)](#). Simple Regular Expressions are explained on [regexp\(5\)](#).

引用名 nm – print name list of an object file

用法概要 /usr/bin/nm [-ACDhLlnPpRrsTVv] [-efox] [-g | -u]
[-t *format*] *file*...

/usr/xpg4/bin/nm [-ACDhLlnPpRrsTVv] [-efox] [-g | -u]
[-t *format*] *file*...

描述 The nm utility displays the symbol table of each ELF object file that is specified by *file*.

If no symbolic information is available for a valid input file, the nm utility reports that fact, but does not consider it an error condition.

选项 The output of nm can be controlled using the following options:

- A Writes the full path name or library name of an object on each line.
- C Demangles C++ symbol names before printing them out.
- D Displays the SHT_DYNSYM symbol information. This is the symbol table used by ld.so.1 and is present even in stripped dynamic executables. If -D is not specified, the default behavior is to display the SHT_SYMTAB symbol information.
- e See NOTES below.
- f See NOTES below.
- g Writes only external (global) symbol information.
- h Does not display the output heading data.
- L Displays the SHT_SUNW_LDYNSYM symbol information. This symbol table contains local function symbols. SHT_SUNW_LDYNSM symbol tables are present even in stripped dynamic executables. These symbols augment the global symbols that are found in SHT_DYNSYM symbol table. If -L is not specified, the default behavior is to display the SHT_SYMTAB symbol information.
- l Distinguishes between WEAK and GLOBAL symbols by appending a * to the key letter for WEAK symbols.
- n Sorts external symbols by name before they are printed.
- o Prints the value and size of a symbol in octal instead of decimal (equivalent to -t o).
- p Produces easy to parse, terse output. Each symbol name is preceded by its value (blanks if undefined) and one of the letters:
 - A Absolute symbol.
 - B bss (uninitialized data space) symbol.
 - C COMMON symbol.

- D Data object symbol.
- F File symbol.
- N Symbol has no type.
- L Thread-Local storage symbol.
- R Register symbol.
- S Section symbol.
- T Text symbol.
- U Undefined.

If the symbol's binding attribute is:

- LOCAL The key letter is lower case.
- WEAK The key letter is upper case. If the `-l` modifier is specified, the upper case key letter is followed by a `*`
- GLOBAL The key letter is upper case.

- P Writes information in a portable output format, as specified in Standard Output.
- r Prepends the name of the object file or archive to each output line.
- R Prints the archive name (if present), followed by the object file and symbol name. If the `-r` option is also specified, this option is ignored.
- s Prints section name instead of section index.
- t *format* Writes each numeric value in the specified format. The format is dependent on the single character used as the *format* option-argument:
 - d The offset is written in decimal (default).
 - o The offset is written in octal.
 - x The offset is written in hexadecimal.
- T See NOTES.
- /usr/bin/nm -u Prints undefined symbols only.
- /usr/xpg4/bin/nm -u Prints long listing for each undefined symbol. See OUTPUT below.
- v Sorts external symbols by value before they are printed.
- V Prints the version of the nm command executing on the standard error output.
- x Prints the value and size of a symbol in hexadecimal instead of decimal (equivalent to `-t x`).

Options can be used in any order, either singly or in combination, and can appear anywhere in the command line. When conflicting options are specified (such as `-v` and `-n`, or `-o` and `-x`) the first is taken and the second ignored with a warning message to the user. (See `-R` for exception.)

操作数

The following operand is supported:

file A path name of an object file, executable file or object-file library.

Output

This section describes the `nm` utility's output options.

Standard Output

For each symbol, the following information is printed:

Index	The index of the symbol. (The index appears in brackets.)
Value	The value of the symbol is one of the following: <ul style="list-style-type: none">▪ A section offset for defined symbols in a relocatable file.▪ Alignment constraints for symbols whose section index is <code>SHN_COMMON</code>.▪ A virtual address in executable and dynamic library files.
Size	The size in bytes of the associated object.
Type	A symbol is of one of the following types: <code>NOTYPE</code> No type was specified. <code>OBJECT</code> A data object such as an array or variable. <code>FUNC</code> A function or other executable code. <code>REGI</code> A register symbol (SPARC only). <code>SECTION</code> A section symbol. <code>FILE</code> Name of the source file. <code>COMMON</code> An uninitialized common block. <code>TLS</code> A variable associated with Thread-Local storage.
Bind	The symbol's binding attributes. <code>LOCAL symbols</code> Have a scope limited to the object file containing their definition. <code>GLOBAL symbols</code> Are visible to all object files being combined. <code>WEAK symbols</code> Are essentially global symbols with a lower precedence than <code>GLOBAL</code> .
Other	An integer corresponding to one of the <code>STV_</code> symbol visibility values defined in <code><sys/elf.h></code> .

Shndx	Except for three special values, this is the section header table index in relation to which the symbol is defined. The following special values exist:
ABS	Indicates the symbol's value does not change through relocation.
COMMON	Indicates an unallocated block and the value provides alignment constraints.
UNDEF	Indicates an undefined symbol.
Name	The name of the symbol.
Object Name	The name of the object or library if -A is specified.

If the -P option is specified, the previous information is displayed using the following portable format. The three versions differ depending on whether -t d, -t o, or -t x was specified, respectively:

```
"%s %s %d %d\n", library/object name, name, type, value, size
```

```
"%s %s %o %o\n", library/object name, name, type, value, size
```

```
"%s %s %x %x\n", library/object name, name, type, value, size
```

where *type* is formatted as described for the -p option, and *library/object name* is formatted as follows:

- If -A is not specified, *library/object name* is an empty string.
- If -A is specified and the corresponding *file* operand does not name a library:


```
"%s: ", file
```
- If -A is specified and the corresponding *file* operand names a library. In this case, *object file* names the object file in the library containing the symbol being described:


```
"%s[%s]: ", file, object file
```

If -A is not specified, then if more than one *file* operand is specified or if only one *file* operand is specified and it names a library, nm writes a line identifying the object containing the following symbols before the lines containing those symbols, in the form:

- If the corresponding *file* operand does not name a library:


```
"%s:\n", file
```
- If the corresponding *file* operand names a library; in this case, *object file* is the name of the file in the library containing the following symbols:


```
"%s[%s]:\n", file, object file
```

If -P is specified, but -t is not, the format is as if -t x had been specified.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of nm: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态 The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/nm

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities

/usr/xpg4/bin/nm

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed

另请参见 [ar\(1\)](#), [as\(1\)](#), [dump\(1\)](#), [ld\(1\)](#), [ld.so.1\(1\)](#), [ar.h\(3HEAD\)](#), [a.out\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注 The following options are obsolete because of changes to the object file format and might be deleted in a future release.

- e Prints only external and static symbols. The symbol table now contains only static and external symbols. Automatic symbols no longer appear in the symbol table. They do appear in the debugging information produced by `cc -g`, which can be examined using [dump\(1\)](#).
- f Produces full output. Redundant symbols (such as `.text`, `.data`, and so forth), which existed previously, do not exist and producing full output is identical to the default output.
- T By default, nm prints the entire name of the symbols listed. Since symbol names have been moved to the last column, the problem of overflow is removed and it is no longer necessary to truncate the symbol name.

引用名 nohup – run a command immune to hangups

用法概要 /usr/bin/nohup *command* [*argument*]...

/usr/bin/nohup -p [-Fa] *pid* [*pid*]...

/usr/bin/nohup -g [-Fa] *gpid* [*gpid*]...

/usr/xpg4/bin/nohup *command* [*argument*]...

描述

The nohup utility invokes the named *command* with the arguments supplied. When the *command* is invoked, nohup arranges for the SIGHUP signal to be ignored by the process.

When invoked with the -p or -g flags, nohup arranges for processes already running as identified by a list of process IDs or a list of process group IDs to become immune to hangups.

The nohup utility can be used when it is known that *command* takes a long time to run and the user wants to log out of the terminal. When a shell exits, the system sends its children SIGHUP signals, which by default cause them to be killed. All stopped, running, and background jobs ignores SIGHUP and continue running, if their invocation is preceded by the nohup command or if the process programmatically has chosen to ignore SIGHUP.

The nohup utility causes processes to ignore SIGHUP but does not in any way protect those processes from other signals. Since modern shells sometimes send signals other than SIGHUP upon logout, it is possible for jobs running under /usr/bin/nohup to be killed when the controlling shell exits.

/usr/bin/nohup Processes run by /usr/bin/nohup are immune to SIGHUP (hangup) and SIGQUIT (quit) signals.

/usr/bin/nohup -p [-Fa] Processes specified by ID are made immune to SIGHUP and SIGQUIT, and all output to the controlling terminal is redirected to nohup.out. If -F is specified, nohup forces control of each process. If -a is specified, nohup changes the signal disposition of SIGHUP and SIGQUIT even if the process has installed a handler for either signal.

/usr/bin/nohup -g [-Fa] Every process in the same process group as the processes specified by ID are made immune to SIGHUP and SIGQUIT, and all output to the controlling terminal is redirected to nohup.out. If -F is specified, nohup forces control of each process. If -a is specified, nohup changes the signal disposition of SIGHUP and SIGQUIT even if the process has installed a handler for either signal.

/usr/xpg4/bin/nohup Processes run by /usr/xpg4/bin/nohup are immune to SIGHUP.

The nohup utility does not arrange to make processes immune to a SIGTERM (terminate) signal, so unless they arrange to be immune to SIGTERM or the shell makes them immune to SIGTERM, they receive it.

If nohup.out is not writable in the current directory, output is redirected to \$HOME/nohup.out. If a file is created, the file has read and write permission (600). See [chmod\(1\)](#). If the standard error is a terminal, it is redirected to the standard output, otherwise it is not redirected. The priority of the process run by nohup is not altered.

选项

The following options are supported:

- a Always changes the signal disposition of target processes. This option is valid only when specified with -p or -g.
- F Force. Grabs the target processes even if another process has control. This option is valid only when specified with -p or -g.
- g Operates on a list of process groups. This option is not valid with -p.
- p Operates on a list of processes. This option is not valid with -g.

操作数

The following operands are supported:

- pid* A decimal process ID to be manipulated by nohup -p.
- pgid* A decimal process group ID to be manipulated by nohup -g.
- command* The name of a command that is to be invoked. If the *command* operand names any of the special [shell_builtins\(1\)](#) utilities, the results are undefined.
- argument* Any string to be supplied as an argument when invoking the *command* operand.

用法

Caution should be exercised when using the -F flag. Imposing two controlling processes on one victim process can lead to chaos. Safety is assured only if the primary controlling process, typically a debugger, has stopped the victim process and the primary controlling process is doing nothing at the moment of application of the proc tool in question.

示例

示例 1 Applying nohup to Pipelines or Command Lists

It is frequently desirable to apply nohup to pipelines or lists of commands. This can be done only by placing pipelines and command lists in a single file, called a shell script. One can then issue:

```
example$ nohup sh file
```


示例 1 Applying nohup to Pipelines or Command Lists (续)

and the nohup applies to everything in *file*. If the shell script *file* is to be executed often, then the need to type `sh` can be eliminated by giving *file* execute permission.

Add an ampersand and the contents of *file* are run in the background with interrupts also ignored (see [sh\(1\)](#)):

```
example$ nohup file &
```

示例 2 Applying nohup -p to a Process

```
example$ long_running_command &
example$ nohup -p 'pgrep long_running_command'
```

示例 3 Applying nohup -g to a Process Group

```
example$ make &
example$ ps -o sid -p $$
      SID
81079
example$ nohup -g 'pgrep -s 81079 make'
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of nohup: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, PATH, NLSPATH, and PATH.

HOME Determine the path name of the user's home directory: if the output file `nohup.out` cannot be created in the current directory, the nohup command uses the directory named by HOME to create the file.

退出状态

The following exit values are returned:

126 *command* was found but could not be invoked.

127 An error occurred in nohup, or *command* could not be found

Otherwise, the exit values of nohup are those of the *command* operand.

文件

`nohup.out` The output file of the nohup execution if standard output is a terminal and if the current directory is writable.

`$HOME/nohup.out` The output file of the nohup execution if standard output is a terminal and if the current directory is not writable.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/nohup

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Enabled

/usr/xpg4/bin/nohup

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[bash\(1\)](#), [batch\(1\)](#), [chmod\(1\)](#), [csh\(1\)](#), [disown\(1\)](#), [ksh88\(1\)](#), [nice\(1\)](#), [pgrep\(1\)](#), [proc\(1\)](#), [ps\(1\)](#), [sh\(1\)](#), [shell_builtins\(1\)](#), [setpgrp\(1\)](#), [signal\(3C\)](#), [proc\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

警告

If you are running the Korn shell ([ksh88\(1\)](#)) as your login shell, and have nohup'ed jobs running when you attempt to log out, you are warned with the message:

```
You have jobs running.
```

You need to log out a second time to actually log out. However, your background jobs continues to run.

附注

The C-shell ([csh\(1\)](#)) has a built-in command nohup that provides immunity from SIGHUP, but does not redirect output to nohup.out. Commands executed with '&' are automatically immune to HUP signals while in the background.

nohup does not recognize command sequences. In the case of the following command,

```
example$ nohup command1; command2
```

the nohup utility applies only to command1. The command,

```
example$ nohup (command1; command2)
```

is syntactically incorrect.

引用名	nroff – format documents for display or line-printer
用法概要	nroff [-ehiq] [-mname] [-nN] [-opagelist] [-raN] [-sN] [-Tname] [-uN] [filename...]
描述	<p>The nroff utility formats text in the named <i>filename</i> for typewriter-like devices. See also troff(1).</p> <p>If no <i>filename</i> argument is present, nroff reads the standard input. An argument consisting of a '-' is taken to be a file name corresponding to the standard input.</p>
选项	<p>The following options are supported. Options can appear in any order so long as they appear <i>before</i> the files.</p> <ul style="list-style-type: none"> -e Produces equally-spaced words in adjusted lines, using full terminal resolution. -h Uses output TAB characters during horizontal spacing to speed output and reduces output character count. TAB settings are assumed to be every 8 nominal character widths. -i Reads the standard input after the input files are exhausted. -q Does not print output that was read from an .rd request. -mname Prepends the macro file /usr/share/lib/tmac/<i>name</i> to the input files. -nN Numbers first generated page <i>N</i>. -opagelist Prints only pages whose page numbers appear in the comma-separated <i>list</i> of numbers and ranges. A range <i>N-M</i> means pages <i>N</i> through <i>M</i>; an initial -<i>N</i> means from the beginning to page <i>N</i>; and a final <i>N-</i> means from <i>N</i> to the end. -raN Sets register <i>a</i> (one-character) to <i>N</i>. -sN Stops every <i>N</i> pages. nroff halts prior to every <i>N</i> pages (default <i>N</i>=1) to allow paper loading or changing, and resumes upon receipt of a NEWLINE. -Tname Prepares output for a device of the specified <i>name</i>. Known <i>names</i> are: <ul style="list-style-type: none"> 37 Teletype Corporation Model 37 terminal — this is the default. lp tn300 GE — any line printer or terminal without half-line capability. 300 DASI-300. 300-12 DASI-300 — 12-pitch. 300S DASI-300S. 300S-12 DASI-300S. 382 DASI-382 (fancy DTC 382). 450 DASI-450 (Diablo Hyterm).

450-12 DASI-450 (Diablo Hyterm) — 12-pitch.

832 AJ 832.

`-uN` Set the emboldening factor for the font mounted in position 3 to *N*. If *N* is missing, then set the emboldening factor to 0.

操作数

The following operand is supported:

filename The file containing text to be processed by nroff.

示例

示例 1 Formatting with a macro package

The following command formats `users.guide` using the `-me` macro package, and stopping every 4 pages:

```
example% nroff -s4 -me users.guide
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of nroff: `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

文件

`/usr/tmp/trtmp*` temporary file (see NOTES)

`/usr/share/lib/tmac/tmac.*` standard macro files

`/usr/share/lib/nterm/*` terminal driving tables for nroff

`/usr/share/lib/nterm/README` index to terminal description files

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools
CSI	Enabled

另请参见

[checknr\(1\)](#), [col\(1\)](#), [eqn\(1\)](#), [man\(1\)](#), [tbl\(1\)](#), [troff\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [me\(5\)](#), [ms\(5\)](#), [term\(5\)](#)

附注

`/usr/tmp` is currently a symbolic link to `/var/tmp`.

Previous documentation incorrectly described the numeric register `yr` as being the *Last two digits of current year*. `yr` is in actuality the number of years since 1900. To correctly obtain the last two digits of the current year through the year 2099, the definition given below of string register `yy` can be included in a document and subsequently used to display a two-digit year. Notice that any other available one- or two-character register name can be substituted for `yy`.

```
.\ " definition of new string register yy--last two digits of year
.\ " use yr (# of years since 1900) if it is < 100
.\ ie \n(yr<100 .ds yy \n(yr
.\ el \{          .\ " else, subtract 100 from yr, store in ny
```

```
.nr ny \n(yr-100
.ie \n(ny>9 \{      .\" use ny if it is two digits
.ds yy \n(ny
.\" remove temporary number register ny
.rr ny \}
.el \{.ds yy 0
.\" if ny is one digit, append it to 0
.as yy \n(ny
.rr ny \} \}
```

引用名 od – octal dump

用法概要

```

/usr/bin/od [-bcCdDfF0oSsvXx] [-] [file] [offset_string]
/usr/bin/od [-bcCdDfF0oSsvXx] [-A address_base] [-j skip]
    [-N count] [-t type_string]... [-] [file]...
/usr/xpg4/bin/od [-bcCdDfF0oSsvXx] [file] [offset_string]
/usr/xpg4/bin/od [-bcCdDfF0oSsvXx] [-A address_base]
    [-j skip] [-N count] [-t type_string]... [file]...

```

描述

The od command copies sequentially each input file to standard output and transforms the input data according to the output types specified by the -t or -bcCdDfF0oSsvXx options. If no output type is specified, the default output is as if -t o2 had been specified. Multiple types can be specified by using multiple -bcCdDfF0oSstvxX options. Output lines are written for each type specified in the order in which the types are specified. If no file is specified, the standard input is used. The [offset_string] operand is mutually exclusive from the -A, -j, -N, and -t options. For the purposes of this description, the following terms are used:

word	Refers to a 16-bit unit, independent of the word size of the machine.
long word	Refers to a 32-bit unit.
double long word	Refers to a 64-bit unit.

选项 The following options are supported:

-A address_base	Specifies the input offset base. The address_base option-argument must be a character. The characters d, o and x specify that the offset base will be written in decimal, octal or hexadecimal, respectively. The character n specifies that the offset will not be written. Unless -A n is specified, the output line will be preceded by the input offset, cumulative across input files, of the next byte to be written. In addition, the offset of the byte following the last byte written will be displayed after all the input data has been processed. Without the -A address_base option and the [offset_string] operand, the input offset base is displayed in octal.
-b	Interprets bytes in octal. This is equivalent to -t o1.

/usr/bin/od	-c	Displays single-byte characters. Certain non-graphic characters appear as C-language escapes:
-------------	----	---

null	\0
backspace	\b
form-feed	\f
new-line	\n
return	\r
tab	\t

Others appear as 3-digit octal numbers. For example:

```
echo "hello world" | od -c
0000000 h e l l o       w o r l d \n
0000014
```

/usr/xpg4/bin/od

- c Interprets bytes as single-byte or multibyte characters according to the current setting of the LC_CTYPE locale category. Printable multibyte characters are written in the area corresponding to the first byte of the character. The two-character sequence ** is written in the area corresponding to each remaining byte in the character, as an indication that the character is continued. Non-graphic characters appear the same as they would using the -C option.
- C Interprets bytes as single-byte or multibyte characters according to the current setting of the LC_CTYPE locale category. Printable multibyte characters are written in the area corresponding to the first byte of the character. The two-character sequence ** is written in the area corresponding to each remaining byte in the character, as an indication that the character is continued. Certain non-graphic characters appear as C escapes:
- | | |
|-----------|----|
| null | \0 |
| backspace | \b |
| form-feed | \f |
| new-line | \n |
| return | \r |
| tab | \t |
- Other non-printable characters appear as one three-digit octal number for each byte in the character.
- d Interprets words in unsigned decimal. This is equivalent to -t u2.
- D Interprets long words in unsigned decimal. This is equivalent to -t u4.
- f Interprets long words in floating point. This is equivalent to -t f4.
- F Interprets double long words in extended precision. This is equivalent to -t f8.
- j *skip* Jumps over *skip* bytes from the beginning of the input. The od command will read or seek past the first *skip* bytes in the concatenated input files. If the combined input is not at least *skip* bytes long, the od command will write a diagnostic message to standard error and exit with a non-zero exit status.

By default, the *skip* option-argument is interpreted as a decimal number. With a leading 0x or 0X, the offset is interpreted as a hexadecimal number; otherwise, with a leading 0, the offset will be interpreted as an octal number. Appending the character b, k, or m to offset will cause it to be interpreted as a

multiple of 512, 1024 or 1 048 576 bytes, respectively. If the *skip* number is hexadecimal, any appended *b* is considered to be the final hexadecimal digit. The address is displayed starting at `00000000`, and its base is not implied by the base of the *skip* option-argument.

- N *count* Formats no more than *count* bytes of input. By default, *count* is interpreted as a decimal number. With a leading `0x` or `0X`, *count* is interpreted as a hexadecimal number; otherwise, with a leading `0`, it is interpreted as an octal number. If *count* bytes of input (after successfully skipping, if *-j skip* is specified) are not available, it will not be considered an error. The `od` command will format the input that is available. The base of the address displayed is not implied by the base of the *count* option-argument.
- o Interprets words in octal. This is equivalent to `-t o2`.
- O Interprets long words in unsigned octal. This is equivalent to `-t o4`.
- s Interprets words in signed decimal. This is equivalent to `-t d2`.
- S Interprets long words in signed decimal. This is equivalent to `-t d4`.
- t *type_string* Specifies one or more output types. The *type_string* option-argument must be a string specifying the types to be used when writing the input data. The string must consist of the type specification characters:
 - a *Named character*. Interprets bytes as named characters. Only the least significant seven bits of each byte will be used for this type specification. Bytes with the values listed in the following table will be written using the corresponding names for those characters.

The following are named characters in `od`:

Value	Name
<code>\000</code>	<code>nul</code>
<code>\001</code>	<code>soh</code>
<code>\002</code>	<code>stx</code>
<code>\003</code>	<code>etx</code>
<code>\004</code>	<code>eot</code>
<code>\005</code>	<code>enq</code>
<code>\006</code>	<code>ack</code>
<code>\007</code>	<code>bel</code>
<code>\010</code>	<code>bs</code>
<code>\011</code>	<code>ht</code>
<code>\012</code>	<code>lf</code>
<code>\013</code>	<code>vt</code>
<code>\014</code>	<code>ff</code>
<code>\015</code>	<code>cr</code>
<code>\016</code>	<code>so</code>


```

\017  si
\020  dle
\021  dc1
\022  dc2
\023  dc3
\024  dc4
\025  nak
\026  syn
\027  etb
\030  can
\031  em
\032  sub
\033  esc
\034  fs
\035  gs
\036  rs
\037  us
\040  sp
\177  del

```

- c *Character*. Interprets bytes as single-byte or multibyte characters specified by the current setting of the LC_CTYPE locale category. Printable multibyte characters are written in the area corresponding to the first byte of the character. The two-character sequence ** is written in the area corresponding to each remaining byte in the character, as an indication that the character is continued. Certain non-graphic characters appear as C escapes: \0, \a, \b, \f, \n, \r, \t, \v. Other non-printable characters appear as one three-digit octal number for each byte in the character.

The type specification characters d, f, o, u, and x can be followed by an optional unsigned decimal integer that specifies the number of bytes to be transformed by each instance of the output type.

- f *Floating point*. Can be followed by an optional F, D, or L indicating that the conversion should be applied to an item of type float, double, or long double, respectively.
- d, o, u, and x *Signed decimal, octal, unsigned decimal, and hexadecimal, respectively*. Can be followed by an optional C, S, I, or L indicating that the conversion should be applied to an item of type char, short, int, or long, respectively.

Multiple types can be concatenated within the same *type_string* and multiple -t options can be specified. Output lines are written for each type specified in the order in which the type specification characters are specified.

- v Shows all input data (verbose). Without the -v option, all groups of output lines that would be identical to the immediately preceding output line (except for byte offsets), will be replaced with a line containing only an asterisk (*).
- x Interprets words in hex. This is equivalent to -t x2.
- X Interprets long words in hex. This is equivalent to -t x4.

操作数

/usr/bin/od

The following operands are supported for /usr/bin/od only:

- Uses the standard input in addition to any files specified. When this operand is not given, the standard input is used only if no *file* operands are specified.
- file* A path name of a file to be read. If no *file* operands are specified, the standard input will be used. If there are no more than two operands, none of the -A, -j, -N, or -t options is specified, and *any* of the following are true:
 1. the first character of the last operand is a plus sign (+)
 2. the first character of the second operand is numeric
 3. the first character of the second operand is x and the second character of the second operand is a lower-case hexadecimal character or digit
 4. the second operand is named "x"
 5. the second operand is named "."

then the corresponding operand is assumed to be an offset operand rather than a file operand.

Without the -N count option, the display continues until an end-of-file is reached.

```
[+][0] offset [.] [b|B]
[+][0][offset] [.]
[+][0x|x][offset]
[+][0x|x] offset[B]
```

The *offset_string* operand specifies the byte offset in the file where dumping is to commence. The offset is interpreted in octal bytes by default. If *offset* begins with "0", it is interpreted in octal. If *offset* begins with "x" or "0x", it is interpreted in hexadecimal and any appended "b" is considered to be the final hexadecimal digit. If "." is appended, the offset is interpreted in decimal. If "b" or "B" is appended, the offset is interpreted in units of 512 bytes. If the file

argument is omitted, the *offset* argument must be preceded by a plus sign (+). The address is displayed starting at the given offset. The radix of the address will be the same as the radix of the offset, if specified, otherwise it will be octal. Decimal overrides octal, and it is an error to specify both hexadecimal and decimal conversions in the same offset operand.

/usr/xpg4/bin/od

The following operands are supported for /usr/xpg4/bin/od only:

file

Same as /usr/bin/od, except only one of the first two conditions must be true.

[+] [0] *offset* [.] [b|B]

+ [*offset*] [.]

[+] [0x] [*offset*]

[+] [0x] *offset* [B]

+x [*offset*]

+x*offset* [B]

Description of *offset_string* is the same as for /usr/bin/od.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of od: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, LC_NUMERIC, and NLSPATH.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/od

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	enabled

/usr/xpg4/bin/od

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[sed\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名 on – execute a command on a remote system with the local environment

用法概要 on [-i] [-d] [-n] *host command [argument] ...*

描述 The on program is used to execute commands on another system, in an environment similar to that invoking the program. All environment variables are passed and the current working directory is preserved. To preserve the working directory, the working file system must be either already mounted on the host or be exported to it. Relative path names will only work if they are within the current file system. Absolute path names may cause problems.

The standard input is connected to the standard input of the remote command. The standard output and the standard error from the remote command are sent to the corresponding files for the on command.

Note that the on program requires that the [rpc.rexd\(1M\)](#) service be running on the remote machine. By default, `rpc.rexd` is present but not running on an Oracle Solaris system. Because of its better security, [ssh\(1\)](#) is the preferred method of invoking commands on remote machines.

选项 The following options are supported:

- d Debug mode. Prints out some messages as work is being done.
- i Interactive mode. Uses remote echoing and special character processing. This option is needed for programs that expect to be talking to a terminal. All terminal modes and window size changes are propagated.
- n No Input. This option causes the remote program to get EOF when it reads from the standard input, instead of passing the standard input from the standard input of the on program. For example, -n is necessary when running commands in the background with job control.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/file-system/nfs

另请参见 [chkey\(1\)](#), [rlogin\(1\)](#), [rsh\(1\)](#), [ssh\(1\)](#), [telnet\(1\)](#), [rpc.rexd\(1M\)](#), [attributes\(5\)](#)

诊断

unknown host	Host name not found.
cannot connect to server	Host down or not running the server.
can't find	Problem finding the working directory.
can't locate mount point	Problem finding current file system.
RPC: Authentication error	The server requires DES authentication and you do not have a secret key registered with

key serv. Perhaps you logged in without a password. Try to keylogin. If that fails, try to set your publickey with chkey.

on server: RPC: can't encode arguments

The 10240 byte limit for arguments to be encoded and passed from the sending to the receiving system has been exceeded.

Other diagnostic messages may be passed back from the server.

已知问题

When the working directory is remote mounted over NFS, a Control-Z hangs the window.

Root cannot use on.

引用名 `optisa` – 确定要使用的最佳变量指令集

用法概要 `optisa instruction_set...`

描述 `optisa` 显示在该命令中指定的哪个 `instruction_set` 在此计算机上的执行性能将是最佳的。在这种情况下，“最佳”是按 `isalist(1)` 返回指令集名称的顺序定义的。`isalist(5)` 中给出了 `instruction_set` 的可能值。

属性 有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	system/core-os

退出状态 将返回以下退出值：

- 0 该命令输出了您指定的 `instruction_set` 值之一。
- 1 没有任何输出，也就是说，此计算机无法使用您随 `optisa` 命令指定的任何 `instruction_set`。

另请参见 `isalist(1)`、`uname(1)`、`attributes(5)`、`isalist(5)`

附注 在确定应在给定计算机上使用给定程序的多个二进制版本中的哪一个时，`optisa` 比 `uname -p` 或 `uname -m`（请参见 `uname(1)`）更为好用。

引用名 pack, pcat, unpack – compress and expand files

用法概要 pack [-f/] [-] file...

pcat file...

unpack [-/] file...

描述

pack

The pack command attempts to store the specified files in a compressed form. Wherever possible (and useful), each input file `file` is replaced by a packed file `file.z` with the same access modes, access and modified dates, and owner as those of `file`. If pack is successful, `file` is removed.

The amount of compression obtained depends on the size of the input file and the character frequency distribution. Because a decoding tree forms the first part of each `.z` file, it is usually not worthwhile to pack files smaller than three blocks, unless the character frequency distribution is very skewed, which can occur with printer plots or pictures.

Typically, text files are reduced to 60-75% of their original size. Load modules, which use a larger character set and have a more uniform distribution of characters, show little compression, the packed versions being about 90% of the original size.

The pack utility returns a value that is the number of files that it failed to compress. If that number exceeds 255, 255 is returned.

No packing occurs if:

- the file appears to be already packed
- the file name is too long to add the `.z` suffix
- the file has links
- the file is a directory
- the file cannot be opened
- the file is empty
- no disk storage blocks are saved by packing
- a file called `file.z` already exists
- the `.z` file cannot be created
- an I/O error occurred during processing.

The last segment of the file name must be short enough to allow space for the appended `.z` extension. Directories cannot be compressed.

pcat

The pcat command does for packed files what [cat\(1\)](#) does for ordinary files, except that pcat cannot be used as a filter. The specified files are unpacked and written to the standard output.

pcat returns the number of files it was unable to unpack. Failure can occur if:

- the file cannot be opened;
- the file does not appear to be the output of pack.

unpack

The `unpack` command expands files created by `pack`. For each `file` specified in the command, a search is made for a file called `file.z` (or just `file`, if `file` ends in `.z`). If this file appears to be a packed file, it is replaced by its expanded version. The new file has the `.z` suffix stripped from its name, and has the same access modes, access and modification dates, and owner as those of the packed file.

`unpack` returns a value that is the number of files it was unable to unpack. Failure can occur for the same reasons that it can in `pcat`, as well as for the following:

- a file with the unpacked name already exists;
- the unpacked file cannot be created.

选项

The following options are supported by `pack`:

- f Forces packing of `file`. This is useful for causing an entire directory to be packed even if some of the files do not benefit. Packed files can be restored to their original form using `unpack` or `pcat`.

The following options are supported by `pack` and `unpack`:

- / When packing or unpacking, copies any ACL and extended system attributes associated with the source file to the target file. If an ACL or extended system attributes cannot be copied, the original file is retained, a diagnostic message is written to `stderr`, and the final exit status is non-zero.

操作数

The following operands are supported:

- `file` A path name of a file to be packed, unpacked, or `pcated`; `file` can include or omit the `.z` suffix.
- `pack` uses Huffman (minimum redundancy) codes on a byte-by-byte basis. If the – argument is used, an internal flag is set that causes the number of times each byte is used, its relative frequency, and the code for the byte to be printed on the standard output. Additional occurrences of – in place of `file` causes the internal flag to be set and reset.

用法

See [largefile\(5\)](#) for the description of the behavior of `pack`, `pcat`, and `unpack` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Viewing a Packed File

To view a packed file named `file.z` use:

```
example% pcat file.z
```

or just:

示例 1 Viewing a Packed File (续)

```
example% pcat file
```

示例 2 Making and Unpacked Copy:

To make an unpacked copy, say `nnn`, of a packed file named `file.z` (without destroying `file.z`) use the command:

```
example% pcat file >nnn
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `pack`, `pcat`, and `unpack`: `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred. The number of files the command failed to pack/unpack is returned. If the number of failures exceeds 255, then 255 is returned.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

另请参见

[cat\(1\)](#), [compress\(1\)](#), [zcat\(1\)](#), [fgetattr\(3C\)](#), [fsetattr\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#)

引用名	packagemanager – 映像包管理系统的 GUI
用法概要	<pre> /usr/bin/packagemanager [options] /usr/bin/packagemanager [-h --help] [-i --info-install file] [-U --update-all] [-R --image-dir dir] /usr/bin/packagemanager [file] </pre>
描述	<p>packagemanager 是映像包管理系统软件 pkg(5) 的图形用户界面。</p> <p>使用软件包管理器，您可以执行以下任务：</p> <ul style="list-style-type: none"> ■ 搜索、安装和删除软件包。 ■ 添加、删除和修改发布者。 ■ 创建、删除和管理引导环境。 <p>如果指定了 <i>file</i> 操作数且其后缀为 <i>.p5i</i>，packagemanager 将在 Web 安装模式下启动，可添加一个或多个发布者并为每个发布者添加多个软件包。</p>
选项	<p>支持以下选项：</p> <pre> -h --help 显示用法消息。 -i file --info-install file 指定 .p5i 文件以在 Web 安装模式下运行 packagemanager。file 必须具有后缀 .p5i。 -R dir --image-dir dir 对根目录为 dir 的映像（而不是自动搜索到的映像）执行操作。 -U --update-all 更新所有具有可用更新的已安装软件包。 注 – 如果 package/pkg、package/pkg/package-manager 或 package/pkg/update-manager 软件包需要更新，则 packagemanager 首先更新这些软件包，然后重新启动以完成其余所有更新。 </pre>
操作数	<p><i>file</i> Web 安装文件。该文件必须具有后缀 <i>.p5i</i>。有关 Web 安装的更多信息，请参见软件包管理器联机帮助。</p>
示例	<p>示例1 对当前映像执行操作</p> <p>对当前映像调用 packagemanager。</p> <pre> \$ packagemanager </pre>

示例2 对指定映像执行操作

对存储在 `/aux0/example_root` 中的映像调用 `packagemanager`。

```
$ packagemanager -R /aux0/example_root
```

示例3 在 Web 安装模式下调用

在 Web 安装模式下调用 `packagemanager`。

```
$ packagemanager ~/test.p5i
```

退出状态

将返回以下退出值：

- 0 一切正常工作。
- 1 出现错误。
- 2 指定的命令行选项无效。

文件

因为 `pkg(5)` 映像可位于任意一个较大的文件系统内，需要使用标记 `$IMAGE_ROOT` 来区分相对路径。对于典型系统安装，`$IMAGE_ROOT` 等效于 `/`。

`$IMAGE_ROOT/var/pkg`
完整或部分映像的元数据目录。

`$IMAGE_ROOT/.org.opensolaris, pkg`
用户映像的元数据目录。

在特定映像的元数据中，某些文件和目录包含修复和恢复期间有用的信息。标记 `$IMAGE_META` 用于指示包含元数据的顶层目录。`$IMAGE_META` 通常是以上给出的两个路径之一。

`$IMAGE_META/gui-cache`
缓存元数据的位置，`packagemanager` 维护高速缓存元数据，以便加速程序启动及发布者之间的转换。

`$IMAGE_META` 目录分层结构中的其他路径是专用的，但可以进行更改。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg/package-manager
接口稳定性	Uncommitted (未确定)

另请参见

[pm-updatemanager\(1\)](#)、[pkg\(1\)](#)、[pkg\(5\)](#)

软件包管理器联机帮助

<http://hub.opensolaris.org/bin/view/Project+pkg/>

附注 需要使用足够的特权调用 `packagemanager` 以便对映像的文件和目录执行操作。

引用名	pagesize – 显示内存页面的大小
用法概要	/usr/bin/pagsize [-a]
描述	pagesize 实用程序输出内存页面的缺省大小（以字节为单位），与 getpagesize(3C) 返回的内容相同。此程序在构造可移植 shell 脚本时非常有用。
选项	支持以下选项： -a 输出系统支持的所有可能的硬件地址转换大小。
属性	有关下列属性的说明，请参见 attributes(5) ：

属性类型	属性值
可用性	system/core-os

另请参见 [ppgsz\(1\)](#)、[getpagesize\(3C\)](#)、[getpagesizes\(3C\)](#)、[attributes\(5\)](#)

引用名	pargs – 输出进程参数、环境变量或辅助向量
用法概要	pargs [-aceFlx] [<i>pid</i> <i>core</i>]...
描述	<p>pargs 实用程序检查目标进程或进程核心文件，并且输出参数、环境变量及值或者进程辅助向量。</p> <p>pargs 将不可打印的字符输出为 <code>\xxx</code> 形式的转义八进制字符，除非该字符是 formats(5) 的转义序列部分中指定的字符，如果是，则会按该部分中指定的方式输出该字符。</p> <p>pargs 尝试对目标进程的语言环境保持敏感性。如果目标进程和 pargs 进程没有使用相同的字符编码，则 pargs 会尝试使用 iconv(3C) 工具来生成所提取的字符串的可打印版本。如果无法进行此类转换，则字符串将显示为 7 位 ASCII。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none">-a 列显 <code>argv[]</code> 中包含的进程参数（缺省）。-c 不管目标的语言环境如何，都将目标进程中的字符串视为以 7 位 ASCII 编码的字符串。iconv(3C) 的使用将被抑制。-e 输出由 <code>_environ</code> 符号或 <code>/proc/<i>pid</i>/psinfo</code> 中的 <code>pr_envp</code> 指向的进程环境变量和值。-F 强制。抓取目标进程，即使另一进程已掌握了控制权。-l 在单个命令行上显示参数。命令行是以适合 <code>/bin/sh</code> 进行解释的方式输出的。如果参数包含不可打印的符号，或者目标进程处于不同的语言环境，则会显示一条警告消息。<code>/bin/sh</code> 可能无法正确地解释结果命令行。-x 输出进程辅助向量。
操作数	<p>支持下列操作数：</p> <p><i>pid</i> 进程 ID 列表。</p> <p><i>core</i> 进程核心文件。</p>
用法	<p>使用 -F 标志时应谨慎。在一个被调试的进程上施加两个控制进程可能会导致混乱。仅当主控制进程（通常是调试器）已停止了被调试的进程，并且在应用 <code>proc</code> 工具的可疑时刻主控制进程未在执行任何操作，才能保证安全。</p>
退出状态	<p>将返回以下退出值：</p> <p>0 操作成功。</p> <p>非零 发生错误，例如没有此类进程、权限被拒绝或者选项无效。</p>

文件 /proc/pid/* 进程信息和控制文件。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed（已确定）

另请参见 [proc\(1\)](#)、[iconv\(3C\)](#)、[proc\(4\)](#)、[ascii\(5\)](#)、[attributes\(5\)](#)、[environ\(5\)](#)、[formats\(5\)](#)

引用名 passwd – change login password and password attributes

用法概要

```
passwd [-r files | -r ldap | -r nis] [name]
passwd [-r files] [-egh] [name]
passwd [-r files] -s [-a]
passwd [-r files] -s [name]
passwd [-r files] [-d | -l | -u | -N] [-f] [-n min]
    [-w warn] [-x max] name
passwd -r ldap [-egh] [name]
passwd [-r ldap ] -s [-a]
passwd [-r ldap ] -s [name]
passwd -r ldap [-d | -l | -u | -N] [-f] [-n min]
    [-w warn] [-x max] name
passwd -r nis [-egh] [name]
```

描述

The `passwd` command changes the password or lists password attributes associated with the user's login *name*. Additionally, authorized users can use `passwd` to install or change passwords and attributes associated with any login *name*.

When used to change a password, `passwd` prompts everyone for their old password, if any. It then prompts for the new password twice. When the old password is entered, `passwd` checks to see if it has aged sufficiently. If aging is insufficient, `passwd` terminates; see [pwconv\(1M\)](#) and [shadow\(4\)](#) for additional information.

The `pwconv` command creates and updates `/etc/shadow` with information from `/etc/passwd`. `pwconv` relies on a special value of `x` in the password field of `/etc/passwd`. This value of `x` indicates that the password for the user is already in `/etc/shadow` and should not be modified.

If aging is sufficient, a check is made to ensure that the new password meets construction requirements. When the new password is entered a second time, the two copies of the new password are compared. If the two copies are not identical, the cycle of prompting for the new password is repeated for, at most, two more times.

Passwords must be constructed to meet the following requirements:

- Each password must have `PASSLENGTH` characters, where `PASSLENGTH` is defined in `/etc/default/passwd` and is set to 6. Setting `PASSLENGTH` to more than eight characters requires configuring [policy.conf\(4\)](#) with an algorithm that supports greater than eight characters.
- Each password must meet the configured complexity constraints specified in `/etc/default/passwd`.

- Each password must not be a member of the configured dictionary as specified in `/etc/default/passwd`.
- For accounts in name services which support password history checking, if prior password history is defined, new passwords must not be contained in the prior password history.

If all requirements are met, by default, the `passwd` command consults `/etc/nsswitch.conf` to determine in which repositories to perform password update. It searches the `passwd` and `passwd_compat` entries. The sources (repositories) associated with these entries are updated. However, the password update configurations supported are limited to the following cases. Failure to comply with the configurations prevents users from logging onto the system. The password update configurations are:

- `passwd: files`
 - `passwd: files ldap`
 - `passwd: files nis`
 - `passwd: compat (==> files nis)`
 - `passwd: compat (==> files ldap)`
- `passwd_compat: ldap`

You can append the `ad` keyword to any of the `passwd` configurations in the above list. However, you cannot use the `passwd` command to change the password of an Active Directory (AD) user. If the `ad` keyword is found in the `passwd` entry during a password update operation, it is ignored.

Network administrators, who own the password table, can change any password attributes. The administrator configured for updating LDAP shadow information can also change any password attributes. See [ldapclient\(1M\)](#).

When a user has a password stored in one of the name services as well as a local `files` entry, the `passwd` command updates both. It is possible to have different passwords in the name service and local `files` entry. Use `passwd -r` to change a specific password repository.

The `passwd` command does not prompt authorized users for the old password.

If LDAP is in effect, an authorized user on any Native LDAP client system can change any password without being prompted for the old LDAP password.

By default, even users authorized to change the password of other users must comply with the configured password policy. See [pam_authok_check\(5\)](#).

Normally, `passwd` entered with no arguments changes the password of the current user. When a user logs in and then invokes [su\(1M\)](#) to become role or another user, `passwd` changes the original user's password, not the password of the role or the new user.

The `-s` argument is restricted to an authorized user.

The format of the display is:

name status mm/dd/yy min max warn

or, if password aging information is not present,

name status

where

name

The login ID of the user.

status

The password status of *name*.

The *status* field can take the following values:

LK

This account is locked account. See Security.

NL

This account is a no login account. See Security.

NP

This account has no password and is therefore open without authentication.

PS

This account has a password.

UN

The data in the password field is unknown. It is not a recognizable hashed password or any of the above entries. See [crypt\(3C\)](#) for valid password hashes.

UP

This account has not yet been activated by the administrator and cannot be used. See Security.

mm/dd/yy

The date password was last changed for *name*. All password aging dates are determined using Greenwich Mean Time (Universal Time) and therefore can differ by as much as a day in other time zones.

min

The minimum number of days required between password changes for *name*. MINWEEKS is found in `/etc/default/passwd` and is set to NULL.

max

The maximum number of days the password is valid for *name*. MAXWEEKS is found in `/etc/default/passwd` and is set to NULL.

warn

The number of days relative to *max* before the password expires and the *name* are warned.

Security

passwd uses [pam\(3PAM\)](#) for password change. It calls PAM with a service name `passwd` and uses service module type `auth` for authentication and `password` for password change.

Locking an account (`-l` option) does not allow its use for password based login or delayed execution (such as [at\(1\)](#), [batch\(1\)](#), or [cron\(1M\)](#)). The `-N` option can be used to disallow password based login, while continuing to allow delayed execution.

locked accounts that have never had a password and no `login` accounts cannot have their status changed directly to an active *password*. See `-d`. Changing a password on a locked account that had a password prior to being locked, changes the password without unlocking the account. See `-u` to unlock the account. An authorized administrator can activate an account in the not yet activated state by giving it a password.

An account can become locked following inactivity. To unlock such an account use the `-u` or `-f` options. With `-u`, the password is not changed; the use of `-f` forces a password change.

选项

The following options are supported:

- a
Shows password attributes for all entries. Use only with the `-s` option. *name* must not be provided. For the `files` and `ldap` repositories, this is restricted to the authorized user.
- e
Changes the login shell. A normal user can change his/her own shell information, an authorized user can change it for any user. The choice of shell is limited by the requirements of [getusershell\(3C\)](#). If the user currently has a shell that is not allowed by `getusershell`, an authorized user can change it.
- g
Changes the `gecos` (finger) information. A normal user can change their own `gecos` information, an authorized user can change it for any user.
- h
Changes the home directory.
- r
Specifies the repository to which an operation is applied. The supported repositories are `files`, `ldap`, or `nis`.
- s *name*
Shows password attributes for the login *name*. For the `files` and `ldap` repositories, this only works for the authorized user. It does not work at all for the `nis` repository, which does not support password aging.

The output of this option, and only this option, is Committed and parsable. The format is *username* followed by white space followed by one of the following codes.

New codes might be added in the future so code that parses this must be flexible in the face of unknown codes. While all existing codes are two characters in length that might not always be the case.

The following are the current status codes:

LK

The account is locked for UNIX authentication. `passwd -l` was run or the account was automatically locked due to the number of authentication failures reaching the configured maximum allowed. See `policy.conf(4)` and `user_attr(4)`.

NL

The account is a no login account. `passwd -N` has been run.

NP

Account has no password. `passwd -d` was run.

PS

The account probably has a valid password.

UN

The data in the password field is unknown. It is not a recognizable hashed password or any of the above entries. See `crypt(3C)` for valid password hashes.

UP

This account has not yet been activated by the administrator and cannot be used. See `Security`.

Authorized User Options

An administrator needs to be granted the User Security profile to be able to lock and unlock an existing account. That profile also provides the ability to activate a newly created account, set password aging options and view password attributes. The following lists shows the authorizations required to perform the various operations.

Only an authorized user can use the following options:

-d

Deletes password for *name* and unlocks the account. The login *name* is not prompted for password. It is only applicable to the `files` and `ldap` repositories.

If the `login(1)` option `PASSREQ=YES` is configured, the account is not able to login. `PASSREQ=YES` is the delivered default.

-f

Forces the user to change password at the next login by expiring the password for *name*. This option is useful for unlocking accounts that have become locked due to inactivity.

-l

Locks account for *name* unless it is already locked or is a no login account. See the `-d` or `-u` option for unlocking the account.

-N

Makes the password entry for *name* a value that cannot be used for login, but does not lock the account. See the `-d` option for removing the value, or to set a password to allow logins.

-n *min*

Sets minimum field for *name*. The *min* field contains the minimum number of days between password changes for *name*. If *min* is greater than *max*, the user can not change the password. Always use this option with the **-x** option, unless *max* is set to -1 (aging turned off). In that case, *min* need not be set.

-u

Unlocks a locked password for entry *name*. See the **-d** option for removing the locked password, or to set a password to allow logins. The **-u** option is useful for unlocking accounts that have become locked due to inactivity.

-w *warn*

Sets warn field for *name*. The *warn* field contains the number of days before the password expires and the user is warned. This option is not valid if password aging is disabled.

-x *max*

Sets maximum field for *name*. The *max* field contains the number of days that the password is valid for *name*. The aging for *name* is turned off immediately if *max* is set to -1 .

操作数

The following operand is supported:

name

User login name.

环境变量

If any of the LC_* variables, that is, LC_CTYPE, LC_MESSAGES, LC_TIME, LC_COLLATE, LC_NUMERIC, and LC_MONETARY (see [environ\(5\)](#)), are not set in the environment, the operational behavior of passwd for each corresponding locale category is determined by the value of the LANG environment variable. If LC_ALL is set, its contents are used to override both the LANG and the other LC_* variables. If none of the above variables is set in the environment, the C (U.S. style) locale determines how passwd behaves.

LC_CTYPE

Determines how passwd handles characters. When LC_CTYPE is set to a valid value, passwd can display and handle text and filenames containing valid characters for that locale. passwd can display and handle Extended Unix Code (EUC) characters where any individual character can be 1, 2, or 3 bytes wide. passwd can also handle EUC characters of 1, 2, or more column widths. In the C locale, only characters from ISO 8859-1 are valid.

LC_MESSAGES

Determines how diagnostic and informative messages are presented. This includes the language and style of the messages, and the correct form of affirmative and negative responses. In the C locale, the messages are presented in the default form found in the program itself (in most cases, U.S. English).

退出状态

The passwd command exits with one of the following values:

0

Success.

- 1 Permission denied.
- 2 Invalid combination of options.
- 3 Unexpected failure. Password file unchanged.
- 4 Unexpected failure. Password file(s) missing.
- 5 Password file(s) busy. Try again later.
- 6 Invalid argument to option.
- 7 Aging option is disabled.
- 8 No memory.
- 9 System error.
- 10 Account expired.
- 11 Password information unchanged.

文件

`/etc/default/passwd`

Default values can be set for the following flags in `/etc/default/passwd`. For example:

`MAXWEEKS=26`

DICTIONBDDIR

The directory where the generated dictionary databases reside. Defaults to `/var/passwd`.

If neither `DICTIONLIST` nor `DICTIONBDDIR` is specified, the system does not perform a dictionary check.

DICTIONLIST

`DICTIONLIST` can contain list of comma separated dictionary files such as `DICTIONLIST=file1, file2, file3`. Each dictionary file contains multiple lines and each line consists of a word and a NEWLINE character (similar to `/usr/share/lib/dict/words`.) You must specify full path names. The words from these files are merged into a database that is used to determine whether a password is based on a dictionary word.

If neither `DICTIONLIST` nor `DICTIONBDDIR` is specified, the system does not perform a dictionary check.

To pre-build the dictionary database, see `mkpwdict(1M)`.

HISTORY

Maximum number of prior password history to keep for a user. Setting the `HISTORY` value to zero (0), or removing the flag, causes the prior password history of all users to be discarded at the next password change by any user. The default is not to define the `HISTORY` flag. The maximum value is 26. Currently, this functionality is enforced only for user accounts defined in the `files` name service (`local passwd(4)/shadow(4)`).

MAXREPEATS

Maximum number of allowable consecutive repeating characters. If `MAXREPEATS` is not set or is zero (0), the default is no checks

MAXWEEKS

Maximum time period that password is valid.

MINALPHA

Minimum number of alpha character required. If `MINALPHA` is not set, the default is 2.

MINDIFF

Minimum differences required between an old and a new password. If `MINDIFF` is not set, the default is 3.

MINDIGIT

Minimum number of digits required. If `MINDIGIT` is not set or is set to zero (0), the default is no checks. You cannot specify `MINDIGIT` if `MINNONALPHA` is also specified.

MINLOWER

Minimum number of lower case letters required. If not set or zero (0), the default is no checks.

MINNONALPHA

Minimum number of non-alpha (including numeric and special) required. If `MINNONALPHA` is not set, the default is 1. You cannot specify `MINNONALPHA` if `MINDIGIT` or `MINSPECIAL` is also specified.

MINWEEKS

Minimum time period before the password can be changed.

MINSPECIAL

Minimum number of special (non-alpha and non-digit) characters required. If `MINSPECIAL` is not set or is zero (0), the default is no checks. You cannot specify `MINSPECIAL` if you also specify `MINNONALPHA`.

MINUPPER

Minimum number of upper case letters required. If `MINUPPER` is not set or is zero (0), the default is no checks.

NAMECHECK

Enable/disable checking of the login name. The default is to do login name checking. A case insensitive value of no disables this feature.

PASSLENGTH

Minimum length of password, in characters.

WARNWEEKS

Time period until warning of date of password's ensuing expiration.

WHITESPACE

Determine if white space characters are allowed in passwords. Valid values are YES and NO. If WHITESPACE is not set or is set to YES, white space characters are allowed.

/etc/oshadow

Temporary file used by passwd and pwconv to update the real shadow file.

/etc/passwd

Password file.

/etc/shadow

Shadow password file.

/etc/shells

Shell database.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	See below.

The human readable output is Uncommitted. The options are Committed.

另请参见

[at\(1\)](#), [batch\(1\)](#), [finger\(1\)](#), [login\(1\)](#), [cron\(1M\)](#), [domainname\(1M\)](#), [eeprom\(1M\)](#), [id\(1M\)](#), [ldapclient\(1M\)](#), [mkpwdict\(1M\)](#), [pwconv\(1M\)](#), [su\(1M\)](#), [useradd\(1M\)](#), [userdel\(1M\)](#), [usermod\(1M\)](#), [crypt\(3C\)](#), [getpwnam\(3C\)](#), [getspnam\(3C\)](#), [getusershell\(3C\)](#), [pam\(3PAM\)](#), [loginlog\(4\)](#), [nsswitch.conf\(4\)](#), [pam.conf\(4\)](#), [passwd\(4\)](#), [policy.conf\(4\)](#), [shadow\(4\)](#), [shells\(4\)](#), [user_attr\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [pam_authok_check\(5\)](#), [pam_authok_get\(5\)](#), [pam_authok_store\(5\)](#), [pam_dhkeys\(5\)](#), [pam_ldap\(5\)](#), [pam_unix_account\(5\)](#), [pam_unix_auth\(5\)](#), [pam_unix_session\(5\)](#), [crypt_unix\(5\)](#)

附注

The `yppasswd` command is a wrapper around `passwd`. Use of `yppasswd` is discouraged. Use `passwd -r repository_name` instead.

Changing a password in the `files` and `ldap` repositories clears the failed login count.

Changing a password reactivates an account deactivated for inactivity for the length of the inactivity period.

Input terminal processing might interpret some key sequences and not pass them to the `passwd` command.

An account with no password, status code `NP`, might not be able to login. See the [login\(1\)](#) `PASSREQ` option.

Authorizations required to perform various options:

<code>-d</code>	delete password	<code>solaris.passwd.assign</code>
<code>-N</code>	set nologin	<code>solaris.passwd.assign</code>
	change any passwd	<code>solaris.passwd.assign</code>
<code>-l</code>	lock account	<code>solaris.account.setpolicy</code>
<code>-u</code>	unlock account	<code>solaris.account.setpolicy</code>
<code>-n</code>	set min field for name	<code>solaris.account.setpolicy</code>
<code>-w</code>	set warn field for name	<code>solaris.account.setpolicy</code>
<code>-x</code>	set max field for name	<code>solaris.account.setpolicy</code>
<code>-f</code>	forces password expiration	<code>solaris.account.setpolicy</code>
<code>-s</code>	display password attributes	<code>solaris.account.setpolicy</code>
<code>-a</code>	display password attributes for all entries	<code>solaris.account.setpolicy</code>
<code>-e</code>	change login shell	<code>solaris.user.manage</code>
<code>-g</code>	change geccos information	<code>solaris.user.manage</code>
<code>-h</code>	change home directory	<code>solaris.user.manage</code>
	set a newly created account's passwd for the first time	<code>solaris.account.activate</code>

All password hash algorithms except `crypt_unix(5)` have a maximum password length of 255.

引用名 paste – merge corresponding or subsequent lines of files

用法概要 /usr/bin/paste [*options*] [*file...*]

描述 The paste utility concatenates the corresponding lines of the given input files, and write the resulting lines to standard output.

The default operation of paste concatenates the corresponding lines of the input files. The NEWLINE character of every line except the line from the last input file is replaced with a TAB character.

If an EOF (end-of-file) condition is detected on one or more input files, but not all input files, paste behaves as though empty lines were read from the files on which EOF was detected, unless the -s option is specified.

选项 The following options are supported:

-d *list* Unless a backslash character (\) appears in list, each character in list is an element specifying a delimiter character. If a backslash character appears in list, the backslash character and one or more characters following it are an element specifying a delimiter character as described below. These elements specify one or more delimiters to use, instead of the default TAB character, to replace the NEWLINE character of the input lines. The elements in list are used circularly. That is, when the list is exhausted, the first element from the list is reused.

When the -s option is specified:

- The last NEWLINE character in a file is not modified.
- The delimiter is reset to the first element of list after each file operand is processed.

When the option is not specified:

- The NEWLINE characters in the file specified by the last file is not modified.
- The delimiter is reset to the first element of list each time a line is processed from each file.

If a backslash character appears in list, it and the character following it is used to represent the following delimiter characters:

- \n NEWLINE character.
- \t TAB character.
- \\ Backslash character.
- \0 Empty string (not a null character). If 0 is immediately followed by the character x, the character X, or any character defined by the LC_CTYPE digit keyword, the results are unspecified.

If any other characters follow the backslash, the results are unspecified.

- s Concatenate all of the lines of each separate input file in command line order. The NEWLINE character of every line except the last line in each input file is replaced with the TAB character, unless otherwise specified by the -d option.

操作数

The following operand is supported:

- file* A path name of an input file. If is specified for one or more of the files, the standard input is used. The standard input is read one line at a time, circularly, for each instance of dot .. Implementations support pasting of at least 12 file operands.

用法

See [largefile\(5\)](#) for the description of the behavior of `paste` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 Listing a Directory in One Column

The following example lists a directory in one column:

```
example% ls | paste -d" " -
```

示例 2 Listing a Directory in Four Columns

The following example lists a directory in four columns:

```
example% ls | paste - - - -
```

示例 3 Combining Pairs of Lines from a File into Single Lines

The following example combines pairs of lines from a file into single lines:

```
example% paste -s -d"\ t\ n" file
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `paste`: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Standard	See standards(5) .

另请参见

[cut\(1\)](#), [grep\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

引用名	patch – apply changes to files
用法概要	<pre>patch [-bLNR] [-c -e -n -u] [-d <i>dir</i>] [-D <i>define</i>] [-i <i>patchfile</i>] [-o <i>outfile</i>] [-p <i>num</i>] [-r <i>rejectfile</i>] [<i>file</i>]</pre>
描述	<p>The <code>patch</code> command reads a source (patch) file containing any of the three forms of difference (diff) listings produced by the <code>diff(1)</code> command (normal, context or in the style of <code>ed(1)</code>) and apply those differences to a file. By default, <code>patch</code> reads from the standard input.</p> <p><code>patch</code> attempts to determine the type of the <code>diff</code> listing, unless overruled by a <code>-c</code>, <code>-e</code>, or <code>-n</code> option.</p> <p>If the patch file contains more than one patch, <code>patch</code> attempts to apply each of them as if they came from separate patch files. (In this case the name of the patch file must be determinable for each <code>diff</code> listing.)</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -b Saves a copy of the original contents of each modified file, before the differences are applied, in a file of the same name with the suffix <code>.orig</code> appended to it. If the file already exists, it is overwritten. If multiple patches are applied to the same file, the <code>.orig</code> file is written only for the first patch. When the <code>-o <i>outfile</i></code> option is also specified, <code>file.orig</code> is not created but, if <code>outfile</code> already exists, <code>outfile.orig</code> is created. -c Interprets the patch file as a context difference (the output of the command <code>diff</code> when the <code>-c</code> or <code>-C</code> options are specified). -d <i>dir</i> Changes the current directory to <i>dir</i> before processing as described in EXTENDED DESCRIPTION. -D <i>define</i> Marks changes with the C preprocessor construct: <pre> #ifdef <i>define</i> . . . #endif</pre> <p>The option-argument <i>define</i> is used as the differentiating symbol.</p> -e Interprets the patch file as an <code>ed</code> script, rather than a <code>diff</code> script. -i <i>patchfile</i> Reads the patch information from the file named by the path name <i>patchfile</i>, rather than the standard input. -l (The letter ell.) Causes any sequence of blank characters in the difference script to match any sequence of blank characters in the input file. Other characters is matched exactly. -n Interprets the script as a normal difference.

- N** Ignores patches where the differences have already been applied to the file; by default, already-applied patches are rejected.
- o *outfile*** Instead of modifying the files (specified by the *file* operand or the difference listings) directly, writes a copy of the file referenced by each patch, with the appropriate differences applied, to *outfile*. Multiple patches for a single file is applied to the intermediate versions of the file created by any previous patches, and results in multiple, concatenated versions of the file being written to *outfile*.
- p *num*** For all path names in the patch file that indicate the names of files to be patched, deletes *num* path name components from the beginning of each path name. If the path name in the patch file is absolute, any leading slashes are considered the first component (that is, **-p 1** removes the leading slashes). Specifying **-p 0** causes the full path name to be used. If **-p** is not specified, only the basename (the final path name component) is used.
- R** Reverses the sense of the patch script. That is, assumes that the difference script was created from the new version to the old version. The **-R** option cannot be used with **ed** scripts. **patch** attempts to reverse each portion of the script before applying it. Rejected differences is saved in swapped format. If this option is not specified, and until a portion of the patch file is successfully applied, **patch** attempts to apply each portion in its reversed sense as well as in its normal sense. If the attempt is successful, the user is prompted to determine if the **-R** option should be set.
- r *rejectfile*** Overrides the default reject file name. In the default case, the reject file has the same name as the output file, with the suffix **.rej** appended to it. See **Patch Application**.
- u** Interprets the patch file as a unified context difference, that is, the output of the command **diff** when the **-u** or **-U** options are specified.

操作数

The following operand is supported:

file A path name of a file to patch.

用法

The **-R** option does not work with **ed** scripts because there is too little information to reconstruct the reverse operation.

The **-p** option makes it possible to customize a patch file to local user directory structures without manually editing the patch file. For example, if the file name in the patch file was **/curds/whey/src/blurfl/blurfl.c**:

- Setting **-p 0** gives the entire path name unmodified.
- Setting **-p 1** gives:

```
curds/whey/src/blurfl/blurfl.c
```

- Without the leading slash, `-p 4` gives:

```
blurfl/blurfl.c
```

- Not specifying `-p` at all gives:

```
blurfl.c
```

When using `-b` in some file system implementations, the saving of a `.orig` file might produce unwanted results. In the case of 12-, 13-, or 14-character file names, on file systems supporting 14-character maximum file names, the `.orig` file overwrites the new file.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `patch`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, and `NLSPATH`.

Affirmative responses are processed using the extended regular expression defined for the `yesexpr` keyword in the `LC_MESSAGES` category of the user's locale. The locale specified in the `LC_COLLATE` category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for `yesexpr`. The locale specified in `LC_CTYPE` determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the `yesexpr`. See [locale\(5\)](#).

Output Files

The output of `patch` the save files (`.orig` suffixes) and the reject files (`.rej` suffixes) are text files.

扩展描述

A patch file can contain patching instructions for more than one file. File names are determined as specified in [Patch Determination](#). When the `-b` option is specified, for each patched file, the original is saved in a file of the same name with the suffix `.orig` appended to it.

For each patched file, a reject file can also be created as noted in [Patch Application](#). In the absence of an `-r` option, the name of this file is formed by appending the suffix `.rej` to the original file name.

Patch File Format

The patch file must contain zero or more lines of header information followed by one or more patches. Each patch must contain zero or more lines of file name identification in the format produced by `diff -c`, and one or more sets of `diff` output, which are customarily called hunks.

`patch` recognizes the following expression in the header information:

```
Index: pathname    The file to be patched is named pathname.
```

If all lines (including headers) within a patch begin with the same leading sequence of blank characters, `patch` removes this sequence before proceeding. Within each patch, if the type of difference is context, `patch` recognizes the following expressions:

- * * * *filename timestamp* The patches arose from *filename*.
- - - *filename timestamp* The patches should be applied to *filename*.

Each hunk within a patch must be the `diff` output to change a line range within the original file. The line numbers for successive hunks within a patch must occur in ascending order.

File Name Determination

If no *file* operand is specified, `patch` performs the following steps to obtain a path name:

1. If the patch contains the strings `***` and `- -`, `patch` strips components from the beginning of each path name (depending on the presence or value of the `-p` option), then tests for the existence of both files in the current directory (or directory specified with the `-d` option).
2. If both files exist, `patch` assumes that no path name can be obtained from this step. If the header information contains a line with the string `Index:`, `patch` strips components from the beginning of the path name (depending on `-p`), then tests for the existence of this file in the current directory (or directory specified with the `-d` option).
3. If an SCCS directory exists in the current directory, `patch` attempts to perform a `get -e SCCS/s .filename` command to retrieve an editable version of the file.
4. If no path name can be obtained by applying the previous steps, or if the path names obtained do not exist, `patch` writes a prompt to standard output and request a file name interactively from standard input.

Patch Application

If the `-c`, `-e`, `-n`, or `-u` option is present, `patch` interprets information within each hunk as a context difference, an `ed` difference, a normal difference, or a unified context difference, respectively. In the absence of any of these options, `patch` determines the type of difference based on the format of information within the hunk.

For each hunk, `patch` begins to search for the place to apply the patch at the line number at the beginning of the hunk, plus or minus any offset used in applying the previous hunk. If lines matching the hunk context are not found, `patch` scans both forwards and backwards at least 1000 bytes for a set of lines that match the hunk context.

If no such place is found and it is a context difference, then another scan takes place, ignoring the first and last line of context. If that fails, the first two and last two lines of context is ignored and another scan is made. Implementations can search more extensively for installation locations.

If no location can be found, `patch` appends the hunk to the reject file. The rejected hunk is written in context-difference format regardless of the format of the patch file. If the input was a normal or `ed`-style difference, the reject file can contain differences with zero lines of context. The line numbers on the hunks in the reject file can be different from the line numbers in the patch file since they reflect the approximate locations for the failed hunks in the new file rather than the old one.

If the type of patch is an ed diff, the implementation can accomplish the patching by invoking the ed command.

退出状态

The following exit values are returned:

- 0 Successful completion.
- 1 One or more lines were written to a reject file.
- >1 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[ed\(1\)](#), [diff\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名 pathchk – check path names

用法概要 pathchk [-p] *path*...

描述 The pathchk command will check that one or more path names are valid (that is, they could be used to access or create a file without causing syntax errors) and portable (that is, no filename truncation will result). More extensive portability checks are provided by the -p option.

By default, pathchk will check each component of each *path* operand based on the underlying file system. A diagnostic will be written for each *path* operand that:

- is longer than PATH_MAX bytes.
- contains any component longer than NAME_MAX bytes in its containing directory
- contains any component in a directory that is not searchable
- contains any character in any component that is not valid in its containing directory.

The format of the diagnostic message is not specified, but will indicate the error detected and the corresponding *path* operand.

It will not be considered an error if one or more components of a *path* operand do not exist as long as a file matching the path name specified by the missing components could be created that does not violate any of the checks specified above.

选项 The following option is supported:

- p Instead of performing checks based on the underlying file system, write a diagnostic for each *path* operand that:
 - is longer than _POSIX_PATH_MAX bytes
 - contains any component longer than _POSIX_NAME_MAX bytes
 - contains any character in any component that is not in the portable filename character set.

操作数 The following operand is supported:

path A path to be checked.

用法 See [largefile\(5\)](#) for the description of the behavior of pathchk when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例 示例 1 Using the pathchk command

To verify that all paths in an imported data interchange archive are legitimate and unambiguous on the current system:

```
example% pax -f archive | sed -e '/ == ./s///' | xargs pathchk
if [ $? -eq 0 ]
then
```

示例1 Using the pathchk command (续)

```

        pax -r -f archive
else
    echo Investigate problems before importing files.
    exit 1
fi

```

To verify that all files in the current directory hierarchy could be moved to any system conforming to the X/Open specification that also supports the `pax(1)` command:

```

example% find . -print | xargs pathchk -p
if [ $? -eq 0 ]
then
    pax -w -f archive .
else
    echo Portable archive cannot be created.
    exit 1
fi

```

To verify that a user-supplied path names a readable file and that the application can create a file extending the given path without truncation and without overwriting any existing file:

```

example% case $- in
    *(C*)   reset="";;
    *)     reset="set +C"
           set -C;;
esac
test -r "$path" && pathchk "$path.out" &&
rm "$path.out" > "$path.out"
if [ $? -ne 0 ]; then
    printf "%s: %s not found or %s.out fails \
creation checks.\n" $0 "$path" "$path"
    $reset # reset the noclobber option in case a trap
           # on EXIT depends on it
    exit 1
fi
$reset
PROCESSING < "$path" > "$path.out"

```

The following assumptions are made in this example:

1. PROCESSING represents the code that will be used by the application to use `$path` once it is verified that `$path.out` will work as intended.
2. The state of the `noclobber` option is unknown when this code is invoked and should be set on exit to the state it was in when this code was invoked. (The `reset` variable is used in this example to restore the initial state.)

示例 1 Using the pathchk command (续)

3. Note the usage of:

```
rm "$path.out" > "$path.out"
```

- a. The pathchk command has already verified, at this point, that `$path.out` will not be truncated.
- b. With the `noclobber` option set, the shell will verify that `$path.out` does not already exist before invoking `rm`.
- c. If the shell succeeded in creating `$path.out`, `rm` will remove it so that the application can create the file again in the PROCESSING step.
- d. If the PROCESSING step wants the file to exist already when it is invoked, the:

```
rm "$path.out" > "$path.out"
```

should be replaced with:

```
> "$path.out"
```

which will verify that the file did not already exist, but leave `$path.out` in place for use by PROCESSING.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of pathchk: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 All *path* operands passed all of the checks.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[pax\(1\)](#), [test\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

引用名 pax – portable archive interchange

用法概要

```
pax [-cdnv] [-H | -L] [-f archive] [-o options]...
    [-s replstr]... [pattern]...

pax -r [-cdiknuv@/] [-H | -L] [-f archive] [-o options]...
    [-p string]... [-s replstr]... [pattern]...

pax -w [-dituvX@/] [-H | -L] [-b blocksize] [-a]
    [-f archive] [-o options]... [-s replstr]...
    [-x format] [file]...

pax -r -w [-diklntuvX@/] [-H | -L] [-o options]...
    [-p string]... [-s replstr]... [file]... directory
```

描述 pax reads, writes, and writes lists of the members of archive files and copies directory hierarchies. A variety of archive formats are supported. See the `-x format` option.

Modes of Operations The action to be taken depends on the presence of the `-r` and `-w` options. The four combinations of `-r` and `-w` are referred to as the four modes of operation: `list`, `read`, `write`, and `copy` modes, corresponding respectively to the four forms shown in the SYNOPSIS.

list In `list` mode, that is, when neither `-r` nor `-w` are specified, pax writes the names of the members of the archive file read from the standard input, with path names matching the specified patterns, to standard output. If a named file has extended attributes, the extended attributes are also listed. If a named file is of type directory, the file hierarchy rooted at that file is listed as well.

read In `read` mode, that is, when `-r` is specified, but `-w` is not, pax extracts the members of the archive file read from the standard input, with path names matching the specified patterns. If an extracted file is of type directory, the file hierarchy rooted at that file is extracted as well. The extracted files are created performing path name resolution with the directory in which pax was invoked as the current working directory.

If an attempt is made to extract a directory when the directory already exists, this is not considered an error. If an attempt is made to extract a FIFO when the FIFO already exists, this is not considered an error.

The ownership, access and modification times, and file mode of the restored files are discussed under the `-p` option.

write In `write` mode, that is, when `-w` is specified, but `-r` is not, pax writes the contents of the *file* operands to the standard output in an archive format. If no *file* operands are specified, a list of files to copy, one per line, are read from the standard input. A file of type directory includes all of the files in the file hierarchy rooted at the file.

copy In `copy` mode, that is, when both `-r` and `-w` are specified, pax copies the *file* operands to the destination directory.

If no *file* operands are specified, a list of files to copy, one per line, are read from the standard input. A file of type directory includes all of the files in the file hierarchy rooted at the file.

The effect of the copy is as if the copied files were written to an archive file and then subsequently extracted, except that there can be hard links between the original and the copied files. If the destination directory is a subdirectory of one of the files to be copied, the results are unspecified. It is an error if *directory* does not exist, is not writable by the user, or is not a directory.

In read or copy modes, if intermediate directories are necessary to extract an archive member, pax performs actions equivalent to the `mkdir(2)` function, called with the following arguments:

- The intermediate directory used as the *path* argument.
- The octal value of 777 or *rxw* (read, write, and execute permissions) as the *mode* argument (see `chmod(1)`).

If any specified *pattern* or *file* operands are not matched by at least one file or archive member, pax writes a diagnostic message to standard error for each one that did not match and exits with a non-zero exit status.

The supported archive formats are automatically detected on input. The default output archive format is `tar(1)`.

A single archive can span multiple files. pax determines what file to read or write as the next file.

If the selected archive format supports the specification of linked files, it is an error if these files cannot be linked when the archive is extracted, except if the files to be linked are symbolic links and the system is not capable of making hard links to symbolic links. In that case, separate copies of the symbolic link are created instead. Any of the various names in the archive that represent a file can be used to select the file for extraction. For archive formats that do not store file contents with each name that causes a hard link, if the file that contains the data is not extracted during this pax session, either the data is restored from the original file, or a diagnostic message is displayed with the name of a file that can be used to extract the data. In traversing directories, pax detects infinite loops, that is, entering a previously visited directory that is an ancestor of the last file visited. When it detects an infinite loop, pax writes a diagnostic message to standard error and terminates.

选项

The following options are supported:

- a Appends files to the end of the archive. This option does not work for some archive devices, such as 1/4-inch streaming tapes and 8mm tapes.

-
- b *blocksize* Blocks the output at a positive decimal integer number of bytes per write to the archive file. Devices and archive formats can impose restrictions on blocking. Blocking is automatically determined on input. Portable applications must not specify a *blocksize* value larger than 32256. Default blocking when creating archives depends on the archive format. See the -x option below.
- c Matches all file or archive members except those specified by the *pattern* or *file* operands.
- d Causes files of type directory being copied or archived or archive members of type directory being extracted or listed to match only the file or archive member itself and not the file hierarchy rooted at the file.
- f *archive* Specifies the path name of the input or output archive, overriding the default standard input (in `list` or `read` modes) or standard output (`write` mode).
- H If a symbolic link referencing a file of type directory is specified on the command line, `pax` archives the file hierarchy rooted in the file referenced by the link, using the name of the link as the root of the file hierarchy. Otherwise, if a symbolic link referencing a file of any other file type which `pax` can normally archive is specified on the command line, then `pax` archives the file referenced by the link, using the name of the link. The default behavior is to archive the symbolic link itself.
- i Interactively renames files or archive members. For each archive member matching a *pattern* operand or file matching a *file* operand, a prompt is written to the file `/dev/tty`. The prompt contains the name of the file or archive member. A line is then read from `/dev/tty`. If this line is blank, the file or archive member is skipped. If this line consists of a single period, the file or archive member is processed with no modification to its name. Otherwise, its name is replaced with the contents of the line. `pax` immediately exits with a non-zero exit status if end-of-file is encountered when reading a response or if `/dev/tty` cannot be opened for reading and writing.
- The results of extracting a hard link to a file that has been renamed during extraction are unspecified.
- k Prevents the overwriting of existing files.
- l Links files. In copy mode, hard links are made between the source and destination file hierarchies whenever possible. If specified in conjunction with -H or -L, when a symbolic link is encountered, the hard link created in the destination file hierarchy is to the file referenced by the symbolic link. If specified when neither -H nor -L is specified, when a symbolic link is encountered, the implementation creates a hard link to the symbolic link in the source file hierarchy or copies the symbolic link to the destination.

- L If a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, pax archives the file hierarchy rooted in the file referenced by the link, using the name of the link as the root of the file hierarchy. Otherwise, if a symbolic link referencing a file of any other file type which pax can normally archive is specified on the command line or encountered during the traversal of a file hierarchy, pax archives the file referenced by the link, using the name of the link. The default behavior is to archive the symbolic link itself.
- n Selects the first archive member that matches each *pattern* operand. No more than one archive member is matched for each pattern, although members of type directory still match the file hierarchy rooted at that file.
- o *options* Provides information to the implementation to modify the algorithm for extracting or writing files. The value of options consists of one or more comma-separated keywords of the form:
- ```
keyword[[:]=value][,keyword[[:]=value], ...]
```
- Some keywords apply only to certain file formats, as indicated with each description. Use of keywords that are inapplicable to the file format being processed produces undefined results.
- Keywords in the *options* argument must be a string that would be a valid portable filename.
- Keywords are not expected to be filenames, merely to follow the same character composition rules as portable filenames.
- Keywords can be preceded with white space. The *value* field consists of zero or more characters. Within *value*, the application precedes any literal comma with a backslash, which is ignored, but preserves the comma as part of *value*. A comma as the final character, or a comma followed solely by white space as the final characters, in *options* is ignored. Multiple -o options can be specified. If keywords given to these multiple -o options conflict, the keywords and values appearing later in command line sequence take precedence and the earlier ones are silently ignored. The following keyword values of *options* are supported for the file formats as indicated:
- `delete=pattern`
- This keyword is applicable only to the -x pax format. When used in write or copy mode, pax omits from extended header records that it produces any keywords matching the string pattern. When used in read or list mode, pax ignores any keywords matching the string pattern in the extended header records. In both cases, matching is performed using the pattern matching notation. For example:
- ```
-o delete=security.*
```


would suppress security-related information.

When multiple `-o delete=pattern` options are specified, the patterns are additive. All keywords matching the specified string patterns are omitted from extended header records that `pax` produces.

`exthdr.name=string`

This keyword is applicable only to the `-x pax` format. This keyword allows user control over the name that is written into the `ustar` header blocks for the extended header. The name is the contents of *string*, after the following character substitutions have been made:

- `%d` The directory name of the file, equivalent to the result of the *dirname* utility on the translated path name.
- `%f` The filename of the file, equivalent to the result of the *basename* utility on the translated path name.
- `%p` The process ID of the `pax` process.
- `%%` A `'%'` character.

Any other `'%'` characters in *string* produce undefined results.

If no `-o exthdr.name=string` is specified, `pax` uses the following default value:

```
%d/PaxHeaders.%p/%f
```

`globexthdr.name=string`

This keyword is applicable only to the `-x pax` format. When used in `write` or `copy` mode with the appropriate options, `pax` creates global extended header records with `ustar` header blocks that are treated as regular files by previous versions of `pax`. This keyword allows user control over the name that is written into the `ustar` header blocks for global extended header records. The name is the contents of *string*, after the following character substitutions have been made:

- `%n` An integer that represents the sequence number of the global extended header record in the archive, starting at 1.
- `%p` The process ID of the `pax` process.
- `%%` A `'%'` character.

Any other `'%'` characters in *string* produce undefined results.

If no `-o globexthdr.name=string` is specified, `pax` uses the following default value:

```
$TMPDIR/GlobalHead.%p.%n
```

where `$TMPDIR` represents the value of the `TMPDIR` environment variable. If `TMPDIR` is not set, `pax` uses `/tmp`.

`invalid=action`

This keyword is applicable only to the `-x pax` format. This keyword allows user control over the action `pax` takes upon encountering values in an extended header record that, in `read` or `copy` mode, are invalid in the destination hierarchy or, in `list` mode, cannot be written in the codeset and current locale of the implementation. The following are invalid values that are recognized by `pax`:

- In `read` or `copy` mode, a filename or link name that contains character encodings invalid in the destination hierarchy. For example, the name can contain embedded NULs.
- In `read` or `copy` mode, a filename or link name that is longer than the maximum allowed in the destination hierarchy, for either a path name component or the entire path name.
- In `list` mode, any character string value (filename, link name, user name, and so on) that cannot be written in the codeset and current locale of the implementation.

The following mutually-exclusive values of the `action` argument are supported:

- | | |
|---------------------|---|
| <code>bypass</code> | In <code>read</code> or <code>copy</code> mode, <code>pax</code> bypasses the file, causing no change to the destination hierarchy. In <code>list</code> mode, <code>pax</code> writes all requested valid values for the file, but its method for writing invalid values is unspecified. |
| <code>rename</code> | In <code>read</code> or <code>copy</code> mode, <code>pax</code> acts as if the <code>-i</code> option were in effect for each file with invalid filename or link name values, allowing the user to provide a replacement name interactively. In <code>list</code> mode, <code>pax</code> behaves identically to the <code>bypass</code> action. |
| <code>UTF-8</code> | <code>pax</code> uses the actual UTF-8 encoding for the name when it is used in <code>read</code> , <code>copy</code> , or <code>list</code> mode and a filename, link name, owner name, or any other field in an extended header record cannot be translated from the <code>pax UTF-8</code> codeset format to the codeset and current locale of the implementation. |
| <code>write</code> | In <code>read</code> or <code>copy</code> mode, <code>pax</code> writes the file, translating the name, regardless of whether this can overwrite an existing file with a valid name. In <code>list</code> mode, <code>pax</code> behaves identically to the <code>bypass</code> action. |

If no `-o invalid=` option is specified, `pax` acts as if `-o invalid=bypass` were specified. Any overwriting of existing files that can be allowed by the

-o *invalid=actions* are subject to permission (-p) and modification time (-u) restrictions, and are suppressed if the -k option is also specified.

linkdata

This keyword is applicable only to the -x pax format. In write mode, pax writes the contents of a file to the archive even when that file is merely a hard link to a file whose contents have already been written to the archive.

listopt=*format*

This keyword specifies the output format of the table of contents produced when the -v option is specified in list mode. (See List Mode Format Specifications below.) To avoid ambiguity, the listopt=*format* is the only or final *keyword=value* pair in an -o option-argument. All characters in the remainder of the option-argument are considered to be part of the format string. When multiple -o listopt=*format* options are specified, the format strings are considered to be a single, concatenated string, evaluated in command line order.

times

This keyword is applicable only to the -x pax and -x xustar formats. When used in write or copy mode, pax includes *atime* and *mtime* extended header records for each file.

In addition to these keywords, if the -x pax format is specified, any of the keywords and values, including implementation extensions, can be used in -o option-arguments, in either of two modes:

keyword=value When used in write or copy mode, these keyword/value pairs are included at the beginning of the archive as `typeflag g` global extended header records. When used in read or list mode, these keyword/value pairs act as if they had been at the beginning of the archive as `typeflag g` global extended header records.

keyword:=value When used in write or copy mode, these keyword/value pairs are included as records at the beginning of a `typeflag x` extended header for each file. This is equivalent to the equal-sign form except that it creates no `typeflag g` global extended header records. When used in read or list mode, these keyword/value pairs act as if they were included as records at the end of each extended header. Thus, they override any global or file-specific extended header record keywords of the same names. For example, in the command:

```
pax -r -o "
gname:=mygroup,
" <archive
```

the group name is forced to a new value for all files read from the archive.

-p *string* Specifies one or more file characteristic options (privileges). The *string* option-argument must be a string specifying file characteristics to be retained or discarded on extraction. The string consists of the specification characters a, e, m, o, and p. Multiple characteristics can be concatenated within the same string and multiple -p options can be specified. The meaning of the specification characters is as follows:

- a Does not preserve file access times.
- e Preserves the user ID, group ID, file mode bits, access time, and modification time.
- m Does not preserve file modification times.
- o Preserves the user ID and group ID.
- p Preserves the file mode bits.

In the preceding list, preserve indicates that an attribute stored in the archive is given to the extracted file, subject to the permissions of the invoking process. Otherwise, the attribute is determined as part of the normal file creation action. The access and modification times of the file is preserved unless otherwise specified with the -p option or not stored in the archive. All attributes that are not preserved are determined as part of the normal file creation action.

If neither the e nor the o specification character is specified, or the user ID and group ID are not preserved for any reason, pax does not set the `setuid` and `setgid` bits of the file mode.

If the preservation of any of these items fails for any reason, pax writes a diagnostic message to standard error. Failure to preserve these items affects the final exit status, but does not cause the extracted file to be deleted.

If file-characteristic letters in any of the *string* option-arguments are duplicated or conflict with each other, the ones given last take precedence. For example, if -p eme is specified, file modification times are preserved.

-r Reads an archive file from standard input.

-s *replstr* Modifies file or archive member names named by *pattern* or *file* operands according to the substitution expression *replstr*, which is based on the `ed(1)` s (substitution) utility, using the regular expression syntax of `regex(5)`. The concepts of "address" and "line" are meaningless in the context of the pax command, and must not be supplied. The format is:

`-s /old/new/ [gp]`

where, as in `ed`, *old* is a basic regular expression and *new* can contain an ampersand (&), a `\n` backreference, where *n* is a digit, or subexpression matching. The *old* string is also permitted to contain newlines.

Any non-null character can be used as a delimiter (/ shown here). Multiple `-s` expressions can be specified. The expressions are applied in the order specified, terminating with the first successful substitution. The optional trailing `g` is as defined in the `ed` command. The optional trailing `p` causes successful substitutions to be written to standard error. File or archive member names that substitute to the empty string are ignored when reading and writing archives.

`-t` When reading files from the file system, and if the user has the permissions required by `utime()` to do so, sets the access time of each file read to the access time that it had before being read by `pax`.

`-u` Ignores files that are older (having a less recent file modification time) than a pre-existing file or archive member with the same name.

`read mode` An archive member with the same name as a file in the file system is extracted if the archive member is newer than the file.

`write mode` An archive file member with the same name as a file in the file system is superseded if the file is newer than the archive member. If option `-a` is also specified, this is accomplished by appending to the archive. Otherwise, it is unspecified whether this is accomplished by actual replacement in the archive or by appending to the archive.

`copy mode` The file in the destination hierarchy is replaced by the file in the source hierarchy or by a link to the file in the source hierarchy if the file in the source hierarchy is newer.

`-v` In `list` mode, produces a verbose table of contents (see `Standard Output`). Otherwise, writes archive member path names and extended attributes to standard error (see `Standard Error`).

`-w` Writes files to the standard output in the specified archive format.

`-x format` Specifies the output archive format. The `pax` utility recognizes the following formats:

`cpio` The extended `cpio(1)` interchange format. See IEEE Std 1003.1–2001. The default *blocksize* for this format for character special archive files is 5120. Implementations support all *blocksize* values less than or equal to 32256 that are multiples of 512.

This archive format allows files with UIDs and GIDs up to 262143 to be stored in the archive. Files with UIDs and GIDs greater than this value are archived with the UID and GID of 60001.

pax The pax interchange format. See IEEE Std 1003.1–2001. The default *blocksize* for this format for character special archive files is 5120. Implementations support all *blocksize* values less than or equal to 32256 that are multiples of 512.

Similar to *ustar*. Also allows archiving and extracting files whose size is greater than 8GB; whose UID, GID, *devmajor*, or *devminor* values are greater than 2097151; whose path (including filename) is greater than 255 characters; or whose *linkname* is greater than 100 characters.

ustar The extended *tar(1)* interchange format. See the IEEE 1003.1(1990) specifications. The default *blocksize* for this format for character special archive files is 10240. Implementations support all *blocksize* values less than or equal to 32256 that are multiples of 512.

This archive format allows files with UIDs and GIDs up to 2097151 to be stored in the archive. Files with UIDs and GIDs greater than this value are archived with the UID and GID of 60001.

xustar Similar to *ustar*. Also allows archiving and extracting files whose size is greater than 8GB; whose UID, GID, *devmajor*, or *devminor* values are greater than 2097151; whose path (including filename) is greater than 255 characters; or whose *linkname* is greater than 100 characters. This option should not be used if the archive is to be extracted by an archiver that cannot handle the larger values.

Any attempt to append to an archive file in a format different from the existing archive format causes *pax* to exit immediately with a non-zero exit status.

In copy mode, if no *-x* format is specified, *pax* behaves as if *-x pax* were specified.

- X** When traversing the file hierarchy specified by a path name, *pax* does not descend into directories that have a different device ID (*st_dev*, see *stat(2)*).
- @** Includes extended attributes in the archive. *pax* does not place extended attributes in the archive by default.

When traversing the file hierarchy specified by a path name, `pax` descends into the attribute directory for any file with extended attributes. Extended attributes go into the archive as special files.

When this flag is used during file extraction, any extended attributes associated with a file being extracted are also extracted. Extended attribute files can only be extracted from an archive as part of a normal file extract. Attempts to explicitly extract attribute records are ignored.

`-/` Includes extended system attributes in the archive. `pax` does not place extended system attributes in the archive by default.

When traversing the file hierarchy specified by a path name, `pax` descends into the attribute directory for any file with extended attributes. Extended attributes go into the archive as special files. When this flag is used during file extraction, any extended attributes associated with a file being extracted are also extracted. Extended attribute files can only be extracted from an archive as part of a normal file extract. Attempts to explicitly extract attribute records are ignored.

Specifying more than one of the mutually-exclusive options `-H` and `-L` is not considered an error. The last option specified determines the behavior of the utility.

The options that operate on the names of files or archive members (`-c`, `-i`, `-n`, `-s`, `-u` and `-v`) interact as follows.

In `read` mode, the archive members are selected based on the user-specified *pattern* operands as modified by the `-c`, `-n` and `-u` options. Then, any `-s` and `-i` options modify, in that order, the names of the selected files. The `-v` option writes names resulting from these modifications.

In `write` mode, the files are selected based on the user-specified path names as modified by the `-n` and `-u` options. Then, any `-s` and `-i` options modify, in that order, the names of these selected files. The `-v` option writes names resulting from these modifications.

If both the `-u` and `-n` options are specified, `pax` does not consider a file selected unless it is newer than the file to which it is compared.

List Mode Format Specifications

In `list` mode with the `-o listopt=format` option, the *format* argument is applied for each selected file. `pax` appends a NEWLINE to the `listopt` output for each selected file. The *format* argument is used as the format string with the following exceptions. (See [printf\(1\)](#) for the first five exceptions.)

1. A SPACE character in the format string, in any context other than a flag of a conversion specification, is treated as an ordinary character that is copied to the output.
2. A ' ' character in the format string is treated as a ' ' character, not as a SPACE.

3. In addition to the escape sequences described in the [formats\(5\)](#) manual page, (`\`, `\a`, `\b`, `\f`, `\n`, `\r`, `\t`, `\v`), `\ddd`, where *ddd* is a one-, two-, or three-digit octal number, is written as a byte with the numeric value specified by the octal number.
4. Output from the `d` or `u` conversion specifiers is not preceded or followed with BLANKs not specified by the format operand.
5. Output from the `o` conversion specifier is not preceded with zeros that are not specified by the format operand.
6. The sequence (*keyword*) can occur before a format conversion specifier. The conversion argument is defined by the value of *keyword*. The following keywords are supported (see IEEE Std 1003.1–2001):
 - Any of the Field Name entries in `ustar Header Block` and `Octet-Oriented cpio Archive Entry`. The implementation supports the `cpio` keywords without the leading `c_` in addition to the form required by `Values for cpio c_ mode Field`.
 - Any keyword defined for the extended header in `pax Extended Header`.
 - Any keyword provided as an implementation-defined extension within the extended header defined in `pax Extended Header`.

For example, the sequence “%(charset)s” is the string value of the name of the character set in the extended header.

The result of the keyword conversion argument is the value from the applicable header field or extended header, without any trailing NULs.

All keyword values used as conversion arguments are translated from the UTF-8 encoding to the character set appropriate for the local file system, user database, and so on, as applicable.

7. An additional conversion specifier character, `T`, is used to specify time formats. The `T` conversion specifier character can be preceded by the sequence (*keyword*=*subformat*), where *subformat* is a date format as defined by `date` operands. The default *keyword* is *mtime* and the default *subformat* is:


```
%b %e %H:%M %Y
```
8. An additional conversion specifier character, `M`, is used to specify the file mode string as defined in `ls Standard Output`. If (*keyword*) is omitted, the mode keyword is used. For example, `%.1M` writes the single character corresponding to the *entry type* field of the `ls -l` command.
9. An additional conversion specifier character, `D`, is used to specify the device for block or special files, if applicable, in an implementation-defined format. If not applicable, and (*keyword*) is specified, then this conversion is equivalent to `%(keyword)u`. If not applicable, and (*keyword*) is omitted, then this conversion is equivalent to `SPACE`.

10. An additional conversion specifier character, F, is used to specify a path name. The F conversion character can be preceded by a sequence of comma-separated keywords:

(keyword[,keyword] ...)

The values for all the keywords that are non-null are concatenated, each separated by a '/'. The default is (path) if the keyword path is defined. Otherwise, the default is (*prefix,name*).

11. An additional conversion specifier character, L, is used to specify a symbolic link expansion. If the current file is a symbolic link, then %L expands to:

"%s -> %s", value of keyword, contents of link

Otherwise, the %L conversion specification is the equivalent of %F.

操作数

The following operands are supported:

<i>directory</i>	The destination directory path name for copy mode.
<i>file</i>	A path name of a file to be copied or archived.
<i>pattern</i>	A pattern matching one or more path names of archive members. A pattern must conform to the pattern matching notation found on the fnmatch(5) manual page. The default, if no <i>pattern</i> is specified, is to select all members in the archive.

Output

Output formats are discussed below:

Standard Output

In `write` mode, if `-f` is not specified, the standard output is the archive formatted according to one of the formats described below. See `-x` format for a list of supported formats.

In `list` mode, when the `-o listopt=format` option has been specified, the selected archive members are written to standard output using the format described above under List Mode Format Specifications. In `list` mode without the `-o listopt=format` option, the table of contents of the selected archive members are written to standard output using the following format:

"%s\n", pathname

If the `-v` option is specified in `list` mode, the table of contents of the selected archive members are written to standard output using the following formats:

- For path names representing hard links to previous members of the archive:

"%s == %s\n", <ls -l listing, linkname

- For all other path names:

"%s\n", <ls -l listing>

where *<ls -l listing>* is the format specified by the `ls` command with the `-l` option. When writing path names in this format, it is unspecified what is written for fields for which the

underlying archive format does not have the correct information, although the correct number of blank-character-separated fields is written.

In `list` mode, standard output is not buffered more than a line at a time.

Standard Error If `-v` is specified in `read`, `write` or `copy` modes, `pax` writes the path names it processes to the standard error output using the following format:

```
"%s\n", pathname
```

These path names are written as soon as processing is begun on the file or archive member, and are flushed to standard error. The trailing NEWLINE character, which is not buffered, is written when the file has been read or written.

If the `-s` option is specified, and the replacement string has a trailing `p`, substitutions are written to standard error in the following format:

```
"%s >> %s\n", <original pathname>, <new pathname>
```

In all operating modes of `pax`, optional messages of unspecified format concerning the input archive format and volume number, the number of files, blocks, volumes, and media parts as well as other diagnostic messages can be written to standard error.

In all formats, for both standard output and standard error, it is unspecified how non-printable characters in path names or link names are written.

When `pax` is in `read` mode or `list` mode, using the `-x pax` archive format, and a file name, link name, owner name, or any other field in an extended header record cannot be translated from the `pax` UTF-8 codeset format to the codeset and current locale of the implementation, `pax` writes a diagnostic message to standard error, processes the file as described for the `-o invalid=` option, and then processes the next file in the archive.

Output Files In `read` mode, the extracted output files are of the archived file type. In `copy` mode, the copied output files are the type of the file being copied. In either mode, existing files in the destination hierarchy are overwritten only when all permission (`-p`), modification time (`-u`), and invalid-value (`-o invalid=`) tests allow it. In `write` mode, the output file named by the `-f` option-argument is a file formatted according to one of the specifications in IEEE Std 1003.1-2001.

错误 If `pax` cannot create a file or a link when reading an archive, or cannot find a file when writing an archive, or cannot preserve the user ID, group ID, or file mode when the `-p` option is specified, a diagnostic message is written to standard error and a non-zero exit status is returned, but processing continues. In the case where `pax` cannot create a link to a file, `pax` does not, by default, create a second copy of the file.

If the extraction of a file from an archive is prematurely terminated by a signal or error, `pax` can have only partially extracted the file or, if the `-n` option was not specified, can have

extracted a file of the same name as that specified by the user, but which is not the file the user wanted. Additionally, the file modes of extracted directories can have additional bits from the read, write, execute mask set as well as incorrect modification and access times.

用法

The `-p` (privileges) option was invented to reconcile differences between historical [tar\(1\)](#) and [cpio\(1\)](#) implementations. In particular, the two utilities use `-m` in diametrically opposed ways. The `-p` option also provides a consistent means of extending the ways in which future file attributes can be addressed, such as for enhanced security systems or high-performance files. Although it can seem complex, there are really two modes that are most commonly used:

- p e Preserve everything. This would be used by the historical superuser, someone with all the appropriate privileges, to preserve all aspects of the files as they are recorded in the archive. The `e` flag is the sum of `o` and `p`, and other implementation-dependent attributes.
- p p Preserve the file mode bits. This would be used by the user with regular privileges who wished to preserve aspects of the file other than the ownership. The file times are preserved by default, but two other flags are offered to disable these and use the time of extraction.

The one path name per line format of standard input precludes path names containing newlines. Although such path names violate the portable filename guidelines, they can exist and their presence can inhibit usage of `pax` within shell scripts. This problem is inherited from historical archive programs. The problem can be avoided by listing file name arguments on the command line instead of on standard input.

It is almost certain that appropriate privileges are required for `pax` to accomplish parts of this. Specifically, creating files of type block special or character special, restoring file access times unless the files are owned by the user (the `-t` option), or preserving file owner, group, and mode (the `-p` option) all probably require appropriate privileges.

In read mode, implementations are permitted to overwrite files when the archive has multiple members with the same name. This can fail if permissions on the first version of the file do not permit it to be overwritten.

When using the `-x xustar` and `-x -pax` archive formats, if the underlying file system reports that the file being archived contains holes, the Solaris `pax` utility records the presence of holes in an extended header record when the file is archived. If this extended header record is associated with a file in the archive, those holes are recreated whenever that file is extracted from the archive. See the `SEEK_DATA` and `SEEK_HOLE` whence values in [lseek\(2\)](#). In all other cases, any NUL (`\0`) characters found in the archive is written to the file when it is extracted.

See [largefile\(5\)](#) for the description of the behavior of `pax` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

Standard Input In *w*rite mode, the standard input is used only if no *f*ile operands are specified. It is a text file containing a list of path names, one per line, without leading or trailing blanks. In *l*ist and *r*ead modes, if *-f* is not specified, the standard input is an archive file. Otherwise, the standard input is not used.

Input Files The input file named by the *archive* option-argument, or standard input when the archive is read from there, is a file formatted according to one of the formats described below. See *Extended Description*. The file `/dev/tty` is used to write prompts and read responses.

示例

示例 1 Copying the Contents of the Current Directory

The following command:

```
example% pax -w -f /dev/rmt1m .
```

copies the contents of the current directory to tape drive 1, medium density. This assumes historical System V device naming procedures. The historical BSD device name would be `/dev/rmt9`.

示例 2 Copying the Directory Hierarchy

The following commands:

```
example% mkdir newdir
example% pax -rw olddir newdir
```

copy the `olddir` directory hierarchy to `newdir`.

示例 3 Reading an Archive Extracted Relative to the Current Directory

The following command:

```
example% pax -r -s ',^/*usr/*,,,' -f a.pax
```

reads the archive `a.pax`, with all files rooted in `/usr` in the archive extracted relative to the current directory.

示例 4 Overriding the Default Output Description

Using the option:

```
-o listopt="%M %(atime)T %(size)D %(name)s"
```

overrides the default output description in Standard Output and instead writes:

```
-rw-rw- - - Jan 12 15:53 2003 1492 /usr/foo/bar
```

Using the options:

```
-o listopt='%L\t%(size)D\n%.7' \
-o listopt='(name)s\n%(atime)T\n%T'
```

示例 4 Overriding the Default Output Description (续)

overrides the default output description in standard output and instead writes:

```
usr/foo/bar -> /tmp          1492
/usr/foo
Jan 12 15:53 1991
Jan 31 15:53 2003
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of pax: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, LC_TIME, and NLS_PATH.

LC_COLLATE	Determine the locale for the behaviour of ranges, equivalence classes, and multi-character collating elements used in the pattern matching expressions for the <i>pattern</i> operand, the basic regular expression for the -s option, and the extended regular expression defined for the yesexpr locale keyword in the LC_MESSAGES category.
TMPDIR	Determine the path name that provides part of the default global extended header record file, as described for the -o globexthdr= keyword as described in the OPTIONS section.
TZ	Determine the timezone used to calculate date and time strings when the -v option is specified. If TZ is unset or null, an unspecified default timezone is used.

退出状态

The following exit values are returned:

- 0 All files were processed successfully.
- >0 An error occurred.

扩展描述

pax Interchange Format

A pax archive tape or file produced in the -xpax format contains a series of blocks. The physical layout of the archive is identical to the *ustar* format described in *ustar Interchange Format*. Each file archived is represented by the following sequence:

- An optional header block with extended header records. This header block is of the form 27403 with a *typeflag* value of x or g. The extended header records is included as the data for this header block.
- A header block that describes the file. Any fields in the preceding optional extended header overrides the associated fields in this header block for this file.
- Zero or more blocks that contain the contents of the file.

At the end of the archive file there are two 512-byte blocks filled with binary zeroes, interpreted as an end-of-archive indicator.

The following is a schematic of an example archive with global extended header records and two actual files in pax format archive. In the example, the second file in the archive has no extended header preceding it, presumably because it has no need for extended attributes.

Description	Block
Global Extended Header	ustar Header [<i>typeflag</i> =g] Global Extended Header Data
File 1: Extended Header is included	ustar Header [<i>typeflag</i> =x] Extended Header Data [<i>typeflag</i> =0] ustar Header Data for File 1
File 2: No Extended Header is included	ustar Header [<i>typeflag</i> =0] Data for File2
End of Archive Indicator	Block of binary zeros Block of binary zeros

pax Header Block

The pax header block is identical to the ustar header block described in ustar Interchange Format except that two additional *typeflag* values are defined:

- g Represents global extended header records for the following files in the archive. The format of these extended header records are as described in pax Extended Header. Each value affects all subsequent files that do not override that value in their own extended header record and until another global extended header record is reached that provides another value for the same field. The *typeflag* g global headers should not be used with interchange media that could suffer partial data loss in transporting the archive.
- x Represents extended header records for the following file in the archive (which has its own ustar header block). The format of these extended header records is as described in pax Extended Header.

For both of these types, the *size* field is the size of the extended header records in octets. The other fields in the header block are not meaningful to this version of pax. However, if this archive is read by pax conforming to a previous version of ISO POSIX-2:1993 Standard, the header block fields are used to create a regular file that contains the extended header records as data. Therefore, header block field values should be selected to provide reasonable file access to this regular file.

A further difference from the ustar header block is that data blocks for files of *typeflag* 1 (the digit one) (hard link) might be included, which means that the *size* field can be greater than zero. Archives created by pax -o linkdata includes these data blocks with the hard links.

pax Extended Header A *pax* extended header contains values that are inappropriate for the *ustar* header block because of limitations in that format: fields requiring a character encoding other than that described in the ISO/IEC 646: 1991 standard, fields representing file attributes not described in the *ustar* header, and fields whose format or length do not fit the requirements of the *ustar* header. The values in an extended header add attributes to the specified file or files or override values in the specified header blocks, as indicated in the following list of keywords. See the description of the *typeflag* header block.

An extended header consists of one or more records, each constructed as follows:

```
"%d %s=%s\n", length, keyword, value
```

The extended header records are encoded according to the ISO/IEC 10646-1: 2000 standard (UTF-8). *length*, BLANK, equals sign (=), and NEWLINE are limited to the portable character set, as encoded in UTF-8. *keyword* and *value* can be any UTF-8 characters. *length* is the decimal length of the extended header record in octets, including the trailing NEWLINE.

keyword is one of the entries from the following list or a keyword provided as an implementation extension. Keywords consisting entirely of lowercase letters, digits, and periods are reserved for future standardization. A keyword does not include an equals sign.

In the following list, the notation of *file(s)* or *block(s)* are used to acknowledge that a keyword affects the specified single file after a *typeflag* *x* extended header, but possibly multiple files after *typeflag* *g*. Any requirements in the list for *pax* to include a record when in write or copy mode applies only when such a record has not already been provided through the use of the *-o* option. When used in copy mode, *pax* behaves as if an archive had been created with applicable extended header records and then extracted.

atime	The file access time for the specified files, equivalent to the value of the <code>st_atime</code> member of the <code>stat</code> structure for a file, as described by the stat(2) function. The access time (<i>atime</i>) is restored if the process has the appropriate privilege required to do so. The format of the <i>value</i> is as described in <i>pax Extended Header File Times</i> .
charset	The name of the character set used to encode the data in the specified files. The entries in the following table are defined to refer to known standards; additional names can be agreed on between the originator and recipient.

<i>value</i>	Formal Standard
ISO-IR 646 1990	ISO/IEC646:1990
ISO-IR 8859 1 1998	ISO/IEC8859-1:1998
ISO-IR 8859 2 1999	ISO/IEC 8859-2:1999
ISO-IR 8859 3 1999	ISO/IEC 8859-3:1999

ISO-IR 8859 4 1999	ISO/IEC8859-4:1998
ISO-IR 8859 5 1999	ISO/IEC8859-5-1999
ISO-IR 8859 6 1999	ISO/IEC8859-6-1999
ISO-IR 8859 7 1987	ISO/IEC8859-7:1987
ISO-IR 8859 8 1999	ISO/IEC8859-8:1999
ISO-IR 8859 9 1999	ISO/IEC8859-9:1999
ISO-IR 8859 10 1998	ISO/IEC8859-10:1999
ISO-IR 8859 13 1998	ISO/IEC8859-13:1998
ISO-IR 8859 14 1998	ISO/IEC8859-14:1998
ISO-IR 8859 15 1999	ISO/IEC8859-15:1999
ISO-IR 10646 2000	ISO/IEC 10646:2000
ISO-IR 10646 2000 UTF-8	ISO/IEC 10646,UTF-8 encoding
BINARY	None

The encoding is included in an extended header for information only; when `pax` is used as described in IEEE Std 1003.1-200x, it does not translate the file data into any other encoding. The BINARY entry indicates unencoded binary data. When used in write or copy mode, it is implementation-defined whether `pax` includes a charset extended header record for a file.

<code>comment</code>	A series of characters used as a comment. All characters in the <i>value</i> field are ignored by <code>pax</code> .
<code>gid</code>	The group ID of the group that owns the file, expressed as a decimal number using digits from the ISO/IEC 646: 1991 standard. This record overrides the <code>gid</code> field in the specified header blocks. When used in write or copy mode, <code>pax</code> includes a <code>gid</code> extended header record for each file whose group ID is greater than 2097151 (octal 7777777).
<code>gname</code>	The group of the files, formatted as a group name in the group database. This record overrides the <code>gid</code> and <code>gname</code> fields in the specified header blocks, and any <code>gid</code> extended header record. When used in read, copy, or list mode, <code>pax</code> translates the name from the UTF-8 encoding in the header record to the character set appropriate for the group database on the receiving system. If any of the UTF-8 characters cannot be translated, and if the <code>-o invalid=UTF-8</code> option is not specified, the results are implementation-defined. When used in write or copy mode, <code>pax</code> includes a

	<i>gname</i> extended header record for each file whose group name cannot be represented entirely with the letters and digits of the portable character set.
<i>linkpath</i>	The pathname of a link being created to another file, of any type, previously archived. This record overrides the <i>linkname</i> field in the specified <i>ustar</i> header blocks. The specified <i>ustar</i> header block determines the type of link created. If <i>typeflag</i> of the specified header block is 1, it is a hard link. If <i>typeflag</i> is 2, it is a symbolic link and the <i>linkpath</i> value is the contents of the symbolic link. <i>pax</i> translates the name of the link (contents of the symbolic link) from the UTF-8 encoding to the character set appropriate for the local file system. When used in write or copy mode, <i>pax</i> includes a <i>linkpath</i> extended header record for each link whose pathname cannot be represented entirely with the members of the portable character set other than NULL.
<i>mtime</i>	The pathname of a link being created to another file, of any type, previously archived. This record overrides the <i>linkname</i> field in the specified <i>ustar</i> header blocks. The specified <i>ustar</i> header block determines the type of link created. If <i>typeflag</i> of the specified header block is 1, it is a hard link. If <i>typeflag</i> is 2, it is a symbolic link and the <i>linkpath</i> value is the contents of the symbolic link. <i>pax</i> translates the name of the link (contents of the symbolic link) from the UTF-8 encoding to the character set appropriate for the local file system. When used in write or copy mode, <i>pax</i> includes a <i>linkpath</i> extended header record for each link whose pathname cannot be represented entirely with the members of the portable character set other than NULL.
<i>path</i>	The pathname of the specified files. This record overrides the name and <i>prefix</i> fields in the specified header blocks. <i>pax</i> translates the pathname of the file from the UTF-8 encoding to the character set appropriate for the local file system. When used in write or copy mode, <i>pax</i> includes a <i>path</i> extended header record for each file whose pathname cannot be represented entirely with the members of the portable character set other than NULL.
<i>realtime.any</i>	The keywords prefixed by <i>realtime</i> are reserved for future standardization.
<i>security.any</i>	The keywords prefixed by <i>security</i> are reserved for future standardization.
<i>size</i>	The size of the file in octets, expressed as a decimal number using digits from the ISO/IEC 646: 1991 standard. This record overrides the <i>size</i> field in the specified header blocks. When used in write or copy mode, <i>pax</i> includes a <i>size</i> extended header record for each file with a <i>size</i> value greater than 8589934591 (octal 7777777777).

<i>uid</i>	The user ID of the file owner, expressed as a decimal number using digits from the ISO/IEC 646:1991 standard. This record overrides the <i>uid</i> field in the following header block(s). When used in write or copy mode, pax includes a <i>uid</i> extended header record for each file whose owner ID is greater than 2097151 (octal 777777).
<i>uname</i>	The owner of the specified files, formatted as a user name in the user database. This record overrides the <i>uid</i> and <i>uname</i> fields in the specified header blocks, and any <i>uid</i> extended header record. When used in read, copy, or list mode, pax translates the name from the UTF-8 encoding in the header record to the character set appropriate for the user database on the receiving system. If any of the UTF-8 characters cannot be translated, and if the <code>-o invalid=UTF-8</code> option is not specified, the results are implementation-defined. When used in write or copy mode, pax includes a <i>uname</i> extended header record for each file whose user name cannot be represented entirely with the letters and digits of the portable character set.

If the *value* field is zero length, it deletes any header block field, previously entered extended header value, or global extended header value of the same name.

If a keyword in an extended header record (or in an `-o` option-argument) overrides or deletes a corresponding field in the *ustar* header block, pax ignores the contents of that header block field.

Unlike the *ustar* header block fields, *NULLs* does not delimit values; all characters within the *value* field are considered data for the field.

pax Extended Header Keyword Precedence

This section describes the precedence in which the various header records and fields and command line options are selected to apply to a file in the archive. When pax is used in read or list modes, it determines a file attribute in the following sequence:

1. If `-o delete=keyword-prefix` is used, the affected attributes is determined from step 7, if applicable, or ignored otherwise.
2. If `-o keyword:=` is used, the affected attributes is ignored.
3. If `-o keyword=value` is used, the affected attribute is assigned the value.
4. If there is a *typeflag* *x* extended header record, the affected attribute is assigned the value. When extended header records conflict, the last one given in the header takes precedence.
5. If `-o keyword=value` is used, the affected attribute is assigned the value.
6. If there is a *typeflag* *g* global extended header record, the affected attribute is assigned the value. When global extended header records conflict, the last one given in the global header takes precedence.
7. Otherwise, the attribute is determined from the *ustar* header block.

pax Extended Header
File Times

pax writes an *mtime* record for each file in write or copy modes if the file's modification time cannot be represented exactly in the *ustar* header logical record described in *ustar* Interchange Format. This can occur if the time is out of *ustar* range, or if the file system of the underlying implementation supports non-integer time granularities and the time is not an integer. All of these time records are formatted as a decimal representation of the time in seconds since the Epoch. If a period (.) decimal point character is present, the digits to the right of the point represents the units of a sub-second timing granularity, where the first digit is tenths of a second and each subsequent digit is a tenth of the previous digit. In read or copy mode, pax truncates the time of a file to the greatest value that is not greater than the input header file time. In write or copy mode, pax outputs a time exactly if it can be represented exactly as a decimal number, and otherwise generates only enough digits so that the same time is recovered if the file is extracted on a system whose underlying implementation supports the same time granularity.

ustar Interchange
Format

A *ustar* archive tape or file contains a series of logical records. Each logical record is a fixed-size logical record of 512 octets. Although this format can be thought of as being stored on 9-track industry-standard 12.7mm (0.5 in) magnetic tape, other types of transportable media are not excluded. Each file archived is represented by a header logical record that describes the file, followed by zero or more logical records that give the contents of the file. At the end of the archive file there are two 512-octet logical records filled with binary zeros, interpreted as an end-of-archive indicator.

The logical records can be grouped for physical I/O operations, as described under the *-bblocksize* and *-x* *ustar* options. Each group of logical records can be written with a single operation equivalent to the `write(2)` function. On magnetic tape, the result of this write is a single tape physical block. The last physical block always is the full size, so logical records after the two zero logical records can contain undefined data.

The header logical record is structured as shown in the following table. All lengths and offsets are in decimal.

表 1 *ustar* Header Block

Field Name	Octet Offset	Length (in Octets)
<i>name</i>	0	100
<i>mode</i>	100	8
<i>uid</i>	108	8
<i>gid</i>	116	8
<i>size</i>	124	12
<i>mtime</i>	136	12
<i>chksum</i>	148	8

表 1 ustar Header Block (续)

Field Name	Octet Offset	Length (in Octets)
<i>typeflag</i>	156	1
<i>linkname</i>	157	100
<i>magic</i>	257	6
<i>version</i>	263	2
<i>uname</i>	265	32
<i>gname</i>	297	32
<i>devmajor</i>	329	8
<i>devminor</i>	337	8
<i>prefix</i>	345	155

All characters in the header logical record is represented in the coded character set of the ISO/IEC 646: 1991 standard. For maximum portability between implementations, names should be selected from characters represented by the portable filename character set as octets with the most significant bit zero. If an implementation supports the use of characters outside of slash and the portable filename character set in names for files, users, and groups, one or more implementation-defined encodings of these characters are provided for interchange purposes.

pax never creates filenames on the local system that cannot be accessed using the procedures described in IEEE Std 1003.1-200x. If a filename is found on the medium that would create an invalid filename, it is implementation-defined whether the data from the file is stored on the file hierarchy and under what name it is stored. *pax* can choose to ignore these files as long as it produces an error indicating that the file is being ignored. Each field within the header logical record is contiguous; that is, there is no padding used.

Each field within the header logical record is contiguous. There is no padding used. Each character on the archive medium is stored contiguously.

The fields *magic*, *uname* and *gname* are character strings, each of which is terminated by a NULL character. The fields *name*, *linkname*, and *prefix* are NULL-terminated character strings except when all characters in the array contain non-NULL characters including the last character. The *version* field is two octets containing the characters 00 (zero-zero) The *typeflag* contains a single character. All other fields are leading zero-filled octal numbers using digits from the ISO/IEC 646:1991 standard IRV. Each numeric field is terminated by one or more SPACE or NULL characters.

Each character on the archive medium is stored contiguously. The fields *magic*, *uname*, and *gname* are character strings each terminated by a NULL character.

name, *linkname*, and *prefix* are NULL-terminated character strings except when all characters in the array contain non-NULL characters including the last character. The *version* field is two octets containing the characters 00 (zero-zero). The *typeflag* contains a single character. All other fields are leading zero-filled octal numbers using digits from the ISO/IEC 646: 1991 standard IRV. Each numeric field is terminated by one or more spaces or NULL characters.

The *name* and the *prefix* fields produce the pathname of the file. A new pathname is formed, if *prefix* is not an empty string (its first character is not NULL), by concatenating *prefix* (up to the first NULL character), a slash character, and name; otherwise, name is used alone. In either case, name is terminated at the first NULL character. If *prefix* begins with a NULL character, it is ignored. In this manner, pathnames of at most 256 characters can be supported. If a pathname does not fit in the space provided, pax notifies the user of the error, and does not store any part of the file-header or data-on the medium.

The *linkname* field does not use the *prefix* to produce a pathname. As such, a *linkname* is limited to 100 characters. If the name does not fit in the space provided, pax notifies the user of the error, and does not attempt to store the link on the medium. The *mode* field provides 12 bits encoded in the ISO/IEC 646: 1991 standard octal digit representation. The encoded bits represent the following values in the *ustar mode* field:

Bit Value	IEE Std 1003.1–2001 Bit	Description
04000	S_ISUID	Set UID on execution
02000	S_ISGID	Set GID on execution
01000	<i>reserved</i>	Reserved for future standardization
00400	S_IRUSR	Read permission for file owner class
00200	S_IWUSR	Write permission for file owner class
00100	S_IXUSR	Execute/search permission for file owner class
00040	S_IRGRP	Read permission for file group class
00020	S_IWGRP	Write permission for file group class
00010	S_IXGRP	Execute/search permission for file group class
00004	S_IROTH	Read permission for file other class
00002	S_IWOTH	Write permission for file other class
00001	S_IXOTH	Execute/search permission for file other class

When appropriate privilege is required to set one of these mode bits, and the user restoring the files from the archive does not have the appropriate privilege, the mode bits for which the user

does not have appropriate privilege are ignored. Some of the mode bits in the archive format are not mentioned elsewhere in volume IEEE Std 1003.1-200x. If the implementation does not support those bits, they can be ignored.

The *uid* and *gid* fields are the user and group ID of the owner and group of the file, respectively.

The *size* field is the size of the file in octets. If the *typeflag* field is set to specify a file to be of type 1 (a link) or 2 (a symbolic link), the *size* field is specified as zero. If the *typeflag* field is set to specify a file of type 5 (directory), the *size* field is interpreted as described under the definition of that record type. No data logical records are stored for types 1, 2, or 5. If the *typeflag* field is set to 3 (character special file), 4 (block special file), or 6 (FIFO), the meaning of the *size* field is unspecified by volume IEEE Std 1003.1-200x, and no data logical records is stored on the medium. Additionally, for type 6, the *size* field is ignored when reading. If the *typeflag* field is set to any other value, the number of logical records written following the header is $(size+511)/512$, ignoring any fraction in the result of the division.

The *mtime* field is the modification time of the file at the time it was archived. It is the ISO/IEC 646: 1991 standard representation of the octal value of the modification time obtained from the `stat()` function.

The *chksum* field is the ISO/IEC 646: 1991 standard IRV representation of the octal value of the simple sum of all octets in the header logical record. Each octet in the header is treated as an unsigned value. These values are added to an unsigned integer, initialized to zero, the precision of which is not less than 17 bits. When calculating the checksum, the *chksum* field is treated as if it were all spaces.

The *typeflag* field specifies the type of file archived. If a particular implementation does not recognize the type, or the user does not have appropriate privilege to create that type, the file is extracted as if it were a regular file if the file type is defined to have a meaning for the *size* field that could cause data logical records to be written on the medium. If conversion to a regular file occurs, pax produces an error indicating that the conversion took place. All of the *typeflag* fields are coded in the ISO/IEC 646: 1991 standard IRV:

- | | |
|---|---|
| 0 | Represents a regular file. For backward compatibility, a <i>typeflag</i> value of binary zero ('0') should be recognized as meaning a regular file when extracting files from the archive. Archives written with this version of the archive file format create regular files with a <i>typeflag</i> value of the ISO/IEC 646: 1991 standard IRV '0'. |
| 1 | Represents a file linked to another file, of any type, previously archived. Such files are identified by each file having the same device and file serial number. The linked-to name is specified in the <i>linkname</i> field with a NULL-character terminator if it is less than 100 octets in length. |
| 2 | Represents a symbolic link. The contents of the symbolic link are stored in the <i>linkname</i> field. |

3, 4	Represents character special files and block special files respectively. In this case the <i>devmajor</i> and <i>devminor</i> fields contain information defining the device, the format of which is unspecified by volume IEEE Std 1003.1-200x. Implementations can map the device specifications to their own local specification or can ignore the entry.
5	Specifies a directory or subdirectory. On systems where disk allocation is performed on a directory basis, the <i>size</i> field contain the maximum number of octets (which can be rounded to the nearest disk block allocation unit) that the directory can hold. A <i>size</i> field of zero indicates no such limiting. Systems that do not support limiting in this manner should ignore the <i>size</i> field.
6	Specifies a FIFO special file. The archiving of a FIFO file archives the existence of this file and not its contents.
7	Reserved to represent a file to which an implementation has associated some high- performance attribute. Implementations without such extensions should treat this file as a regular file (type 0).
A - Z	The letters A through Z inclusive are reserved for custom implementations. All other values are reserved for future versions of IEEE Std 1003.1-200x.
SUN.devmajor	A Solaris extension to pax extended header keywords. Specifies the major device number of the file. When used in write or copy mode and the <i>xustar</i> or <i>pax</i> format (see <i>-x format</i>) was specified, <i>pax</i> includes a <i>SUN.devmajor</i> extended header record for each file whose major device number is too large to fit in 8 octets.
SUN.devminor	A Solaris extension to pax extended header keywords. Specifies the minor device number of the file. When used in write or copy mode and the <i>xustar</i> or <i>pax</i> format (see <i>-x format</i>) is specified, <i>pax</i> includes a <i>SUN.devminor</i> extended header record for each file whose minor device number is too large to fit in 8 octets.
SUN.holesdata	A Solaris extension to pax extended header keywords. Specifies the data and hole pairs for a sparse file. In write or copy modes and when the <i>xustar</i> or <i>pax</i> format (see <i>-x format</i>) is specified, <i>pax</i> includes a <i>SUN.holesdata</i> extended header record if the underlying file system supports the detection of files with holes (see fpathconf(2)) and reports that there is at least one hole in the file being archived. <i>value</i> consists of two or more consecutive entries of the following form:

SPACEdata_offsetSPACEhole_offset

where the data and hole offsets are the long values returned by passing `SEEK_DATA` and `SEEK_HOLE` to `lseek(2)`, respectively. For example, the following entry is an example of the `SUN.holesdata` entry in the extended header for a file with data offsets at bytes 0, 24576, and 49152, and hole offsets at bytes 8192, 32768, and 49159: `49 SUN.holesdata= 0 8192 24576 32768 49152 49159:`

```
49 SUN.holesdata= 0 8192 24576 32768 49152 49159
```

When extracting a file from an archive in read or copy modes, if a `SUN.holesdata = pair` is found in the extended header for the file, then the file is restored with the holes identified using this data. For example, for the `SUN.holesdata` provided in the example above, bytes from 0 to 8192 are restored as data, a hole is created up to the next data position (24576), bytes 24576 to 32768 is restored as data, and so forth.

- X A Solaris custom `typeflag` implementation which specifies an `xstar` format (see `-x` format) extended header. The `typeflag 'x'` extended header is treated as a `ustar typeflag 'x'` extended header.
- E A Solaris custom `typeflag` implementation which specifies an extended attributes header. See `fsattr(5)`.

Attempts to archive a socket using `ustar` interchange format produce a diagnostic message. Handling of other file types is implementation-defined.

The *magic* field is the specification that this archive was output in this archive format. If this field contains `ustar` (the five characters from the ISO/IEC 646: 1991 standard IRV shown followed by `NULL`), the *uname* and *gname* fields contain the ISO/IEC 646: 1991 standard IRV representation of the owner and group of the file, respectively (truncated to fit, if necessary). When the file is restored by a privileged, protection-preserving version of the utility, the user and group databases are scanned for these names. If found, the user and group IDs contained within these files are used rather than the values contained within the *uid* and *gid* fields.

cpio Interchange Format

The octet-oriented `cpio` archive format are a series of entries, each comprising a header that describes the file, name of the file, and contents of the file.

An archive can be recorded as a series of fixed-size blocks of octets. This blocking is be used only to make physical I/O more efficient. The last group of blocks are always at the full size.

For the octet-oriented `cpio` archive format, the individual entry information are in the order indicated and described by the following table: Octet-Oriented `cpio` Archive Entry. See the `cpio.h` header for additional details.

Header Field Name	Length (in Octets)	Interpreted as
<i>c_magic</i>	6	Octal number
<i>c_dev</i>	6	Octal number
<i>c_ino</i>	6	Octal number
<i>c_mode</i>	6	Octal number
<i>c_uid</i>	6	Octal number
<i>c_gid</i>	6	Octal number
<i>c_nlink</i>	6	Octal number
<i>c_rdev</i>	6	Octal number
<i>c_mtime</i>	11	Octal number
<i>c_namesize</i>	6	Octal number
<i>c_filesize</i>	11	Octal number
Filename Field Name	Length	Interpreted as
<i>c_name</i>	<i>c_namesize</i>	Pathname string
Filename Field Name	Length	Interpreted as
<i>c_filedata</i>	<i>c_filesize</i>	Data

cpio Header

For each file in the archive, a header as defined previously written. The information in the header fields is written as streams of the ISO/IEC 646: 1991 standard characters interpreted as octal numbers. The octal numbers are extended to the necessary length by appending the ISO/IEC 646: 1991 standard IRV zeros at the most-significant-digit end of the number. The result is written to the most-significant digit of the stream of octets first. The fields are interpreted as follows:

<i>c_magic</i>	Identifies the archive as being a transportable archive by containing the identifying value "070707".
<i>c_dev,c_ino</i>	Contains values that uniquely identify the file within the archive (that is, no files contain the same pair of <i>c_dev</i> and <i>c_ino</i> values unless they are links to the same file). The values are determined in an unspecified manner.
<i>c_mode</i>	Contains the file type and access permissions as defined in the following table. Directories, FIFOs, symbolic links, and regular files are supported on a system conforming to volume IEEE Std 1003.1-200x; additional values defined

previously are reserved for compatibility with existing systems. Additional file types can be supported. Such files should not be written to archives intended to be transported to other systems.

File Permissions Name	Value	Indicates
C_IRUSR	000400	by owner
C_IWUSR	000200	by owner
C_IXUSR	000100	by owner
C_IRGRP	000040	by group
CW_IWFGP	000020	by group
CW_IXGRP	000010	by group
CW_IROTH	000004	by others
CW_IWOTH	000002	by others
CW_IXOTH	000001	by others
CW_ISUID	004000	Set uid
W_ISGID	002000	Set gid
W_ISVTX	001000	Reserved

File Type Name	Value	Indicates
C_ISDIR	040000	Directory
C_ISFIFO	010000	FIFO
C_ISREG	0100000	Regular file
C_ISLNK	0120000	Symbolic link
C_ISBLK	060000	Block special file
C_ISCHR	020000	Character special file
C_ISSOCK	0140000	Socket
C_ISCTG	0110000	Reserved

c_uid Contains the user ID of the owner.

c_gid Contains the group ID of the group

<i>c_nlink</i>	Contains a number greater than or equal to the number of links in the archive referencing the file. If the <i>-a</i> option is used to append to a <i>cpio</i> archive, <i>pax</i> does need not to account for the files in the existing part of the archive when calculating the <i>c_nlink</i> values for the appended part of the archive. It does also need not alter the <i>c_nlink</i> values in the existing part of the archive if additional files with the same <i>c_dev</i> and <i>c_ino</i> values are appended to the archive.
<i>c_rdev</i>	Contains implementation-defined information for character or block special files.
<i>c_mtime</i>	Contains the latest time of modification of the file at the time the archive was created.
<i>c_namesize</i>	Contains the length of the pathname, including the terminating NULL character.
<i>c_filesize</i>	Contains the length of the file in octets. This is the length of the data section following the header structure.

cpio Filename The *c_name* field contains the pathname of the file. The length of this field in octets is the value of *c_namesize*. If a filename is found on the medium that would create an invalid pathname, it is implementation-defined whether the data from the file is stored on the file hierarchy and under what name it is stored. All characters are represented in the ISO/IEC 646: 1991 standard IRV. For maximum portability between implementations, names should be selected from characters represented by the portable filename character set as octets with the most significant bit zero. If an implementation supports the use of characters outside the portable filename character set in names for files, users, and groups, one or more implementation-defined encodings of these characters are provided for interchange purposes. *pax* does not create filenames on the local system that cannot be accessed by way of the procedures described in volume IEEE Std 1003.1-200x. If a filename is found on the medium that would create an invalid filename, it is implementation-defined whether the data from the file is stored on the local file system and under what name it is stored. *pax* can choose to ignore these files as long as it produces an error indicating that the file is being ignored.

cpio File Data Following *c_name*, there is *c_filesize* octets of data. Interpretation of such data occurs in a manner dependent on the file. If *c_filesize* is zero, no data is contained in *c_filedata*. When restoring from an archive:

- If the user does not have the appropriate privilege to create a file of the specified type, *pax* ignores the entry and writes an error message to standard error.
- Only regular files have data to be restored. Presuming a regular file meets any selection criteria that might be imposed on the format-reading utility by the user, such data is restored.
- If a user does not have appropriate privilege to set a particular *mode* flag, the flag is ignored. Some of the *mode* flags in the archive format are not mentioned in volume IEEE Std 1003.1-200x. If the implementation does not support those flags, they can be ignored.

cpio Special Entries FIFO special files, directories, and the trailer are recorded with *c_filesize* equal to zero. For other special files, *c_filesize* is unspecified in volume IEEE Std 1003.1-200x. The header for the next file entry in the archive are written directly after the last octet of the file entry preceding it. A header denoting the filename trailer indicates the end of the archive; the contents of octets in the last block of the archive following such a header are undefined.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [chmod\(1\)](#), [cpio\(1\)](#), [ed\(1\)](#), [printf\(1\)](#), [tar\(1\)](#), [mkdir\(2\)](#), [lseek\(2\)](#), [stat\(2\)](#), [write\(2\)](#), [archives.h\(3HEAD\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fnmatch\(5\)](#), [formats\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#), [standards\(5\)](#)

IEEE Std 1003.1-200x, ISO/IEC 646: 1991, ISO POSIX-2:1993 Standard

引用名 perl – Practical Extraction and Report Language

用法概要 perl [-sTuU] [-hv] [-V [: *configvar*]] [-cw]
 [-d [: *debugger*]] [-D [*number/list*]] [-pna]
 [-F *pattern*] [-l [*octal*]] [-0 [*octal*]] [-I *dir*]
 [-m [-] *module*] [-M [-] '*module...*'] [-P] [-S]
 [-x [*dir*]] [-i [*extension*]] [-e '*command*'] [--]
 [*programfile*] [*argument*]...

描述 For ease of access, the Perl manual has been split up into the following sections.

OVERVIEW

perl	Perl overview (this section)
perlintro	Perl introduction for beginners
perltoc	Perl documentation table of contents

TUTORIALS

Tutorials

perlreftut	Perl references short introduction
perldsc	Perl data structures intro
perllo1	Perl data structures: arrays of arrays
perlrequick	Perl regular expressions quick start
perlretut	Perl regular expressions tutorial
perlboot	Perl OO tutorial for beginners
perltoot	Perl OO tutorial, part 1
perltooc	Perl OO tutorial, part 2
perlbot	Perl OO tricks and examples
perlstyle	Perl style guide
perlcheat	Perl cheat sheet
perltrap	Perl traps for the unwary
perldebtut	Perl debugging tutorial
perlfaq	Perl frequently asked questions
perlfaq1	General Questions About Perl
perlfaq2	Obtaining and Learning about Perl
perlfaq3	Programming Tools
perlfaq4	Data Manipulation
perlfaq5	Files and Formats
perlfaq6	Regexes
perlfaq7	Perl Language Issues
perlfaq8	System Interaction
perlfaq9	Networking

REFERENCE MANUAL

perlsyn	Perl syntax
perldata	Perl data structures
perlop	Perl operators and precedence

perlsub	Perl subroutines
perlfunc	Perl built-in functions
perlopentut	Perl open() tutorial
perlpacktut	Perl pack() and unpack() tutorial
perlpod	Perl plain old documentation
perlpodspec	Perl plain old documentation format specification
perlrun	Perl execution and options
perldiag	Perl diagnostic messages
perllexwarn	Perl warnings and their control
perldebug	Perl debugging
perlvar	Perl predefined variables
perlre	Perl regular expressions, the rest of the story
perlref	Perl regular expressions quick reference
perlref	Perl references, the rest of the story
perlform	Perl formats
perlobj	Perl objects
perltie	Perl objects hidden behind simple variables
perldbfilter	Perl DBM filters
perlipc	Perl interprocess communication
perlfork	Perl fork() information
perlnumber	Perl number semantics
perlthrtut	Perl threads tutorial
perlothrtut	Old Perl threads tutorial
perlport	Perl portability guide
perllocale	Perl locale support
perluniintro	Perl Unicode introduction
perlunicode	Perl Unicode support
perlebcdic	Considerations for running Perl on EBCDIC platforms
perlsec	Perl security
perlmod	Perl modules: how they work
perlmodlib	Perl modules: how to write and use
perlmodstyle	Perl modules: how to write modules with style
perlmodinstall	Perl modules: how to install from CPAN
perlnewmod	Perl modules: preparing a new module for distribution
perlutil	utilities packaged with the Perl distribution
perlcompile	Perl compiler suite intro
perlfiler	Perl source filters

INTERNALS AND C LANGUAGE INTERFACE

perlembed	Perl ways to embed perl in your C or C++ application
perldebugts	Perl debugging guts and tips
perlxsut	Perl XS tutorial
perlxs	Perl XS application programming interface
perlclib	Internal replacements for standard C library functions

perlguts	Perl internal functions for those doing extensions
perlcall	Perl calling conventions from C
perlapi	Perl API listing (autogenerated)
perlintern	Perl internal functions (autogenerated)
perliol	C API for Perl's implementation of IO in Layers
perlpio	Perl internal IO abstraction interface
perlhack	Perl hackers guide

MISCELLANEOUS

perlbook	Perl book information
perltodo	Perl things to do
perldoc	Look up Perl documentation in Pod format
perlhist	Perl history records
perldelta	Perl changes since previous version
perl583delta	Perl changes in version 5.8.3
perl582delta	Perl changes in version 5.8.2
perl581delta	Perl changes in version 5.8.1
perl58delta	Perl changes in version 5.8.0
perl573delta	Perl changes in version 5.7.3
perl572delta	Perl changes in version 5.7.2
perl571delta	Perl changes in version 5.7.1
perl570delta	Perl changes in version 5.7.0
perl561delta	Perl changes in version 5.6.1
perl56delta	Perl changes in version 5.6
perl5005delta	Perl changes in version 5.005
perl5004delta	Perl changes in version 5.004
perlartistic	Perl Artistic License
perlgpl	GNU General Public License

LANGUAGE-SPECIFIC

perlcn	Perl for Simplified Chinese (in EUC-CN)
perljp	Perl for Japanese (in EUC-JP)
perlko	Perl for Korean (in EUC-KR)
perltw	Perl for Traditional Chinese (in Big5)

PLATFORM-SPECIFIC

perlsolaris	Perl notes for Solaris
-------------	------------------------

Platform-Specific If you're new to Perl, you should start with `perlintro`, which is a general intro for beginners and provides some background to help you navigate the rest of Perl's extensive documentation. For ease of access, the Perl manual has been split up into several sections.

The manpages listed above are installed in the `/usr/perl5/man/` directory.

Extensive additional documentation for Perl modules is available. This additional documentation is in the `/usr/perl5/man` directory. Some of this additional documentation is distributed standard with Perl, but you'll also find documentation for any customer-installed third-party modules there.

You can view Perl's documentation with `man(1)` by including `/usr/perl5/man` in the `MANPATH` environment variable. Notice that running `catman(1M)` on the Perl manual pages is not supported. For other Solaris-specific details, see the `NOTES` section below.

You can also use the supplied `/usr/perl5/bin/perl5doc` script to view Perl information.

If something strange has gone wrong with your program and you're not sure where you should look for help, try the `-w` switch first. It will often point out exactly where the trouble is.

Perl is a language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It's also a good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal).

Perl combines (in the author's opinion, anyway) some of the best features of C, sed, awk, and sh, so people familiar with those languages should have little difficulty with it. (Language historians will also note some vestiges of csh, Pascal, and even BASIC-PLUS.) Expression syntax corresponds closely to C expression syntax. Unlike most Unix utilities, Perl does not arbitrarily limit the size of your data -if you've got the memory, Perl can slurp in your whole file as a single string. Recursion is of unlimited depth. And the tables used by hashes (sometimes called "associative arrays") grow as necessary to prevent degraded performance. Perl can use sophisticated pattern matching techniques to scan large amounts of data quickly. Although optimized for scanning text, Perl can also deal with binary data, and can make dbm files look like hashes. Setuid Perl scripts are safer than C programs through a dataflow tracing mechanism that prevents many stupid security holes.

If you have a problem that would ordinarily use sed or awk or sh, but it exceeds their capabilities or must run a little faster, and you don't want to write the silly thing in C, then Perl may be for you. There are also translators to turn your sed and awk scripts into Perl scripts.

But wait, there's more...

Begun in 1993 (see `perlh1st`), Perl version 5 is nearly a complete rewrite that provides the following additional benefits:

- Modularity and reusability using innumerable modules Described in `perlmod`, `perlmodlib`, and `perlmodinstall`.
- Embeddable and extensible Described in `perlembed`, `perlxsut`, `perlx`, `perlcall`, `perlguits`, and `xsubpp`.
- Roll-your-own magic variables (including multiple simultaneous DBM implementations). Described in `perltie` and `AnyDBM_File`.

- Subroutines can now be overridden, autoloaded, and prototyped. Described in `perlsub`.
- Arbitrarily nested data structures and anonymous functions. Described in `perlreftut`, `perlref`, `perldsc`, and `perllo1`.
- Object-oriented programming. Described in `perlobj`, `perlboot`, `perltoot`, `perltoc`, and `perlbot`.
- Support for light-weight processes (threads). Described in `perlthrtut` and `threads`.
- Support for Unicode, internationalization, and localization. Described in `perluniintro`, `perllocale` and `Locale::Maketext`.
- Lexical scoping. Described in `perlsub`.
- Regular expression enhancements. Described in `perlre`, with additional examples in `perlop`.
- Enhanced debugger and interactive Perl environment, with integrated editor support. Described in `perldebtut`, `perldebug` and `perldebbugs`.
- POSIX 1003.1 compliant library. Described in `POSIX`.

Okay, that's *definitely* enough hype.

环境变量

The Perl shipped with Solaris is installed under `/usr/perl5` rather than the default `/usr/local` location. This is so that it can coexist with a customer-installed Perl in the default `/usr/local` location.

Any additional modules that you choose to install will be placed in the `/usr/perl5/site_perl/5.8.4` directory. The `/usr/perl5/vendor_perl` directory is reserved for SMI-provided modules.

Notice that the Perl utility scripts such as `perldoc` and `perlbug` are in the `/usr/perl5/bin` directory, so if you wish to use them you need to include `/usr/perl5/bin` in your `PATH` environment variable.

See also the `perlrun` mapage.

Author

Larry Wall, with the help of oodles of other folks.

If your Perl success stories and testimonials may be of help to others who wish to advocate the use of Perl in their applications, or if you wish to simply express your gratitude to Larry and the Perl developers, please write to `perl-thanks@perl.org`.

文件

"@INC" Locations of Perl libraries

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	runtime/perl-584, runtime/perl-584/extra, runtime/perl-584/manual, SUNWpl5u, SUNWpl5v SUNWpl5p, SUNWpl5m See below.
Interface Stability	See below.

Perl is available for most operating systems, including virtually all Unix-like platforms. See "Supported Platforms" in `perlport` for a listing.

The Script interface, the XSUB interface, and the Directory layout are Committed. The Binary interface is Uncommitted.

另请参见

a2p awk to perl translator
s2p sed to perl translator
<http://www.perl.com> Perl home page
<http://www.perl.com/CPAN> The Comprehensive Perl Archive
<http://www.perl.org> Perl Mongers (Perl user groups)

诊断

The 'use warnings' pragma (and the `-w` switch) produce some lovely diagnostics.

See `perldiag` for explanations of all Perl's diagnostics. The 'use diagnostics' pragma automatically turns Perl's normally terse warnings and errors into these longer forms.

Compilation errors will tell you the line number of the error, with an indication of the next token or token type that was to be examined. (In a script passed to Perl via `-e` switches, each `-e` is counted as one line.)

Setuid scripts have additional constraints that can produce error messages such as "Insecure dependency". See `perlsec`.

Did we mention that you should definitely consider using the `-w` switch?

附注

Perl 5.8.4 has been built to be largefile-aware and to use 64-bit integers, although the interpreter itself is a 32-bit application (LP32). To view detailed configuration information, use `perl -V` and `perlbug -dv`.

If you wish to build and install add-on modules from CPAN using `gcc`, you can do so using the `/usr/perl5/5.8.4/bin/perlgcc` script – see `perlgcc(1)` for details.

If you wish to build and install your own version of Perl, you should NOT remove the 5.8.4 version of perl under `/usr/perl5`, as it is required by several system utilities. The Perl package names are as follows:

SUNWperl584core	Perl 5.8.4 (Core files)
SUNWperl584usr	Perl 5.8.4 (Non-core files)
SUNWperl584man	Perl 5.8.4 (Manual pages)

Solaris 10 also ships with the 5.6.1 version of Perl that was included in Solaris 9. If you are upgrading your system and wish to continue to use Perl 5.6.1 as the default Perl version you should refer to the `perlsolaris` manpage for details of how to do this. Note that you should upgrade your installation to use Perl 5.8.4 as soon as is practicable, as Perl 5.6.1 may be removed in a future release.

The Perl motto is "There's more than one way to do it." Divining how many more is left as an exercise to the reader.

The three principal virtues of a programmer are Laziness, Impatience, and Hubris. See the Camel Book for why.

已知问题

The `-w` switch is not mandatory.

Perl is at the mercy of your machine's definitions of various operations such as type casting, `atoi()`, and floating-point output with `sprintf()`.

If your `stdio` requires a seek or eof between reads and writes on a particular stream, so does Perl. (This doesn't apply to `sysread()` and `syswrite()`.)

While none of the built-in data types have any arbitrary size limits (apart from memory size), there are still a few arbitrary limits: a given variable name may not be longer than 251 characters. Line numbers displayed by diagnostics are internally stored as short integers, so they are limited to a maximum of 65535 (higher numbers usually being affected by wraparound).

You may mail your bug reports (be sure to include full configuration information as output by the `myconfig` program in the perl source tree, or by `'perl -V'`) to `perlbug@perl.org`. If you've succeeded in compiling perl, the `perlbug` script in the `utils/` subdirectory can be used to help mail in a bug report.

Perl actually stands for Pathologically Eclectic Rubbish Lister, but don't tell anyone I said that.

引用名 pfexec, pfbash, pfcsh, pfksh, pfsh, pftcsh, pfzsh – 执行配置文件中的命令

用法概要

```
/usr/bin/pfexec command
/usr/bin/pfexec -P privspec command [ arg ]...
/usr/bin/pfsh [ options ] [ argument ]...
/usr/bin/pfcsh [ options ] [ argument ]...
/usr/bin/pfksh [ options ] [ argument ]...
```

描述

pfexec 程序可设置 PRIV_PFEEXEC 进程标志，并将当前进程标记为配置文件 shell。然后，它执行指定的命令。内核将查询 [exec_attr\(4\)](#) 数据库，并通过相应的属性执行。

将按 [user_attr\(4\)](#) 数据库中的用户条目指定的顺序搜索配置文件。如果同一命令出现在多个配置文件中，则配置文件 shell 将使用第一个匹配的条目。

使用第二种形式 (pfexec -P *privspec*)，用户可以获取在 [prof_attr\(4\)](#) 中授予用户配置文件的附加特权。在命令行上指定的特权是使用 [priv_str_to_set\(3C\)](#) 解析的。得到的特权将先与在 [prof_attr\(4\)](#) 中使用 `privs` 关键字为所有用户配置文件指定的特权合并在一起并添加到可继承集，然后再执行命令。

用法

pfexec 用于使用预定义的进程属性（例如特定用户或组 ID）执行命令。

有关配置文件 shell 的完整使用说明，请参见 [sh\(1\)](#)、[csh\(1\)](#) 和 [ksh\(1\)](#) 手册页。

示例

示例 1 获取附加的用户特权

```
example% pfexec -P all chown user file
```

此命令仅以分配给当前用户的所有特权运行 `chown user file`，不必使用所有特权。

退出状态

将返回以下退出值：

- 0 成功完成。
- 1 出现错误。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[bash\(1\)](#)、[csh\(1\)](#)、[ksh\(1\)](#)、[ksh88\(1\)](#)、[profiles\(1\)](#)、[sh\(1\)](#)、[tcsh\(1\)](#)、[zsh\(1\)](#)、[exec_attr\(4\)](#)、[prof_at](#)

引用名	pg – files perusal filter for CRTs
用法概要	pg [-number] [-p string] [-cefnrs] [+ linenumber] [+/ pattern /] [filename]...
描述	<p>The pg command is a filter that allows the examination of <i>filenames</i> one screenful at a time on a CRT. If the user types a RETURN, another page is displayed; other possibilities are listed below.</p> <p>This command is different from previous paginators in that it allows you to back up and review something that has already passed. The method for doing this is explained below.</p> <p>To determine terminal attributes, pg scans the terminfo(4) data base for the terminal type specified by the environment variable TERM. If TERM is not defined, the terminal type dumb is assumed.</p>
选项	<p>-number An integer specifying the size (in lines) of the window that pg is to use instead of the default. (On a terminal containing 24 lines, the default window size is 23).</p> <p>-pstring pg uses <i>string</i> as the prompt. If the prompt string contains a %d, the first occurrence of %d in the prompt will be replaced by the current page number when the prompt is issued. The default prompt string is “:”.</p> <p>-c Home the cursor and clear the screen before displaying each page. This option is ignored if <code>clear_screen</code> is not defined for this terminal type in the terminfo(4) data base.</p> <p>-e pg does <i>not</i> pause at the end of each file.</p> <p>-f Normally, pg splits lines longer than the screen width, but some sequences of characters in the text being displayed (for instance, escape sequences for underlining) generate undesirable results. The -f option inhibits pg from splitting lines.</p> <p>-n Normally, commands must be terminated by a <newline> character. This option causes an automatic end of command as soon as a command letter is entered.</p> <p>-r Restricted mode. The shell escape is disallowed. pg prints an error message but does not exit.</p> <p>-s pg prints all messages and prompts in the standard output mode (usually inverse video).</p> <p>+linenumber Start up at <i>linenumber</i>.</p> <p>+/pattern/ Start up at the first line containing the regular expression pattern.</p>

操作数

The following operands are supported:

filename A path name of a text file to be displayed. If no *filename* is given, or if it is `-`, the standard input is read.

用法

Commands

The responses that may be typed when `pg` pauses can be divided into three categories: those causing further perusal, those that search, and those that modify the perusal environment.

Commands that cause further perusal normally take a preceding *address*, an optionally signed number indicating the point from which further text should be displayed. This *address* is interpreted in either pages or lines depending on the command. A signed *address* specifies a point relative to the current page or line, and an unsigned *address* specifies an address relative to the beginning of the file. Each command has a default address that is used if none is provided.

The perusal commands and their defaults are as follows:

- (+1)<*newline*> or <*blank*> This causes one page to be displayed. The address is specified in pages.
- (+1) l With a relative address this causes `pg` to simulate scrolling the screen, forward or backward, the number of lines specified. With an absolute address this command prints a screenful beginning at the specified line.
- (+1) d or ^D Simulates scrolling half a screen forward or backward.
- if* Skip *i* screens of text.
- iz* Same as <*newline*> except that *i*, if present, becomes the new default number of lines per screenful.

The following perusal commands take no *address*.

- . or ^L Typing a single period causes the current page of text to be redisplayed.
- \$ Displays the last full window in the file. Use with caution when the input is a pipe.

The following commands are available for searching for text patterns in the text. The regular expressions are described on the [regex\(5\)](#) manual page. They must always be terminated by a <*newline*>, even if the `-n` option is specified.

- i/pattern/* Search forward for the *i*th (default *i*=1) occurrence of *pattern*. Searching begins immediately after the current page and continues to the end of the current file, without wrap-around.

i[^]*pattern*[^]

i?*pattern*? Search backwards for the *i*th (default *i*=1) occurrence of *pattern*. Searching begins immediately before the current page and continues to the beginning of the current file, without wrap-around. The [^] notation is useful for Adds 100 terminals which will not properly handle the ?.

After searching, pg will normally display the line found at the top of the screen. This can be modified by appending *m* or *b* to the search command to leave the line found in the middle or at the bottom of the window from now on. The suffix *t* can be used to restore the original situation.

The user of pg can modify the environment of perusal with the following commands:

<i>in</i>	Begin perusing the <i>i</i> th next file in the command line. The <i>i</i> is an unsigned number, default value is 1.
<i>ip</i>	Begin perusing the <i>i</i> th previous file in the command line. <i>i</i> is an unsigned number, default is 1.
<i>iw</i>	Display another window of text. If <i>i</i> is present, set the window size to <i>i</i> .
<i>s filename</i>	Save the input in the named file. Only the current file being perused is saved. The white space between the <i>s</i> and <i>filename</i> is optional. This command must always be terminated by a < <i>newline</i> >, even if the <i>-n</i> option is specified.
<i>h</i>	Help by displaying an abbreviated summary of available commands.
<i>q</i> or <i>Q</i>	Quit pg.
<i>!command</i>	<i>Command</i> is passed to the shell, whose name is taken from the SHELL environment variable. If this is not available, the default shell is used. This command must always be terminated by a < <i>newline</i> >, even if the <i>-n</i> option is specified.

At any time when output is being sent to the terminal, the user can hit the quit key (normally CTRL-**) or the interrupt (break) key. This causes pg to stop sending output, and display the prompt. The user may then enter one of the above commands in the normal manner. Unfortunately, some output is lost when this is done, because any characters waiting in the terminal's output queue are flushed when the quit signal occurs.

If the standard output is not a terminal, then pg acts just like [cat\(1\)](#), except that a header is printed before each file (if there is more than one).

Large File Behavior

See [largefile\(5\)](#) for the description of the behavior of pg when encountering files greater than or equal to 2 Gbyte (2³¹ bytes).

示例

示例 1 An example of the pg command.

The following command line uses pg to read the system news:

```
example% news | pg -p "(Page %d) :"
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of pg: LC_CTYPE, LC_MESSAGES, and NLSPATH.

The following environment variables affect the execution of pg:

COLUMNS Determine the horizontal screen size. If unset or NULL, use the value of TERM, the window size, baud rate, or some combination of these, to indicate the terminal type for the screen size calculation.

LINES Determine the number of lines to be displayed on the screen. If unset or NULL, use the value of TERM, the window size, baud rate, or some combination of these, to indicate the terminal type for the screen size calculation.

SHELL Determine the name of the command interpreter executed for a !command.

TERM Determine terminal attributes. Optionally attempt to search a system-dependent database, keyed on the value of the TERM environment variable. If no information is available, a terminal incapable of cursor-addressable movement is assumed.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

文件

/tmp/pg* temporary file when input is from a pipe

/usr/share/lib/terminfo/??/* terminal information database

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

另请参见

[cat\(1\)](#), [grep\(1\)](#), [more\(1\)](#), [terminfo\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#)

附注

While waiting for terminal input, pg responds to BREAK, CTRL-C, and CTRL-\ by terminating execution. Between prompts, however, these signals interrupt pg's current task and place the user in prompt mode. These should be used with caution when input is being read from a pipe, since an interrupt is likely to terminate the other commands in the pipeline.

The terminal /, ^, or ? may be omitted from the searching commands.

If terminal tabs are not set every eight positions, undesirable results may occur.

When using `pg` as a filter with another command that changes the terminal I/O options, terminal settings may not be restored correctly.

引用名 pgrep, pkill – 通过名称及其他属性查找进程或发送进程信号

用法概要

```
pgrep [-flvx] [-n | -o] [-d delim] [-P ppidlist]
      [-g pgrplist] [-s sidlist] [-u euidlist] [-U uidlist]
      [-G gidlist] [-J projidlist] [-t termlist]
      [-T taskidlist] [-c ctidlist] [-z zoneidlist]
      [pattern]

pkill [-signal] [-fvx] [-n | -o] [-P ppidlist]
      [-g pgrplist] [-s sidlist] [-u euidlist] [-U uidlist]
      [-G gidlist] [-J projidlist] [-t termlist]
      [-T taskidlist] [-c ctidlist] [-z zoneidlist]
      [pattern]
```

描述

Pgrep 实用程序检查系统上的活动进程，并报告属性与命令行中指定标准相匹配的进程 ID。每个进程 ID 打印为一个十进制值，并通过分隔符与下个 ID 分隔，其缺省为换行。对于每个属性选项，用户可以在命令行上指定一组可能值，由逗号分隔。例如，

```
pgrep -G other,daemon
```

与其真实组 ID 为 `other` OR `daemon` 的进程相匹配。如果指定多个标准选项，pgrep 与其属性符合标准选项的逻辑 AND 的进程相匹配。例如，

```
pgrep -G other,daemon -U root,daemon
```

与其属性为以下项的进程相匹配：

```
(真实组 ID 为 other OR daemon) AND
(真实用户 ID 为 root OR daemon)
```

Pkill 与 pgrep 功能几乎相同，除了由 `kill(1)` 向每个匹配进程发送信号，而不是打印其进程 ID。信号名称或数字可以指定为 pkill 的第一个命令行选项。

选项

支持以下选项：

- c *ctidlist* 仅与其合同 ID 位于列表中的进程相匹配。
- d *delim* 指定要在每个匹配进程 ID 之间打印的输出分隔符字符串。如果没有指定 -d 选项，缺省值为换行符。-d 选项仅当指定为 pgrep 的选项时才有效。
- f 常规表达模式应与完整进程参数字符串匹配（从 `/proc/nnnnn/psinfo` 文件的 `pr_psargs` 字段获取）。如果没有指定 -f 选项，表达模式仅匹配可执行文件的名称（从 `/proc/nnnnn/psinfo` 文件的 `pr_fname` 字段获取）。
- g *pgrplist* 仅与其组 ID 位于给定列表中的进程相匹配。如果组 0 包括在列表中，则解释为 pgrep 或 pkill 进程的组 ID。
- G *gidlist* 仅与其真实组 ID 位于给定列表中的进程相匹配。每个组 ID 可以指定为组名或数字组 ID。

- J *projidlist* 仅与其项目 ID 位于给定列表中的进程相匹配。每个项目 ID 可以指定为项目名或数字项目 ID。
- l 长输出格式。打印每个匹配进程的名称和进程 ID。根据是否指定 -f 选项（请参见上文），从 `pr_psargs` 或 `pr_fname` 字段获取进程名。-l 选项仅当指定为 `pgrep` 的选项时才有效。
- n 仅与符合所有其他指定匹配标准的最新（最新创建的）进程相匹配。不能和选项 -o 同时使用。
- o 仅与符合所有其他指定匹配标准的最旧（最早创建的）进程相匹配。不能和选项 -n 同时使用。
- P *ppidlist* 仅与其父进程 ID 位于给定列表中的进程相匹配。
- s *sidlist* 仅与其进程会话 ID 位于给定列表中的进程相匹配。如果 ID 0 包括在列表中，则解释为 `pgrep` 或 `pkill` 进程的会话 ID。
- t *termlist* 仅匹配与给定列表中的终端相关联的进程。每个终端将指定为 `/dev` 中终端设备路径名 "dev/" 的后缀。例如，`term/a` 或 `pts/0`。
- T *taskidlist* 仅与其任务 ID 位于给定列表中的进程相匹配。如果 ID 0 包括在列表中，则解释为 `pgrep` 或 `pkill` 进程的任务 ID。
- u *euidlist* 仅与其有效用户 ID 位于给定列表中的进程相匹配。每个用户 ID 可以指定为登录名或数字用户 ID。
- U *uidlist* 仅与其真实用户 ID 位于给定列表中的进程相匹配。每个用户 ID 可以指定为登录名或数字用户 ID。
- v 逆向读取匹配。匹配所有进程，除了符合指定匹配标准的进程。
- x 仅将其参数字符串或可执行文件名与指定模式完全匹配的进程视为匹配进程。当进程参数字符串或可执行文件名中的所有字符都与模式匹配时，才将该模式匹配视为完全匹配。
- z *zoneidlist* 仅与其区域 ID 位于给定列表中的进程相匹配。每个区域 ID 可以指定为区域名或数字区域 ID。此选项仅在全局区域中执行时有效。如果 `pkill` 功能用于向其他区域中的进程发送信号，该进程必须已声明 `{PRIV_PROC_ZONE}` 特权（请参见 [privileges\(5\)](#)）。
- signal* 指定发送到每个匹配进程的信号。如果未指定信号，将缺省发送 `SIGTERM`。*signal* 的值可以是 [signal.h\(3HEAD\)](#) 中定义的一个符号名称，没有 `SIG` 前缀，或者是对应信号数字的十进制值。-*signal* 选项仅当指定为 `pkill` 的第一个选项时才有效。

操作数

支持下列操作数：

- pattern* 指定扩展常规表达 (Extended Regular Expression, ERE) 模式，以匹配可执行文件名或完整进程参数字符串。有关 ERE 语法的完整说明，请参见 [regex\(5\)](#)。

示例 1 获取进程 ID
获取 sendmail 的进程 ID:

```
example% pgrep -x -u root sendmail
283
```

示例 2 终止进程
终止最新创建的 xterm:

```
example% pkill -n xterm
```

退出状态 将返回以下退出值:

- 0 一个或多个进程匹配。
- 1 无进程匹配。
- 2 指定的命令行选项无效。
- 3 发生了致命错误。

文件 `/proc/nnnnn/psinfo` 进程信息文件

属性 有关下列属性的说明, 请参见 [attributes\(5\)](#):

属性类型	属性值
可用性	system/core-os

另请参见 [kill\(1\)](#)、[proc\(1\)](#)、[ps\(1\)](#)、[truss\(1\)](#)、[kill\(2\)](#)、[signal.h\(3HEAD\)](#)、[proc\(4\)](#)、[attributes\(5\)](#)、[privl](#)

附注 这两个实用程序将 ERE *pattern* 参数与 `/proc/nnnnn/psinfo` 文件的 `pr_fname` 或 `pr_psargs` 字段进行匹配。这些字符串的长度根据 `<sys/procfs.h>` 中的定义进行限制。如果模式可匹配比当前限制长的字符串, 则可能无法匹配指定的进程集。

如果 *pattern* 参数包含 ERE 元字符 (同时也是 shell 元字符), 可能需要使用适当的 shell 引号将此模式括起来。

从不使用 `pgrep` 或 `pkill` 匹配死进程。

当前 `pgrep` 或 `pkill` 进程绝不会将自身视为可能匹配。

引用名	pkcs11_inspect – 打印证书内容
用法概要	<code>/usr/lib/pam_pkcs11/pkcs11_inspect [debug] [config_file=filename]</code>
描述	<p><code>pkcs11_inspect</code> 使用 <code>pam_pkcs11</code> 库基础结构获取及显示证书内容。</p> <p><code>pkcs11_inspect</code> 使用与 <code>pam_pkcs11(5)</code> PAM 模块相同的配置文件及参数。它装入定义的映射器模块，并使用它们在证书中查看需要的条目，即 <code>ms_mapper</code> 查找 <code>ms UPN</code> 条目等。</p> <p>映射器模块在证书中找到正确的条目时，它会转换为 UTF-8 并将其打印到 <code>stdout</code>。</p>
选项	<p>支持以下选项：</p> <p><code>config_file=filename</code> 设置配置文件。缺省值是 <code>/etc/security/pam_pkcs11/pam_pkcs11.conf</code>。</p> <p><code>debug</code> 启用调试输出。</p> <p>因为它使用与 <code>pam_pkcs11(5)</code> 相同的配置文件，所以所有 <code>pam_pkcs11</code> 选项均可用。其中某些选项在非 PAM 环境中没有意义，因此被忽略。某些映射器选项（<code>mapfile</code>、<code>ignorecase</code>）对证书内容不起作用，因此也被忽略。</p>
退出状态	<p>将返回以下退出值：</p> <p>0 成功完成。</p> <p> <code>pkcs11_inspect</code> 在 <code>stdout</code> 中打印为映射器找到的所有证书内容。</p> <p>1 出现错误。</p>
示例	<p>示例1 使用 <code>pkcs_inspect</code></p> <p>以下示例运行 <code>pkcs_inspect</code> 命令而没有任何选项：</p> <pre>% pkcs11_inspect</pre> <p>示例2 使用具有选项的 <code>pkcs_inspect</code></p> <p>以下示例运行 <code>pkcs_inspect</code> 命令时具有选项：</p> <pre>% pkcs11_inspect debug config_file=\${HOME}/.pam_pkcs11.conf</pre>
文件	<code>/etc/security/pam_pkcs11/pam_pkcs11.conf</code>
作者	Juan Antonio Martinez, jonsito@teleline.es
属性	有关下列属性的说明，请参见 attributes(5) ：

属性类型	属性值
可用性	library/security/pam/module/pam-pkcs11
接口稳定性	Uncommitted (未确定)

另请参见

[pklogin_finder\(1\)](#)、[attributes\(5\)](#)、[pam_pkcs11\(5\)](#)

PAM-PKCS11 用户手册，http://www.opensc-project.org/pam_pkcs11

引用名

pkg - 映像包管理系统的检索客户端

用法概要

```

/usr/bin/pkg [options] command [cmd_options] [operands]
/usr/bin/pkg refresh [-q] [--full] [publisher ...]

/usr/bin/pkg install [-nvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-be-activate] [--no-index]
  [--no-refresh] [--no-backup-be | --require-backup-be]
  [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  [--reject pkg_fmri_pattern ...] pkg_fmri_pattern ...

/usr/bin/pkg uninstall [-nvq] [-C n] [--no-be-activate]
  [--no-index] [--no-backup-be | --require-backup-be]
  [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  pkg_fmri_pattern ...

/usr/bin/pkg update [-fnvq] [-C n] [-g path_or_uri ...]
  [--accept] [--licenses] [--no-be-activate] [--no-index]
  [--no-refresh] [--no-backup-be | --require-backup-be]
  [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  [--reject pkg_fmri_pattern ...] [pkg_fmri_pattern ...]

/usr/bin/pkg list [-Hafnsuv] [-g path_or_uri ...]
  [--no-refresh] [pkg_fmri_pattern ...]

/usr/bin/pkg info [-lr] [-g path_or_uri ...] [--license]
  [pkg_fmri_pattern ...]

/usr/bin/pkg contents [-Hmr] [-a attribute=pattern ...]
  [-g path_or_uri ...] [-o attribute ...] [-s sort_key]
  [-t action_name ...] [pkg_fmri_pattern ...]

/usr/bin/pkg search [-HIaf\lpr] [-o attribute ...]
  [-s repo_uri] query

/usr/bin/pkg verify [-Hqv] [pkg_fmri_pattern ...]

/usr/bin/pkg fix [--accept] [--licenses]
  [pkg_fmri_pattern ...]

/usr/bin/pkg revert [-nv] [--no-be-activate]
  [--no-backup-be | --require-backup-be]
  [--backup-be-name name]
  [--deny-new-be | --require-new-be] [--be-name name]
  (--tagged tag-name ... | path-to-file ...)

/usr/bin/pkg mediator [-aH] [-F format] [mediator ...]

usr/bin/pkg set-mediator [-nv] [-I implementation]
  [-V version] [--no-be-activate]
  [--no-backup-be | --require-backup-be]

```

```

    [--backup-be-name name]
    [--deny-new-be | --require-new-be] [--be-name name]
    mediator ...

/usr/bin/pkg unset-mediator [-nIV] [--no-be-activate]
    [--no-backup-be | --require-backup-be]
    [--backup-be-name name]
    [--deny-new-be | --require-new-be] [--be-name name]
    mediator ...

/usr/bin/pkg variant [-H] [variant.variant_name ...]

/usr/bin/pkg change-variant [-nvq] [-C n] [-g path_or_uri ...]
    [--accept] [--licenses] [--no-be-activate]
    [--no-backup-be | --require-backup-be]
    [--backup-be-name name]
    [--deny-new-be | --require-new-be] [--be-name name]
    variant_name=value ...

/usr/bin/pkg facet [-H] [facet_name ...]

/usr/bin/pkg change-facet [-nvq] [-C n] [-g path_or_uri ...]
    [--accept] [--licenses] [--no-be-activate]
    [--no-backup-be | --require-backup-be]
    [--backup-be-name name]
    [--deny-new-be | --require-new-be] [--be-name name]
    facet_name=[True|False|None] ...

/usr/bin/pkg avoid [pkg_fmri_pattern ...]

/usr/bin/pkg unavoid [pkg_fmri_pattern ...]

/usr/bin/pkg freeze [-n] [-c reason] [pkg_fmri_pattern] ...

/usr/bin/pkg unfreeze [-n] [pkg_name_pattern] ...

/usr/bin/pkg property [-H] [propname ...]

/usr/bin/pkg set-property propname propvalue

/usr/bin/pkg add-property-value propname propvalue

/usr/bin/pkg remove-property-value propname propvalue

/usr/bin/pkg unset-property propname ...

/usr/bin/pkg publisher [-HPn] [-F format] [publisher ...]

/usr/bin/pkg set-publisher [-Ped] [-k ssl_key] [-c ssl_cert]
    [-g origin_to_add | --add-origin origin_to_add ...]
    [-G origin_to_remove | --remove-origin origin_to_remove ...]
    [-m mirror_to_add | --add-mirror mirror_to_add ...]
    [-M mirror_to_remove | --remove-mirror mirror_to_remove ...]
    [--enable] [--disable] [--no-refresh] [--reset-uuid]
    [--non-sticky] [--sticky] [--search-after publisher]
    [--search-before publisher] [--search-first]

```



```

[--approve-ca-cert path_to_CA]
[--revoke-ca-cert hash_of_CA_to_remove]
[--unset-ca-cert hash_of_CA_to_remove]
[--set-property name_of_property=value]
[--add-property-value name_of_property=value_to_add]
[--remove-property-value name_of_property=value_to_remove]
[--unset-property name_of_property_to_delete]
[--proxy proxy_to_use] publisher

/usr/bin/pkg set-publisher -p repo_uri [-Ped]
[-k ssl_key] [-c ssl_cert] [--non-sticky] [--sticky]
[--search-after publisher] [--search-before publisher]
[--search-first] [--approve-ca-cert path_to_CA]
[--revoke-ca-cert hash_of_CA_to_remove]
[--unset-ca-cert hash_of_CA_to_remove]
[--set-property name_of_property=value]
[--add-property-value name_of_property=value_to_add]
[--remove-property-value name_of_property=value_to_remove]
[--unset-property name_of_property_to_delete]
[--proxy proxy_to_use] [publisher]

/usr/bin/pkg unset-publisher publisher ...

/usr/bin/pkg history [-Hn\] [-t [time | time-time],...]
[-o column,...] [-n number]

/usr/bin/pkg purge-history

/usr/bin/pkg rebuild-index

/usr/bin/pkg update-format

/usr/bin/pkg version

/usr/bin/pkg help

/usr/bin/pkg image-create [-FPufz] [--force]
[--full | --partial | --user] [--zone]
[-k ssl_key] [-c ssl_cert] [--no-refresh]
[--variant variant_name=value ...]
[-g path_or_uri | --origin path_or_uri ...]
[-m uri | --mirror uri ...]
[--set-property name_of_property=value]
[--facet facet_name=(True|False) ...]
[(-p | --publisher) [name]=repo_uri] dir

```

描述

`pkg` 是映像包管理系统的检索客户端。在有效配置下，可以调用 `pkg` 来为要安装的软件包创建位置（称为“映像”），然后将软件包安装到这些映像中。软件包由发布者发布。发布者可使其软件包在一个或多个系统信息库上可用，或者在软件包归档中可用。`pkg` 从发布者的系统信息库或软件包归档中检索软件包，然后将软件包安装到映像中。

发布者名称将个人、个人组或组织标识为一个或多个软件包的源。为避免发布者名称冲突并有帮于标识发布者，最佳做法是使用代表发布软件包实体的域名作为发布者名称。

系统信息库是客户端可在其中发布和检索软件包内容（该软件包包含的文件，例如程序和文档）和元数据（有关该软件包的信息，例如其名称和描述）的位置。例如，有一个名为 `example.org` 的发布者，其系统信息库位于 URI `http://example.org/repository`。

`pkg` 还可以卸载软件包、刷新发布者元数据（例如可用软件包的列表）、验证映像中的软件包安装，以及查询映像中的各个标记。也可以对 `pkg(5)` 系统信息库执行这些查询。

映像有三种类型：完整映像：能够提供完整的系统；部分映像：与完整映像（父映像）链接，但本身无法提供完整的系统；用户映像。

选项

支持以下选项：

-?

--help

显示用法消息。

-R *dir*

--image-dir *dir*

对根目录为 *dir* 的映像进行操作。如果未根据环境指定或确定目录，则缺省值为 `/`。有关更多信息，请参见“环境变量”部分。

子命令

支持以下子命令：

`pkg refresh [-q] [- -full] [publisher ...]`

对于所有发布者，更新客户机可用软件包列表和发布者元数据。

publisher

仅对指定的发布者，更新客户机可用软件包列表和发布者元数据。

-q

在执行请求的操作期间隐藏进度消息。

--full

强制完全检索所有发布者元数据（而不是尝试增量更新），并请求操作期间使用的所有代理忽略高速缓存的数据。此选项用于进行故障排除，正常情况下不应使用。

`pkg install [-nvq] [-C n] [-g path_or_uri ...] [--accept] [--licenses] [--no-be-activate] [--no-index] [--no-refresh] [--no-backup-be | --require-backup-be] [--backup-be-name name] [--deny-new-be | --require-new-be] [--be-name name] [--reject pkg_fmri_pattern ...] pkg_fmri_pattern ...`

安装指定软件包并将其更新到最新版本，该版本与映像中安装的软件包所允许的 *pkg_fmri_pattern* 匹配。要显式请求安装软件包的最新版本，请在 *pkg_fmri_pattern* 的版本部分使用 `latest`。例如，指定 `vim@latest`。

软件包基于发布者搜索顺序和粘性进行选择。有关搜索顺序和粘性的信息，请参见 `pkg publisher` 和 `pkg set-publisher` 命令。如果 `pkg_fmri_pattern` 未指定发布者，则会将第一个提供匹配软件包的发布者用作安装源。如果该发布者未提供可以安装在此映像中的软件包版本，则安装操作将失败。使用 `pkg list -a` 命令可查看哪些发布者提供可以安装在此映像中的软件包版本。

指定了多个 `pkg_fmri_pattern` 时，如果指定的软件包中有任何一个无法在此映像中安装，则指定的所有软件包都不会安装。

安装过程中，某些配置文件可能被重命名或替换。有关软件包系统如何确定要保留的文件，以及在软件包操作期间如何保留这些文件的更多信息，请参见 `pkg(5)` 手册页中的“文件操作”。

如果某个软件包在避免列表中，则安装该软件包会将它从该列表中删除。

`-g path_or_uri`

暂时将指定的软件包系统信息库或归档文件添加到从中检索软件包数据的映像的源列表内。需要客户机 SSL 证书的系统信息库不能与此选项一起使用。可以多次指定此选项。

确定要使用的软件包版本时，优先选择映像中配置的发布者，而非指定 `path_or_uri` 源中找到的发布者。如果映像中配置的发布者和 `path_or_uri` 源都提供要安装的软件包版本，则客户机将从 `path_or_uri` 源中检索该软件包的内容。安装或更新后，未在映像中配置的发布者提供的所有软件包都将添加到映像配置中，但添加的这些软件包没有源。使用 `pkg publisher` 命令可查看在映像中配置的发布者。

`-n`

试运行操作而不进行软件包更改。

`-v`

在执行请求的操作期间发出详细进度消息，并显示详细的规划信息（例如更改侧面、中介和变量）。可以多次指定此选项，以增加显示的规划信息量。

`-q`

在执行请求的操作期间隐藏进度消息。

`-C n`

指定要并行更新的子映像数。在递归检查子映像（通常为区域）时，最多可以并行更新 n 个子映像。并行更新的缺省子映像数为 1。如果 n 为 0 或负数，则并行更新所有子映像。另请参见“环境变量”部分中的 `PKG_CONCURRENCY`。

`--accept`

表示同意并接受所更新或安装的软件包的许可证条款。如果存在任何需要您接受的软件包许可证，而您未提供此选项，则安装操作将失败。

`--licenses`

显示在此操作过程中所安装或更新的软件包的所有许可证。

- `--no-backup-be`
不创建备份引导环境。
- `--no-be-activate`
如果创建了一个引导环境，不将其设置为下次引导时的活动引导环境。有关更多信息，请参见 `beadm(1M)` 手册页。
- `--no-index`
不在操作成功完成后更新搜索索引。
- `--no-refresh`
不尝试联系映像发布者的系统信息库来检索最新的可用软件包列表和其他元数据。
- `--backup-be-name name`
使用给定参数命名创建的备份引导环境。使用 `--backup-be-name` 表示 `--require-backup-be`。另请参见 `beadm(1M)` 手册页。
- `--be-name name`
将新创建的引导环境重命名为给定的参数。使用 `--be-name` 表示 `--require-new-be`。另请参见 `beadm(1M)` 手册页。
- `--require-backup-be`
如果不创建新的引导环境，则始终创建一个备份引导环境。如果不使用该选项，则根据映像策略创建备份引导环境。有关何时自动创建备份引导环境的说明，请参见下文“映像属性”中的 `be-policy`。
- `--require-new-be`
始终创建新的引导环境。如果不使用该选项，则根据映像策略创建引导环境。有关何时自动创建引导环境的说明，请参见下文“映像属性”中的 `be-policy`。该选项不能与 `-require-backup-be` 一起使用。
- `--deny-new-be`
不创建新的引导环境。如果需要新的引导环境，则不执行此操作。
- `--reject pkg_fmri_pattern`
阻止安装名称与给定模式匹配的软件包。如果已安装匹配的软件包，则在此操作过程会将其删除。作为组依赖性目标的被拒绝软件包将放置在避免列表中。

```
pkg uninstall [-nvq] [-C n] [--no-be-activate] [--no-index] [--no-backup-be |
--require-backup-be] [--backup-be-name name] [--deny-new-be | --require-new-be]
[--be-name name] pkg_fmri_pattern ...
```

删除与 `pkg_fmri_pattern` 匹配的已安装软件包。

如果某个软件包是组依赖性的主体，则卸载该软件包会将它放置在避免列表中。请参见下文的 `avoid` 子命令。

有关选项的说明，请参见上述 `install` 命令。

在命令输出中，请注意任何指出已创建新引导环境的消息。如果已创建并激活新引导环境，缺省情况下，下次重新引导时会引导该环境。有关管理引导环境的信息，请参见 `beadm(1M)` 手册页。

```
pkg update [-fnvq] [-C n] [-g path_or_uri ...] [--accept] [--licenses] [--no-be-activate]
[--no-index] [--no-refresh] [--no-backup-be | --require-backup-be]
[--backup-be-name name] [--deny-new-be | --require-new-be] [--be-name name]
[--reject pkg_fmri_pattern ...] [pkg_fmri_pattern ...]
```

将当前映像中安装的所有软件包更新到最新版本，该版本需符合已安装的软件包和发布者配置对系统施加的约束。

pkg_fmri_pattern

仅更新当前映像中安装的指定软件包。如果所提供的 *pkg_fmri_pattern* 模式中包含星号(*)，则会按如上所述更新当前映像中安装的所有软件包。

要显式请求安装软件包的最新版本，请在 *pkg_fmri_pattern* 的版本部分使用 `latest`。例如，指定 `vim@latest`。

可以指定比已安装版本更高或更低的版本，以便对特定软件包执行就地降级或升级。不支持跨越软件包重命名边界或过时边界更新特定的软件包。

作为将要降级的软件包的一部分、自原始版本安装以来更改了的任何保留配置文件，将使用扩展名 `.update` 进行重命名。有关软件包系统如何确定要保留的文件，以及在软件包升级期间如何保留这些文件的更多信息，请参见 `pkg(5)` 手册页中的“文件操作”。

指定了多个 *pkg_fmri_pattern* 时，如果指定的软件包中有任何一个无法在此映像中更新，则指定的所有软件包都不会更新。

-f

不在更新已安装的所有软件包时执行客户机最新状态检查。

对于其他所有选项，请参见上面的 `install` 命令。

在命令输出中，请注意任何指出已创建新引导环境的消息。如果已创建并激活新引导环境，在未指定 `--no-be-activate` 选项时，缺省情况下，下次重新引导时会引导该环境。有关管理引导环境的信息，请参见 `beadm(1M)` 手册页。

```
pkg list [-Hafnsuv] [-g path_or_uri ...] [--no-refresh] [pkg_fmri_pattern ...]
```

显示当前映像中安装的所有软件包的列表，包括版本和安装状态等信息。缺省情况下，会排除不同体系结构或区域类型的软件包变量。输出通常包括三个列：

NAME (PUBLISHER)	VERSION	IFO
system/core-os	0.5.11-0.175.0.0.0.2.1	i--
x11/wm/fvwm (fvwm.org)	2.6.5	i--

第一列包含软件包的名称。如果安装（或者提供）该软件包的发布者在发布者搜索顺序上不是第一个，则该发布者名称将列在软件包名称的后面，并括在括号中。第二列包含软件包的发行版本和分支版本。有关发行版本和分支版本以及变量的信息，请参见 `pkg(5)` 手册页。

最后一列包含一组标志，用于显示软件包的状态：

- **I** 列中的 **i** 表明软件包已安装。
- **F** 列中的 **f** 表明软件包已冻结。
- **O** 列中的 **o** 表明软件包已过时。**O** 列中的 **r** 表明软件包已重命名（一种形式的过时）。

pkg_fmri_pattern

仅列出指定的软件包。

-H

在列出时省略标题。

-a

列出已安装的软件包，以及可用于安装的软件包的最新版本。如果软件包是已安装 **incorporation** 和映像变量所允许的，则认为这些软件包可用于安装。如果指定了一个或多个模式，则会列出与指定模式匹配的且任何已安装 **incorporation** 和映像变量允许的最新版本。如果不使用 **-a**，则仅列出已安装的软件包。

-af

列出所有变量的所有软件包的所有版本，而不管 **incorporation** 约束或安装状态如何。使用这些选项时，要显式列出某个软件包的最新版本，请在 *pkg_fmri_pattern* 的版本部分使用 **latest**。例如，指定 **vim@latest**。

-g path_or_uri

使用指定的软件包系统信息库或归档文件作为操作的软件包数据源。需要客户机 SSL 证书的系统信息库不能与此选项一起使用。可以多次指定此选项。如果未指定 **-n**，则可使用 **-g** 表示 **-a**。

-n

显示所有已知软件包的最新版本，而不管安装状态如何。

-s

显示单行短格式，用于提供软件包名称和摘要。此选项可与 **-a**、**-n**、**-u** 或 **-v** 一起使用。

-u

仅列出有更高版本可用的软件包。此选项不能与 **-g** 一起使用。

-v

在第一列中显示完整的软件包 FMRI，包括发布者和完整版本（**VERSION** 列将消失）。此选项可与 **-a**、**-n** 或 **-u** 一起使用。

--no-refresh

不尝试联系映像发布者的系统信息库来检索最新的可用软件包列表。

pkg info [-lr] [-g path_or_uri ...] [--license] [pkg_fmri_pattern ...]
以用户可读的格式显示当前映像中安装的所有软件包的相关信息。

pkg_fmri_pattern

仅显示指定软件包的相关信息。

-g *path_or_uri*

使用指定的软件包系统信息库或归档文件作为操作的软件包数据源。需要客户机 SSL 证书的系统信息库不能与此选项一起使用。可以多次指定此选项。使用 -g 表示 -r。

-l

仅显示已安装的软件包的相关信息。这是缺省值。

-r

根据最新的可用版本匹配软件包，并从映像中配置的发布者系统信息库检索当前未安装的软件包的相关信息（如有必要）。使用此选项时，必须至少指定一个软件包。如果不指定 -r，则缺省情况下仅显示已安装的软件包。

--license

显示软件包的许可证文本。此选项可与 -l 或 -r 结合使用。

pkg contents [-Hmr] [-a *attribute=pattern ...*] [-g *path_or_uri ...*] [-o *attribute,...*] [-s *sort_key*] [-t *action_name ...*] [*pkg_fmri_pattern ...*]

显示映像中安装的所有软件包的内容（操作属性）。未指定选项时，显示当前映像中安装的操作的 *path* 属性值，按属性值的字母顺序排列。有关操作及其属性的信息，请参见 [pkg\(5\)](#) 手册页中的“操作”。另请参见以下伪属性名称列表。

pkg_fmri_pattern

仅显示指定软件包的内容。

-H

在输出中省略标题。

-a *attribute=pattern*

将输出限制为特定操作，这些操作具有在选项参数中指定的属性，且其属性值与选项参数中的 (*glob*) 模式匹配（位于属性名称之后，带一个等号）。如果指定了多个 -a 选项，则会显示与其中任一选项匹配的操作。

-g *path_or_uri*

显示指定软件包系统信息库或归档文件中可以安装在此映像中的软件包的信息。需要客户机 SSL 证书的系统信息库不能与此选项一起使用。可以安装的软件包包含当前已安装的软件包以及其他满足此映像安装条件（例如变量和侧面限制）的软件包。可以多次指定此选项。使用 -g 表示 -r。

-m

显示指定软件包中所有操作的所有属性，包括无法安装在此映像中的操作。

-o *attribute*

显示指定属性，按列出的第一个属性的值排序。可以多次指定 -o 选项；也可以通过使用逗号分隔属性名称，将多个属性指定为一个 -o 选项的参数。仅显示包含所请求属性的操作。

-r

显示此映像中配置的发布者系统信息库中可以安装在此映像中的软件包的最新可用版本信息。可以安装的软件包包含当前已安装的软件包以及其他满足此映像安装条件（例如变量和侧面限制）的软件包。使用此选项时，必须至少指定一个软件包。

-s *sort_key*

按指定的操作属性对操作进行排序。如果未提供此选项，则缺省设置是按照路径进行排序或者按照 -o 选项指定的第一个属性进行排序。可以多次指定 -s 选项。

-t *action_name*

仅列出指定的操作。可以在一个逗号分隔的列表中指定多个操作。*action_name* 的值是 pkg(5) 手册页的“操作”中列出的操作之一，例如 `file`、`directory`、`driver`、`depend`、`set`。可以多次指定此选项。

为方便起见，可以使用多个特殊的伪属性名称：

`action.hash`

操作的散列值（如果该操作承载了有效负荷）。

`action.key`

该操作的关键属性值。例如，对于 `file` 操作，关键属性是文件的路径。某些操作不具备关键属性。

`action.name`

操作的名称。例如，对于某个文件操作，这是 `file`。

`action.raw`

匹配操作的所有属性。

`pkg.fmri`

包含操作的软件包的完整 FMRI，例如

`pkg://solaris/group/feature/amp@0.5.11,5.11-0.175.0.0.0.2.1:20120705T153434Z。`

`pkg.name`

包含操作的软件包的名称，例如 `web/amp`。

`pkg.publisher`

包含操作的软件包的发布者，例如 `solaris`。

`pkg.shortfmri`

包含操作的软件包的短格式 FMRI，例如

`pkg://solaris/group/feature/amp@0.5.11,5.11-0.175。`

`contents` 和 `search` 子命令是彼此相关的：两者都可在系统中查询软件包的内容。`contents` 子命令显示一个或多个已安装或可安装软件包中的操作，根据指定的选项过滤输出。`search` 子命令从另一个方向处理查询，显示包含用户提供的标记的所有软件包的名称。

每个子命令都能够表达另一个子命令也能表达的某些查询。应该谨慎选择所需的子命令，因为使用其中一个子命令来表达某个给定查询，可能比使用另一个子命令更为自然。

`pkg search [-HIaflpr] [-o attribute,...] [-s repo_uri] query`
搜索 *query* 的匹配项并显示结果。请参见下文的 *query* 说明。

-H
在输出中省略标题。

-I
使用区分大小写的搜索。

-a
执行搜索并显示有关匹配操作的信息。这是缺省值。

-f
显示所有结果，而不考虑软件包版本。缺省情况下，`search` 会从结果中去除低于当前安装版本的软件包以及当前 `incorporation` 中排除的软件包版本。

-l
搜索映像的已安装软件包。

可以同时指定 `-l` 和 `-r`（或 `-s`），在此情况下，将会同时执行本地搜索和远程搜索。

-p
显示其某些操作与每个查询词匹配的软件包。使用此选项相当于在查询中的每个词两侧添加尖括号 (`<>`)。有关 `<>` 运算符的更多说明，请参见下文的 *query*。

-r
搜索对应于映像发布者的系统信息库。这是缺省值。

可以同时指定 `-l` 和 `-r`（或 `-s`），在此情况下，将会同时执行本地搜索和远程搜索。

-o attribute
控制结果列。可以多次指定 `-o` 选项；也可以通过使用逗号分隔属性名称，将多个属性指定为一个 `-o` 选项的参数。除了上面概述的伪属性外，还为搜索结果定义了下列属性：

`search.match`
与搜索查询匹配的字符串。

`search.match_type`
包含与搜索查询匹配的字符串的属性。

-s repo_uri
搜索位于给定 URI 的 `pkg(5)` 系统信息库。可以多次指定此选项。不支持软件包归档。

query

缺省情况下，会将 *query* 解释为要精确匹配的一系列条件。可将 *?* 和 *** 字符用作 glob(3C) 式通配符，以更灵活地获取查询匹配项。

除了支持简单的标记匹配和通配符搜索外，还支持更复杂的查询语言。可通过使用单引号或双引号（*'* 或 *"*）来搜索短语。请务必考虑所用的 shell，使 pkg 能够真正识别 *'* 或 *"*。

支持使用 AND 和 OR 的布尔搜索。

要为哪些标记建立索引是与操作相关的，不过可以包括内容散列和路径名称。有关操作及其属性的信息，请参见 pkg(5) 手册页中的“操作”。另请参见上面 pkg contents 和 -o 中的伪属性名称列表。

结构化查询支持以下语法：

```
pkg_name:action_name:index:token
```

action_name 的值是 pkg(5) 手册页的“操作”中列出的操作之一。*index* 是操作的属性。*index* 的值必须与 *token* 匹配。

并非所有操作属性都是可搜索的。例如，*mode* 是 *file* 操作的一个属性，但是 *mode* 不是 *index* 的有效值。

index 的某些值并不是操作属性，而是从其他属性派生的值。例如，*index* 可以是 *basename*，其不是任何操作的属性，而是派生自 *file* 或 *dir* 操作的 *path* 属性，它采用了路径的最后组成部分。

不同的操作类型有不同的 *index* 有效值。本文档并未列出所有可能的值。一些更为有用的 *index* 值包括，对应于文件系统操作的 *basename* 和 *path*、对应于 *depend* 操作的依赖性类型（例如 *require*、*optional*、*group*），以及对应于 *driver* 操作的 *driver_name* 和 *alias*。

index 的一个特殊值是对应于 *set* 操作的 *name* 属性值。在这种情况下，*token* 与对应于指定 *name* 属性的 *value* 属性的值相匹配。例如，以下搜索将查找分类为 "Development/Databases" 或 "System/Databases" 的软件包。请参见“示例”部分中查找 SMF 服务的示例。

```
$ pkg search info.classification:databases
```

结构化查询中缺少的字段使用隐式通配符。搜索 *basename:pkg* 将会匹配 *index* 为 *basename* 且与 *token* *pkg* 匹配的所有软件包中的所有操作，如以下部分输出中所示：

```
$ pkg search basename:pkg
INDEX      ACTION VALUE          PACKAGE
basename  dir      usr/share/pkg      pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1
basename  dir      var/sadm/pkg       pkg:/package/svr4@0.5.11-0.175.0.0.0.2.1
basename  dir      var/spool/pkg      pkg:/package/svr4@0.5.11-0.175.0.0.0.2.1
basename  file     usr/bin/pkg        pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1
```

添加其他字段将会缩小搜索范围，如以下完整输出中所示：

```
$ pkg search file:basename:pkg
INDEX   ACTION VALUE      PACKAGE
basename file   usr/bin/pkg pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1
```

pkg_name 和 *token* 字段中支持显式通配符。*action_name* 和 *index* 必须完全匹配。

有关搜索文件和依赖项的示例，请参见“示例”部分。

要将操作转换为包含这些操作的软件包，请使用 <>，如以下部分输出中所示：

```
$ pkg search \

```

使用 *-a* 选项（缺省设置）时，搜索 *token* 会导致返回与 *token* 匹配的操作的信息，而搜索 <token> 会导致返回一个软件包列表，这些软件包包含与 *token* 匹配的操作。

pkg verify [-Hqv] [*pkg_fmri_pattern* ...]

验证当前映像中安装的所有软件包的安装情况。如果相关发布者的当前签名策略不是 *ignore*，则会根据策略验证每个软件包的签名。有关如何应用签名策略的说明，请参见下文“映像属性”中的 *signature-policy*。

pkg_fmri_pattern

仅验证当前映像中安装的指定软件包的安装情况。

-H

在验证输出中省略标题。

-q

如果出现任何致命错误，则不显示任何信息，而是返回失败消息。

-v

包括有关软件包的信息性消息。

pkg fix [--accept] [--licenses] [*pkg_fmri_pattern* ...]

修复 *pkg verify* 报告的任何错误。已安装软件包内容的验证基于定制的内容分析，该分析可能会返回与其他程序不同的结果。

pkg_fmri_pattern

仅修复 *pkg verify* 报告的有关当前映像中安装的指定软件包的错误。

--accept

表示同意并接受所更新或安装的软件包的许可证条款。如果存在任何需要您接受的软件包许可证，而您未提供此选项，则操作将失败。

--licenses

显示在此操作过程中要安装或更新的软件包的所有许可证。

```
pkg revert [-nv] [- -no-be-activate] [--no-backup-be | --require-backup-be ]
[--backup-be-name name] [--deny-new-be | --require-new-be] [--be-name name]
(--tagged tag-name ... | path-to-file ...)
```

将 pkg(5) 软件包提供的文件恢复为其交付时的状态。文件所有权和保护也会得到恢复。

注意 – 将某些可编辑文件恢复为其缺省值可能会使系统无法引导，或导致其他故障。

--tagged *tag-name*
恢复所有标记有 *tag-name* 的文件。

path-to-file
恢复指定的文件。

对于其他所有选项，请参见上面的 `install` 命令。

```
pkg mediator [-aH] [-F format] [mediator ...]
```

显示所有中介的当前选定版本和/或实现。

mediator
仅显示指定中介的当前选定版本和/或实现。

-a
列出可为当前安装的软件包设置的中介。

-F
指定备用输出格式。当前，只有 `tsv`（Tab Separated Values，制表符分隔值）有效。

-H
在列出时省略标题。

```
pkg set-mediator [-nv] [-I implementation] [-V version] [--no-be-activate]
[--no-backup-be | --require-backup-be] [--backup-be-name name] [--deny-new-be |
--require-new-be] [--be-name name] mediator ...
```

设置当前映像中指定中介的版本和/或实现。

-I *implementation*
设置要使用的中介接口的实现。缺省情况下，如果未指定版本，则允许所有实现版本。要指定一个没有版本的实现，请附加 `@` 符号。

-v *version*
设置要使用的中介接口的版本。

如果指定的中介版本和/或实现当前不可用，则会删除使用指定中介的任何链接。

对于其他所有选项，请参见上面的 `install` 命令。

```
pkg unset-mediator [-nvIV] [--no-be-activate] [--no-backup-be |
--require-backup-be] [--backup-be-name name] [--deny-new-be | --require-new-be]
[--be-name name] mediator ...
```

将指定中介的版本和/或实现恢复为系统缺省值。

-I
仅恢复中介接口的实现。

-V
仅恢复中介接口的版本。

对于其他所有选项，请参见上面的 `install` 命令。

```
pkg variant [-H] [ variant.variant_name ...]
```

显示在此映像中设置的所有变量的当前值。有关变量的更多信息，请参见 `pkg(5)` 手册页中的“侧面和变量”。

variant.variant_name
仅显示在此映像中设置的指定变量的当前值。

-H
在列出时省略标题。

```
pkg change-variant [-nvq] [-C n] [-g path_or_uri ...] [--accept] [--licenses]
[--no-be-activate] [--no-backup-be | --require-backup-be] [--backup-be-name name]
[--deny-new-be | --require-new-be] [--be-name name] variant_name=value ...
```

更改在当前映像中设置的指定变量的值。

更改变量的值可能会导致删除、更新或安装某些软件包内容。更改变量值还可能会导致安装、更新或删除整个软件包，以满足新的映像配置。有关变量的更多信息，请参见 `pkg(5)` 手册页中的“侧面和变量”。

有关选项的说明，请参见上述 `install` 命令。

```
pkg facet [-H] [ facet_name ...]
```

显示使用 `pkg change-facet` 命令在此映像中显式设置的所有侧面的当前值。有关侧面的更多信息，请参见 `pkg(5)` 手册页中的“侧面和变量”。

facet_name
仅显示在此映像中设置的指定侧面的当前值。

-H
在列出时省略标题。

```
pkg change-facet [-nvq] [-C n] [-g path_or_uri ...] [--accept] [--licenses]
[--no-be-activate] [--no-backup-be | --require-backup-be] [--backup-be-name name]
[--deny-new-be | --require-new-be] [--be-name name] facet_name=[ True|False|None] ...
```

更改当前映像中设置的指定侧面的值。

可以将侧面设置为 `True` 或 `False`。将侧面设置为 `None` 时，系统会将缺省值 `True` 应用于该侧面；因此将安装任何由此侧面约束的操作。有关操作的信息，请参见 `pkg(5)` 手册页中的“操作”。

更改侧面的值可能会导致删除、更新或安装某些软件包内容。更改侧面值还可能会导致安装、更新或删除整个软件包，以满足新的映像配置。有关侧面的更多信息，请参见 `pkg(5)` 手册页中的“侧面和变量”。

有关选项的说明，请参见上述 `install` 命令。

`pkg avoid [pkg_fmri_pattern ...]`

显示每个避免安装的软件包，以及对该软件包存在组依赖性的任何软件包。

如果需要满足 `require` 依赖性，则会安装避免列表中的软件包。如果删除了该依赖性，则会卸载相应的软件包。

pkg_fmri_pattern

如果指定的软件包是某个组依赖性的目标，则会通过将匹配指定模式的软件包名称放置在避免列表中来避免安装这些软件包。只能避免安装当前尚未安装的软件包。如果某个软件包当前是某个组依赖性的目标，则卸载该软件包会将它放置在避免列表中。

`pkg unavoid [pkg_fmri_pattern ...]`

显示避免安装的软件包的列表。

pkg_fmri_pattern

从避免列表中删除指定的软件包。使用此子命令无法删除避免列表中与某个已安装软件包的组依赖性相匹配的软件包。要从避免列表中删除与某个组依赖性相匹配的软件包，请安装该软件包。

`pkg freeze [-n] [-c reason] [pkg_fmri_pattern] ...`

显示有关当前已冻结软件包的信息：软件包名称、版本、软件包冻结时间以及任何冻结软件包的相关原因。

冻结某个软件包不会阻止删除该软件包。删除软件包时不会显示警告。

pkg_fmri_pattern

将指定的软件包冻结到指定的版本。如果未提供版本，则必须安装该软件包，然后将它冻结在该已安装版本。冻结一个已经冻结的软件包会将冻结版本替换为新指定的版本。

如果安装或更新冻结的软件包，则其最终版本必须与冻结时使用的版本匹配。例如，如果某个软件包在冻结时的版本为 1.2，则可以将它更新到 1.2.1、1.2.9、1.2.0.0.1，等等。但该软件包的最终版本不能为 1.3 或

1.1。*pkg_fmri_pattern* 中提供的发布者用于查找匹配的软件包。但是，在冻结过程中不会记录发布者信息。软件包只针对其版本（而不是发布者）进行冻结。

-c reason

记录冻结软件包的 **原因**。当安装或更新因冻结而不能继续进行时，将会显示该原因。

-n
 试运行冻结操作，并显示要冻结的软件包的列表，但不冻结任何软件包。

pkg unfreeze [-n] [*pkg_name_pattern*] ...

显示有关当前已冻结软件包的信息：软件包名称、版本、软件包冻结时间以及任何冻结软件包的相关原因。

pkg_fmri_pattern

从指定的软件包中删除冻结操作施加的约束。将会忽略提供的版本。

-n
 试运行解冻操作，并显示要解冻的软件包的列表，但不解冻任何软件包。

pkg property [-H] [*propname* ...]

显示所有映像属性的名称和值。有关映像属性的说明，请参见下面的“映像属性”。

propname

仅显示指定属性的名称和值。

-H
 在列出时省略标题。

pkg set-property *propname propvalue*

更新现有映像属性或添加新的映像属性。

pkg add-property-value *propname propvalue*

向现有映像属性添加值，或添加新的映像属性。

pkg remove-property-value *propname propvalue*

从现有映像属性中删除值。

pkg unset-property *propname* ...

删除一个或多个现有映像属性。

pkg publisher [-HPn] [-F *format*] [*publisher* ...]

按搜索优先顺序显示所有发布者、其源 URI 和镜像的列表。

publisher

仅显示指定发布者的详细配置信息。

-H
 在列出时省略标题。

-P
 仅显示发布者搜索顺序中的第一个发布者。

-n
 仅显示已启用的发布者。

-F
 指定备用输出格式。当前，只有 **tsv**（Tab Separated Values，制表符分隔值）有效。

```
pkg set-publisher [-Ped] [-k ssl_key] [-c ssl_cert] [-g origin_to_add] --add-origin
origin_to_add... [-G origin_to_remove] --remove-origin origin_to_remove... [-m
mirror_to_add] --add-mirror mirror_to_add... [-M mirror_to_remove] --remove-mirror
mirror_to_remove... [--enable] [--disable] [--no-refresh] [--reset-uuid]
[--non-sticky] [--sticky] [--search-after publisher] [--search-before publisher]
[--search-first] [--approve-ca-cert path_to_CA] [--revoke-ca-cert
hash_of_CA_to_remove] [--unset-ca-cert hash_of_CA_to_remove] [--set-property
name_of_property=value] [--add-property-value name_of_property=value_to_add]
[--remove-property-value name_of_property=value_to_remove] [--unset-property
name_of_property_to_delete] [--proxy proxy_to_use] publisher
```

更新现有发布者，或添加发布者。如果未指定任何影响搜索顺序的选项，则会将新发布者附加到搜索顺序，因此，会最后搜索这些新发布者。

-P

--search-first

将指定的发布者设置为搜索顺序中第一个。安装新软件包时，将首先搜索此发布者。对已安装软件包的更新将来自最初提供该软件包的同一发布者，前提是该发布者保持粘滞。

--non-sticky

对于最初从此发布者安装的软件包，更新可以从级别比此发布者更高的发布者处获取。

--sticky

对于从此发布者安装的软件包，更新必须也来自此发布者。这是缺省行为。

--search-before *publisher*

更改发布者搜索顺序，以便在此选项中指定的发布者之前搜索添加或修改的发布者。

--search-after *publisher*

更改发布者搜索顺序，以便在此选项中指定的发布者之后搜索添加或修改的发布者。

--approve-ca-cert *path_to_CA*

将指定的证书添加为可信的 CA 证书。pkg publisher 命令的详细输出中列出了用户批准的 CA 证书的 PEM 表示形式中的散列。

--revoke-ca-cert *hash_of_CA_to_remove*

将其 PEM 表示形式具有给定散列的证书视为已撤销。pkg publisher 命令的详细输出中列出了用户撤销的 CA 证书的散列。

--unset-ca-cert *hash_of_CA_to_remove*

从已批准证书列表和已撤销证书列表中删除具有给定散列的证书。

--set-property *name_of_property*=*value*

更新现有发布者属性或添加新的发布者属性。

--add-property-value *name_of_property*=*value_to_add*

向现有发布者属性添加值，或添加新的发布者属性。

`--remove-property-value name_of_property=value_to_remove`
从现有发布者属性中删除值。

`--unset-property name_of_property_to_delete`
删除现有的发布者属性。

`-k ssl_key`
指定客户机 SSL 密钥。

`-c ssl_cert`
指定客户机 SSL 证书。

`-g origin_to_add`
`--add-origin origin_to_add`
将指定的 URI 或路径添加为给定发布者的源。这应该是软件包系统信息库或归档的位置。

`-G origin_to_remove`
`--remove-origin origin_to_remove`
从给定发布者的源列表中删除 URI 或路径。可以使用特殊值 * 来删除所有源。

`--no-refresh`
不尝试联系映像发布者的系统信息库来检索最新的可用软件包列表和其他元数据。

`--reset-uuid`
选择一个新的唯一标识符，用于向其发布者标识此映像。

`-m mirror_to_add`
`--add-mirror mirror_to_add`
将 URI 添加为给定发布者的镜像。

`-M mirror_to_remove`
`--remove-mirror mirror_to_remove`
从给定发布者的镜像列表中删除 URI。可以使用特殊值 * 来删除所有镜像。

`-e`
`--enable`
启用发布者。

`-d`
`--disable`
禁用发布者。填充软件包列表时，或者执行某些软件包操作（安装、卸载和更新）时，将不使用被禁用的发布者。但是，仍可以设置和查看被禁用的发布者的属性。如果只存在一个发布者，则不能将其禁用。

`--proxy proxy_to_use`
使用指定的 Web 代理 URI 检索指定源 (-g) 或镜像 (-m) 的内容。代理值存储在发布者配置中。在运行时，`$http_proxy` 或相关环境变量会覆盖此代理设置。有关接受的环境变量名列表，请参见 `curl(1)` 手册页。

```
pkg set-publisher -p repo_uri [-Ped] [-k ssl_key] [-c ssl_cert] [- -non-sticky] [--sticky]
[--search-after publisher] [--search-before publisher] [--search-first]
[--approve-ca-cert path_to_CA] [--revoke-ca-cert hash_of_CA_to_remove]
[--unset-ca-cert hash_of_CA_to_remove] [--set-property name_of_property=value]
[--add-property-value name_of_property=value_to_add]
[--remove-property-value name_of_property=value_to_remove] [--unset-property
name_of_property_to_delete] [--proxy proxy_to_use] [publisher]
  从 repo_uri 系统信息库 URI 中检索发布者配置信息。
```

如果为此 `set-publisher` 子命令指定了发布者操作数，则仅添加或更新该发布者。如果未指定发布者，则相应添加或更新 *repo_uri* 中的所有发布者。

有关选项的说明，请参见上述 `set-publisher` 命令。与 `-p` 一起使用时，`-P`、`--search-first`、`--search-before` 以及 `--search-after` 选项仅适用于添加的发布者，而不适用于更新的发布者。

`-p` 选项不能与以下选项结合使

用：`-g`、`--add-origin`、`-G`、`--remove-origin`、`-m`、`--add-mirror`、`-M`、`--remove-mirror`、`--d` 或 `--reset-uuid`。

```
pkg unset-publisher publisher ...
```

删除与一个或多个指定发布者相关联的配置。

```
pkg history [-HNL] [-t [time | time-time],...] [-o column,...] [-n number]
```

显示适用映像的命令历史记录。

`-H`

在列出时省略标题。

`-t time`

`-t time-time`

显示以逗号分隔列表指定的时间戳（格式为 `%Y-%m-%dT%H:%M:%S`）对应的日志记录（请参见 `strftime(3C)`）。要指定时间范围，请在开始和完成时间戳之间使用连字符（`-`）。可以将关键字 `now` 用作当前时间的别名。如果指定的时间戳包含重复的时间戳或重叠的日期范围，则每个重复的历史记录事件仅显示一次。

`-l`

以长格式显示日志记录，也就是说，除了显示标准格式的内容外，还包括命令的结果、命令完成时间、所用客户机的版本和名称、执行操作的用户名，以及执行命令时遇到的任何错误。

`-N`

显示发行说明文本。

`-n number`

仅显示指定数目的最近条目。

`-o column`

按照以逗号分隔的指定列名列表显示输出。有效的列名为：

be
在其上启动此操作的引导环境的名称。

be_uuid
在其上启动此操作的引导环境的 `uuid`。

client
客户端的名称。

client_ver
客户端的版本。

command
用于此操作的命令行。

finish
完成此操作时的时间。

id
启动此操作的用户 ID。

new_be
此操作创建的新引导环境。

new_be_uuid
此操作创建的新引导环境的 `uuid`。

operation
操作的名称。

outcome
此操作的结果摘要。

reason
有关此操作的结果的其他信息。

release_note
指示此操作是否已生成发行说明。

snapshot
执行此操作期间创建的快照。仅当成功完成操作后未自动删除快照时，才会记录此信息。

start
启动此操作时的时间。

time
执行此操作花费的总时间。对于用时不到一秒的操作，将显示 `0:00:00`。

user
启动此操作的用户名。

如果指定了 `command` 或 `reason` 列，则它们必须是 `-o` 列表中的最后一项，这样才能让输出字段彼此分隔。这两列不能显示在同一个 `history` 命令中。

如果系统上不再存在该引导环境，则在 `be` 或 `new_be` 值的后面会显示一个星号 (*)。

可通过使用 `be_uid` 或 `new_be_uid` 字段查找当前引导环境名称，来获取 `be` 和 `new_be` 的值。如果某个引导环境后来已重命名，随后又被删除，则显示的 `be` 和 `new_be` 值是执行 `pkg` 操作时记录的值。

`pkg purge-history`

删除所有现有历史记录信息。

`pkg rebuild-index`

重建由 `pkg search` 使用的索引。这是一项恢复操作，不适用于一般用途。

`pkg update-format`

将映像格式更新到当前版本。完成此操作后，该映像无法再在早期版本的 `pkg(5)` 系统上使用。

`pkg version`

显示一个用于唯一标识 `pkg` 版本的字符串。不保证在不同版本中，此字符串在任何方面都具有类似性。

`pkg help`

显示用法消息。

`pkg image-create [-FPUz] [-f|--force] [--full|--partial|--user] [--zone] [-k ssl_key] [-c ssl_cert] [--no-refresh] [--variant variant_name=value ...] [-g path_or_uri] [--origin path_or_uri ...] [-m uri] [--mirror uri ...] [--set-property name_of_property=value] [--facet facet_name=(True|False) ...] [(-p|--publisher) [name=] repo_uri] dir`

在 *dir* 指定的位置，创建一个适合于软件包操作的映像。缺省的映像类型为“用户”，即 `-U` (`--user`) 选项指定的类型。可以将映像类型设置为完整映像 (`--F` 或 `--full`)，或者设置为与完整映像（包括给定的 *dir* 路径）链接的部分映像 (`-P` 或 `--partial`)。可以使用 `-g` 或 `--origin` 指定其他源。可以使用 `-m` 或 `--mirror` 指定其他镜像。

必须使用 `-p` 或 `--publisher` 选项提供软件包系统信息库 URI。如果还提供了某个发布者名称，则创建映像时仅添加该发布者。如果未提供发布者名称，则会将指定的系统信息库已知的所有发布者添加到映像。完成初始创建操作后，将会尝试检索与此发布者关联的目录。

对于使用客户端 SSL 验证的发布者，可以通过 `-c` 和 `-k` 选项注册客户端密钥和客户端证书。此密钥和证书用于映像创建期间添加的所有发布者。

如果要在非全局区域上下文中运行映像，则可以使用 `-z` (`--zone`) 选项设置相应的变量。

- f
- force
基于现有映像强制创建一个映像。请慎用此选项。
- no-refresh
不尝试联系映像发布者的系统信息库来检索最新的可用软件包列表和其他元数据。
- variant *variant_name*= *value*
将指定的变量设置为指示值。有关变量的更多信息，请参见 pkg(5) 手册页中的“侧面和变量”。
- facet *facet_name*=(True|False)
将指定的侧面设置为指示值。有关侧面的更多信息，请参见 pkg(5) 手册页中的“侧面和变量”。
- set-property *name_of_property*= *value*
将指定的映像属性设置为指示值。有关映像属性的说明，请参见下面的“映像属性”。

映像属性

以下属性可定义映像的特征。这些属性存储有关映像的用途、内容和行为的信息。要查看映像中这些属性的当前值，请使用 `pkg property` 命令。要修改这些属性的值，请使用 `pkg set-property` 和 `pkg unset-property` 命令。

be-policy

(字符串) 指定在打包操作期间何时创建引导环境。允许使用以下值：

default

应用缺省引导环境创建策略 `create-backup`。

always-new

所有软件包操作均需要重新引导：在下次引导时设为活动状态的新引导环境中执行这些操作。除非显式请求，否则不创建备份引导环境。

该策略最为安全，但是它比大多数站点的需要更为严格，因为在不重新引导的情况下无法添加任何软件包。

create-backup

对于需要重新引导的软件包操作，在下次引导时，将创建一个新的引导环境并将其设置为活动状态。如果修改了软件包或安装了可能影响内核的内容，并且该操作影响实时引导环境，将创建备份引导环境，但不会将其设置为活动状态。也可以显式请求创建备份引导环境。

仅当新安装的软件导致系统不稳定时（有可能发生，但比较少见），该策略才可能存在风险。

when-required

对于需要重新引导的软件包操作，在下次引导时，将创建一个新的引导环境并将其设置为活动状态。除非显式请求，否则不创建备份引导环境。

该策略的风险最高，因为如果对实时引导环境的打包更改使得以后无法再进行更改，则可能没有可回退的最近引导环境。

ca-path

(字符串) 一个路径名称，指向为执行 SSL 操作而将 CA 证书保存到的目录。此目录的格式特定于底层 SSL 实现。要对可信 CA 证书使用替代位置，请将此值更改为指向另一个目录。有关 CA 目录的要求，请参见 `SSL_CTX_load_verify_locations(3openssl)` 的 `CPath` 部分。

缺省值： `/etc/openssl/certs`

check-certificate-revocation

(布尔型) 如果此属性设置为 `True`，则软件包客户机将尝试访问用于签名验证的证书中的任何 CRL 分发点，以确定证书自颁发以来是否已被撤销。

缺省值： `False`

flush-content-cache-on-success

(布尔型) 如果此属性设置为 `True`，则完成安装或更新操作后，软件包客户机将删除其内容高速缓存中的文件。对于更新操作，仅从源引导环境中删除内容。如果随后目标引导环境中发生了打包操作，并且此选项未发生更改，则软件包客户机将刷新其内容高速缓存。

在磁盘空间有限的系统上，可以使用此属性使内容高速缓存保持为较小的大小。此属性可能会导致花费更长的时间来完成操作。

缺省值： `True`

mirror-discovery

(布尔型) 此属性通知客户端使用 mDNS 和 DNS-SD 发现本地链路内容镜像。如果此属性设置为 `True`，则客户机尝试从其动态发现的镜像中下载软件包内容。要运行一个通过 mDNS 通告其内容的镜像，请参见 `pkg.depotd(1M)` 手册页。

缺省值： `False`

send-uuid

(布尔型) 执行网络操作时发送映像的通用唯一标识符 (Universally Unique Identifier, UUID)。尽管用户可以禁用此选项，但是某些网络系统信息库可能会拒绝与不提供 UUID 的客户机通信。

缺省值： `True`

signature-policy

(字符串) 确定在映像中安装、更新、修改或验证软件包时，要对清单执行哪些检查。应用于软件包的最终策略取决于映像策略和发布者策略的组合。该策略组合的严格程度至少相当于这两个策略单独执行时较严格的那一个。缺省情况下，软件包客户机不检查证书是否已撤销。要启用这些检查（可能需要客户机访问外部 Web 站点），请将 `check-certificate-revocation` 映像属性设置为 `True`。允许使用以下值：

ignore

忽略所有清单的签名。

verify

验证所有具有签名的清单的签名是否有效，但不要求签名所有安装的软件包。这是缺省值。

require-signatures

要求所有新安装的软件包至少具有一个有效签名。如果安装的软件包不具备有效签名，`pkg fix` 和 `pkg verify` 命令也会发出警告。

require-names

与 `require-signatures` 遵循相同的要求，但还要求 `signature-required-names` 属性中列出的字符串显示为用于验证签名信任链的证书的通用名称。

signature-required-names

(字符串列表) 在验证软件包签名时必须视为证书通用名称的名称列表。

trust-anchor-directory

(字符串) 包含映像信任锚的目录的路径名称。此路径是映像的相对路径。缺省值为 `ignore`。

use-system-repo

(布尔型) 此属性指示映像是否应使用系统信息库作为映像和发布者配置的源，以及作为与提供的发布者通信的代理。缺省值为 `False`。有关系统信息库的信息，请参见 `pkg.sysrepo(1M)` 手册页。

发布者属性

以下属性定义了特定发布者的签名策略。具有相同名称的映像属性定义了该映像的签名策略。要查看特定发布者的这些属性的当前值，请使用 `pkg publisher publisher_name` 命令。要修改发布者的这些签名策略属性的值，请使用 `pkg set-publisher` 命令的 `--set-property` 和 `--unset-property` 选项。

signature-policy

(字符串) 此属性的作用与同名映像属性的功能相同，不过它仅适用于来自特定发布者的软件包。

signature-required-names

(字符串列表) 此属性的作用与同名映像属性的功能相同，不过它仅适用于来自特定发布者的软件包。

示例

示例 1 在配置了发布者的情况下创建映像

使用 `/aux0/example_root` 中存储的发布者 `example.com` 创建一个新的完整映像。

```
$ pkg image-create -F -p example.com=http://pkg.example.com:10000 \
/aux0/example_root
```

示例 2 创建一个映像并指定附加源和镜像

使用发布者 `example.com` 创建一个新的完整映像。该映像还有一个附加镜像、两个附加源，并存储在 `/aux0/example_root` 中。

示例2 创建一个映像并指定附加源和镜像 (续)

```
$ pkg image-create -F -p example.com=http://pkg.example.com:10000 \
-g http://alternate1.example.com:10000/ \
-g http://alternate2.example.com:10000/ \
-m http://mirror.example.com:10000/ \
/aux0/example_root
```

示例3 在未配置发布者的情况下创建映像

在未配置发布者的情况下，在 /aux0/example_root 中创建一个新的完整映像。

```
$ pkg image-create -F /aux0/example_root
```

示例4 安装软件包

在当前映像中安装最新版本的 widget 软件包。

```
$ pkg install application/widget
```

示例5 列出软件包的指定内容

列出 system/file-system/zfs 软件包的内容。显示操作名称、文件模式（如果已定义）、大小（如果已定义）、路径和目标（如果为链接）。将操作限制为类型 dir、file、link 和 hardlink，因为指定可用于所有操作的 action.name 属性将显示一个列出了所有操作的行，而此处并不需要显示它。

```
$ pkg contents -t dir,file,link,hardlink \
-o action.name,mode,pkg.size,path,target system/file-system/zfs
ACTION.NAME MODE PKG.SIZE PATH TARGET
dir 0755 etc
dir 0755 etc/fs
dir 0755 etc/fs/zfs
link etc/fs/zfs/mount ../../../../usr/sbin/zfs
link etc/fs/zfs/umount ../../../../usr/sbin/zfs
dir 0755 etc/zfs
dir 0755 kernel
dir 0755 kernel/drv
dir 0755 kernel/drv/amd64
file 0755 1706744 kernel/drv/amd64/zfs
file 0644 980 kernel/drv/zfs.conf
dir 0755 kernel/fs
dir 0755 kernel/fs/amd64
hardlink kernel/fs/amd64/zfs ../../../../kernel/drv/amd64/zfs
...
```

示例6 列出两个软件包的指定内容

列出 web/browser/firefox 和 mail/thunderbird 的内容，将显示的内容仅限于软件包名称，以及其 path 属性以 .desktop 或 .png 结尾的操作的路径属性。

示例6 列出两个软件包的指定内容 (续)

```
$ pkg contents -o pkg.name,path -a path=*.desktop \
-a path=*.png web/browser/firefox mail/thunderbird
PKG.NAME          PATH
web/browser/firefox usr/share/applications/firefox.desktop
mail/thunderbird   usr/share/applications/thunderbird.desktop
web/browser/firefox usr/share/pixmaps/firefox-icon.png
mail/thunderbird   usr/share/pixmaps/thunderbird-icon.png
...
```

示例7 搜索软件包

在软件包数据库中搜索标记 bge。

```
$ pkg search bge
INDEX      ACTION VALUE                                PACKAGE
driver_name driver bge                                pkg:/driver/network/ethernet/bge@0.5
basename   file  kernel/drv/sparcv9/bge                    pkg:/driver/network/ethernet/bge@0.5
basename   file  kernel/drv/amd64/bge                      pkg:/driver/network/ethernet/bge@0.5
basename   file  platform/sun4v/kernel/drv/sparcv9/bge    pkg:/system/kernel/platform@0.5.11-0
pkg.fmri   set   solaris/driver/network/bge                pkg:/driver/network/bge@0.5.11-0.173
pkg.fmri   set   solaris/driver/network/ethernet/bge      pkg:/driver/network/ethernet/bge@0.5
```

该标记在软件包 driver/network/bge 中，既用作代表 /kernel/drv/arch/bge 的文件操作的基名，又用作驱动程序名称。

示例8 搜索文件

要搜索提供某个文件的软件包，请指定该文件的全路径名，包括开头的斜杠字符。

```
$ pkg search -o path,pkg.name -l /usr/bin/vim
PATH          PKG.NAME
usr/bin/vim   editor/vim/vim-core
```

要搜索某个文件和提供该文件的软件包，请指定 file (对于 *action_name*)、path 或 basename (对于 *index*)，以及完整或部分文件名 (对于 *token*)。

```
$ pkg search -o path,pkg.name -l file:basename:vim
PATH          PKG.NAME
usr/bin/vim   editor/vim/vim-core
```

示例9 搜索文件和目录

要搜索文件和目录以及提供它们的软件包，请指定 path 或 basename (对于 *index*) 以及完整或部分文件名 (对于 *token*)。根据您的 shell，您可能需要对通配符进行转义。

```
$ pkg search -o path,pkg.name -l path:*/vim
PATH          PKG.NAME
usr/bin/vim   editor/vim/vim-core
```

示例9 搜索文件和目录 (续)

```
usr/share/vim editor/vim
usr/share/vim editor/vim/vim-core
$ pkg search -o path, pkg.name -l basename:vim
PATH          PKG.NAME
usr/share/vim editor/vim
usr/share/vim editor/vim/vim-core
usr/bin/vim   editor/vim/vim-core
```

示例10 显示哪些软件包提供特定的 SMF 服务

要显示哪些软件包提供特定的 SMF 服务，请在结构化搜索中为 *index* 指定 `org.opensolaris.smf.fmri` 值，并为 *token* 指定要查找的服务名称。`org.opensolaris.smf.fmri` 值是 `set` 操作的某个属性的名称。请注意对服务名称中的 `:` 进行转义。

例如，要显示哪些 HTTP 服务器可用，请为 *token* 指定 `svc:/network/http` 值。

```
$ pkg search 'org.opensolaris.smf.fmri:svc\:/network/http*'
INDEX          ACTION VALUE          PACKAGE
org.opensolaris.smf.fmri set   svc:/network/http     pkg:/web/server/lighttpd-14@1.4.23-0.1
org.opensolaris.smf.fmri set   svc:/network/http     pkg:/web/proxy/privoxy@3.0.17-0.175.0.0
org.opensolaris.smf.fmri set   svc:/network/http     pkg:/web/proxy/squid@3.1.18-0.175.0.0
org.opensolaris.smf.fmri set   svc:/network/http     pkg:/web/java-servlet/tomcat@6.0.35-0.0
org.opensolaris.smf.fmri set   svc:/network/http     pkg:/web/server/apache-22@2.2.22-0.175.0.0
org.opensolaris.smf.fmri set   svc:/network/http:apache22 pkg:/web/server/apache-22@2.2.22-0.175.0.0
org.opensolaris.smf.fmri set   svc:/network/http:lighttpd14 pkg:/web/server/lighttpd-14@1.4.23-0.1
org.opensolaris.smf.fmri set   svc:/network/http:privoxy pkg:/web/proxy/privoxy@3.0.17-0.175.0.0
org.opensolaris.smf.fmri set   svc:/network/http:squid pkg:/web/proxy/squid@3.1.18-0.175.0.0
org.opensolaris.smf.fmri set   svc:/network/http:tomcat6 pkg:/web/java-servlet/tomcat@6.0.35-0.0
```

示例11 搜索依赖于指定软件包的软件包

搜索依赖于 `package/pkg` 的已安装软件包。

```
$ pkg search -l depend::package/pkg
INDEX          ACTION VALUE          PACKAGE
incorporate depend package/pkg@0.5.11-0.175.0.0.0.2.1 pkg:/consolidation/ips/ips-incorporation@0.5.11-0.175.0.0.0.2.1
require        depend pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1 pkg:/package/pkg/package-manager@0.5.11-0.175.0.0.0.2.1
require        depend pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1 pkg:/system/library/install@0.5.11-0.175.0.0.0.2.1
require        depend pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1 pkg:/package/pkg/update-manager@0.5.11-0.175.0.0.0.2.1
require        depend pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1 pkg:/system/library/boot-management@0.5.11-0.175.0.0.0.2.1
require        depend package/pkg          pkg:/system/zones/brand/brand-solaris@0.5.11-0.175.0.0.0.2.1
require        depend pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1 pkg:/install/distribution-constructor@0.5.11-0.175.0.0.0.2.1
require        depend pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1 pkg:/system/boot-environment-utilities@0.5.11-0.175.0.0.0.2.1
require        depend pkg:/package/pkg@0.5.11-0.175.0.0.0.2.1 pkg:/package/pkg/system-repository@0.5.11-0.175.0.0.0.2.1
```

示例 11 搜索依赖于指定软件包的软件包 (续)

示例 12 搜索依赖性

在安装的软件包中搜索所有 `incorporate` 依赖性。

```
$ pkg search -l depend:incorporate:
INDEX          ACTION VALUE          PACKAGE
incorporate    depend pkg:/BRcMbnx@0.5.11-0.175.0.0.0.2.1 pkg:/consolidation/osnet/osnet-incorp
incorporate    depend pkg:/BRcMbnxe@0.5.11-0.175.0.0.0.2.1 pkg:/consolidation/osnet/osnet-incorp
...
```

示例 13 添加发布者

添加新的发布者 `example.com`，该发布者的系统信息库位于 `http://www.example.com/repo`。

```
$ pkg set-publisher -g http://www.example.com/repo example.com
```

示例 14 添加具有密钥和证书的发布者

添加新的发布者 `example.com`，该发布者的安全系统信息库位于 `https://secure.example.com/repo`，其密钥和证书存储在目录 `/root/creds` 中。

```
$ pkg set-publisher -k /root/creds/example.key \
-c /root/creds/example.cert -g https://secure.example.com/repo \
example.com
```

示例 15 添加并自动配置发布者

使用自动配置功能添加一个新的发布者，该发布者的系统信息库位于 `/export/repo`。

```
$ pkg set-publisher -p /export/repo
```

示例 16 添加并手动配置发布者

使用手动配置功能添加新的发布者 `example.com`，该发布者的系统信息库位于 `/export/repo/example.com`。

```
$ pkg set-publisher -g /export/repo example.com
```

示例 17 添加发布者并配置代理

添加源为 `http://server/repo`、代理为 `http://webcache:8080` 的新发布者 `mypub`。

```
$ pkg set-publisher -g http://server/repo \
--proxy http://webcache:8080 mypub
```

示例 18 验证所有签名的软件包

配置一个映像以验证所有签名的软件包。

```
$ pkg set-property signature-policy verify
```

示例 19 要求签名所有软件包

配置一个映像，以要求签名所有软件包，并要求字符串 `example.com` 显示为信任链中某一个证书的通用名称。

```
$ pkg set-property signature-policy require-names example.com
```

示例 20 要求对来自指定发布者的所有软件包进行签名

配置一个映像，以便必须对通过发布者 `example.com` 安装的所有软件包进行签名。

```
$ pkg set-publisher --set-property signature-policy=require-signatures \  
example.com
```

示例 21 要求信任链中存在指定的字符串

将字符串 `foo` 添加到映像的通用名称列表，这些通用名称必须显示在签名的信任链中才能视为有效。

```
$ pkg add-property-value signature-require-names foo
```

示例 22 从指定发布者的信任链中删除某个字符串

从通用名称列表中删除字符串 `foo`，必须显示这些通用名称才能验证发布者 `example.com` 的签名。

```
$ pkg set-publisher --remove-property-value signature-require-names=foo \  
example.com
```

示例 23 添加可信 CA 证书

添加 `/tmp/example_file.pem` 中存储的证书，作为发布者 `example.com` 的可信 CA 证书。

```
$ pkg set-publisher --approve-ca-cert /tmp/example_file.pem \  
example.com
```

示例 24 撤销证书

撤销发布者 `example.com` 的、包含散列 `a12345` 的证书，防止该证书验证来自 `example.com` 的软件包的任何签名。

```
$ pkg set-publisher --revoke-ca-cert a12345 example.com
```

示例 25 忘记针对某个证书执行的操作

使 `pkg` 忘记用户曾经添加或撤销了证书 `a12345`。

```
$ pkg set-publisher --unset-ca-cert a12345 example.com
```

示例 26 将软件包降级

将安装的软件包 `foo@1.1` 降级到更低的版本。

```
$ pkg update foo@1.0
```

示例 27 切换发生冲突的软件包安装

当两个软件包发生冲突时，切换所安装的软件包。假定软件包 A 依赖于软件包 B 或软件包 C，而 B 和 C 是互斥的。如果安装了 A 和 B，则使用以下命令即可改为使用 C 而不是 B，且无需卸载 A：

```
$ pkg install --reject B C
```

示例 28 列出软件包归档中的软件包

列出某个软件包归档中所有软件包的所有版本。

```
$ pkg list -f -g /my/archive.p5p
```

示例 29 列出软件包系统信息库中的软件包

列出某个系统信息库中所有软件包的所有版本。

```
$ pkg list -f -g http://example.com:10000
```

示例 30 显示有关软件包归档中某个软件包的信息

显示软件包归档中某个软件包的最新版本的软件包信息。该软件包当前不一定已安装。

```
$ pkg info -g /my/archive.p5p pkg_name
```

示例 31 显示软件包归档中某个软件包的内容

显示软件包归档中某个软件包的内容。该软件包当前未安装。

```
$ pkg contents -g /my/archive.p5p pkg_name
```

示例 32 删除发布者的所有源和镜像

删除某个发布者的所有源和镜像，并添加新的源。

```
$ pkg set-publisher -G '*' -M '*' -g http://example.com:10000 \
example.com
```

环境变量**PKG_IMAGE**

用于软件包操作的映像所在的目录。如果指定了 -R，则会忽略此属性。

PKG_CLIENT_CONNECT_TIMEOUT

传输操作期间尝试建立连接时等待的秒数（针对每次尝试），达到此秒数后，客户端会异常中止操作。值 0 表示无限期等待。

缺省值：60

PKG_CLIENT_LOWSPEED_TIMEOUT

传输操作期间低于 `lowspeed` 限制（1024 字节/秒）的秒数，达到此秒数后，客户端会异常中止操作。值 0 表示不中止运行。

缺省值：30

PKG_CLIENT_MAX_CONSECUTIVE_ERROR

客户端异常中止操作之前发生瞬态传输错误的最大次数。值 0 表示不中止运行。

缺省值：4

PKG_CLIENT_MAX_REDIRECT

在传输操作期间，异常中止某个连接之前允许的最大 HTTP 或 HTTPS 重定向次数。值 0 表示不中止运行。

缺省值：5

PKG_CONCURRENCY

要并行更新的子映像数。如果指定了 -c 选项，则会忽略此属性。

在递归检查子映像（通常为区域）时，最多可以并行更新 \$PKG_CONCURRENCY 个子映像。如果 \$PKG_CONCURRENCY 为 0 或负数，则并行更新所有子映像。

缺省值：1

PKG_CLIENT_MAX_TIMEOUT

客户端异常中止操作之前每台主机上的最大传输尝试次数。值 0 表示不中止运行。

缺省值：4

http_proxy、https_proxy

HTTP 或 HTTPS 代理服务器。

退出状态

将返回以下退出值：

- 0 命令成功。
- 1 出现错误。
- 2 指定的命令行选项无效。
- 3 请求了多项操作，但只有一部分操作成功。
- 4 未进行更改—没有要执行的操作。
- 5 无法对实时映像执行请求的操作。
- 6 无法完成请求的操作，因为尚未接受所安装或更新的软件包的许可证。
- 7 该映像当前已被另一个进程使用，无法修改。
- 99 发生了意外的异常。

文件

可将 pkg(5) 映像放置在较大文件系统中的任意位置。在以下文件描述中，标记 \$IMAGE_ROOT 用于区分相对路径。对于典型的系统安装，\$IMAGE_ROOT 等效于 /。

\$IMAGE_ROOT/var/pkg

完整或部分映像的元数据目录。

`$IMAGE_ROOT/.org.opensolaris,pkg`
 用户映像的元数据目录。

在特定映像的元数据中，某些文件和目录可能包含修复和恢复期间有用的信息。标记 `$IMAGE_META` 引用元数据所在的顶层目录。`$IMAGE_META` 通常是上述两个路径之一。

`$IMAGE_META/lost+found`

在软件包操作期间移动的有冲突目录和文件的位置。某个已删除目录的未打包内容的位置。

`$IMAGE_META/publisher`

为每个发布者包含一个目录。每个目录存储特定于发布者的元数据。

`$IMAGE_META` 目录分层结构中的其他路径是专用的，可能会进行更改。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted (未确定)

另请参见

[pkgsend\(1\)](#)、[pkg.depotd\(1M\)](#)、[glob\(3C\)](#)、[pkg\(5\)](#)、[beadm\(1M\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名 pkgdepend – 映像包管理系统依赖项分析器

用法概要 /usr/bin/pkgdepend [*options*] *command* [*cmd_options*] [*operands*]

```
/usr/bin/pkgdepend generate [-IMm] -d dir [-d dir]
[-D name=value] [-k path] manifest_file
```

```
/usr/bin/pkgdepend resolve [-moSv] [-d output_dir]
[-s suffix] manifest_file ...
```

描述 pkgdepend 命令生成并解析软件包的依赖项。某个软件包可能会依赖于其他软件包中的文件。pkgdepend 命令通常使用两次：一次用于文件依赖项生成和一次用于文件到软件包的解析。

generate 子命令将检查软件包的内容，并确定该软件包所需的外部文件。

resolve 子命令使用执行 generate 步骤后输出的文件列表，然后搜索软件包的引用集来确定包含这些依赖文件的软件包的名称。为依赖文件搜索的软件包的引用集为当前在发布者的系统上安装的软件包。

提供的文件的多个组件用作依赖项信息的源：

ELF 分析提供的文件中的 ELF 头以获取依赖项信息，-k 和 -D 选项可用于修改获取的信息。有关 ELF 依赖项的更多详细信息，请参见 ldd 和《链接程序和库指南》。

脚本 包含引用某个解释程序的 #! 行的 Shell 脚本会导致对提供该解释程序的程序包出现依赖项。

Python Python 脚本首先作为脚本进行分析。此外，脚本声明的任何导入可能还会充当依赖项信息的源。

硬链接 清单中的硬链接会导致对提供链接目标的软件包出现依赖项。

SMF 提供的包含 require_all 依赖项的 SMF 服务清单会导致对提供 SMF 清单（这些清单提供这些 FMRI）的软件包出现依赖项。

选项 支持以下选项：

-?

--help
显示用法消息。

-R *dir*

--image-dir *dir*

对根目录为 *dir* 的映像进行操作。如果未根据环境指定或确定目录，则缺省值为 /。有关更多信息，请参见“环境变量”部分。

子命令 支持以下子命令：

```
pkgdepend generate [-IMm] -d dir [-d dir] [-D name=value] [-k path] manifest_file
生成由 manifest_file 指定的清单文件的依赖项。
```


- I
显示 *manifest_file* 中满足条件的依赖项。请勿使用带 **-I** 选项的 `pkgdepend resolve` 命令产生的结果。
- M
显示无法分析的文件类型的列表。
- m
重复显示原始清单，并在其后添加发现的任何依赖项。
- d *dir*
将 *dir* 添加到搜索清单文件的目录列表。
- D *name= value*
添加 *value*，将其作为在 ELF 文件依赖项的运行路径中扩展标记 *name* 的一种方法。
- k *path*
将 *path* 添加到搜索内核模块的运行路径列表。使用 **-k** 选项会删除缺省路径 `/kernel` 和 `/usr/kernel`。

运行路径（例如 **-k** 选项指定的那些运行路径）也可以通过使用操作或清单属性 `pkg.depend.runpath` 来按操作或清单指定。`pkg.depend.runpath` 属性的值是要使用的路径的冒号分隔字符串。在清单或操作中设置任何 `pkg.depend.runpath` 属性时均会覆盖使用 **-k** 选项指定的路径。

特殊标记 `$PKGDEPEND_RUNPATH` 可用作 `pkg.depend.runpath` 属性值的一个组件，以包括所分析文件的标准系统运行路径。

在某些情况下，您可能需要防止自动生成依赖项。例如，如果软件包提供了一个用于导入一组模块的样例 Python 脚本，则该样例脚本导入的那些模块不是提供该样例脚本的软件包的依赖项。使用操作或清单属性 `pkg.depend.bypass-generate` 可以防止针对指定的文件生成依赖项。

`pkg.depend.bypass-generate` 值是与文件名匹配的 Python 正则表达式。正则表达式隐式固定在文件路径的开头和结尾。以下示例中提供的值与 `this/that` 匹配，但与 `something/this/that/the/other` 不匹配。

```
pkg.depend.bypass-generate=this/that
```

有关 Python 正则表达式语法的更多信息，请使用命令 `pydoc re` 或参见 <http://docs.python.org/dev/howto/regex.html> 中更为完整的文档。

当 `pkgdepend generate` 输入清单包含 SMF 清单文件时，由这些 SMF 清单文件声明的任何 SMF 服务或实例都将包括在 `pkgdepend` 输出中。这些 SMF 服务或实例以名称为 `org.opensolaris.smf.fmri` 的 `set` 操作的形式包括在输出中。

`pkgdepend resolve [-moSv] [-d output_dir] [-s suffix] manifest_file ...`

将文件中的依赖项转换为提供这些文件的软件包中的依赖项。先根据命令行中给定的清单解析依赖项，然后再根据系统上安装的软件包进行解析。缺省情况下，每个清单的依赖项放置在名为 *manifest_file.res* 的文件中。

- m
将重复清单，删除 `generate` 步骤生成的任何依赖项，然后添加已解析的依赖项。
- o
将结果写入到标准输出。该选项旨在供用户使用。将此输出附加到某个文件可能会导致产生无效的清单。在用于清单处理的管道中使用时，强烈建议使用 `-d` 或 `-s` 选项，而不要使用 `-o`。
- d *output_dir*
将单独文件中提供的每个清单的已解析依赖项写入 *output_dir* 中。缺省情况下，每个文件与清单（该清单是写入该文件的依赖项的源）具有相同的基名。
- s *suffix*
对于每个输出文件，将 *suffix* 附加到文件（该文件是解析的依赖项的源）的基名。如果 *suffix* 不是 *.suffix* 格式，则会将句点 (.) 附加到 *suffix* 的前面。
- S
只根据命令行上指定的清单进行解析，而不根据系统上安装的清单进行解析。
- v
将包括其他软件包依赖项调试元数据。

示例

示例1 生成依赖项

为 `foo` 中写入的清单（其内容目录在 `./bar/baz` 中）生成依赖项，并将结果存储在 `foo.fdeps` 中。

```
$ pkgdepend generate -d ./bar/baz foo > foo.fdeps
```

示例2 解析依赖项

根据彼此的情况和当前系统上安装的软件包来解析 `foo.fdeps` 和 `bar.fdeps` 中的文件依赖项。

```
$ pkgdepend resolve foo.fdeps bar.fdeps
$ ls *.res
foo.fdeps.res  bar.fdeps.res
```

示例3 生成并解析两个清单的依赖项

生成两个清单（`foo` 和 `bar`）的文件依赖项，并保留原始清单中的所有信息。然后解析文件依赖项，并将生成的清单放置在 `./res` 中。这些生成的清单可以和 `pkgsend publish` 一起使用。

示例3 生成并解析两个清单的依赖项 (续)

```
$ pkgdepend generate -d /proto/foo -m foo > ./deps/foo
$ pkgdepend generate -d /proto/bar -m bar > ./deps/bar
$ pkgdepend resolve -m -d ./res ./deps/foo ./deps/bar
$ ls ./res
foo    bar
```

示例4 将值添加到ELF文件依赖项的标记

在为foo中写入的清单（其内容目录在/中）生成依赖项时，将ELF文件中运行路径内的所有PLATFORM标记替换为sun4v和sun4u。

```
$ pkgdepend generate -d / -D 'PLATFORM=sun4v' -D 'PLATFORM=sun4u' foo
```

示例5 指定内核模块目录

在为foo中写入的清单（其内容目录在/中）生成依赖项时，将/kmod指定为要在其中查找内核模块的目录。

```
$ pkgdepend generate -d / -k /kmod foo
```

示例6 绕过依赖项生成

将opt/python附加到给定Python脚本的标准Python运行路径，然后根据名称为test的所有Python模块绕过作为opt/python/foo/file.py提供的文件的依赖项生成。

避免针对usr/lib/python2.6/vendor-packages/xdg中提供的任何文件生成依赖项。

```
$ cat manifest.py
set name=pkg.fmri value=pkg:/mypackage@1.0,1.0
set name=pkg.summary value="My test package"
dir path=opt mode=0755 group=sys owner=root
dir path=opt/python mode=0755 group=sys owner=root
dir path=opt/python/foo mode=0755 group=sys owner=root
file NOHASH path=opt/python/__init__.py mode=0644 group=sys owner=root
file NOHASH path=opt/python/foo/__init__.py mode=0644 group=sys owner=root
#
# Add runpath and bypass-generate attributes:
#
file NOHASH path=opt/python/foo/file.py mode=0644 group=sys owner=root \
  pkg.depend.bypass-generate=.* /test.py.* \
  pkg.depend.bypass-generate=.* /testmodule.so \
  pkg.depend.bypass-generate=.* /test.so \
  pkg.depend.bypass-generate=usr/lib/python2.6/vendor-packages/xdg/. * \
  pkg.depend.runpath=$PKGDEPEND_RUNPATH:/opt/python

$ pkgdepend generate -d proto manifest.py
```

环境变量 **PKG_IMAGE** 指定包含要用于软件包操作的映像的目录。如果指定 **-R**，则忽略该值。

退出状态 将返回以下退出值：

- 0** 一切正常工作。
- 1** 出现错误。
- 2** 指定的命令行选项无效。
- 99** 发生了意外的异常。

属性 有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted（未确定）

另请参见 [pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名	pkgdiff - 比较软件包清单
用法概要	<pre>/usr/bin/pkgdiff [-i attribute ...] [-o attribute] [-v name=value ...] file1 file2</pre>
描述	<p>pkgdiff 可比较两个软件包清单并报告差异。在进行比较之前，pkgdiff 会将每个清单和操作按一致的顺序排序。</p> <p>输出采用以下格式：</p> <p>+ <i>complete_action</i> 此操作在 <i>file2</i> 中而不在 <i>file1</i> 中。</p> <p>- <i>complete_action</i> 此操作在 <i>file1</i> 中而不在 <i>file2</i> 中。</p> <p><i>actionname keyvalue [variant values, if any]</i></p> <p>- <i>attribute1=value1</i> 此 <i>attribute,value</i> 在 <i>file1</i> 中而不在 <i>file2</i> 中。</p> <p>+ <i>attribute2=value2</i> 此 <i>attribute,value</i> 在 <i>file2</i> 中而不在 <i>file1</i> 中。</p> <p>为了完成比较，带有不同变体、但带有相同类型和关键属性值的操作将被视为不同的操作。因此，可更改属性的操作将以其完整格式显示，而不是作为属性更改显示。</p>
选项	<p>支持以下选项：</p> <p>--help 显示用法消息。</p> <p>-i <i>attribute</i> 比较期间将忽略 <i>attribute</i>（如果存在）。可以使用 -i <i>hash</i> 忽略文件散列值。该选项不能与 -o 选项一起使用。可以重复该选项。</p> <p>-o <i>attribute</i> 仅报告 <i>attribute</i> 的差异。该选项不能与 -i 选项一起使用。该选项会省略不影响操作 <i>attribute</i> 的任何操作更改。</p> <p>-v <i>name= value</i> 仅计算该变体值的差异。例如，只计算 <i>arch=sparc</i> 的差异。在进行比较之前，将删除所有操作的该变体标记。只能为每个变体指定一个值。可对不同的变体重复该选项。</p>
退出状态	<p>将返回以下退出值：</p> <p>0 未找到差异。</p> <p>1 找到了差异。</p> <p>>1 出现错误。</p>

99 发生了意外的异常。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted (未确定)

另请参见

[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名	pkgfmt – 格式化软件包清单
用法概要	<code>/usr/bin/pkgfmt [-c -d -u] [package-manifest-file]</code>
描述	<p>不带 <code>-c</code> 或 <code>-d</code> 选项的 <code>pkgfmt</code> 将以一致的方式对软件包清单进行格式化，包括每 80 个字符自动换行，按类型对操作排序，以及对属性排序。未解析成操作（如宏、注释或转换）的行不会按排序顺序显示。</p> <p>如果未提供参数，<code>pkgfmt</code> 将会读取 <code>stdin</code>，直到遇到 EOF，然后将格式化的清单写入 <code>stdout</code>。在命令行上指定的任何清单将在原位格式化。</p> <p>带 <code>-c</code> 选项的 <code>pkgfmt</code> 将会检查清单是否格式化为 <code>pkgfmt</code> 样式。如果文件未正确格式化，<code>-d</code> 选项将显示差异。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> <code>-?</code> <code>--help</code> 显示用法消息。 <code>-c</code> 检查清单是否格式化为 <code>pkgfmt</code> 样式。 <code>-d</code> 以统一格式显示与格式化版本的清单差异。 <code>-u</code> 达到 80 个字符时不自动换行。将传统的文本处理工具应用到软件包清单时，该选项十分有用。
退出状态	<p>将返回以下退出值：</p> <ul style="list-style-type: none"> 0 命令成功。 1 指定了 <code>-c</code> 或 <code>-d</code> 选项，并且一个或多个清单不处于 <code>pkgfmt</code> 正常格式，或出现了错误。 2 指定的命令行选项无效。 99 发生了意外的异常。
属性	有关下列属性的说明，请参见 <code>attributes(5)</code> ：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted（未确定）

另请参见

[pkg\(5\)](#)<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名 pkginfo – 显示软件包信息

用法概要 pkginfo [-q | -x | -l] [-p | -i] [-r] [-a *arch*]
 [-v *version*] [-c *category*]... [*pkginst*]...

pkginfo -d *device* [-R *root_path*] [-q | -x | -l] [-a *arch*]
 [-v *version*] [-c *category*]... [*pkginst*]...

描述 pkginfo 显示安装在系统中的软件包信息（首先说明），或位于特定设备或目录中的软件包信息（其次说明）。

没有选项时，pkginfo 列出主要类别、软件包实例和所有完全安装及部分安装的软件包的名称。针对每个选择的软件包显示一行。

选项 -p 和 -i 选项与 -d 选项结合使用时无意义。

选项 -q、-x 和 -l 是互斥的。

-a *arch*
 将软件包体系结构指定为 *arch*。

-c *category*
 显示匹配 *category* 的软件包。类别在 pkginfo(4) 文件中以 CATEGORY 参数定义。如果提供了多个类别，软件包只需匹配列表中的一个类别。匹配并非特定于案例的。

-d *device*
 定义软件包所在的设备 *device*。device 可以是磁带、可移除磁盘等的绝对目录路径名或标识符。特殊令牌 *spool* 可用来表示缺省的安装假脱机目录 (/var/spool/pkg)。

-i
 仅显示完全安装的软件包的信息。

-l
 指定长格式，其包含指定软件包的所有可用信息。

-p
 仅显示部分安装的软件包的信息。

-q
 不列出任何信息。通过程序使用，检查是否已安装某个软件包。

-r
 列出可重定位的软件包的安装库。

-R *root_path*
 定义要用作 **根路径** 的目录的全路径名。所有文件，包括软件包系统信息文件，都重定位到指定的 *root_path* 下开始的目录树。

-v *version*

指定软件包版本为 *version*。版本在 [pkginfo\(4\)](#) 文件中以 `VERSION` 参数定义。可通过在版本名前加波浪号 (`≈`) 请求所有兼容版本。在版本比较时多个空格将替换为一个空格。

-x

指定软件包的提取列表。列表包括软件包缩写、软件包名称、软件包体系结构（如果有）和软件包版本（如果有）。

操作数***pkginst***

按软件包实例指定软件包。实例可以是软件包缩写或特定的实例（例如，`inst.1` 或 `inst.2`）。可以通过 `inst.*` 请求软件包的所有实例。星号字符 (*) 对于某些 shell 来说是特殊字符，可能需要转义。在 C-Shell 中，"*" 必须由单引号 (') 括起来或者在前面加反斜杠 (\)。

退出状态**0**

成功完成。

>0

出现错误。

文件**/var/spool/pkg**

缺省的安装假脱机目录

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[pkgtrans\(1\)](#)、[pkgadd\(1M\)](#)、[pkgask\(1M\)](#)、[pkgchk\(1M\)](#)、[pkgrm\(1M\)](#)、[pkginfo\(4\)](#)、[attributes\(5\)](#)

《Application Packaging Developer's Guide》

附注

软件包命令可识别 [largefile\(5\)](#)。它们处理大于 2 GB 的文件的方法与处理较小文件的方法相同。在当前的执行中，[pkgadd\(1M\)](#)、[pkgtrans\(1\)](#) 和其他软件包命令可以处理最大 4 GB 的数据流。

引用名	pkglint – 映像包管理系统软件包 lint
用法概要	<pre>/usr/bin/pkglint [-c <i>cache_dir</i>] [-r <i>repo_uri</i>] [-p <i>regex</i>] [-f <i>config_file</i>] [-b <i>build_no</i>] [-v] [-l <i>lint_uri</i>] <i>manifest</i> ... /usr/bin/pkglint -L [-v]</pre>
描述	<p>pkglint 对一个或多个软件包清单运行一系列检查，并有选择性地引用其他系统信息库。</p> <p>应该在发布软件包之前，在软件包构建期间使用 pkglint。pkglint 将对清单执行全面的测试，在 pkgsend 或 pkg.depotd 正常操作期间，执行这种测试可能会消耗过多的资源。pkglint 检查包括对重复操作、缺少的属性和非正常文件权限的测试。</p> <p>可以在命令行上将用于 lint 的清单作为本地文件的空格分隔列表进行传递，也可以从系统信息库检索清单。</p> <p>从系统信息库检索清单时，首次运行时 pkglint 将在指定的高速缓存目录中创建并填充 pkg(5) 用户映像。如果提供了 -r 选项，则为引用系统信息库创建名为 <i>cache_dir/ref_image</i> 的用户映像。如果提供了 -l 选项，则为 lint 系统信息库创建名为 <i>cache_dir/lint_image</i> 的用户映像。不会在这些映像中安装任何内容。pkglint 只使用这些映像从系统信息库中检索清单。</p> <p>pkglint 的后续调用可以重新使用高速缓存目录，并可以省略任何 -r 或 -l 参数。</p> <p>pkglint 对在高速缓存目录中配置发布者提供有限的支持。使用 pkg 可对这些映像执行更复杂的发布者配置。</p> <p>pkglint 使软件包作者能够绕过对给定清单或操作进行的检查。其中的属性 pkg.linted 设置为 True 的清单或操作不会为该清单或操作生成任何 lint 输出。</p> <p>使用 pkglint 检查名称的子字符串可以进行更高粒度的 pkg.linted 设置。例如，设置为 True 的 pkg.linted.<i>check.id</i> 将对给定的清单或操作绕过名称为 <i>check.id</i> 的所有检查。</p> <p>可通过指定一个 pkglintrc 文件来配置 pkglint 的行为。缺省情况下，pkglint 在 /usr/share/lib/pkg/pkglintrc 和 \$HOME/.pkglintrc 中搜索配置选项。使用 -f 选项可以指定其他配置文件。</p> <p>lint 运行期间，遇到的任何错误或警告将输出到 stderr。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none">-h--help <p>显示用法消息。</p>

- b *build_no*
指定内部版本号，用于细化从 `lint` 和引用系统信息库执行 `lint` 期间使用的软件包列表。如果不指定 `-b` 选项，将使用最新版本的软件包。另请参见 `version.pattern` 配置属性。
- c *cache_dir*
指定用于从 `lint` 和引用系统信息库高速缓存软件包元数据的本地目录。
- l *lint_uri*
指定表示 `lint` 系统信息库位置的 URI。支持基于 HTTP 和基于文件系统的发布。如果指定 `-l`，则还必须指定 `-c`。
- L
列出已知的和排除的 `lint` 检查，然后退出。显示每个检查的短名称和描述。与 `-v` 标志结合使用时，将显示实现检查的方法而不显示描述。
- f *config_file*
使用 *config_file* 配置文件来配置 `pkglint` 会话。
- p *regexp*
指定正则表达式，用于细化要从 `lint` 系统信息库检查的软件包列表。将会装入来自引用系统信息库的所有清单（假定它们与提供的 `-b` 值匹配），并忽略此模式。
- r *repo_uri*
指定表示引用系统信息库位置的 URI。如果指定 `-r`，则还必须指定 `-c`。
- v
以详细模式运行 `pkglint`，覆盖配置文件中的任何 `log_level` 设置。

文件

`pkglintrc` 配置文件采用以下键/值参数：

`log_level`

发出 `lint` 消息的最低级别。低于该级别的 `lint` 消息将被放弃。缺省值为 `INFO`。

严重性从低到高的日志级别依次为 `DEBUG`、`INFO`、`WARNING`、`ERROR` 和 `CRITICAL`。

`do_pub_checks`

如果为 `True`，则执行可能只对已发布的软件包有意义的检查。缺省值为 `True`。

`pkglint.ext.*`

`pkglint` 的插件机制允许在运行时添加其他 `lint` 模块。以 `pkglint.ext.` 开头的任何键采用必须是完全指定的 Python 模块的值。有关更多信息，请参见“开发工具”一节。

`pkglint.exclude`

要从执行的检查集中省略的完全指定的 Python 模块、类或函数名称的空格分隔列表。

`use_progress_tracker`

如果为 `True`，则在 `lint` 运行期间迭代清单时，会使用进度跟踪器。缺省值为 `True`。

version.pattern

指定内部版本号以针对 (-b) 执行 lint 时使用的版本模式。如果未在配置文件中指定，则 -b 选项将使用模式 `*,5.11-0.`，并使用分支前缀 0 来匹配 5.11 内部版本的所有组件。

开发工具

扩展 `pkglint`、子类 `pkg.lint.base.Checker` 及其子类 `ManifestChecker`、`ActionChecker` 和 `ContentChecker` 执行的检查集。将包含这些类的 Python 模块名称添加到配置文件中的新 `pkglint.ext` 键。

这些新子类的实例由 `pkglint` 在启动时创建。lint 会话过程中，将会调用带有特殊关键字参数 `pkglint_id` 的每个子类中的方法。这些方法应该与超类中的相应 `check()` 方法具有相同的签名。还应该为方法指定一个 `pkglint_desc` 属性，该属性用作 `pkglint -L` 输出的描述。

参数可用于 `Checker` 子类，使这些子类能够调优其行为。建议的参数命名约定为 `pkglint_id.name`。参数值可以存储在配置文件中，或者在使用 `LintEngine.get_param()` 方法检索的清单或操作中访问。从清单访问参数时，将在键名的前面附加前缀 `pkg.lint`，以确保 `pkglint` 参数不会与任何现有的操作或清单值重叠。

示例

示例1 对特定系统信息库的首次运行

对给定系统信息库上首次运行 `pkglint` 会话。

```
$ pkglint -c /space/cache -r http://localhost:10000 mymanifest.mf
```

示例2 对相同系统信息库的后续运行

针对示例1中使用的同一系统信息库的后续运行。

```
$ pkglint -c /space/cache mymanifest-fixed.mf
```

示例3 将Lint系统信息库和细化的清单集一起使用

将 `pkglint` 会话和 lint 系统信息库结合运行，并指定要检查的清单的子集。

```
$ pkglint -c /space/othercache -l http://localhost:10000 \
-p '.*firefox.*'
```

示例4 指定内部版本

在详细模式下针对给定的内部版本运行 `pkglint` 会话。

```
$ pkglint -c /space/cache -r http://localhost:10000 \
-l http://localhost:12000 -b 147 -v
```

示例5 修改配置文件

带有新的 lint 模块的配置文件，排除某些检查。

```
$ cat ~/.pkglintrc
[pkglint]
```

示例5 修改配置文件 (续)

```
log_level = DEBUG
# log_level = INFO

pkglint.ext.mycheck = org.timf.mychecks
pkglint.ext.opensolaris = pkg.lint.opensolaris
pkglint.exclude: pkg.lint.opensolaris.OpenSolarisActionChecker
pkg.lint.pkglint.PkgActionChecker.unusual_perms pkg.lint.pkglint.PkgManifestChecker
pkg.lint.opensolaris.OpenSolarisManifestChecker
```

退出状态

将返回以下退出值：

- 0 命令成功。
- 1 一项或多项 lint 检查提供了输出。
- 2 指定的命令行选项无效。
- 99 发生了意外的异常。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted (未确定)

另请参见

[pkg\(1\)](#)、[pkg.depotd\(1M\)](#)、[pkgsend\(1\)](#)、[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名	pkgmerge – 映像包管理系统软件包合并实用程序
用法概要	<pre> /usr/bin/pkgmerge [-n] -d <i>dest_repo</i> -s <i>variant=value[,...],src_repo ...</i> [<i>pkg_fmri_pattern ...</i>] </pre>
描述	<p>pkgmerge 是一种用于创建多变量软件包的软件包发布工具。它通过合并具有相同名称和版本的软件包（时间戳除外），使用给定源的指定变量名称和值标记合并的版本中的唯一操作，然后将新的软件包发布到目标系统信息库，从而实现此操作。仅使用每个源中各软件包的最新版本。</p> <p>如果将某一操作的 <code>pkg.merge.blend</code> 属性设置为要合并变量的名称，则在合并之前将该操作复制到其他清单，以便在最终输出中显示该操作时不显示任何已添加的变量标记。请注意，属性 <code>pkg.merge.blend</code> 本身已从输出清单的所有操作中删除。对于多个传递合并，可以使用不同的值重复该属性。</p> <p>在输入清单中交付到同一路径的不同操作将导致 pkgmerge 因错误而退出。</p>
选项	<p>支持以下选项：</p> <pre> -? --help 显示用法消息。 -d <i>dest_repo</i> 指定要将合并软件包发布到的目标系统信息库的文件系统路径或 URI。目标系统信息库必须已存在。可以使用 <code>pkgrepo</code> 创建新的系统信息库。 -n 执行试运行，不对目标系统信息库进行任何更改。 -s <i>variant=value[...],src_repo</i> 指定用于该源的软件包的变量名称和值，后跟要从中检索软件包的源系统信息库或软件包归档文件的文件系统路径或 URI。可指定多个变量，以逗号分隔。必须为所有源命名相同的变量。可以多次指定此选项。 </pre>
环境变量	<p>支持以下环境变量：</p> <p>TMPDIR 在程序执行期间用于存储临时数据的目录的绝对路径。如果未设置，则存储临时数据的缺省路径为 <code>/var/tmp</code>。</p>
示例	<p>示例1 指定变量名称和值</p> <p>使用为从中检索到软件包的源指定的给定变量名称和值标记在指定源中发现的每个软件包：</p> <pre> \$ pkgmerge -s arch=sparc,http://src.example.com \ -d http://dest.example.com </pre> <p>样例软件包：</p>

示例1 指定变量名称和值 (续)

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
dir group=sys mode=0755 owner=root path=usr
```

操作后的样例软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
set name=variant.arch value=sparc
dir group=sys mode=0755 owner=root path=usr
```

示例2 合并和发布软件包

合并给定源中每个软件包的最新版本并将新软件包发布到目标系统信息库:

```
$ pkgmerge -s arch=sparc,http://src1.example.com \
-s arch=i386,http://src2.example.com \
-d /path/to/target/repository
```

源 1 (SPARC) 中的样例软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T121410Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

源 2 (i386) 中的样例软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

合并软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
set name=variant.arch value=sparc value=i386
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=sparc
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=i386
dir group=sys mode=0755 owner=root path=usr
```

示例3 合并 i386 和 SPARC 系统的调试和非调试软件包

在一组用于 i386 和 SPARC 系统的调试及非调试系统信息库中, 合并每个软件包的最新版本:

```
$ pkgmerge -s arch=sparc,debug=false,/repo/sparc-nondebug \
-s arch=sparc,debug=true,/repo/sparc-debug \
-s arch=i386,debug=false,/repo/i386-nondebug \
-s arch=i386,debug=true,/repo/i386-debug \
-d /path/to/target/repository
```

源 1 (SPARC 非调试) 中的样例软件包:

示例3 合并 i386 和 SPARC 系统的调试和非调试软件包 (续)

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T121410Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

源 2 (SPARC 调试) 中的样例软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T121411Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

源 3 (i386 非调试) 中的样例软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

源 4 (i386 调试) 中的样例软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163428Z
file id mode=0555 owner=root group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

合并软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163428Z
set name=variant.arch value=sparc value=i386
set name=variant.debug value=false value=true
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=sparc variant.debug=false
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=sparc variant.debug=true
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=i386 variant.debug=false
file id mode=0555 owner=root group=bin path=usr/bin/foo variant.arch=i386 variant.debug=true
dir group=sys mode=0755 owner=root path=usr
```

示例4 使用 pkg.merge.blend 合并

使用 pkg.merge.blend 属性合并两个不相互冲突的体系结构的软件包。

```
$ pkgmerge -s arch=sparc,http://src1.example.com \
-s arch=i386,http://src2.example.com \
-d /path/to/target/repository
```

源 1 (SPARC) 中的样例软件包:

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T121410Z
file 1d5eac1aab628317f9c088d21e4afda9c754bb76 mode=0555 owner=root \
  group=bin path=usr/bin/sparc/foo pkg.merge.blend=arch
file d285ada5f3cae14ea00e97a8d99bd3e357caadc0 mode=0555 owner=root \
  group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```


示例 4 使用 pkg.merge.blend 合并 (续)

源 2 (i386) 中的样例软件包：

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
file a285ada5f3cae14ea00e97a8d99bd3e357cb0dca mode=0555 owner=root \
    group=bin path=usr/bin/i386/foo pkg.merge.blend=arch
file d285ada5f3cae14ea00e97a8d99bd3e357caadc0 mode=0555 owner=root \
    group=bin path=usr/bin/foo
dir group=sys mode=0755 owner=root path=usr
```

合并软件包：

```
set name=pkg.fmri value=pkg://example.com/foo@5.11,5.11-0.200:20381001T163427Z
set name=variant.arch value=sparc value=i386
file d285ada5f3cae14ea00e97a8d99bd3e357caadc0 mode=0555 owner=root \
    group=bin path=usr/bin/foo
file a285ada5f3cae14ea00e97a8d99bd3e357cb0dca mode=0555 owner=root \
    group=bin path=usr/bin/i386/foo
file 1d5eac1aab628317f9c088d21e4afda9c754bb76 mode=0555 owner=root \
    group=bin path=usr/bin/sparc/foo
dir group=sys mode=0755 owner=root path=usr
```

退出状态

将返回以下退出值：

- 0 命令成功。
- 1 出现错误。
- 2 指定的命令行选项无效。
- 99 发生了意外的异常。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted (未确定)

另请参见

[pkgrepo\(1\)](#)、[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名	pkgmk - 生产可安装的软件包
用法概要	<pre>pkgmk [-o] [-a arch] [-b base_src_dir] [-d device] [-f prototype] [-l limit] [-p pstamp] [-r root_path] [-v version] [variable=value]... [pkginst]</pre>
描述	<p>pkgmk 实用程序生产可安装的软件包，用作 pkgadd(1M) 命令的输入。软件包内容是目录结构格式。</p> <p>命令将软件包 prototype(4) 文件作为输入，创建 pkgmap(4) 文件。prototype 文件中每个条目的内容将复制到相应的输出位置。有关内容（校验和、文件大小、修改日期）的信息都在 pkgmap 文件中计算和存储，还包括在 prototype 文件中指定的信息。</p> <p>pkgmk 搜索 prototype(4) 文件中列出的文件，如以下条件所述。注意：如果样例文件包含文件的显式位置以包括在软件包中，则以下搜索说明不适用。</p> <ol style="list-style-type: none"> 1. 如果 -b 或 -r 选项都未指定，prototype(4) 文件中列出的每个文件路径的文件名部分应可以在与 prototype(4) 文件相同的目录中找到。 2. 如果 -b 指定为相对路径（无前导"/"），则 base_src_dir 会放到针对 prototype(4) 文件的相对路径前面。将在 root_path 目录中搜索生成的路径。如果未指定 root_path，它缺省为"/"。 3. 如果 -b 指定为绝对路径（具有前导"/"），则 base_src_dir 会放到针对 prototype(4) 文件的相对路径前面，结果即为文件的位置。不搜索 root_path。 4. 如果指定了 -r 选项，则使用针对 prototype(4) 文件的全文件路径。相对路径前面都有 base_src_dir。如果未指定 base_src_dir，它缺省为""。在 root_path 的每个目录中搜索生成的路径。 <p>如果使用 "pkgproto a/relative/path" 或 "pkgproto a/relative/path=install/path" 创建了样例文件，则应该使用 -r root_path 选项指定 a/relative/path 的位置，使 pkgmk 可以正确定位源文件。</p> <p>软件包命令，包括 pkgmk，都可以识别 largefile(5)。它们处理大于 2 GB 的文件的方法与处理较小文件的方法相同。在当前的执行中，pkgadd(1M)、pkgtrans(1) 和其他软件包命令可以处理最大 4 GB 的数据流。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -a <i>arch</i> 使用 <i>arch</i> 覆盖 pkginfo(4) 文件中提供的体系结构信息。 -b <i>base_src_dir</i> 在前面添加指定的 <i>base_src_dir</i>，来定位源计算机上的可重定位目标文件。使用此选项搜索样例文件中的所有目标文件。pkgmk 应分别在 <i>/base_src_dir</i> 中查找目标文件，或者使用 -b 和 -r 选项定位目标文件。 -d <i>device</i> 在 <i>device</i> 上创建软件包。<i>device</i> 可以是可移除磁盘的绝对目录路径名或标识符。缺省设备是安装假脱机目录 (<i>/var/spool/pkg</i>)。 -f <i>prototype</i> 使用 <i>prototype</i> 文件作为命令的输入。缺省的 <i>prototype</i> 文件名是 [Pp]rototype。

- `-l limit` 将输出设备的 512 字节块的最大大小指定为 `limit`。在缺省情况下，如果输出文件是一个目录或可挂载设备，`pkgmk` 使用 `df(1M)` 命令动态地计算输出设备中的可用空间容量。此选项在与 `pkgtrans(1)` 结合使用时可用来创建数据流格式的软件包。
- `-o` 覆盖相同的实例；已存在的软件包实例会被覆盖。
- `-p pstamp` 使用 `pstamp` 覆盖 `pkginfo(4)` 文件中的生产标记定义。
- `-r root_path` 使用附加了源路径名的指定 `root_path` 来定位源计算机上的目标文件，可以用逗号 (,) 作为路径元素的分隔符。如果指定了此选项，在指定的每个目录中搜索全目标路径。如果 `-b` 或 `-r` 选项都未指定，则在当前目录中搜索叶文件名。
- `-v version` 使用 `version` 覆盖 `pkginfo(4)` 文件中提供的版本信息。
- `variable=value` 将指定的变量放在软件包环境中。（有关变量规范的定义，请参见 `prototype(4)`。）

操作数

支持下列操作数：

- `pkginst` 按软件包实例指定软件包。实例可以是软件包缩写或特定的实例（例如，`inst.1` 或 `inst.2`）。可以通过 `inst.*` 请求软件包的所有实例。星号字符 (*) 对于某些 shell 来说是特殊字符，可能需要转义。在 C-Shell 中，* 必须由单引号 (') 括起来或者在前面加反斜杠 (\)。

退出状态

将返回以下退出值：

- 0 成功完成。
- >0 出现错误。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	system/core-os

另请参见

`pkgparam(1)`、`pkgproto(1)`、`pkgtrans(1)`、`uname(1)`、`df(1M)`、`pkgadd(1M)`、`pkginfo(4)`、`pkgmk`
 《Application Packaging Developer's Guide》

附注

体系结构信息在命令行中通过 `-a` 选项提供，或在 `prototype(4)` 文件中提供。如果没有提供体系结构信息，`pkgmk` 使用 `uname -m` 的输出。（请参见 `uname(1)`）。

版本信息在命令行中通过 `-v` 选项提供，或在 `pkginfo(4)` 文件中提供。如果没有提供版本信息，将提供基于当前日期的缺省信息。

体系结构和版本的命令行定义将覆盖 `prototype(4)` 的定义。

如果使用以下区域相关参数的无效组合之一，`pkgmk` 将失败。

1. SUNW_PKG_ALLZONES 和 SUNW_PKG_THISZONE 都设置为 TRUE。
2. SUNW_PKG_HOLLOW 设置为 TRUE，且 SUNW_PKG_ALLZONES 设置为 FALSE。
3. 软件包包含请求脚本，且 SUNW_PKG_THISZONE 设置为 TRUE。

有关这些参数的其他信息，请参见 [pkginfo\(4\)](#)。

引用名	pkgmogrify - 映像包管理系统清单变换程序
用法概要	<pre> /usr/bin/pkgmogrify [-vi] [-I includedir ...] [-D macro=value ...] [-O outputfile] [-P printfile] [inputfile ...] </pre>
描述	<p>pkgmogrify 允许对软件包清单进行程式编辑，以简化自动生成软件和自动发布软件包时所需的典型变换。</p> <p>pkgmogrify 提供：</p> <ul style="list-style-type: none"> ■ 宏替换，以便于在各种体系结构和平台间共享单个清单。 ■ 包含其他清单或清单片段，例如标准组件和变换。 ■ 变换软件包操作，包括修改、删除或添加操作属性。
选项	<p>支持以下选项：</p> <p>-?</p> <p>--help 显示用法消息。</p> <p>-D <i>macro</i>= <i>value</i> 将 <i>macro</i> 定义为一个宏，其值为 <i>value</i>。宏在输入文件中显示为 $\\$(macro)$。将重复执行替换操作，直到再也找不到变换。常见用语包括：</p> <ul style="list-style-type: none"> ■ 通过在行的开头使用特定于体系结构的标记来删除其他体系结构清单中的行。 $\\$(sparc_ONLY)file \dots$ 处理 SPARC 体系结构时，该宏将被设置为空字符串。而处理其他体系结构时，该宏在命令行中会被设置为 #，从而从当前体系结构的清单中删除此操作。 ■ 指定路径名称中特定于平台的部分，例如，可执行文件和库的 64 位体系结构目录的名称： $file NOHASH path=usr/bin/\\$(ARCH64)/cputrack \dots$ 应当将这些宏设置为命令行中所需的值。不存在预定义的宏值。 <p>-I <i>includedir</i> 将指定目录添加到在命令行和在嵌入式 include 指令中指定的文件的搜索路径中。</p> <p>-O <i>outputfile</i> 将清单输出写入指定的文件中。如果出现错误或 transform 指令引起中止操作，则不会写入该文件。缺省情况下，清单输出会写入 stdout。</p> <p>-P <i>printfile</i> 将 transform 指令输出操作生成的输出写入指定的文件。如果出现错误或 transform 指令引起中止操作，则不会写入该文件。缺省情况下，输出会写入 stdout。</p> <p>-i 忽略文件中的 include 指令。仅处理在命令行（或 stdin）中指定的文件。</p>

-v

将注释写入显示变换效果的输出清单。该信息可以帮助调试。

嵌入式指令

清单文件支持两种类型的指令：`include` 指令和 `transform` 指令。

Include 指令的格式为：

```
<include file>
```

该指令能够使 `pkgmogrify` 首先在当前目录中搜索名为 `file` 的文件，然后在 `-I` 选项指定的目录中搜索该文件。如果找到，则会将该文件的内容插入清单中遇到该指令的位置。如果未找到，`pkgmogrify` 将退出并显示错误。

Transform 指令的格式为：

```
<transform matching-criteria -> operation>
```

这些指令会进行累积（直到将所有输入都读入内存中），然后按遇到这些指令的顺序应用到操作。

匹配条件的格式为：

```
[action-name ... ] [attribute=<value-regexp> ...]
```

必须至少匹配一个指定的 `action-name`。必须匹配指定的每个 `attribute`。操作名称和属性列在 `pkg(5)` 手册页的“操作”中。使用的正则表达式语法是 Python 语法。有关 Python 正则表达式语法的更多信息，请使用命令 `pydoc re` 或请参见 <http://docs.python.org/dev/howto/regex.html> 中更为完整的文档。该正则表达式固定于开头而非结尾。因此，通过文件扩展名进行匹配的正则表达式匹配文件必须包含前导 `*`，且应当包含尾随 `$`。

可以指定多个条件，以空格分隔。

有以下操作可用：

- add** 将值添加到属性中。此操作使用两个参数。第一个参数是属性的名称，第二个参数是对应值。
- default** 如果不存在属性值，则设置该属性值。此操作与 `add` 操作使用相同的两个参数。
- delete** 删除属性值。此操作使用两个参数。第一个参数是属性的名称。第二个参数是要与删除的属性值匹配的正则表达式。与用于匹配操作的正则表达式不同，该正则表达式并不是固定的。
- drop** 放弃该操作。
- edit** 修改操作的属性。此操作使用三个参数。第一个参数是属性的名称，第二个参数是与属性值匹配的正则表达式。第三个参数是用于替换正则表达式匹配的部分值的替换字符串。与用于匹配操作的正则表达式不同，该正则表达式并不是固定的。如果在正则表达式中定义了组，则在替换字符串中可以使用 `\1`、`\2` 等格式的一般正则表达式向后引用。

<code>emit</code>	向清单输出流中发出一行。此行必须是有效操作字符串、为空（导致空白行）或注释（ <code>#</code> 后面跟有任意文本）。
<code>exit</code>	终止清单处理过程。不输出任何清单，也不应用任何 <code>print</code> 操作。如果给定一个参数，该参数必须是整数并且用作退出代码。缺省为0。如果给定两个参数，则第一个参数是退出代码，第二个参数是要输出到 <code>stderr</code> 的消息。
<code>print</code>	将消息输出到 <code>-P</code> 指定的输出文件中。
<code>set</code>	设置属性的值。此操作与 <code>add</code> 操作使用相同的两个参数。

除了 `delete` 和 `drop` 以外的所有操作都使用其内容输出到输出流的参数（可能是可选的）。这些字符串可能包含三种不同类型的特殊标记，这些标记允许输出包含不基于每种操作的固定变换的信息。

第一种替换通过将属性的名称放置在括号内（跟随在百分号后面），允许操作引用当前操作的属性值。例如，`%(alias)` 引用操作的 `alias` 属性的值。

存在几个合成属性。以下两个对 `pkgmogrify` 而言是唯一的：

- `pkg.manifest.filename` 引用在其中找到操作的文件的名称。
- `pkg.manifest.lineno` 引用在其中找到操作的行。

以下三个合成属性类似于 `pkg` 中使用的属性：

- `action.hash` 引用操作的散列值（如果该操作携带有效负荷）。对于携带有效负荷的操作，`set` 操作可以通过对 `action.hash` 属性进行操作来更改该操作的散列。
- `action.key` 引用关键属性的值。
- `action.name` 引用操作的名称。

如果请求其值的属性不存在，`pkgmogrify` 将退出并显示错误。为防止出现错误退出，请在属性名称后面附加 `;notfound=` 以及要用于替换属性值的值。例如，如果存在 `alias` 属性，则 `%(alias;notfound='no alias')` 输出该属性的值，否则输出 `no alias`。

如果请求其值的属性具有多个值，则输出每个值，以空格隔开。与 `notfound` 标记类似，标记 `prefix`、`suffix` 和 `sep` 也可用于更改此行为。由 `prefix` 表示的字符串放置在每个值之前，由 `suffix` 表示的字符串放置在每个值之后，而 `sep` 放置在某个值的后缀与下一个值的前缀之间。

与操作属性类似，`pkgmogrify` 指令也可以使用花括（而非圆括号）引用件包属性：`%(pkg.fmri)`。应用 `transform` 指令时，必须已在 `set` 操作中定义了属性，否则会将其视为 `notfound`，如上所述。当处理过程到达清单文件（介绍软件包）的结尾处时，将清除下一个软件包的属性。

这不仅在将软件包属性视为操作属性来引用方面，而且在匹配甚至暂时修改这些属性方面，都很有用。因此，在这些情况下都可以使用合成属性名称 `pkg`（仅在 `pkgmogrify` 上下文中）。

如果 `pkgmogrify` 完成读取在命令行中指定的清单并且该清单定义了 `pkg.fmri` 软件包属性，`pkgmogrify` 会创建此合成 `pkg` 操作，其属性为软件包的属性。`<transform>` 指令随后会对此操作进行匹配，正如它会对任何其他操作进行匹配一样。

`pkg` 操作中的操作是特殊的，它们仅在内存中进行，不会直接影响发出的清单。例如，尝试通过 `add`、`default` 或 `set` 操作设置 `pkg` 操作的属性时，不会使 `set` 操作添加到清单中，虽然该操作将可用于其他 `<transform>` 指令进行匹配。尝试 `emitpkg` 操作会导致错误。要添加软件包属性，改为 `emitset` 操作。

第三种替换是逆向引用功能。该替换与在 `edit` 操作中可使用的替换不同，它是对 `->` 左侧的变换匹配中列出组的引用。它们由 `%<1>`、`%<2>` 等表示（以在匹配条件中显示的顺序）。

处理顺序如下所示：

1. 从输入文件中读取行。
2. 应用宏。
3. 处理 `<include ...>` 和 `<transform>` 指令，从而找到并读取更多文件。
4. 累积所有输入之后，输出中的每行都会转换为操作并应用所有变换。
5. 成功完成处理后，写入输出。

示例

示例1 将标记添加到 SMF 清单中

将标记添加到服务管理工具 (Service Management Facility, SMF) 清单中，以便在活动系统上安装软件包时导入这些标记。

```
<transform file path=(var|lib)/svc/manifest/*.xml -> \
    add restart_fmri svc:/system/manifest-import:default>
```

示例2 移动文件

将文件从 `usr/sfw/bin` 移至 `usr/bin`。

```
<transform file -> edit path usr/sfw/bin usr/bin>
```

示例3 指定需要重新引导

将 `reboot-needed` 标记添加到 `/kernel` 下 `.conf` 文件以外的文件中。请注意，以下示例利用了按照在输入文件中看见的顺序将变换应用到每个操作的方式。

```
<transform file path=kernel/*. -> set reboot-needed true>
<transform file path=kernel/*.conf -> delete reboot-needed .*>
```

这还可以通过包含正则表达式的单个变换规则完成。

示例4 将FMRI属性转换为Depend操作

将软件包属性 `pkg.fmri` 转换为 `depend` 操作，使其成为 `incorporation` 的一部分。

```
<transform set name=pkg.fmri -> \
    emit depend type=incorporate fmri=%(value)>
<transform set name=pkg.fmri -> drop>
```

示例5 输出错误编号列表

输出带有双引号和前缀的错误编号的逗号分隔列表。

```
set name=bugs value=12345 value=54321 value=13579 value=97531
<transform set name=bugs -> \
    print %(value;sep=",";prefix="bug='";suffix="'")>
```

示例6 允许丢失属性

即使是在丢失属性时，也可以安全输出消息。

```
<transform driver -> print Found aliases: %(alias;notfound=<none>)>
```

示例7 设置缺省值

设置缺省所有者、组以及权限的值。

```
<transform file dir -> default owner root>
<transform file dir -> default group bin>
<transform file -> default mode 0444>
<transform dir -> default mode 0755>
```

示例8 将依赖项添加到未标记为已过时的软件包中

对于未标记为已过时的任何软件包，为提供软件包的合并添加对 `incorporation` 的依赖。必须在读入清单后执行该组变换，否则始终发出依赖项。因为修改 `pkg` 操作不会永久起作用，所以无需清除匹配 `pkg.obsolete=false` 的属性。

```
<transform pkg -> default pkg.obsolete false>
<transform pkg pkg.obsolete=false -> emit depend \
    fmri=consolidation/$(CONS)/$(CONS)-incorporation type=require>
```

示例9 发现问题时退出并输出消息

在清单中发现已过时属性时退出并输出错误消息。

```
<transform file dir link hardlink opensolaris.zone=.* -> \
    exit 1 The opensolaris.zone attribute is obsolete.>
```

示例10 设置合适的语言环境侧面

设置适用于正在考虑的路径名称的语言环境侧面。

示例 10 设置合适的语言环境侧面 (续)

```
<transform dir file link hardlink path=.*/locale/([^/]+).* -> \
  default facet.locale.%<1> true>
```

退出状态

将返回以下退出值：

- 0 一切正常工作。
- 1 出现预料中的错误情况。
- 2 指定的命令行选项无效。
- 99 意外的处理错误。

文件

`/usr/share/pkg/transforms`

该目录包含使用有用变换设置侧面、执行器和其他属性的文件。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted (未确定)

另请参见

[pkg\(1\)](#)、[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名 pkgparam – 显示软件包参数值

用法概要 pkgparam [-v] [-d *device*] [-R *root_path*] *pkginst* [*param*]...
 pkgparam -f *filename* [-v] [*param*]...

描述 pkgparam 显示与命令行中请求的参数关联的值。这些值位于 *pkginst* 的 [pkginfo\(4\)](#) 文件中或通过 -f 选项命名的特定文件中。

每行显示一个参数值。除非使用 -v 选项，否则只给出参数值。使用此选项，命令的输出为以下格式：

```
parameter1='value1'
parameter2='value2'
parameter3='value3'
```

如果命令行未指定任何参数，则显示与软件包关联的所有参数的值。

选项 此命令的选项和参数是：

-d *device* 指定存储 *pkginst* 的 *device*。它可以是磁带或可移除磁盘的目录路径名或标识符（例如，`/var/tmp` 或 `/dev/dsk/c1d0s0`）。特殊的令牌 `spool` 可以用来表示缺省的安装假脱机目录（`/var/spool/pkg`）。

-f *filename* 读取参数值的 *filename*。

-R *root_path* 定义要用作 *root_path* 的子目录全路径名。所有文件，包括软件包系统信息文件，都重定位到指定的 *root_path* 下开始的目录树。

-v 详细模式。显示参数名称及其值。

操作数 *pkginst* 定义应为其显示参数值的特定软件包实例。

param 定义应显示其值的特定参数。

错误 如果指定软件包的参数信息不可用，命令以非零状态退出。

退出状态 0 成功完成。

>0 出现错误。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[pkgmk\(1\)](#)、[pkgproto\(1\)](#)、[pkgtrans\(1\)](#)、[pkgadd\(1M\)](#)、[pkginfo\(4\)](#)、[attributes\(5\)](#)、[largefile\(5\)](#)

《Application Packaging Developer's Guide》

附注

使用 `-f` 选项，可以指定应从中提取参数值的文件。此文件应与 `pkginfo(4)` 文件格式相同。例如，此类文件可能在软件包开发过程中创建，而在软件测试的此阶段使用。

软件包命令可识别 `largefile(5)`。它们处理大于 2 GB 的文件的方法与处理较小文件的方法相同。在当前的执行中，`pkgadd(1M)`、`pkgtrans(1)` 和其他软件包命令可以处理最大 4 GB 的数据流。

引用名	pkgproto - 生成原型文件条目，以用作 pkgmk 命令的输入
用法概要	pkgproto [-i] [-c class] [path1] pkgproto [-i] [-c class] [path1=path2]...
描述	<p>pkgproto 扫描指定的路径并生成 prototype(4) 文件条目（可用作 pkgmk(1) 命令的输入）。</p> <p>如果未在命令行中指定路径，则会采用标准输入作为路径列表。如果命令行中列出的路径名是目录，则会对目录的内容进行搜索。不过，如果是从 <code>stdin</code> 读取输入，则不会搜索指定为路径名的目录。</p> <p>pkgproto 之类的软件包命令能够识别 largefile(5)。它们处理大于 2 GB 的文件的方法与处理较小文件的方法相同。在当前的执行中，pkgadd(1M)、pkgtrans(1) 和其他软件包命令可以处理最大 4 GB 的数据流。</p>
选项	<p>-i 忽略符号链接并将路径记录为 <code>f</code>type=f（文件）而不是与之相对的 <code>f</code>type=s（符号链接）。</p> <p>-c class 将所有路径的类映射到 <code>class</code>。</p>
操作数	<p><code>path1</code> 目标文件所在的路径名。</p> <p><code>path2</code> 输出中应取代 <code>path1</code> 的路径名。</p>
示例	<p>示例1 基本用法</p> <p>以下示例显示了 pkgproto 的常见用法和所产生的输出的部分列表。</p> <pre>example% pkgproto /bin=bin /usr/bin=usrbin /etc=etc f none bin/sed=/bin/sed 0775 bin bin f none bin/sh=/bin/sh 0755 bin daemon f none bin/sort=/bin/sort 0755 bin bin f none usrbin/sdb=/usr/bin/sdb 0775 bin bin f none usrbin/shl=/usr/bin/shl 4755 bin bin d none etc/master.d 0755 root daemon f none etc/master.d/kernel=/etc/master.d/kernel 0644 root daemon f none etc/rc=/etc/rc 0744 root daemon</pre> <p>示例2 在管道中使用 pkgproto</p> <p>以下命令显示了接受 find 命令的输出的 pkgproto。</p> <pre>example% find / -type d -print pkgproto d none / 755 root root d none /bin 755 bin bin d none /usr 755 root root d none /usr/bin 775 bin bin d none /etc 755 root root d none /tmp 777 root root</pre>

退出状态 0 成功完成。
 >0 出现错误。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [pkgmk\(1\)](#)、[pkgparam\(1\)](#)、[pkgtrans\(1\)](#)、[pkgadd\(1M\)](#)、[prototype\(4\)](#)、[attributes\(5\)](#)、[largefile\(5\)](#)

 《Application Packaging Developer's Guide》

附注 缺省情况下，`pkgproto` 为所遇到的任何符号链接创建符号链接条目 (`ftype=s`)。当使用 `-i` 选项时，`pkgproto` 为符号链接创建文件条目 (`ftype=f`)。必须对 [prototype\(4\)](#) 文件进行编辑以分配 `v`（可变）、`e`（可编辑）或 `x`（专用目录）等文件类型。`pkgproto` 检测链接的文件。如果多个文件链接在一起，则遇到的第一个路径将被视为链接的源。

 缺省情况下，`pkgproto` 将原型条目输出到标准输出中。不过，应将输出保存在文件中（方便起见，命名为 `Prototype` 或 `prototype`），以用作 [pkgmk\(1\)](#) 命令的输入。

引用名	pkgrecv - 映像包管理系统内容检索实用程序
用法概要	<pre> /usr/bin/pkgrecv [-s <i>src_uri</i>] [-a] [-d (<i>path dest_uri</i>)] [-c <i>cache_dir</i>] [-kr] [-m <i>match</i>] [-n] [--raw] [--key <i>keyfile</i> --cert <i>certfile</i>] (<i>fmri pattern</i>) ... /usr/bin/pkgrecv [-s <i>src_uri</i>] --newest </pre>
描述	<p>pkgrecv 允许用户检索 pkg(5) 系统信息库或软件包归档文件中的软件包。pkgrecv 还可以选择性地检索到的软件包重新发布到另一个软件包系统信息库中或者对其进行归档。缺省情况下，会以 pkg、pkg.depotd 以及软件包发布工具可使用的软件包系统信息库格式检索软件包。</p> <p>完成 pkgrecv 操作后，在系统信息库上运行 pkgrepo refresh 或 pkgrepo rebuild 以构建搜索索引。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -h 显示用法消息。 -a 将检索的软件包数据存储在 -d 指定的位置中的 pkg(5) 归档文件中。该文件不能已存在。此选项仅可以与基于文件系统的目标一起使用。尽管不要求，但还是强烈建议使用文件扩展名 .p5p（例如：archive.p5p）。该选项不能与 --raw 结合使用。 -c <i>cache_dir</i> 指定将用于缓存已下载内容的目录的路径。如果没有提供此目录，客户机将自动选择一个高速缓存目录。如果下载中断并且已自动选择了一个高速缓存目录，可使用该选项继续执行下载。有关如何设置用于存储临时数据的位置的详细信息，请参见下文的“环境变量”部分。 -d (<i>path dest_uri</i>) 指定要将软件包重新发布到的目标的文件系统路径或 URI。如果指定了 -a，则目标为尚不存在的新软件包归档文件。否则，目标必须是已存在的软件包系统信息库。可以使用 pkgrepo 创建新的系统信息库。 -k 使检索的软件包内容保持压缩状态。重新发布时会忽略该选项。压缩的软件包内容不得与 pkgsend 一起使用。 -m <i>match</i> 使用以下值控制匹配行为： <ul style="list-style-type: none"> all-timestamps 包括所有匹配的时间戳，而不仅仅是最新的时间戳（意味着 all-versions）。 all-versions 包括所有匹配的版本，而不仅仅是最新的版本。

- n
执行试运行，不进行任何更改。
- r
递归检索提供的软件包列表的所有依赖项。
- s *src_uri*
指定一个 URI，代表要从中接收软件包数据的 pkg(5) 系统信息库或软件包归档文件的位置。
- cert *certfile*
指定用于从 HTTPS 系统信息库进行软件包检索的客户机 SSL 证书文件。
- key *keyfile*
指定用于从 HTTPS 系统信息库进行软件包检索的客户机 SSL 密钥文件。
- newest
列出指定系统信息库中提供的最新版本的软件包，然后退出。（忽略 -s 以外的所有其他选项。）
- raw
按主干和版本，检索一组目录结构中的原始软件包数据并将其存储在 -d 指定的位置中。此选项仅可以与基于文件系统的目标一起使用。此软件包数据可用于方便地修改和重新发布软件包，也许通过更正文件内容，也许通过提供附加的软件包元数据。该选项不能与 -a 结合使用。

示例

示例1 列出最新的软件包

列出名为 `test` 的系统上的系统信息库中的最新软件包。

```
$ pkgrecv -s http://test --newest
pkg://solaris/system/library/c++-runtime@0.5.11,5.11-0.175.0.0.0.2.1:20120921T190358Z
pkg://solaris/system/library/freetype-2@2.4.8,5.11-0.175.1.0.0.7.1234:20120109T215840Z
pkg://solaris/system/library/math@0.5.11,5.11-0.175.0.0.0.2.1:20120921T190432Z
```

示例2 检索原始软件包数据

以能够与 `pkgsend publish` 结合使用的合适格式从示例 1 中接收 `c++-runtime` 软件包。

```
$ pkgrecv -s http://test \
-d /local/repo --raw \
c++-runtime@0.5.11,5.11-0.175.0.0.0.2.1:20120921T190358Z
Processing packages for publisher solaris ...
Retrieving and evaluating 1 package(s)...
PROCESS      ITEMS      GET (MB)      SEND (MB)
Completed    1/1        3.5/3.5       0.0/0.0
$ ls /local/repo
pkg5.repository publisher system%2Flibrary%2Fc%2B%2B-runtime
```


示例3 从系统中检索依赖项

从名为 test 的系统中接收软件包 editor/vim 及其所有依赖项。

```
$ pkgrecv -s http://test -d /local/repo -r editor/vim
```

示例4 检索所有版本

从名为 test 的系统中接收软件包 editor/vim 的所有版本。

```
$ pkgrecv -s http://test -d /local/repo -m all-versions \
editor/vim
```

```
Processing packages for publisher solaris ...
```

```
Retrieving and evaluating 2 package(s)...
```

PROCESS	ITEMS	GET (MB)	SEND (MB)
Completed	2/2	16.7/16.7	44.9/44.9

示例5 检索所有版本并远程重新发布

从名为 test 的系统中接收软件包 library/zlib 的所有版本，然后将其重新发布到名为 remote 的系统上的远程系统信息库中。

```
$ pkgrecv -s http://test -d http://remote:10000 \
-m all-versions library/zlib
```

示例6 从系统信息库中检索依赖项

从位于 /export/repo 的系统信息库中接收软件包 editor/gnu-emacs 及其所有依赖项。

```
$ pkgrecv -s /export/repo -d /local/repo -r editor/gnu-emacs
```

示例7 检索其他软件包和更改的内容

从位于 http://pkg.oracle.com/solaris/release/ 的系统信息库中将并非已存在的所有软件包和所有已更改内容接收到位于 /export/repoSolaris11 的系统信息库。

```
$ pkgrecv -s http://pkg.oracle.com/solaris/release/ \
-d /export/repoSolaris11 -m all-timestamps '*'
```

将位于 http://pkg.oracle.com/solaris/support/ 的安全系统信息库中的所有尚不存在的软件包和所有更改内容接收到位于 /export/repoSolaris11 的系统信息库。

```
$ pkgrecv -s http://pkg.oracle.com/solaris/support/ \
-d /export/repoSolaris11 -m all-timestamps \
--key /var/pkg/ssl/Oracle_Solaris_11_Support.key.pem \
--cert /var/pkg/ssl/Oracle_Solaris_11_Support.certificate.pem '*'
```

示例8 创建软件包归档文件

根据位于 http://example.com:10000 的系统信息库创建包含软件包 editor/gnu-emacs 及其所有依赖项的软件包归档文件。

```
$ pkgrecv -s http://example.com:10000 -d /my/emacs.p5p -a \
-r editor/gnu-emacs
```

示例 9 将软件包从归档文件复制到系统信息库中
 将软件包归档文件中的所有软件包复制到位于 `/export/repo` 中的现有系统信息库。

```
$ pkgrecv -s /my/archive.p5p -d /export/repo '*'
```

环境变量

支持以下环境变量：

PKG_DEST 要将检索到的软件包保存到的目录的路径，或者要复制软件包的系统信息库或软件包归档文件的文件系统路径或 URI。

PKG_SRC URI 或文件系统路径代表要从中检索软件包的 `pkg(5)` 系统信息库或软件包归档文件的位置。

TMPDIR 在程序执行期间用于存储临时数据的目录的绝对路径。如果未设置，则存储临时数据的缺省路径为 `/var/tmp`。

退出状态

将返回以下退出值：

0 命令成功。

1 出现错误。

2 指定的命令行选项无效。

3 请求了多项操作，但只有一部分操作成功。

99 发生了意外的异常。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted (未确定)

另请参见

[pkgrepo\(1\)](#)、[pkgsend\(1\)](#)、[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名	pkgrepo – 映像包管理系统的系统信息库管理实用程序
用法概要	<pre> /usr/bin/pkgrepo create [--version ver] uri_or_path /usr/bin/pkgrepo add-publisher -s repo_uri_or_path publisher ... /usr/bin/pkgrepo get [-F format] [-H] [-p publisher ...] -s repo_uri_or_path [section/property ...] /usr/bin/pkgrepo info [-F format] [-H] [-p publisher ...] -s repo_uri_or_path /usr/bin/pkgrepo list [-F format] [-H] [-p publisher ...] -s repo_uri_or_path [pkg_fmri_pattern ...] /usr/bin/pkgrepo rebuild [-p publisher ...] -s repo_uri_or_path [--no-catalog] [--no-index] /usr/bin/pkgrepo refresh [-p publisher ...] -s repo_uri_or_path [--no-catalog] [--no-index] /usr/bin/pkgrepo remove [-n] [-p publisher ...] -s repo_uri_or_path pkg_fmri_pattern ... /usr/bin/pkgrepo set [-p publisher] -s repo_uri_or_path section/property=[value] /usr/bin/pkgrepo set [-p publisher] -s repo_uri_or_path section/property=(<i>[value]</i>) ... /usr/bin/pkgrepo help /usr/bin/pkgrepo version </pre>
描述	<p>通过 <code>pkgrepo</code> 可以创建和管理 <code>pkg(5)</code> 软件包系统信息库。软件包系统信息库是一组预定义的目录和文件，允许 <code>pkg</code> 和发布客户机（例如 <code>pkgsend</code> 或 <code>pkgrcv</code>）存储和检索软件包数据。此外，当需要对软件包系统信息库进行基于网络的访问时，<code>pkg.depotd</code> 可以提供对该系统信息库的客户机访问权限，以存储和/或检索软件包数据。</p>
选项	<p>支持以下选项：</p> <pre> -? --help 显示用法消息。 </pre>
子命令	<p>支持以下子命令：</p> <pre> pkgrepo create [--version ver] uri_or_path 在指定的位置创建 <code>pkg(5)</code> 系统信息库。 该子命令仅可以与基于文件系统的系统信息库一起使用。 --version 使用与指定版本兼容的格式创建系统信息库。缺省情况下，会创建版本为 4 的系统信息库。支持的版本包括： </pre>

- 3 支持为单个发布者存储软件包，目录版本为 1，搜索版本为 1。
- 4 支持为多个发布者存储软件包，目录版本为 1，搜索版本为 1。

`pkgrepo add-publisher -s repo_uri_or_path publisher ...`

将指定的发布者添加到系统信息库中。新的发布者没有软件包或内容。

该子命令仅可以与基于第 4 版文件系统的系统信息库一起使用。

`pkgrepo get [-F format] [-H] [-p publisher ...] -s repo_uri_or_path [section/property ...]`

显示系统信息库或其发布者的属性信息。

缺省情况下，会在单独的行中显示每个属性及其值。空的 ASCII 字符串值用一对双引号 ("") 表示。ASCII 字符串值中的以下 Bourne shell 元字符以及换行符、空格符和制表符都必须使用反斜杠 (\) 进行转义：

`; & () | ^ < > \ " ' ``

有关显示发布者和系统信息库属性的示例，请参见“示例”部分。

有关属性列表以及每个属性的用途和值，请参见下文的 `set` 子命令。

-F format

指定备用输出格式。*format* 的值可以是 `tsv`（以制表符分隔的值）、`json`（单行 JavaScript 对象表示法）或 `json-formatted`（格式易于阅读的 JavaScript 对象表示法）。

-H

在列出时省略标题。

-p publisher

显示给定发布者的属性信息。特殊值 `all` 显示所有发布者的属性。可以多次指定此选项。

-s repo_uri_or_path

对位于给定 URI 或文件系统路径的系统信息库进行操作。

section/property

仅显示指定属性的值，例如 `publisher/prefix` 或 `repository/version`。有关完整的属性列表，请参见 `set` 子命令。

`pkgrepo info [-F format] [-H] [-p publisher ...] -s repo_uri_or_path`

显示系统信息库已知的软件包发布者的列表。该列表包括每个发布者的软件包数量、最后一次更新发布者的软件包数据的时间以及发布者的软件包数据的状态（例如当前是否处于正在处理状态）。

-F format

指定备用输出格式。*format* 的值可以是 `tsv`（以制表符分隔的值）、`json`（单行 JavaScript 对象表示法）或 `json-formatted`（格式易于阅读的 JavaScript 对象表示法）。

-H
在列出时省略标题。

-p *publisher*
仅显示给定发布者的数据。如果没有提供该选项，将显示所有发布者的数据。可以多次指定此选项。

-s *repo_uri_or_path*
对位于给定 URI 或文件系统路径的系统信息库进行操作。

`pkgrepo list [-F format] [-H] [-p publisher ...] -s repo_uri_or_path [pkg_fmri_pattern ...]`
列出 *repo_uri_or_path* 系统信息库中与指定 *pkg_fmri_pattern* 模式匹配的软件包。如果未指定任何模式，将列出系统信息库中的所有软件包。

在缺省输出中，第一列包含软件包发布者的名称。第二列包含软件包的名称。第三列是显示软件包状态的标志。状态列中的 *o* 值表示软件包已过时。状态列中的 *r* 值表示已重命名软件包，但格式已过时。第四列包含软件包的发行版本和分支版本。有关发行版本和分支版本的信息，请参见 `pkg(5)`。

-F *format*
指定备用输出格式。*format* 的值可以是 `tsv`（以制表符分隔的值）、`json`（单行 JavaScript 对象表示法）或 `json-formatted`（格式易于阅读的 JavaScript 对象表示法）。

-H
在列出时省略标题。

-p *publisher*
仅显示给定发布者的软件包。如果没有提供该选项，将显示所有发布者的软件包。可以多次指定此选项。

-s *repo_uri_or_path*
对位于给定 URI 或文件系统路径的系统信息库进行操作。

`pkgrepo rebuild [-p publisher ...] -s repo_uri_or_path [- no-catalog] [--no-index]`
放弃在系统信息库中找到的所有目录、搜索以及其他缓存信息，然后根据系统信息库的当前内容重新创建这些信息。

-p *publisher*
仅针对给定发布者执行操作。如果没有提供该选项或者指定了特定值 `all`，则对所有发布者执行操作。可以多次指定此选项。

-s *repo_uri_or_path*
对位于给定 URI 或文件系统路径的系统信息库进行操作。

--no-catalog
不重新生成软件包数据。

--no-index
不重新生成搜索索引。

`pkgrepo refresh [-p publisher ...] -s repo_uri_or_path [- --no-catalog] [--no-index]`

将在系统信息库中找到的所有新软件包编入目录并更新所有搜索索引。这主要供延迟的发布使用（`pkgsend` 的 `--no-catalog` 或 `--no-index` 选项）。

`-p publisher`

仅针对给定发布者执行操作。如果没有提供该选项或者指定了特定值 `all`，则对所有发布者执行操作。可以多次指定此选项。

`-s repo_uri_or_path`

对位于给定 URI 或文件系统路径的系统信息库进行操作。

`--no-catalog`

不添加任何新软件包。

`--no-index`

不更新搜索索引。

`pkgrepo remove [-n] [-p publisher ...] -s repo_uri_or_path pkg_fmri_pattern ...`

从系统信息库中删除与指定模式匹配的软件包，其中包括这些软件包引用的且其他任何软件包没有使用的所有文件。

注-删除关联发布者的所有搜索索引数据。

该子命令仅可以与基于文件系统的系统信息库一起使用。

注意-此操作不可逆并且不得在其他客户机正在访问系统信息库时使用，因为这样会使得它们在执行检索操作期间出现故障。

`-n`

试运行操作而不进行软件包更改。在退出之前，会显示要删除的软件包的列表。

`-p publisher`

仅删除给定发布者的匹配软件包。如果没有提供该选项，会删除所有发布者的所有匹配软件包。可以多次指定此选项。

`-s repo_uri_or_path`

对位于给定 URI 或文件系统路径的系统信息库进行操作。

`pkgrepo set [-p publisher] -s repo_uri_or_path section/property=[value] ...`

`set [-p publisher] -s repo_uri_or_path section/property=([value]) ...`

为系统信息库或发布者设置指定属性的值。

该子命令仅可以与基于文件系统的系统信息库一起使用。

`-p publisher`

仅为给定发布者设置属性数据。如果发布者尚未存在，将添加该发布者。特殊值 `all` 可用于设置所有发布者的属性。

`-s repo_uri_or_path`

对位于给定 URI 或文件系统路径的系统信息库进行操作。

可以使用以下格式之一指定属性和值：

section/property=
清除属性值。

section/property= value
将属性值替换为给定值。

section/property=(value1 value2 valueN)
将属性值替换为值列表。

对于系统信息库第 3 和 4 版，可以为系统信息库设置以下属性：

publisher/prefix 代表缺省发布者名称的字符串。第一个字符必须是 a-z、A-Z 或 0-9。该字符串的剩余部分只能包含字符 0-9、-、.、a-z 以及 A-Z。该值指示存在多个发布者的软件包时或者将软件包发布到系统信息库但没有指定发布者时，应当使用的发布者。

对于系统信息库第 3 和 4 版，可以为系统信息库中的各发布者设置以下属性：

publisher/alias 字符串，代表在使用系统信息库的配置数据添加发布者时客户机应当使用的缺省别名。第一个字符必须是 a-z、A-Z 或 0-9。该字符串的剩余部分只能包含字符 0-9、-、.、a-z 以及 A-Z。

repository/collection_type 可以使用值 `core` 或 `supplemental`，以表明此系统信息库中提供的软件包类型。

`core` 类型表明系统信息库包含该库中的软件包所声明的所有依赖项。`core` 类型主要用于操作系统的系统信息库。

`supplemental` 类型表明系统信息库包含依赖于另一个系统信息库中的软件包或要与另一个系统信息库中软件包一起使用的软件包。

repository/description 纯文本段落，描述系统信息库的用途和内容。

repository/detailed_url URI，代表提供更多有关系统信息库信息的文档的位置（例如网页）。

repository/legal_uris 文档的位置列表 (URI)，提供关于系统信息库的其他法律信息。

repository/mirrors 系统信息库的位置列表 (URI)，这些系统信息库包含系统信息库的软件包内容的副本但不包含软件包元数据。

repository/name 纯文本字符串，包含系统信息库的名称。

<code>repository/origins</code>	系统信息库的位置列表 (URI)，这些系统信息库包含该系统信息库的软件包元数据和内容的完整副本。
<code>repository/refresh_seconds</code>	整数值，表示客户机在每次更新检查之后和检查系统信息库以查找更新的软件包数据之前应当等待的秒数。
<code>repository/registration_uri</code>	代表资源位置的 URI，必须使用该位置才能获取访问系统信息库的证书。注册网页就是一个示例。
<code>repository/related_uris</code>	系统信息库的位置列表 (URI)，这些系统信息库包含用户可能感兴趣的软件包。

此处没有记录但列在 `get` 子命令输出中的属性保留供内部使用，不得对其进行设置。

`pkgrepo help`
显示用法消息。

`pkgrepo version`
显示一个用于唯一标识 `pkg(5)` 系统版本的字符串。由 `version` 操作生成的值不能进行排序，并且对于在不平等情况下的比较而言是不安全的。

示例

示例1 创建软件包系统信息库

```
$ pkgrepo create /my/repository
```

示例2 显示信息

显示发布者摘要以及系统信息库中软件包的数量。

```
$ pkgrepo info -s /my/repository
PUBLISHER PACKAGES STATUS UPDATED
example.com 5          online 2011-07-22T18:09:09.769106Z
$ pkgrepo info -s http://pkg.oracle.com/solaris/release/
PUBLISHER PACKAGES STATUS UPDATED
solaris 3941         online 2010-11-12T19:24:25.967246Z
```

示例3 重新生成目录和搜索数据

重新生成系统信息库的目录和搜索数据。

```
$ pkgrepo rebuild -s /my/repository
```

示例4 刷新目录和搜索数据

刷新系统信息库的目录和搜索数据。

```
$ pkgrepo refresh -s /my/repository
$ pkgrepo refresh -s http://example.com/repository
```


示例5 显示所有系统信息库属性

```
$ pkgrepo get -s /export/repoSolaris11
SECTION    PROPERTY    VALUE
publisher  prefix      solaris
repository description Local\ copy\ of\ the\ Oracle\ Solaris\ 11\ repository
repository name      Oracle\ Solaris\ 11
repository version    4
$ pkgrepo get -s http://pkg.oracle.com/solaris/release/
SECTION    PROPERTY    VALUE
deployment content    s11_11-11
deployment pubdate    20111102T222051Z
publisher  prefix      solaris
repository version    4
```

示例6 显示所有发布者属性

```
$ pkgrepo get -s http://pkg.oracle.com/solaris/release/ -p all
PUBLISHER SECTION    PROPERTY    VALUE
solaris  publisher alias
solaris  publisher prefix      solaris
solaris  repository collection-type core
solaris  repository description This\ repository\ serves\ the\ Oracle\
Solaris\ 11\ Package\ repository.
solaris  repository legal-uris  ()
solaris  repository mirrors    (http://pkg-cdn1.oracle.com/solaris.release/)
solaris  repository name      Oracle\ Solaris\ 11\ Package\ Repository
solaris  repository origins   ()
solaris  repository refresh-seconds
solaris  repository registration-uri ""
solaris  repository related-uris  ()
```

示例7 设置缺省发布者

```
$ pkgrepo set -s /my/repository publisher/prefix=example.com
```

示例8 设置发布者属性

```
$ pkgrepo set -s /my/repository -p example.com \
repository/origins=http://example.com/repository
```

示例9 将新的发布者添加到系统信息库中

```
$ pkgrepo add-publisher -s /my/repository example.com
```

退出状态

将返回以下退出值：

- 0 命令成功。
- 1 出现错误。
- 2 指定的命令行选项无效。

- 3 请求了多项操作，但只有一部分操作成功。
- 4 没有进行更改时，无需执行任何操作。
- 99 发生了意外的异常。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted（未确定）

另请参见

[pkg\(1\)](#)、[pkgrecv\(1\)](#)、[pkgsend\(1\)](#)、[pkg.depotd\(1M\)](#)、[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名	pkgsend - 映像包管理系统发布客户机
用法概要	<pre> /usr/bin/pkgsend [options] command [cmd_options] [operands] /usr/bin/pkgsend generate [-T pattern] [--target file] source ... /usr/bin/pkgsend publish [-b bundle ...] [-d source ...] [-s repo_uri_or_path] [-T pattern] [--n</pre>
描述	<p>通过 <code>pkgsend</code>，可使用软件包清单将新软件包和新软件包版本发布到映像包管理系统信息库。要创建或管理系统信息库，请参见 <code>pkgrepo(1)</code>。要从现有系统信息库的软件包中创建软件包归档文件，请参见 <code>pkgrecv(1)</code>。有关软件包清单的更多信息，请参见 <code>pkg(5)</code>。</p> <p>完成 <code>pkgsend</code> 操作后，在系统信息库上运行 <code>pkgrepo refresh</code> 或 <code>pkgrepo rebuild</code> 以构建搜索索引。</p>
选项	<p>支持以下选项：</p> <pre> -? --help 显示用法消息。</pre>
子命令	<p>支持以下子命令：</p> <pre> pkgsend generate [-T pattern] [--target file] source ...</pre> <p>读取每个 <code>source</code>（例如，SVR4 软件包、目录或 <code>tar</code> 文件）并将介绍 <code>source</code> 的清单发送到 <code>stdout</code>。在输出清单中，<code>file</code> 和 <code>dir</code> 操作将所有者设置为 <code>root</code>，并将组设置为 <code>bin</code>。</p> <p>然后可以注释输出清单，使用 <code>pkgdepend</code> 添加或分析依赖项，并在将其传递到 <code>publish</code> 子命令之前使用 <code>pkglint</code> 验证它的正确性。</p> <p>以下是支持的源：</p> <ul style="list-style-type: none"> ■ 文件系统格式 SVR4 软件包 ■ 数据流格式 SVR4 软件包 ■ <code>tar</code> 文件 ■ 目录 <p>如果源中文件的基本名称与使用 <code>-T</code> 指定的模式匹配，则将文件的时间戳添加到该文件的操作中。<code>pattern</code> 使用 <code>shell</code> 匹配规则：</p> <pre> * 匹配所有内容。 ? 匹配任何单个字符。 [seq] 匹配 <i>seq</i> 中的任何字符。 ![seq] 匹配不在 <i>seq</i> 中的任何字符。</pre> <p>如果指定源是一个目录，当单个 <code>inode</code> 具有多个路径名称时，则没有明确的方式区分 <code>file</code> 操作与 <code>hardlink</code> 操作。通常，在文件系统遍历中发现的第一个视为文</p>

件，其余的视为硬链接。这可以是任意的，具体取决于文件系统的实现方式。要指定哪些路径名称应视为文件，请将每个路径名称作为参数传递到 `--target` 选项。该选项不会影响其他类型的源，因为它们可以表明哪些路径名称是文件，哪些是硬链接。

当提供 SVR4 软件包作为源时，`pkgsend` 会确认不存在具有类操作脚本的文件，以及不存在安装前、安装后、删除前或删除后脚本。但与 `manifest` 类一起安装的任何 SMF 清单除外。将从所有可重定位路径中删除 `BASEDIR`。

SVR4 DESC 参数将转换为 `pkg.description` 值。SVR4 NAME 参数将转换为 `pkg.summary` 值。

```
pkgsend publish [-b bundle ...] [-d source ...] [-s repo_uri_or_path] [-T pattern]
[--no-catalog] [manifest ...]
```

将使用指定软件包清单的软件包发布到目标软件包系统信息库，并从提供的源中检索该软件包的文件。如果指定了多个清单，它们将以提供的顺序联接。如果未指定清单，则从 `stdin` 中读取清单。

如果未指定，则 `pkgsend publish` 向软件包 FMRI 添加内部版本。`publish` 工具还向软件包 FMRI 添加时间戳（UTC 中的当前时间）。有关软件包 FMRI 的版本字符串的信息，请参见 `pkg(5)` 手册页。

`-b bundle`

将指定的包添加到源列表，以便在查找清单中的文件时搜索该包。包是 `tar` 文件和 SVR4 软件包等源。如果多次指定该选项，则以在命令行显示的顺序对源进行搜索。如果同时指定 `-b` 和 `-d`，则首先搜索 `-d` 源。有关支持的包及其使用方法的说明，请参见以上的 `generate` 子命令。

`-d source`

将指定的目录添加到源列表，以便在查找清单中的文件时搜索该目录。如果多次指定该选项，则以在命令行显示的顺序对源进行搜索。有关支持的源及其使用方法的说明，请参见以上的 `generate` 子命令。

`-s repo_uri_or_path`

将软件包发布到位于给定 URI 或文件系统路径的系统信息库。有关发布限制和建议的更多信息，请参见下文的“附注”部分。另请参见“环境变量”部分。

`--no-catalog`

不将软件包添加到发布者的目录。当一次发布多个软件包，并且必须连续执行发布者目录更新时，建议使用该选项。完成发布后，可使用 `pkgrepo` 的 `refresh` 子命令将新软件包添加到相应的发布者目录。

有关 `-T` 选项的说明，请参见以上的 `generate` 子命令。

环境变量

`PKG_REPO` 目标系统信息库的路径或 URI。

示例

示例1 生成并发布软件包

使用 `pkgsend generate` 创建软件包并将其发布。

```
$ pkgsend generate /path/to/proto > /path/to/manifests/foo.p5m
```

将 `example.com` 发布者的软件包 FMRI 添加到 `foo.p5m` 的开头。

```
set name=pkg.fmri value=pkg://example.com/foo@1.0
```

结果清单应类似于以下内容：

```
set name=pkg.fmri value=pkg://example.com/foo@1.0
dir group=sys mode=0755 owner=root path=usr
dir group=bin mode=0755 owner=root path=usr/bin
file usr/bin/foo group=bin mode=0555 owner=root path=usr/bin/foo

$ pkgsend publish -s http://example.com:10000 -d /path/to/proto \
/path/to/manifests/foo.p5m
```

示例2 创建和发布普通软件包

为包含以下行的发布者 `example.com` 创建清单：

```
set name=pkg.fmri value=pkg://example.com/foo@1.0-1
file /exdir/foo mode=0555 owner=root group=bin path=/usr/bin/foo
```

发布软件包：

```
$ pkgsend publish -s http://example.com:10000 -d /exdir
```

示例3 使用已经存在的清单

使用基于文件系统的发布和已经存在的清单发布软件包。

```
$ pkgsend publish -s /tmp/example_repo -d /tmp/pkg_files \
/tmp/pkg_manifest
```

退出状态

将返回以下退出值：

- 0 命令成功。
- 1 出现错误。
- 2 指定的命令行选项无效。
- 99 发生了意外的异常。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg

属性类型	属性值
接口稳定性	Uncommitted (未确定)

另请参见

[pkgdepend\(1\)](#)、[pkgrepo\(1\)](#)、[pkg.depotd\(1M\)](#)、[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

附注

由于发布协议限制，当发布大小超过 128 MB 的单个软件包文件时，必须使用基于文件系统的发布。当需要系统信息库的访问控制时，也建议使用基于文件系统的发布。

当使用基于文件系统的发布时，在完成发布后必须重新启动提供目标系统信息库服务的任何 `pkg.depotd` 进程，以便在其 Web 界面或搜索响应中反映更改。有关更多信息，请参见 [pkg.depotd\(1M\)](#)。

引用名	pkgsign - 映像包管理系统签名实用程序
用法概要	<pre> /usr/bin/pkgsign [-a <i>hash_algorithm</i>] [-c <i>path_to_signing_certificate</i>] [-i <i>path_to_intermediate_cert</i>] ... [-k <i>path_to_private_key</i>] [-n] -s <i>path_or_uri</i> [--help] [--no-index] [--no-catalog] (<i>fmri pattern</i>) ... </pre>
描述	通过使用提供的密钥和证书添加签名操作， <code>pkgsign</code> 可在系统信息库中更新给定 FMRI 的清单。修改后的软件包保留原来的时间戳。
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> <code>--help</code> 显示用法消息。 <code>-a <i>hash_algorithm</i></code> 使用签名算法 <i>hash_algorithm</i> 而不是缺省值。缺省签名算法为 <code>rsa-sha256</code>。支持的签名算法包括 <code>rsa-sha256</code>、<code>rsa-sha384</code>、<code>rsa-sha512</code>、<code>sha256</code>、<code>sha384</code> 和 <code>sha512</code>。仅指定散列算法的签名算法会导致签名值为软件包清单的散列。指定 <code>rsa</code> 和散列算法的签名算法会导致签名值为使用提供的私钥进行签名的清单的散列（请参见 <code>-c</code> 和 <code>-k</code> 选项）。 <code>-c <i>path_to_signing_certificate</i></code> 添加证书 <i>path_to_signing_certificate</i> 作为验证操作中的签名值时所使用的证书。<code>-c</code> 选项仅可以与 <code>-k</code> 选项一起使用。 <code>-i <i>path_to_intermediate_cert</i></code> 添加证书 <i>path_to_intermediate_cert</i> 作为验证证书 <i>path_to_signing_certificate</i>（作为参数提供给 <code>-c</code>）时所使用的证书。通过多次指定 <code>-i</code>，可提供多个证书。 <code>-k <i>path_to_private_key</i></code> 使用存储在 <i>path_to_private_key</i> 中的私钥对清单进行签名。<code>-k</code> 选项只能与 <code>-c</code> 选项一起使用。如果未设置 <code>-k</code>，则签名值为清单的散列。 <code>-n</code> 执行试运行，不以任何方式更改系统信息库。 <code>-s <i>path_or_uri</i></code> 对位于 <i>path_or_uri</i> 的系统信息库中的软件包进行签名。 <code>--no-index</code> 在重新发布已签名的清单后不更新系统信息库搜索索引。 <code>--no-catalog</code> 在重新发布已签名的清单后不更新系统信息库目录。

示例

示例1 使用清单的散列值进行签名

使用清单的散列值对发布到 `http://localhost:10000` 的软件包进行签名。这通常用于测试。

```
$ pkgsign -s http://localhost:10000 -a sha256 \
example_pkg@1.0,5.11-0:20100626T030108Z
```

示例2 使用密钥和证书进行签名

使用 `rsa-sha384` 对发布到位于 `/foo/bar` 的文件系统信息库的软件包进行签名，以便对清单执行散列和签名操作。签名密钥位于 `/key/usr2.key` 中，其关联的证书位于 `/key/usr2.cert` 中，而用于验证证书的证书位于 `/icerts/usr1.cert` 中。

```
$ pkgsign -s file:///foo/bar/ -a rsa-sha384 \
-k /key/usr2.key -c /key/usr2.cert -i /icerts/usr1.cert \
example_pkg@1.0,5.11-0:20100626T031341Z
```

退出状态

将返回以下退出值：

- 0 命令成功。
- 1 出现错误。
- 2 指定的命令行选项无效。
- 3 请求了多项操作，但只有一部分操作成功。
- 99 发生了意外的异常。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg
接口稳定性	Uncommitted (未确定)

另请参见

[pkg\(1\)](#)、[pkgrecv\(1\)](#)、[pkgsend\(1\)](#)、[pkgrepo\(1\)](#)、[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

引用名	pkgtrans – 转换软件包格式
用法概要	pkgtrans [-inosg] [-k <i>keystore</i>] [-a <i>alias</i>] [-P <i>passwd</i>] <i>device1 device2</i> [<i>pkginst</i>]...
描述	<p>pkgtrans 实用程序用于将可安装的软件包从一种格式转换为另一种格式。它可进行下列转换：</p> <ul style="list-style-type: none"> ■ 从文件系统格式到数据流 ■ 从文件系统格式到已签名的数据流 ■ 从数据流到文件系统格式 ■ 从一种文件系统格式到另一种文件系统格式
选项	<p>此命令的选项和参数如下：</p> <p>-a <i>alias</i> 使用与友好名称别名相关联的公钥证书以及对应的私钥。有关更多信息，请参见 pkgadd(1M) 中的证书位置与密钥库和证书格式。</p> <p>-g 对结果数据流进行签名。</p> <p>-i 仅复制 pkginfo(4) 和 pkgmap(4) 文件。</p> <p>-k <i>keystore</i> 使用密钥库检索用于生成签名的私钥。如果未指定此选项，将搜索缺省位置来查找由 -a 指定的指定私钥。如果没有给定别名且密钥库中存在多个密钥，则 pkgtrans 将中止。有关搜索位置和格式的更多信息，请参见 pkgadd(1M) 中的密钥库位置与密钥库和证书格式。</p> <p> 当以非 root 用户运行时，用于证书搜索的缺省基目录是 <code>~/.pkg/security</code>，其中 <code>~</code> 是调用 pkgtrans 的用户的起始目录。</p> <p>-n 如果目标设备上已存在该软件包的任何实例，则会创建软件包的新实例，最大实例数由 pkginfo(4) 文件中的 MAXINST 变量指定。</p> <p>-o 覆盖目标设备上的相同实例。如果软件包实例已经存在，将覆盖现有实例。</p> <p>-P <i>passwd</i> 提供用来解密密钥库的口令。有关此选项参数的语法的详细信息，请参见 pkgadd(1M) 中的口令短语参数。</p> <p>-s 指示软件包应该作为数据流（而不是作为文件系统）写入到 <i>device2</i>。缺省行为是以文件系统格式写入到支持这两种格式的设备中。</p>
操作数	<p><i>device1</i> 指示源设备。该设备上的软件包将被转换并放置到 <i>device2</i> 中。请参见下文的“设备说明符”。</p> <p><i>device2</i> 指示目标设备。转换后的软件包将放置到该设备中。请参见下文的“设备说明符”。</p>

pkginst 指定应转换 *device1* 中的哪个（些）软件包实例。可以使用标记 `all` 来指示所有软件包。可以使用 `pkginst.*` 来指示某个软件包的所有实例。如果没有定义软件包，则会出现一个提示，它显示设备上的所有软件包，并询问要转换哪一个。

星号字符(*) 对于某些 shell 来说是特殊字符，可能需要转义。在 C-Shell 中，* 必须放在单引号(') 中或者以反斜杠(\) 为前缀。

设备说明符

打包工具（包括 `pkgtrans`、`pkgadd(1M)` 和 `pkgchk(1M)`）都具有相应的选项，可以使用这些选项通过指定软件包所在的设备来指定软件包位置。下面列出了可以存储和从中检索软件包的设备类型。请注意，源设备和目标设备不能相同。

device 通过将设备标识符指定为设备，可以将软件包存储为字符或块设备。该设备类型的一个常见示例是可移除磁带的 `/dev/rmt/0`。`pkgtrans` 还可以生成流格式的常规文件系统文件，该格式的文件适合存储在字符设备、Web 服务器上或者作为 `pkgadd(1M)` 的输入。

directory（目录） 通过指定某个文件系统目录的绝对路径可以将软件包存储到某个目录中。软件包内容将驻留在指定目录内的一个目录中。软件包目录名称必须与其在 `pkginfo(4)` 文件中的 PKG 定义相同。此类型的一个设备定义示例为 `/export/packages`。

示例

示例1 转换 /tmp 中的软件包

以下示例将 /tmp 中的软件包 `pkg1` 和 `pkg2` 转换为数据流格式：

```
example% pkgtrans -s /tmp /tmp/datastream.pkg pkg1 pkg2
```

示例2 创建已签名的软件包

以下示例基于 `pkg1` 和 `pkg2` 创建已签名的软件包，并从 `$PASS` 环境变量读取口令：

```
example% pkgtrans -sg -k /tmp/keystore.p12 -a foo \  
-p env:PASS /tmp /tmp/signedpkg pkg1 pkg2
```

示例3 转换软件包数据流

以下示例将软件包数据流转换为文件系统格式软件包：

```
example% pkgtrans /tmp/pkg1.pkg ~/tmp pkg1
```

环境变量

`MAXINST` 变量是在 `pkginfo(4)` 文件中设置的，它并声明软件包实例的最大数量。

退出状态

0 成功完成。

>0 出现错误。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/svr4
接口稳定性	Committed (已确定)

另请参见

[pkginfo\(1\)](#)、[pkgmk\(1\)](#)、[pkgparam\(1\)](#)、[pkgproto\(1\)](#)、[installf\(1M\)](#)、[pkgadd\(1M\)](#)、[pkgask\(1M\)](#)

《[Application Packaging Developer's Guide](#)》

附注

缺省情况下，如果目标设备上已存在某个软件包的任何实例，则 `pkgtrans` 不会转换该软件包的任何实例。如果已经存在该软件包的一个实例，使用 `-n` 选项将创建一个新实例。如果已经存在该软件包的一个实例，使用 `-o` 选项将覆盖该实例。如果目标设备是数据流，这两个选项都没有用。

软件包命令可识别 [largefile\(5\)](#)。它们处理大于 2 GB 的文件的方法与处理较小文件的方法相同。在其当前实现中，[pkgadd\(1M\)](#)、`pkgtrans` 和其他软件包命令最多可以处理 4 GB 的数据流。

引用名	pklogin_finder – 将证书映射到用户
用法概要	<code>/usr/lib/pam_pkcs11/pklogin_finder [debug] [config_file=filename]</code>
描述	<p>pklogin_finder 使用 pam_pkcs11 库基础结构以交互方式将所提供的 PKCS#11 证书映射到用户。</p> <p>pklogin_finder 使用与 pam_pkcs11(5) PAM 模块相同的配置文件和参数。它装入所定义的映射器模块，并尝试查找所发现的证书与用户登录之间的映射。</p>
选项	<p>支持以下选项：</p> <p><code>config_file=filename</code> 设置配置文件。 缺省值是 <code>/etc/security/pam_pkcs11/pam_pkcs11.conf</code>。</p> <p><code>debug</code> 启用调试输出。 缺省值是不调试。</p> <p>因为它使用与 pam_pkcs11(5) 相同的配置文件，所以所有 pam_pkcs11 选项均可用。其中某些选项在非 PAM 环境中没有意义，因此被忽略。某些映射器选项（<code>mapfile</code>、<code>ignorecase</code>）对证书内容不起作用，因此也被忽略。</p>
退出状态	<p>将返回以下退出值：</p> <p>0 成功完成。</p> <p> pkcs11_inspect 在 <code>stdout</code> 上输出登录名，然后退出。</p> <p>1 出现错误。 发现用户映射错误。</p> <p>2 出现错误。 找不到用户匹配。</p>
示例	<p>示例1 使用 pklogin_finder</p> <p>以下示例在不使用任何选项的情况下运行 pklogin_finder 命令：</p> <pre>% pkcs11_inspect</pre> <p>示例2 带选项使用 pklogin_finder</p> <p>以下示例带选项使用 pkcs_finder 命令：</p> <pre>% pklogin_finder debug config_file=\${HOME}/.pam_pkcs11.conf</pre>

文件 /etc/security/pam_pkcs11/pam_pkcs11.conf
作者 Juan Antonio Martinez, jonsito@teleline.es
属性 有关下列属性的说明, 请参见 [attributes\(5\)](#):

属性类型	属性值
可用性	library/security/pam/module/pam-pkcs11
接口稳定性	Uncommitted (未确定)

另请参见 [pkcs11_inspect\(1\)](#)、[attributes\(5\)](#)、[pam_pkcs11\(5\)](#)

PAM-PKCS11 用户手册, http://www.opensc-project.org/pam_pkcs11

引用名 pktool – 管理证书和密钥

用法概要 pktool [-f *option_file*] [-i] *subcommand subcommand_options ...*

描述 使用 `pktool` 命令，用户可以管理多个密钥库中的证书和密钥，包括 PKCS#11 令牌（即，加密框架）、Netscape 安全服务 (Netscape Security Services, NSS) 令牌以及 OpenSSL 的基于标准文件的密钥库。

`pktool` 还支持列出、删除和导入证书撤销列表 (Certificate Revocation List, CRL)。`pktool` 不支持创建 CRL、签署 CRL 或导出 CRL。对 PKCS#11 密钥库的 CRL 支持是基于文件的。

选项 支持以下命令选项：

-f *option_file*

允许用户在文件中设置选项，而不是在命令行上输入选项。

提供此选项是为了方便用户，因为使用 `pktool` 时可能需要在命令行上指定大量子命令及相关选项。

option_file 的格式是每行一个选项或值对。

下面是 *option_file* 的一个示例：

```
list
keystore=nss
dir=/export/foo
objtype=key
```

-i

允许用户以交互方式为 `gencert` 和 `gencsr` 子命令指定 subject-DN。指定 `-i` 后，系统将提示用户输入一些数据以组成 subject-DN。

使用 `-i` 选项的示例如下所示：

```
Country Name (2 letter code) [US]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []:Menlo Park
Organization Name (eg, company):Sun Microsystems Inc.
Organizational Unit Name (eg, section):OPG
Common Name (eg, YOUR name):John Smith
Email Address []: john.smith@sun.com
```

最后得到的 subject-DN 为：

```
"C=US, ST=CA, L=Menlo Park, O=Sun Microsystems Inc., \
OU=OPG, emailAddress=john.smith@sun.com, \
CN=John Smith"
```

子命令

支持以下子命令：

delete

delete 子命令的格式如下：

```
pktool delete [token=token[:manuf[:serial]]]
              [objtype=private|public|both]
              [label=object-label]
```

```
pktool delete keystore=pkcs11
              objtype=cert[:public | private | both]]
              [token=token[:manuf[:serial]]]
              [label=cert-label]
              [serial=hex-serial-number]
              [issuer=issuer-DN]
              [subject=subject-DN]
```

```
pktool delete keystore=nss
              objtype=cert
              [subject=subject-DN]
              [issuer=issuer-DN]
              [serial=hex-serial-number]
              [nickname=cert-nickname]
              [token=token[:manuf[:serial]]]
              [dir=directory-path]
              [prefix=DBprefix]
```

```
pktool delete keystore=nss
              objtype=crl
              [nickname=cert-nickname]
              [subject=subject-DN]
              [token=token[:manuf[:serial]]]
              [dir=directory-path]
              [prefix=DBprefix]
```

```
pktool delete keystore=pkcs11
              objtype=key[:public | private | both]]
              [token=token[:manuf[:serial]]]
              [label=key-label]
```

```
pktool delete keystore=pkcs11
              objtype=crl
              infile=input-fn
```

```
pktool delete keystore=file
              objtype=cert
              [infile=input-fn]
              [dir=directory-path]
```

```

[serial=hex-serial-number]
[issuer=issuer-DN]
[subject=subject-DN]

```

```

pktool delete keystore=file
               objtype=key
               [infile=input-fn]
               [dir=directory-path]

```

```

pktool delete keystore=file
               objtype=crl
               infile=input-fn

```

删除证书、密钥或证书撤销列表 (certificate revocation list, CRL)。

要从 PKCS#11 令牌中删除专用证书或私钥，系统将提示用户输入正确的个人识别号 (Personal Identification Number, PIN) 来向 PKCS#11 表明身份。

download

download 子命令的格式如下所示：

```

pktool download url=url_str
                 [objtype=crl|cert]
                 [http_proxy=proxy_str]
                 [outfile=output-fn]
                 [dir=directory-path]

```

从指定的 URL 位置下载 CRL 文件或证书文件。在成功下载文件后，将检查下载的 CRL 或证书文件的有效性。如果 CRL 或证书已过期，download 将发出警告。

export

export 子命令的格式如下：

```

pktool export [token=token[:manuf[:serial]]]
              outfile=output-fn

pktool export keystore=pkcs11
              outfile=output-fn
              [objtype=cert|key]
              [label=label]
              [subject=subject-DN]
              [issuer=issuer-DN]
              [serial=hex-serial-number]
              [outformat=pem|der|pkcs12|raw]
              [token=token[:manuf[:serial]]]

pktool export keystore=nss
              outfile=output-fn
              [subject=subject-DN]

```



```

[issuer=issuer-DN]
[serial=hex-serial-number]
[nickname=cert-nickname]
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]
[outformat=pem|der|pkcs12]

```

```

pktool export keystore=file
certfile=cert-input-fn
keyfile=key-input-fn
outfile=output-pkcs12-fn

```

将 PKCS#11 令牌的内容、NSS 令牌中的证书的内容或基于文件的密钥库的内容保存到指定文件。

gencert

gencert 子命令的格式如下所示：

```

pktool gencert [-i] keystore=nss
label=cert-nickname
subject=subject-DN
serial=hex_serial_number
[altname=[critical:]subjectAltName,subjectAltName...]
[keyusage=[critical:]usage,usage...]
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]
[keytype=rsa | ec [curve=ECC Curve Name] \
[hash=md5 | sha1 | sha224 | sha256 | sha384 | sha512]]
[keytype=dsa [hash=sha1 | sha224 ]]
[keylen=key-size]
[trust=trust-value]
[eku=[critical:]EKU_name,...]
[listcurves ]
[lifetime=number-hour|number-day|number-year]

pktool gencert [-i] [ keystore=pkcs11]
label=key/cert-label
subject=subject-DN
serial=hex_serial_number
[altname=[critical:]subjectAltName,subjectAltName...]
[keyusage=[critical:]usage,usage...]
[token=token[:manuf[:serial]]]
[ keytype=rsa | ec [curve=ECC Curve Name] \
[hash=md5 | sha1 | sha224 | sha256 | sha384 | sha512]]
[ keytype=dsa [hash=sha1 | sha224 | sha256 ]]
[keylen=key-size]
[eku=[critical:]EKU_name,...]

```

```
[listcurves]
[lifetime=number-hour|number-day|number-year]
```

```
pktool gencert [-i] keystore=file
outcert=cert-fn
outkey=key-fn
subject=subject-DN
serial=hex_serial_number
[altname=[critical:]subjectAltName,subjectAltName...]
[keyusage=[critical:]usage,usage...]
[format=der|pem]
[ keytype=rsa [hash=md5 | sha1 | sha224 | sha256 | sha384 | sha512]]
[ keytype=dsa [hash=sha1 | sha224 | sha256 ]]
[keylen=key-size]
[eku=[critical:]EKU_name,...]
[lifetime=number-hour|number-day|number-year]
```

生成一个自签名证书并将该证书及相关私钥安装到指定的密钥库。

对于基于令牌的密钥库，gencert 会提示用户输入 PIN。

gencsr

gencsr 子命令的格式如下所示：

```
pktool gencsr [-i] keystore=nss
nickname=key-nickname
outcsr=csr-fn
subject=subject-DN
[altname=[critical:]subjectAltName,subjectAltName...]
[keyusage=[critical:]usage,usage...]
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]
[keytype=rsa | ec [curve=ECC Curve Name] \
[hash= md5 | sha1 | sha224 | sha256 | sha384 | sha512]]
[keytype=dsa [hash=sha1 | sha224]]
[keylen=key-size]
[format=pem|der]
[eku=[critical:]EKU_name,...]
[listcurves]

pktool gencsr [-i] keystore=pkcs11
label=key-label
outcsr=csr-fn
subject=subject-DN
[altname=[critical:]subjectAltName,subjectAltName...]
[keyusage=[critical:]usage,usage...]
[token=token[:manuf[:serial]]]
[ keytype=rsa | ec [curve=ECC Curve Name] \
[hash=md5 | sha1 | sha224 | sha256 | sha384 | sha512]]
```

```
[keylen=key-size]
[format=pem|der]
[eku=[critical:]EKU_name,...]
[listcurves]
```

```
pktool gensr [-i] keystore=file
outcsr=csr-fn
outkey=key-fn
subject=subject-DN
[altname=[critical:]subjectAltName,subjectAltName...]
[keyusage=[critical:]usage,usage...]
[dir=directory-path]
[ keytype=rsa [hash=md5 | sha1 | sha224 | sha256 | sha384 | sha512]]
[ keytype=dsa [hash=sha1 | sha224 | sha256 ] ]
[keylen=key-size]
[format=pem|der]
[eku=[critical:]EKU_name,...]
```

创建 PKCS#10 证书签名请求 (certificate signing request, CSR) 文件。可以将该 CSR 发送给颁证机构 (Certifying Authority, CA) 进行签名。对于基于令牌的密钥库，genscr 子命令会提示用户输入 PIN。

genkey

genkey 子命令的格式如下所示：

```
pktool genkey [keystore=pkcs11]
label=key-label
[keytype=aes|arcfour|des|3des|generic]
[keylen=key-size (for aes, arcfour, or \
generic keytypes only)]
[token=token[:manuf[:serial]]]
[sensitive=y|n]
[extractable=y|n]
[print=y|n]
```

```
pktool genkey keystore=nss
label=key-label
[keytype=aes|arcfour|des|3des|generic]
[keylen=key-size (for aes, arcfour, or \
generic keytypes only)]
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]
```

```
pktool genkey keystore=file
outkey=key-fn
[keytype=aes|arcfour|des|3des|generic]
[keylen=key-size (for aes, arcfour, \
or generic keytypes only)]
[print=y|n]
```

在指定密钥库中生成对称密钥。对于基于令牌的密钥库，genkey 子命令会提示用户输入 PIN。

genkeypair

genkeypair 子命令的格式如下所示：

```
pktool genkeypair keystore=nss
                    label=key-nickname
                    [token=token[:manuf[:serial]]]
                    [dir=directory-path]
                    [prefix=DBprefix]
                    [keytype=rsa|dsa|ec [curve=ECC Curve Name]]
                    [keylen=key-size]
                    [listcurves]
```

```
pktool genkeypair [keystore=pkcs11]
                    label=key-label
                    [token=token[:manuf[:serial]]]
                    [keytype=rsa|dsa|ec [curve=ECC Curve Name]]
                    [keylen=key-size]
                    [listcurves]
```

```
pktool genkeypair keystore=file
                    outkey=key_filename
                    [format=der|pem]
                    [keytype=rsa|dsa]
                    [keylen=key-size]
```

import

import 子命令的格式如下：

```
pktool import [token=token>[:manuf>[:serial>]]]
              infile=input-fn
```

```
pktool import [keystore=pkcs11]
              infile=input-fn
              label=object-label
              [keytype=aes|arcfour|des|3des|generic]
              [sensitive=y|n]
              [extractable=y|n]
              [token=token[:manuf[:serial]]]
              [objtype=cert|key]
```

```
pktool import keystore=pkcs11
              objtype=crl
              infile=input-fn
              outcrl=output-crl-fn
              outformat=pem|der
```

```
pktool import keystore=nss
```

```

objtype=cert
infile=input-fn
label=cert-label
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]
[trust=trust-value]

```

```

pktool import keystore=nss
objtype=crl
infile=input-fn
[verifycrl=y|n]
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]

```

```

pktool import keystore=file
infile=input-fn
outkey=output-key-fn
outcert=output-key-fn
[outputformat=pem|der]

```

```

pktool import keystore=file
objtype=crl
infile=input-fn
outcrl=output-crl-fn
[outputformat=pem|der]

```

将证书、密钥或 CRL 从指定输入文件装入指定密钥库。

inittoken

inittoken 子命令的格式如下所示：

```

pktool inittoken [ slotid=slot number ]
                  [ currlabel=token[:manuf[:serial]]]
                  [ newlabel=new token label ]

```

该命令使用 C_InitToken API 初始化 PKCS#11 令牌。定位令牌的首选方法是指定其缺省标签。还可以通过使用 newLabel 参数向令牌分配新标签。如果没有提供 newLabel，则不会修改令牌标签。系统将提示用户输入安全员 (security officer, SO) PIN，然后此命令才能继续执行。

list

list 子命令的格式如下：

```

pktool list [token=token[:manuf[:serial]]]
            [objtype=private|public|both]
            [label=label]

```

```

pktool list [keystore=pkcs11]

```

```
[objtype=cert[:public | private | both]]
[token=token[:manuf[:serial]]]
[label=cert-label]
[serial=hex-serial-number]
[issuer=issuer-DN]
[subject=subject-DN]

pktool list [keystore=pkcs11]
objtype=key[:public | private | both]]
[token=token[:manuf[:serial]]]
[label=key-label]
[keyvalue=y|n]

pktool list keystore=pkcs11
objtype=crl
infile=input-fn

pktool list keystore=nss
objtype=cert
[subject=subject-DN]
[issuer=issuer-DN]
[serial=hex-serial-number]
[nickname=cert-nickname]
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]

pktool list keystore=nss
objtype=key
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]

pktool list keystore=file
objtype=cert
[infile=input-fn]
[dir=directory-path]
[serial=hex-serial-number]
[issuer=issuer-DN]
[subject=subject-DN]

pktool list keystore=file
objtype=key
[infile=input-fn]
[dir=directory-path]
[keyvalue=y|n]
```

列出证书、密钥或证书撤销列表 (certificate revocation list, CRL)。在显示 PKCS#11 令牌中的专用证书或私钥时，系统将提示用户输入正确的 PIN 以便向 PKCS#11 令牌表明身份。

setpin

setpin 子命令的格式如下所示：

```
pktool setpin keystore=nss
    [token=token]
    [dir=directory-path]
    [prefix=DBprefix]

pktool setpin [ keystore=pkcs11]
    [token=token[:manuf[:serial]]]
    [usertype=user | so]
```

更改用于向 PKCS#11 或 NSS 令牌表明用户身份的口令短语。口令短语可以是长度为 1 到 256 之间且不含空值的字符串。

setpin 提示用户输入旧口令短语（如果有）。如果旧口令短语匹配，pktool 将提示用户输入新口令短语两次。如果两次输入的新口令短语匹配，它将成为该令牌的当前口令短语。

对于 Sun Software PKCS#11 softtoken 密钥库（缺省），用户在使用 setpin 命令来更改对象存储的口令短语时，必须使用缺省口令短语 changeme 作为旧口令短语。在初始化新创建的令牌对象存储并为其设置口令短语时，需要执行此操作。

如果为基于 PKCS#11 的令牌指定了 usertype=so 选项，则会更改与普通用户 PIN 相对的安全员 (Security Officer, SO) 用户 PIN。缺省情况下，usertype 假定为 user。

signcsr

signcsr 子命令的格式如下所示：

```
signcsr keystore=pkcs11
    signkey=label (label of key to use for signing)
    csr=CSR_filename
    serial=serial_number_hex_string_for_final_certificate
    outcert=filename_for_final_certificate
    issuer=issuer-DN
    [store=y|n] (store the new cert in NSS DB, default=n)
    [outlabel=certificate label]
    [format=pem|der] (certificate output format)
    [subject=subject-DN] (override the CSR subject name)
    [altname=subjectAltName,subjectAltName...] (add subjectAltName )
    [keyusage=[critical:]usage,...] (add key usage bits)
    [eku=[critical:]EKU_Name,...] (add Extended Key Usage )
    [lifetime=number-hour|number-day|number-year]
    [token=token[:manuf[:serial]]]

signcsr keystore=file
    signkey=filename
```

```

csr=CSR_filename
serial=serial_number_hex_string_for_final_certificate
outcert=filename_for_final_certificate
issuer=issuer-DN
[format=pem|der] (certificate output format)
[subject=subject-DN] (override the CSR subject name)
[altname=subjectAltName,subjectAltName...] (add a subjectAltName)
[keyusage=[critical:]usage,...] (add key usage bits)
[lifetime=number-hour|number-day|number-year]
[eku=[critical:]EKU_Name,...] (add Extended Key Usage)
signcsr keystore=nss
signkey=label (label of key to use for signing)
csr=CSR_filename
serial=serial_number_hex_string_for_final_certificate
outcert=filename_for_final_certificate
issuer=issuer-DN
[store=y|n] (store the new cert in NSS DB, default=n)
[outlabel=certificate label]
[format=pem|der] (certificate output format)
[subject=subject-DN] (override the CSR subject name)
[altname=subjectAltName,subjectAltName...] (add a subjectAltName)
[keyusage=[critical:]usage,...] (add key usage bits)
[eku=[critical:]EKU_Name,...] (add Extended Key Usage)
[lifetime=number-hour|number-day|number-year]
[token=token[:manuf[:serial]]]
[dir=directory-path]
[prefix=DBprefix]

```

tokens

tokens 子命令的格式如下所示：

```
pktool tokens
```

tokens 子命令列出所有可见的 PKCS#11 令牌。

-?

-? 子命令的格式如下所示：

```
pktool -?
pktool --help
```

-? 选项显示用法和帮助信息。--help 是 -? 的同义词。

用法

pktool 的子命令支持以下选项：

```
altname=[critical:]subjectAltName,subjectAltName...
```

证书的拥有者替代名称。altname 选项后的参数应使用 tag=value 格式。有效的标记为 IP、DNS、EMAIL、URI、KRB、UPN 和 RID。如果 altname 字符串带有前缀 critical，则 SubjectAltName 扩展被标记为 critical。

critical 标志为所有 altname 共享，应放在第一个拥有者替代名称的开头。

`altname` 标记（例如 IP、EMAIL、DN 等）之间的分隔符是逗号 (,)。DN 类型的组成部分之间的分隔符是分号 (;)。

示例 1：将 IP 地址添加到 `subjectAltName` 扩展。`altname="IP=1.2.3.4"` 示例 2：将电子邮件地址添加到 `subjectAltName` 扩展，并将其标记为 `critical`。`altname="critical:EMAIL=first.last@company.com"`

示例 3：向 `subjectAltName` 扩展添加多个所有者替代名称 IP 地址、电子邮件地址和标识名，并将其标记为 `critical`。

```
altname="critical:IP=1.2.3.4,EMAIL=first.last@company.com,\
DN=C=US;O=Oracle;OU=Security;CN=John Smith"
```

`currlabel=token label`

此选项仅供 `inittoken` 命令使用。这用于定位正在被初始化的缺省令牌。有关要使用的令牌名称的格式的详细信息，请参见 `token` 选项。

`curve=Elliptic_Curve_Name`

此选项用于指定在生成 X.509 证书或证书签名请求时或生成椭圆曲线密钥对时要使用的椭圆曲线参数。

支持以下指定曲线：

```
secp112r1, secp112r2, secp128r1, secp128r2, secp160k1
secp160r1, secp160r2, secp192k1, secp192r1, secp224k1
secp224r1, secp256k1, secp256r1, secp384r1, secp521r1
sect113r1, sect113r2, sect131r1, sect131r2, sect163k1
sect163r1, sect163r2, sect193r1, sect193r2, sect233k1
sect233r1, sect239k1, sect283k1, sect283r1, sect409k1
sect409r1, sect571k1, sect571r1, c2pnb163v1, c2pnb163v2
c2pnb163v3, c2pnb176v1, c2tnb191v1, c2tnb191v2, c2tnb191v3
c2pnb208w1, c2tnb239v1, c2tnb239v2, c2tnb239v3, c2pnb272w1
c2pnb304w1, c2tnb359v1, c2pnb368w1, c2tnb431r1, prime192v2
prime192v3
```

通过将 `listcurves` 选项与 `gencert`、`gencsr` 或 `genkeypair` 子命令一起使用，还可以查看指定曲线的列表。

`dir=directory_path`

指定存储所请求的对象的 NSS 数据库目录或 OpenSSL 密钥库目录。

`eku=[critical:]EKU_Name,[critical:]EKU_Name, ...]`

指定要添加到证书或证书请求的扩展密钥用法 X.509v3 扩展值。

将 `EKU_Name` 指定为以下任一

值：`serverAuth`、`clientAuth`、`codeSigning`、`emailProtection`、`ipsecEndSystem`、`ipsecTransport`、`timeStamping`、`OCSPSigning`、`KPClientAuth`、`KPKdc` 或 `scLogon`。

举例如下：

```
eku=KPClientAuth,clientAuth
```

`extractable=y | n`

指定 PKCS#11 令牌中的结果对称密钥是否为可提取的。有效值是 `y` 和 `n`。缺省值是 `y`。

`format=pem | der | pkcs12`

对于 `gencert` 子命令，此选项仅适用于基于文件的密钥库，如 OpenSSL。它用于指定要创建的密钥或证书文件的输出格式。有效格式是 `pem` 或 `der`。缺省格式为 `pem`。

对于 `genscr` 子命令，此选项指定 CSR 文件的输出编码格式。有效格式是 `pem` 或 `der`。缺省格式为 `pem`。

`hash=md5 | sha1 | sha224 | sha256 | sha384 | sha512`

对于 `gencert` 和 `genscr` 子命令，此选项允许调用者指定用于生成 X.509 证书签名的散列算法。使用 NSS 或 PKCS#11 密钥库创建基于 EC 或 RSA 的证书时可以使用此选项。使用基于 OpenSSL 文件的密钥库时，椭圆曲线支持不可用。

`infile=input-fn`

当 `objtype=cert` 且 `keystore=file` 时，为 `list` 和 `delete` 子命令指定证书文件名称。对于 `import` 子命令，此选项指定要导入的文件名。当 `objtype=crl` 时，为 `list`、`delete` 和 `import` 子命令指定输入 CRL 文件名。

`issuer=issuer-DN`

指定证书的颁发者。

`keylen=key-size`

指定要生成的私钥或对称密钥的大小（位数）。

对于 `gencert` 和 `genscr` 子命令，缺省密钥长度是 1024 位。

对于 `genkey` 子命令，使用 AES 算法生成的对称密钥的最小位数和最大位数分别是 128 和 256。使用 ARCFOUR 算法时，最小位数和最大位数分别是 8 和 2048。普通密钥的最小位数是 8 位，最大位数任意。AES、ARCFOUR 或普通密钥的缺省密钥长度是 128。对于 DES 密钥或 3DES 密钥，密钥长度是固定的；如果指定了此选项，将忽略此选项。

`keystore=nss | pkcs11 | file`

指定基础密钥库的类型：NSS 令牌、PKCS#11 令牌或基于文件的插件。

`keytype=rsa | dsa | ec | aes | arcfour | des | 3des | generic`

指定要生成的私钥或对称密钥的类型。

对于 `gencert` 和 `genscr` 子命令，有效私钥类型是 `rsa`、`ec` 或 `dsa`。缺省密钥类型是 `rsa`。

对于 `genkey` 子命令，有效的对称密钥类型是 `aes`、`arcfour`、`des`、`3des` 或 `generic`。缺省密钥类型是 `aes`。

`keyusage=[critical:]usage,usage,usage,...`

```

Key Usage strings:
* digitalSignature
* nonRepudiation
* keyEncipherment
* dataEncipherment
* keyAgreement
* keyCertSign
* cRLSign
* encipherOnly
* decipherOnly

```

示例 1: 设置 `KeyUsage`, 以便 `cert` (或 `csr`) 可以用于对非证书或 CRL (`digitalSignature`) 数据进行签名和验证, 还可以用于对非加密密钥 (`dataEncipherment`) 数据进行加密和解密。 `keyusage=digitalSignature,dataEncipherment`

示例 2: 与上面的示例 1 相同, 但设置了 `critical` 位。 `keyusage=critical:digitalSignature,dataEncipherment`

`keyvalue=y|n`

此选项显示基于文件的密钥库和 PKCS#11 密钥库中对称密钥的十六进制格式密钥值。有效值是 `y` 和 `n`。缺省值为 `n`。

`label=key-label | cert-label`

对于 `gencert` 子命令, 此选项指定 PKCS#11 令牌中的私钥和自签名证书的标签。

对于 `genscr` 子命令, 此选项指定 PKCS#11 令牌中的私钥的标签。

对于 `list` 子命令, 此选项指定 PKCS#11 令牌中的 X.509 证书的标签 (`objtype=key` 时) 或私钥的标签 (`objtype=cert` 时) 以进一步限制列表。

对于 `delete` 子命令, 此选项指定 X.509 证书的标签 (`objtype=key` 时) 或私钥的标签 (`objtype=cert` 时) 以从 PKCS#11 令牌中删除指定对象。

`listcurves`

此选项用于显示受支持的椭圆曲线名称的列表。仅 `gencert`、`genscr` 或 `genkeypair` 子命令可以使用此选项。

`lifetime=number-hour|number-day|number-year`

指定证书的有效期。可以通过 `number-hour`、`number-day` 或 `number-year` 指定证书使用期限。只能指定一种格式。缺省值是 `1-year`。此选项的示例如下: `lifetime=1-hour`、`lifetime=2-day`、`lifetime=3-year`

`newlabel=token label`

此选项仅供 `inittoken` 命令使用。这用于更改分配给正在被初始化的令牌的标签。有关要使用的令牌名称的格式的信息, 请参见 `token` 选项。

`nickname=cert-nickname`

对于 `gencert` 子命令, 此选项是必需的, 用以为 NSS 密钥库指定证书的呢称。

对于 `list` 子命令，此选项指定 NSS 令牌中的证书的呢称以显示其内容。对于 `delete` 子命令，要从 NSS 令牌中删除 CRL，可使用此选项指定颁发者的证书的呢称。对于 `delete` 子命令，要从 NSS 令牌中删除证书，可使用此选项来指定证书的呢称。对于 `import` 子命令，要将指定的输入文件导入 NSS 令牌，可使用此选项来指定结果证书的呢称。

`objtype=cert | key | crl`

指定对象类：`cert`、`key` 或 `crl`。对于 `download` 子命令，如果未指定此选项，则缺省类是 `crl`。

`objtype=public | private | both`

指定对象的类型：专用对象、公用对象或两者。当指定了 `objtype=key` 时，此选项仅适用于针对 PKCS#11 令牌的 `list` 和 `delete` 子命令。缺省值是 `public`。

对于 `list` 子命令，可以将 `label` 选项与此选项组合使用以进一步限制密钥列表。对于 `delete` 子命令，可使用此选项将要删除的密钥缩小到仅公钥或仅私钥。此外，还可以省略 `label` 选项以指示删除所有公钥、所有私钥或所有这两种类型的密钥。可供 `objtype` 参数选用的 `public`、`private` 和 `both` 仅适用于 PKCS#11 密钥库，保留它们是为了保持与 `pktool` 命令的早期版本的兼容性。

`outcert=cert-fn`

指定要写入到的输出证书文件名。对于基于文件的插件（如 OpenSSL），此选项是必需的。此选项必须与 `outkey=key-fn` 选项一起使用。

`outcrl=output-crl-fn`

指定要写入到的输出 CRL 文件名。

`outcsr=csr-fn`

指定要写入到的输出 CSR 文件名。

`outfile=output-fn`

对于 `export` 子命令，此选项指定要创建的输出文件名。对于 `import` 子命令，此选项指定证书或 CRL 的输出文件名。此选项仅适用于基于文件的插件，如 OpenSSL。对于 `download` 子命令，如果未指定此选项，则下载的文件名是 URL 字符串的基名。

`outformat=pem | der | pkcs12`

对于 `import` 子命令，此选项指定从指定的 PKCS#12 文件提取到基于文件的插件中的证书或密钥的输出格式。有效值是 `pem` 或 `der`。缺省值是 `pem`。将 CRL 导入基于 CRL 文件的密钥库时，此选项指定 CRL 的输出格式。有效值是 `pem` 或 `der`。缺省值是 `der`。对于 `export` 子命令，此选项指定要创建的指定输出文件的格式。支持的格式为 `pem`、`der` 或 `pkcs12`。缺省值是 `pkcs12`。

`outkey=key-fn`

指定要写入到的输出私钥文件名。只有使用 `files` 密钥库时需要此选项。

`prefix=DBprefix`

指定 NSS 数据库前缀。此选项仅适用于 NSS 令牌。

print=y | n

此选项用于 `genkey` 子命令中，且适用于 PKCS11 密钥库和基于文件的密钥库。如果 `print=y`，`genkey` 子命令将在十六进制的单个行中输出生成的密钥的密钥值。缺省值为 `n`。对于 PKCS11 密钥库，如果创建对称密钥时使用了 `sensitive=y` 或 `extractable=n` 设置，则不会显示密钥值，即使 `print` 选项设置为 `y` 也是如此。仍然会创建密钥，但发出如下警告：`cannot reveal the key value`。

sensitive=y | n

指定 PKCS#11 令牌中的结果对称密钥是否为敏感的。有效值是 `y` 和 `n`。缺省值为 `n`。

serial=hex-serial-number

指定证书的唯一序列号。序列号必须指定为十六进制值。示例：`0x0102030405060708090a0b0c0d0e0f`

subject=subject-DN

为证书或证书请求指定具体的证书所有者。`subject=` 设置示例如下：

```
subject=O=Sun Microsystems Inc., \
OU=Solaris Security Technologies Group, \
L=Ashburn, ST=VA, C=US, CN=John Smith
```

token=token[:manuf[:serial]]

当令牌标签包含结尾空格时，为方便起见，此选项不要求用户键入这些空格。

冒号分隔的令牌标识字符串 `token:manuf:serial`。如果任何部分中有：文本字符，则需要使用反斜杠 (`\`) 进行转义。如果没有发现 `:`，则整个字符串（最多 32 个字符）将用作令牌标签。如果只发现一个 `:`，则字符串将用作令牌标签和生产商。当指定了 `keystore=nss` 时，如果未指定此选项，则缺省值为 NSS 内部令牌。当指定了 `keystore=pkcs11` 时，如果未指定此选项，则缺省值为 `pkcs11_softtoken`。

trust=trust-value

指定证书信任属性。此选项仅适用于 NSS 证书，将应用标准的 NSS 语法。

usertype=user | so

指定为其执行 `setpin` 命令的用户的类型。缺省情况下，是为标准用户执行，但可以指定 `so` 来为令牌安全员设置 PIN。

url=url_string

指定用来下载 CRL 或证书文件的 URL。

verifycrl=y | n

将 CRL 导入 NSS 密钥库时，此选项指定是否执行 CRL 验证。有效值是 `y` 和 `n`。缺省值为 `n`。

http_proxy=proxy_str

指定代理服务器主机名和端口号。格式可以是 `http://hostname[:port]` 或 `hostname[:port]`。如果没有指定此选项，则 `download` 子命令将检查 `http_proxy` 环境变量。命令行选项的优先级高于环境变量。

示例

示例1 生成自签名证书

以下示例将创建证书并将其存储在命令中指示的密钥库中：

```
$ pktool gencert keystore=nss nickname=WebServerCert \
  subject="O=Sun Microsystems Inc., OU=Solaris Security Technologies Group, \
  L=Ashburn, ST=VA, C=US, CN=John Smith" dir=/etc/certs \
  keytype=rsa keylen=2048 hash=sha512
```

示例2 生成证书签名请求

以下示例将创建 CSR 并将其存储在命令中指示的密钥库中：

```
$ pktool genscr keystore=nss subject="O=Sun Microsystems Inc., \
  OU=Solaris Security Technologies Group, L=Ashburn, ST=VA, C=US, \
  CN=John Smith" keytype=rsa keylen=2048 hash=sha256 outcsr=csr.dat
```

示例3 导入证书

以下示例将一个证书对象从指定输入文件导入到命令中指示的密钥库中：

```
$ pktool import keystore=nss objtype=cert infile=mycert.pem \
  nickname=mycert
```

退出状态

将返回以下退出值：

```
0
  成功完成。
>0
  出现错误。
```

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed (已确定)

另请参见

[attributes\(5\)](#)、[pkcs11_softtoken\(5\)](#)

RSA PKCS#11 v2.11 <http://www.rsasecurity.com>

RSA PKCS#12 v1.0 <http://www.rsasecurity.com>

SECG 建议的椭圆曲线域参数 <http://www.secg.org>

引用名	plabel - 获取进程的标签
用法概要	/usr/bin/plabel [-sS] [pid...]
描述	plabel 是一个 proc 工具命令，用于获取进程的标签。如果未指定 <i>pid</i> ，则显示的标签是 plabel 命令的标签。未指定这些选项时，标签的输出格式会以缺省格式显示。
选项	-s 以短格式显示与 <i>pid</i> 关联的标签。 -S 以长格式显示与 <i>pid</i> 关联的标签。
退出状态	plabel 退出时返回下列值之一： <ul style="list-style-type: none"> 0 成功完成。 1 由于用法错误而无法成功完成。 2 无法转换标签。 3 无法分配内存。
属性	有关下列属性的说明，请参见 attributes(5) ：

属性类型	属性值
可用性	system/trusted
接口稳定性	请参见下文。

plabel 实用程序是 "Committed"（已确定）。输出为 "Not-an-Interface"（不是接口）。

另请参见 [proc\(1\)](#)、[getplabel\(3TSOL\)](#)、[attributes\(5\)](#)

附注 仅当系统配置有 Trusted Extensions 时，本手册页中介绍的功能才可用。

引用名 plgrp – 观察和影响线程的主 lgroup 和 lgroup 关联

用法概要

```
plgrp [-F] [-h] pid | core [/lwps] ...
plgrp [-F] -i pid[/lwps] ...
plgrp [-F] -a lgroup_list pid[/lwps] ...
plgrp [-F] [-I default | none | future]
    -H lgroup_list pid[/lwps] ...
plgrp [-F] [-I default | none | future] -H lgroup_list -e command [arguments]
plgrp [-F] [-I default | none | future]
    -A lgroup_list/none | weak | strong [,...] pid[/lwps] ...
plgrp [-F] [-I default | none | future]
    -A lgroup_list/none | weak |strong [,...] -e command [arguments]
plgrp [-F] -I default | none | future pid[/lwps]
plgrp [-F] -I default | none | future -e command [arguments] ...
```

描述 plgrp 为一个或多个进程、线程或 LWP 显示或设置主 lgroup 和 lgroup 关联。

lgroup 表示彼此之间最多相隔某个间隔（等待时间）的 CPU 和内存之类的硬件设备的集合。系统中的每个 lgroup 由一个唯一的 lgroup ID 来标识。lgroups 组织到分层结构中，以便于查找最近的资源。有关 lgroups 和 lgroup 分层结构的更多信息，请参见 lgrpinfo(1)。

缺省情况下，在创建每个线程时都会为其分配一个主 lgroup。当系统需要为线程分配 CPU 或内存资源时，它将从线程的主 lgroup 中搜索 lgroup 分层结构以查找离线程的主 lgroup 最近的可用资源。

通常，线程的主 lgroup 是与线程的关联性最强的 lgroup。最初，系统为每个线程选择一个主 lgroup，但将该 lgroup 的线程关联保留设置为 none。如果一个线程为其处理器集合中的某个 lgroup（非其主 lgroup）设置了更强的关联，则只要该线程没有绑定到某个 CPU，它将重新认该 lgroup 为主。如果线程的主 lgroup 的关联（如果有）被删除（设置为 none），则线程会重新认其处理器集中具有次高关联的 lgroup 为主。

[lgrp_affinity_set\(3LGRP\)](#) 中详细介绍了不同级别的 lgroup 关联及其语义。

用法

指定 lgroups lgroup_list 是包含以下一项或多项内容的逗号分隔列表：

- lgroup_ID
- Range of lgroup_IDs specified as
 <start lgroup_ID>-<end lgroup_ID>
- all
- root
- leaves

`all` 关键字表示系统中的所有 `lgroup` ID。`root` 关键字表示根 `lgroup` 的 ID。`leaves` 关键字表示所有 `leaflgroups`（即没有子项的 `lgroups`）的 ID。

指定进程和线程

`plgrp` 接受一个或多个以空格分隔的进程或线程作为参数。可以用与 `proc(1)` 工具类似的方式指定进程和线程。进程 ID 可以指定为整数 `pid` 或 `/proc/pid`。使用 `/proc/pid` 时，可以使用 Shell 扩展来指定进程。例如，可以使用 `/proc/*` 指定系统中的所有进程。如果只是单独给定了进程 ID，则会将该进程的所有线程提供为 `plgrp` 的参数。

可以显式指定线程，将其进程 ID 和线程 ID 分别指定为 `pid/lwpid`。通过使用连字符 (-) 和逗号 (,)，一次可以选择一个进程的多个线程。例如，`pid/1,2,7-9` 以 `pid` 作为其进程 ID，并指定进程的线程 1、2、7、8 和 9。

选项

支持以下选项：

- `-a lgroup_list` 为指定 `lgroup_list` 显示指定进程或线程的 `lgroup` 关联。
- `-A lgroup_list/none|weak|strong[, ...]` 为指定 `lgroup_list` 设置指定进程或线程的关联。
可以给定以逗号分隔的 `lgroups/affinity` 分配列表来一次设置多个关联。
- `-F` 通过抓取目标进程来强制执行，即使另一进程已掌握了控制权。使用 `-F` 标志时应谨慎。在一个被调试的进程上施加两个控制进程可能会导致混乱。仅当主控制进程（通常是调试器）已停止了被调试的进程，并且在应用该 `proc` 工具的过程中主控制进程未在执行任何操作，才能保证安全。有关更多详细信息，请参见“警告”部分。
- `-e` 创建一个新进程，将 `plgrp` 应用到该进程，然后执行指定的命令和参数。
- `-h` 获取指定进程和/或线程的主 `lgroup`。如果没有指定任何选项，则此选项是缺省值。
- `-H lgroup_list` 设置指定进程和线程的主 `lgroup`。
这将为目标 `lgroup` 设置一个很强的关联以便让线程重新认主。如果指定了多个 `lgroup`，则 `plgrp` 将尝试以循环方式让线程认 `lgroup` 为主。
- `-i` 显示指定进程或线程的 `lgroup` 关联继承。
- `-I default | none | future` 设置指定进程或线程的 `lgroup` 关联继承。

操作数

支持下列操作数：

lwps 指定线程。请参见“用法”部分。

pid 指定进程 ID。请参见“用法”部分。

示例

示例 1 获取 Shell 的主 lgroup

以下示例获取 Shell 的主 lgroup：

```
% plgrp $$
PID/LWPID  HOME
3401/1      1
```

示例 2 将多个线程的主 lgroup 设置为根 lgroup

以下示例将多个线程的主 lgroup 设置为根 lgroup：

```
% plgrp -H root 'pgrep firefox'
PID/LWPID  HOME
918/1      1 => 0
934/1      2 => 0
934/2      1 => 0
934/3      2 => 0
934/625    1 => 0
934/626    2 => 0
934/624    2 => 0
934/623    2 => 0
934/630    1 => 0
```

示例 3 执行 plgrp，根 lgroup 为多个线程的主 lgroup

以下示例执行 firefox，根为多个线程的主 lgroup：

```
% plgrp -H root -e /usr/bin/firefox
```

示例 4 获取 lgroups 0-2 的两个线程的关联

以下示例获取 lgroups 1-2 的两个线程的关联：

```
% plgrp -a 0-2 101398/1 101337/1
PID/LWPID  HOME  AFFINITY
101398/1    1     0-2/none
101337/1    1     0-2/none
```

示例 5 设置 lgroup 关联

以下示例设置 lgroup 关联：

```
% plgrp -A 0/weak,1/none,2/strong 101398
PID/LWPID  HOME      AFFINITY
101398/1    1 => 2     0,2/none => 2/strong,0/weak
```

退出状态 将返回以下退出值：

- 0 成功完成。
- 1 语法错. 没有更改任何内容。
- 2 发生非致命错误或中断. 更改了一些内容。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

命令语法和输出格式是 "Ucommitted"（未确定）。

另请参见 [lgrpinfo\(1\)](#)、[madv.so.1\(1\)](#)、[pmadvise\(1\)](#)、[pmap\(1\)](#)、[proc\(1\)](#)、[ps\(1\)](#)、[prstat\(1M\)](#)、[lgrp_affi...](#)

警告 与 [proc\(1\)](#) 工具一样，`plgrp` 实用程序在检查其目标进程时会停止这些进程，如果调用该实用程序时使用了任何选项，则它会报告结果。

在某些情况下，进程可能会发生死锁。进程被停止后将无法执行任何操作。在生产环境中停止某个频繁使用的进程（即使仅停止很短时间）可能会导致严重的瓶颈，甚至导致这些进程挂起，使得用户无法使用这些进程。因此，应该避免在生产环境中停止 UNIX 进程。请参见 [proc\(1\)](#)。

可以通过以下方法来识别由该工具停止的进程：发出以下命令

```
/usr/bin/ps -efLL
```

并在输出的第一列中查找 T。缺省情况下，某些进程（例如，`sched`）大多数情况下都可以显示 T 状态。

引用名 plimit – 获取或设置正在运行的进程的资源限制

用法概要 plimit [-km] pid...

plimit {-cdfnstv} *soft,hard*... pid...

描述 如果指定了一个或多个 *cdfnstv* 选项，*plimit* 会在由进程 ID 列表 *pid* 标识的进程中设置指定资源的软（当前）限制和/或硬（最大）限制。否则，*plimit* 会报告由进程 ID 列表 *pid* 标识的进程的资源限制。

只允许进程所有者或超级用户获取或设置进程的资源限制。只有超级用户可以增大硬限制。

选项 支持以下选项：

-k 在输出中，以千字节（1024 个字节）而非 512 个字节的块显示文件大小。

-m 在输出中，以兆字节（1024*1024 个字节）显示文件和内存大小。

其余选项用于更改指定的资源限制。它们接受以下形式的参数：

soft、*hard*

soft 指定软（当前）限制，*hard* 指定硬（最大）限制。如果未指定硬限制，则逗号可以省略。如果软限制是空字符串，则仅设置硬限制。每个限制是一个**无限制**的文字字符串，或者是一个数字，具有一个可选的比例因子，如下所示：

nk *n* 千字节

nm *n* 兆字节（CPU 时间的分钟数）

nh *n* 小时（仅适用于 CPU 时间）

mm:ss 分和秒（仅适用于 CPU 时间）

软限制不能超过硬限制。

-c *soft,hard* 设置核心文件的大小限制（缺省单位是 512 字节的块）。

-d *soft,hard* 设置数据段（堆）的大小限制（缺省单位是千字节）。

-f *soft,hard* 设置文件的大小限制（缺省单位是 512 字节的块）。

-n *soft,hard* 设置文件描述符的限制（没有缺省单位）。

-s *soft,hard* 设置栈段的大小限制（缺省单位是千字节）。

-t *soft,hard* 设置 CPU 时间限制（缺省单位是秒）。

-v *soft,hard* 设置虚拟内存的大小限制（缺省单位是千字节）。

- 操作数** 支持以下操作数。
 pid 进程 ID 列表。
- 退出状态** plimit 在成功时返回退出值零，失败时（例如，没有这样的进程、权限遭拒或选项无效）返回非零的退出值。
- 文件** /proc/pid/* 进程信息和控制文件
- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [ulimit\(1\)](#)、[proc\(1\)](#)、[getrlimit\(2\)](#)、[setrlimit\(2\)](#)、[proc\(4\)](#)、[attributes\(5\)](#)

引用名	pmapadvise – 将关于内存的建议应用于某个进程
用法概要	pmapadvise -o <i>option</i> [, <i>option</i>] [-F] [-l] [-v] <i>pid</i> ...
描述	<p>pmapadvise 使用 madvise(3C) 在指定的进程中应用有关如何使用内存的建议。</p> <p>pmapadvise 允许用户在一个特定时刻向一个子范围应用建议。pmapadvise 与 madv.so.1(1) 的区别在于，madv.so.1(1) 在目标程序的整个执行过程中将建议应用于指定类型的所有段。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -F 通过抓取目标进程来强制执行，即使另一进程已掌握了控制权。 您应当慎用 -F 选项。请参见 proc(1)。 -l 显示未解析的动态链接程序映射名称。 -o 以下面的格式指定要应用的建议： <ul style="list-style-type: none"> private=advice shared=advice heap=advice stack=advice address[:length]=advice <p>其中，advice 可以是下列值之一：</p> <ul style="list-style-type: none"> normal random sequential willneed dontneed free access_lwp access_many access_many_pset access_default <p>可以提供 address 和 length 来指定应用建议的子范围。缺省情况下，address 应该为十六进制，并且 length 应以字节计。</p> <p>如果未指定 length，且起始地址引用了某个段的开头，则会将建议应用于该段。length 可使用 K、M、G、T、P 或 E 加以限定，以分别指定千字节、兆字节、千兆字节、兆兆字节或艾字节作为度量单位。</p> -v 显示详细输出。像 pmap(1) 一样显示输出，显示哪个建议应用于何处。当建议应用于某个指定区域（例如专用的、共享的，等等）时，要精确获取关于建议应用于何处的反馈时，这可能比较有用。

`pmap` 尝试处理所有合法选项。如果指定了非法的地址范围，则会输出一条错误消息，且跳过违规选项。当存在语法错误时，`pmap` 将不处理任何选项而退出，并且会输出用法消息。

如果在某个区域上给定了冲突的建议，则优先顺序是从针对性最强的建议到针对性最弱的（即最通用的）建议。换句话说，为特定地址范围指定的建议优先于为堆和栈指定的建议，其次，为堆和栈指定的建议优先于为专用和共享内存指定的建议。

此外，下列每个组中的建议与同一组中的其他建议是互斥的：

```
MADV_NORMAL, MADV_RANDOM, MADV_SEQUENTIAL
MADV_WILLNEED, MADV_DONTNEED, MADV_FREE
MADV_ACCESS_DEFAULT, MADV_ACCESS_LWP, MADV_ACCESS_MANY
```

操作数

支持下列操作数：

pid 进程 ID。

示例

示例1 将建议应用于位于指定地址的段

以下示例将建议应用于位于指定地址的段：

```
% pmap $$
100666: tcsh
00010000    312K r-x-- /usr/bin/tcsh
0006C000     48K rwx-- /usr/bin/tcsh
00078000    536K rwx-- [ heap ]
FF100000    856K r-x-- /lib/libc.so.1
FF1E6000     32K rwx-- /lib/libc.so.1
FF1EE000      8K rwx-- /lib/libc.so.1
FF230000    168K r-x-- /lib/libcurses.so.1
FF26A000     32K rwx-- /lib/libcurses.so.1
FF272000      8K rwx-- /lib/libcurses.so.1
FF280000    576K r-x-- /lib/libnsl.so.1
FF310000     40K rwx-- /lib/libnsl.so.1
FF31A000     24K rwx-- /lib/libnsl.so.1
FF364000      8K rwx-  [ anon ]
FF370000     48K r-x-- /lib/libsocket.so.1
FF38C000      8K rwx-- /lib/libsocket.so.1
FF3B0000    176K r-x-- /lib/ld.so.1
FF3EC000      8K rwx-- /lib/ld.so.1
FF3EE000      8K rwx-- /lib/ld.so.1
FFBE6000    104K rw--- [ stack ]
%
% pmap -o 78000=access_lwp $$
%
```

示例 2 使用 -v 选项

以下示例显示了来自 pmapadvise 的详细输出：

```
% pmapadvise -o heap=access_lwp,stack=access_default -v $$
1720:  -sh
00010000    88K r-x-- /usr/sbin/sh
00036000     8K rwx-- /usr/sbin/sh
00038000    16K rwx-- [ heap ]           <= access_lwp
FF250000    24K r-x-- /lib/libgen.so.1
FF266000     8K rwx-- /lib/libgen.so.1
FF272000     8K rwx-  [ anon ]
FF280000   840K r-x-- /lib/libc.so.1
FF362000    32K rwx-- /lib/libc.so.1
FF36A000    16K rwx-- /lib/libc.so.1
FF390000    64K rwx-- [ anon ]
FF3B0000   168K r-x-- /lib/ld.so.1
FF3EA000     8K rwx-- /lib/ld.so.1
FF3EC000     8K rwx-- /lib/ld.so.1
FFBFE000     8K rw--- [ stack ]           <= access_default
```

退出状态

将返回以下退出值：

0 成功完成。

非零值 出现错误。

文件

/proc/* 进程文件

/usr/prob/lib/* proc 工具支持文件

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

命令语法是 "Committed"（已确定）。输出格式是 "Uncommitted"（未确定）。

另请参见

[madv.so.1\(1\)](#)、[pmap\(1\)](#)、[proc\(1\)](#)、[madvise\(3C\)](#)、[attributes\(5\)](#)

引用名	pmap - 显示进程的地址空间的相关信息
用法概要	<pre> /usr/bin/pmap [-rslF] [-A address_range] [pid core]... /usr/bin/pmap -L [-rslF] [-A address_range] [pid] ... /usr/bin/pmap -x [-aslF] [-A address_range] [pid core]... /usr/bin/pmap -S [-alF] [-A address_range] [pid core]... </pre>
描述	pmap 实用程序显示进程的地址空间的相关信息。
选项	<p>支持以下选项：</p> <p>-a 输出用于共享映射的匿名和交换预留空间。</p> <p>-A address_range 指定要显示的地址空间子范围。<i>address_range</i> 是以下列格式之一指定的：</p> <p><i>start_addr</i> 单个地址，将输出限制为包含该地址的段（或页面，如果提供了 -L 选项）。如果指定的地址对应于某个段的起始地址，则输出将始终包括整个段，即使指定了 -L 选项。</p> <p><i>start_addr,</i> 后跟逗号但没有结束地址的地址，将输出限制为从包含指定地址的段开始的所有段（或页面，如果提供了 -L 选项）。</p> <p><i>start_addr,end_addr</i> 由起始地址和结束地址指定的地址范围，将输出限制为从包含起始地址的段或页面到包含结束地址的段或页面之间的所有段（或页面，如果提供了 -L 选项）。</p> <p><i>,end_addr</i> 前接逗号但没有起始地址的地址范围，将输出限制为从第一个段（或页面，如果提供了 -L 选项）到包含指定地址的段（或页面，如果提供了 -L 选项）之间的所有段或页面。</p> <p>-F 强制。抓取目标进程，即使另一进程已掌握了控制权。 请参见“用法”部分。</p> <p>-l 显示未解析的动态链接程序映射名称。</p> <p>-L 输出其中包含为虚拟内存提供后备支持的物理内存的 lgroup。</p> <p>-r 输出进程的保留地址。</p> <p>-s 输出 HAT 页面大小信息。</p> <p>-S 显示每个映射的交换预留空间信息。有关更多信息，请参见“用法”部分。</p> <p>-x 显示每个映射的其他信息。有关更多信息，请参见“用法”部分。</p>

用法 `pmap` 实用程序显示进程的地址空间的相关信息。

进程映射

```
/usr/bin/pmap [ -rslF ] [-A address_range] [ pid | core ] ...
```

缺省情况下，`pmap` 按映射被映射到进程的虚拟地址顺序显示所有映射。将显示映射大小、标志和被映射对象的名称。

可以使用 `-A` 选项将输出限制到指定的地址范围。指定的地址向上或向下舍入到段边界，输出包括由这些地址界定的段。

进程 Lgroup 映射

```
/usr/bin/pmap -L [ -rslF ] [-A address_range] pid ...
```

可使用 `-L` 选项来确定包含为指定虚拟内存提供后备支持的物理内存的 `lgroup`。与 `-A` 选项一起使用时，指定的地址向上或向下舍入到页面边界，输出被限制到由这些地址界定的页面。

这可以与 `plgrp(1)` 结合使用来查明所关注的线程的主 `lgroup` 是否与内存所在位置一样，以及该位置是否应当为该线程的内存地址。`lgrpinfo(1)` 命令与该 `pmap` 选项结合使用时也很有用。它显示 `lgroup` 分层结构、内容和特征，这提供了内存分布于其中的各个 `lgroup` 的详细信息、这些 `lgroup` 相互之间的关系以及它们与所关注的任何其他 `lgroup` 的关系。

此外，可以通过使用 `plgrp(1)`、`pmadvise(1)` 或 `madv.so.1(1)` 更改线程和内存的布置。

线程匿名/被锁定映射的详细信息

```
/usr/bin/pmap -x [ -aslF ] [-A address_range] [ pid | core ] ...
```

`-x` 选项显示每个映射的其他信息。使用此选项可显示每个映射的大小、驻留物理内存 (resident physical memory, RSS) 量、匿名内存量以及锁定的内存量。此选项不包括内核地址空间因该进程而占用的匿名内存。

交换预留空间

```
/usr/bin/pmap -S [ -alF ] [-A address_range] [ pid | core ] ...
```

`-S` 选项显示每个映射的交换预留空间信息。

使用 `-F` 标志时应谨慎。在一个被调试的进程上施加两个控制进程可能会导致混乱。仅当主控制进程（通常是调试器）已停止了被调试的进程，并且在应用 `proc` 工具的可疑时刻主控制进程未在执行任何操作，才能保证安全。

显示格式

除非指定了 `--s` 或 `--L` 选项，否则将为进程内的每个映射显示一行输出。使用 `-s` 选项时，将为每个硬件转换页面大小的连续映射显示一行。使用 `-L` 选项时，将为属于同一 `lgroup` 的连续映射显示一行。同时使用 `-L` 和 `-s` 选项时，将为属于同一 `lgroup` 的每个硬件转换页面大小的连续映射显示一行。下面的括号中显示的是列标题。

虚拟地址 (Address) 输出的第一列表示每个映射的起始虚拟地址。虚拟地址是以升序显示的。

虚拟映射大小 (Kbytes) 每个映射的虚拟大小（以千字节为单位）。

驻留物理内存 (RSS) 为每个映射驻留的物理内存量（以千字节为单位），包括与其他地址空间共享的物理内存。

匿名内存 (Anon) 使用系统页面大小计数的与指定映射相关联的匿名内存的页面数。不包括与其他地址空间共享的匿名内存，除非指定了 `-a` 选项。

为包含通过 `MAP_PRIVATE`（请参见 [mmap\(2\)](#)）映射的映射的“写入时复制”页面的进程堆、栈报告匿名内存。

锁定 (Locked) 映射内被锁定的页面数。典型示例有通过 `mlock()` 锁定的内存和通过 `SHM_SHARE_MMU` 创建的 System V 共享内存。

权限/标志 (Mode) 显示每个映射的虚拟内存权限。有效权限有：

`r:` 进程可以读取映射。

`w:` 进程可以写入映射。

`x::` 进程可以执行驻留在映射内的指令。

可以显示为每个映射显示其他信息的标志：

`s:` 映射是共享的，因此在所观察的地址空间中所做的更改将提交到映射的文件，且可以从共享该映射的所有其他进程中看到这些更改。

`R:` 没有为此映射预留交换空间。使用 `MAP_NORESERVE` 创建的映射和 System V ISM 共享内存映射不预留交换空间。

`*` 核心文件中不存在映射数据（仅当应用于核心文件时适用）。有关配置核心文件内容的信息，请参见 [coreadm\(1M\)](#)。

Lgroup (Lgrp) 包含为指定映射提供后备支持的物理内存的 `lgroup`。

映射名称 (Mapped File)	<p>每个映射的说明性名称。将为映射显示下列主要的名称类型：</p> <ul style="list-style-type: none"> ▪ 映射的文件：对于进程与文件之间的映射，<code>pmap</code> 命令会尝试解析每个映射的文件名。如果无法解析文件名，<code>pmap</code> 将显示包含该文件的主设备号和从设备号，以及文件的文件系统 <code>inode</code> 编号。 ▪ 匿名内存：将与文件系统内的任何指定对象或文件都不相关的内存报告为 <code>[anon]</code>。 <p><code>pmap</code> 命令显示某些已知匿名内存映射的通用名称：</p> <p><code>[heap]</code> 映射是进程堆。</p> <p><code>[stack]</code> 映射是主栈。</p> <p><code>[stack tid=<i>n</i>]</code> 映射是线程 <i>n</i> 的栈。</p> <p><code>[altstack tid=<i>n</i>]</code> 映射用作线程 <i>n</i> 的备用信号栈。</p> <p>如果映射的通用名称未知，<code>pmap</code> 将 <code>[anon]</code> 显示为映射名称。</p> <ul style="list-style-type: none"> ▪ System V 共享内存：使用 System V 共享内存系统调用创建的映射是通过下面显示的名称报告的： <ul style="list-style-type: none"> <code>shmid=<i>n</i></code>： 映射是 System V 共享内存映射。报告了创建映射时使用的共享内存标识符。 <code>ism shmid=<i>n</i></code>： 映射是 System V 共享内存的“锁定共享内存” (ISM) 变体。ISM 映射是在设置了 <code>SHM_SHARE_MMU</code> 标志的情况下依照 <code>shmat(2)</code> 创建的（请参见 <code>shmop(2)</code>）。 <code>dism shmid=<i>n</i></code>： 映射是 ISM 的可分页变体。可分页的 ISM 是在设置了 <code>SHM_PAGEABLE</code> 标志的情况下依照 <code>shmat(2)</code> 创建的（请参见 <code>shmop(2)</code>）。 ▪ 其他：其他对象（包括帧缓存器等设备）的映射。对于其他被映射的对象，没有显示映射名称。
页面大小 (Pgsz)	用于该映射的硬件地址转换的页面大小（以千字节为单位）。有关详细信息，请参见 <code>memcntl(2)</code> 。
交换空间 (Swap)	为该映射预留的交换空间量（以千字节为单位）。即，从由 <code>swap -s</code> 命令显示的总的可用预留交换空间池中扣除的

交换空间。请参见 [swap\(1M\)](#)。

示例

示例1 显示进程映射

缺省情况下，`pmap` 为目标进程的地址空间内的每个映射输出一行。下面的示例显示了典型 `bourne shell` 的地址空间：

```
example$ pmap 102905
102905:  sh
00010000  192K r-x-- /usr/bin/ksh
00040000    8K rwx-- /usr/bin/ksh
00042000   40K rwx-- [ heap ]
FF180000  664K r-x-- /usr/lib/libc.so.1
FF236000   24K rwx-- /usr/lib/libc.so.1
FF23C000    8K rwx-- /usr/lib/libc.so.1
FF250000    8K rwx-- [ anon ]
FF260000   16K r-x-- /usr/lib/en_US.ISO8859-1.so.2
FF272000   16K rwx-- /usr/lib/en_US.ISO8859-1.so.2
FF280000  560K r-x-- /usr/lib/libnsl.so.1
FF31C000   32K rwx-- /usr/lib/libnsl.so.1
FF324000   32K rwx-- /usr/lib/libnsl.so.1
FF350000   16K r-x-- /usr/lib/libmp.so.2
FF364000    8K rwx-- /usr/lib/libmp.so.2
FF380000   40K r-x-- /usr/lib/libsocket.so.1
FF39A000    8K rwx-- /usr/lib/libsocket.so.1
FF3A0000    8K r-x-- /usr/lib/libdl.so.1
FF3B0000    8K rwx-- [ anon ]
FF3C0000  152K r-x-- /usr/lib/ld.so.1
FF3F6000    8K rwx-- /usr/lib/ld.so.1
FFBFC000   16K rw--- [ stack ]
total    1864
```

示例2 显示内存分配和映射类型

可使用 `-x` 选项提供有关每个映射的内存分配和映射类型的信息。为每个映射显示了驻留的、非共享匿名的和锁定的内存量：

```
example$ pmap -x 102908
102908:  sh
Address  Kbytes  RSS  Anon  Locked Mode  Mapped File
00010000   88    88    -    -  r-x--  sh
00036000    8     8     8    -  rwx--  sh
00038000   16    16    16    -  rwx--  [ heap ]
FF260000   16    16    -    -  r-x--  en_US.ISO8859-1.so.2
FF272000   16    16    -    -  rwx--  en_US.ISO8859-1.so.2
FF280000  664   624    -    -  r-x--  libc.so.1
FF336000   32    32     8    -  rwx--  libc.so.1
FF380000   24    24    -    -  r-x--  libgen.so.1
FF396000    8     8    -    -  rwx--  libgen.so.1
```

示例2 显示内存分配和映射类型 (续)

```

FF3A0000      8      8      -      - r-x-- libdl.so.1
FF3B0000      8      8      8      - rwx-- [ anon ]
FF3C0000     152     152     -      - r-x-- ld.so.1
FF3F6000      8      8      8      - rwx-- ld.so.1
FFBFE000      8      8      8      - rw--- [ stack ]
-----
total Kb    1056    1016     56     -

```

通过使用每个映射的驻留内存计数和匿名内存计数可以估算进程的每个附加实例使用的增量内存量。

在上面的示例中，bourne shell 具有 1032 千字节的驻留内存大小。不过，该 shell 使用的大量物理内存是与 shell 的其他实例共享的。shell 的另一相同实例会尽可能地与该 shell 共享物理内存，并为任何非共享部分分配匿名内存。在上面的示例中，每个附加 bourne shell 使用了大约 56 千字节的附加物理内存。

下面的较复杂示例显示了包含不同映射类型的进程的输出格式。在此示例中，映射如下所示：

```

0001000: Executable text, mapped from 'maps' program

0002000: Executable data, mapped from 'maps' program

0002200: Program heap

0300000: A mapped file, mapped MAP_SHARED
0400000: A mapped file, mapped MAP_PRIVATE

0500000: A mapped file, mapped MAP_PRIVATE | MAP_NORESERVE

0600000: Anonymous memory, created by mapping /dev/zero

0700000: Anonymous memory, created by mapping /dev/zero
         with MAP_NORESERVE

0800000: A DISM shared memory mapping, created with SHM_PAGEABLE
         with 8MB locked via mlock(2)

0900000: A DISM shared memory mapping, created with SHM_PAGEABLE,
         with 4MB of its pages touched.

0A00000: A DISM shared memory mapping, created with SHM_PAGEABLE,
         with none of its pages touched.

0B00000: An ISM shared memory mapping, created with SHM_SHARE_MMU

```

示例2 显示内存分配和映射类型 (续)

```
example$ pmap -x 15492
15492: ./maps
  Address  Kbytes    RSS   Anon  Locked Mode   Mapped File
00010000      8      8     -    -  r-x--  maps
00020000      8      8      8    -  rwx--  maps
00022000  20344  16248  16248    -  rwx--  [ heap ]
03000000   1024   1024     -    -  rw-s-  dev:0,2 ino:4628487
04000000   1024   1024    512    -  rw---  dev:0,2 ino:4628487
05000000   1024   1024    512    -  rw--R  dev:0,2 ino:4628487
06000000   1024   1024   1024    -  rw---  [ anon ]
07000000    512    512    512    -  rw--R  [ anon ]
08000000   8192   8192     -   8192  rwx-s-  [ dism shmid=0x5]
09000000   8192   4096     -    -  rwx-s-  [ dism shmid=0x4]
0A000000   8192   8192     -   8192  rwx-sR  [ ism shmid=0x2 ]
0B000000   8192   8192     -   8192  rwx-sR  [ ism shmid=0x3 ]
FF280000    680    672     -    -  r-x--  libc.so.1
FF33A000     32     32     32    -  rwx--  libc.so.1
FF3A0000      8      8     -    -  r-x--  libdl.so.1
FF3B0000      8      8      8    -  rwx--  [ anon ]
FF3C0000    152    152     -    -  r-x--  ld.so.1
FF3F6000      8      8      8    -  rwx--  ld.so.1
FFBFA000     24     24     24    -  rwx--  [ stack ]
-----
total Kb   50456  42256  18888  16384
```

示例3 显示页面大小信息

可以使用 `-s` 选项为地址空间的每一部分显示硬件转换页面大小。(有关 Solaris 多页面大小支持的详细信息, 请参见 [memcntl\(2\)](#))。

在下面的示例中, 我们可以看到绝大部分映射使用 8 KB 的页面大小, 而堆使用 4 MB 的页面大小。

请注意, 相同页面大小的驻留页面的非连续区域报告为单独的映射。在下面的示例中, `libc.so` 库被报告为多个单独的映射, 因为只有部分 `libc.so` 文本是驻留的:

```
example$ pmap -xs 15492
15492: ./maps
  Address  Kbytes    RSS   Anon  Locked Pgsz Mode   Mapped File
00010000      8      8     -    -   8K  r-x--  maps
00020000      8      8      8    -   8K  rwx--  maps
00022000  3960  3960  3960    -   8K  rwx--  [ heap ]
00400000   8192   8192  8192    -   4M  rwx--  [ heap ]
00C00000   4096     -     -    -    -  rwx--  [ heap ]
01000000   4096  4096  4096    -   4M  rwx--  [ heap ]
03000000   1024   1024     -    -   8K  rw-s-  dev:0,2 ino:4628487
```

示例3 显示页面大小信息 (续)

```

04000000    512    512    512    -    8K rw--- dev:0,2 ino:4628487
04080000    512    512    -    -    - rw--- dev:0,2 ino:4628487
05000000    512    512    512    -    8K rw--R dev:0,2 ino:4628487
05080000    512    512    -    -    - rw--R dev:0,2 ino:4628487
06000000   1024   1024   1024    -    8K rw--- [ anon ]
07000000    512    512    512    -    8K rw--R [ anon ]
08000000   8192   8192    -   8192    - rwxS- [ dism shmid=0x5 ]
09000000   4096   4096    -    -    8K rwxS- [ dism shmid=0x4 ]
0A000000   4096    -    -    -    - rwxS- [ dism shmid=0x2 ]
0B000000   8192   8192    -   8192   4M rwxSR [ ism shmid=0x3 ]
FF280000    136    136    -    -    8K r-x-- libc.so.1
FF2A2000    120    120    -    -    - r-x-- libc.so.1
FF2C0000    128    128    -    -    8K r-x-- libc.so.1
FF2E0000    200    200    -    -    - r-x-- libc.so.1
FF312000    48     48     -    -    8K r-x-- libc.so.1
FF31E000    48     40     -    -    - r-x-- libc.so.1
FF33A000    32     32     32    -    8K rwx-- libc.so.1
FF3A0000     8      8      -    -    8K r-x-- libdl.so.1
FF3B0000     8      8      8     -    8K rwx-- [ anon ]
FF3C0000    152    152    -    -    8K r-x-- ld.so.1
FF3F6000     8      8      8     -    8K rwx-- ld.so.1
FFBFA000    24     24     24    -    8K rwx-- [ stack ]
-----
total Kb   50456   42256   18888   16384

```

示例4 显示交换预留空间

可使用 `-s` 选项来描述进程的交换预留空间。将为进程内的每个映射显示预留的交换空间量。对于共享的映射，交换预留空间报告为零，因为这类预留空间仅在系统范围内计入一次。

```

example$ pmap -S 15492
15492: ./maps
Address Kbytes Swap Mode Mapped File
00010000     8    - r-x-- maps
00020000     8     8 rwx-- maps
00022000  20344  20344 rwx-- [ heap ]
03000000   1024    - rw-s- dev:0,2 ino:4628487
04000000   1024   1024 rw--- dev:0,2 ino:4628487
05000000   1024    512 rw--R dev:0,2 ino:4628487
06000000   1024   1024 rw--- [ anon ]
07000000    512    512 rw--R [ anon ]
08000000   8192    - rwxS- [ dism shmid=0x5]
09000000   8192    - rwxS- [ dism shmid=0x4]
0A000000   8192    - rwxS- [ dism shmid=0x2]
0B000000   8192    - rwxSR [ ism shmid=0x3]

```


示例4 显示交换预留空间 (续)

```

FF280000    680    - r-x-- libc.so.1
FF33A000    32    32 rwx-- libc.so.1
FF3A0000     8    - r-x-- libdl.so.1
FF3B0000     8    8 rwx-- [ anon ]
FF3C0000   152    - r-x-- ld.so.1
FF3F6000     8    8 rwx-- ld.so.1
FFBFA000    24    24 rwx-- [ stack ]
-----
total Kb   50456  23496

```

可以使用交换预留空间信息估算每个附加进程使用的虚拟交换量。每个进程都使用全局虚拟交换池中的虚拟交换。全局交换预留空间是由 `swap(1M)` 命令的“avail”字段报告的。

示例5 标记多线程进程中的栈

```

example$ pmap 121969
121969: ./stacks
00010000    8K r-x-- /tmp/stacks
00020000    8K rwx-- /tmp/stacks
FE8FA000    8K rwx-R [ stack tid=11 ]
FE9FA000    8K rwx-R [ stack tid=10 ]
FEAFA000    8K rwx-R [ stack tid=9 ]
FEBFA000    8K rwx-R [ stack tid=8 ]
FECFA000    8K rwx-R [ stack tid=7 ]
FEDFA000    8K rwx-R [ stack tid=6 ]
FEEFA000    8K rwx-R [ stack tid=5 ]
FEFFA000    8K rwx-R [ stack tid=4 ]
FF0FA000    8K rwx-R [ stack tid=3 ]
FF1FA000    8K rwx-R [ stack tid=2 ]
FF200000   64K rw--- [ altstack tid=8 ]
FF220000   64K rw--- [ altstack tid=4 ]
FF240000  112K rw--- [ anon ]
FF260000   16K rw--- [ anon ]
FF280000  672K r-x-- /usr/lib/libc.so.1
FF338000   24K rwx-- /usr/lib/libc.so.1
FF33E000    8K rwx-- /usr/lib/libc.so.1
FF35A000    8K rwxS- [ anon ]
FF360000  104K r-x-- /usr/lib/libthread.so.1
FF38A000    8K rwx-- /usr/lib/libthread.so.1
FF38C000    8K rwx-- /usr/lib/libthread.so.1
FF3A0000    8K r-x-- /usr/lib/libdl.so.1
FF3B0000    8K rwx-- [ anon ]
FF3C0000  152K r-x-- /usr/lib/ld.so.1
FF3F6000    8K rwx-- /usr/lib/ld.so.1
FFBFA000   24K rwx-- [ stack ]

```

示例5 标记多线程进程中的栈 (续)

```
total      1384
```

示例6 显示lgroup内存分配

下面的示例按映射显示lgroup内存分配：

```
example$ pmap -L 'pgrep nscd'
100095: /usr/sbin/nscd
00010000      8K r-x--  2 /usr/sbin/nscd
00012000     48K r-x--  1 /usr/sbin/nscd
0002E000      8K rwx--  2 /usr/sbin/nscd
00030000     16K rwx--  2 [ heap ]
00034000      8K rwx--  1 [ heap ]
.
.
.
FD80A000     24K rwx--  2 [ anon ]
FD820000      8K r-x--  2 /lib/libmd5.so.1
FD840000     16K r-x--  1 /lib/libmp.so.2
FD860000      8K r-x--  2 /usr/lib/straddr.so.2
FD872000      8K rwx--  1 /usr/lib/straddr.so.2
FD97A000      8K rw--R  1 [ stack tid=24 ]
FD990000      8K r-x--  2 /lib/nss_nis.so.1
FD992000     16K r-x--  1 /lib/nss_nis.so.1
FD9A6000      8K rwx--  1 /lib/nss_nis.so.1
FD9C0000      8K rwx--  2 [ anon ]
FD9D0000      8K r-x--  2 /lib/nss_files.so.1
FD9D2000     16K r-x--  1 /lib/nss_files.so.1
FD9E6000      8K rwx--  2 /lib/nss_files.so.1
FDAFA000      8K rw--R  2 [ stack tid=23 ]
FDBFA000      8K rw--R  1 [ stack tid=22 ]
FDCFA000      8K rw--R  1 [ stack tid=21 ]
FDDFA000      8K rw--R  1 [ stack tid=20 ]
.
.
.
FEFFA000      8K rw--R  1 [ stack tid=2 ]
FF000000      8K rwx--  2 [ anon ]
FF004000     16K rwx--  1 [ anon ]
FF00A000     16K rwx--  1 [ anon ]
.
.
.
FF3EE000      8K rwx--  2 /lib/ld.so.1
FFBFE000      8K rw---  2 [ stack ]
total      2968K
```

退出状态 将返回以下退出值：

0 操作成功。

非零 出现错误。

文件 /proc/* 进程文件
 /usr/proc/lib/* proc 工具支持文件

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

命令语法是 "Committed"（已确定）。-L 选项和输出格式是 "Uncommitted"（未确定）。

另请参见 [ldd\(1\)](#)、[lgrpinfo\(1\)](#)、[madv.so.1\(1\)](#)、[mdb\(1\)](#)、[plgrp\(1\)](#)、[pmadvise\(1\)](#)、[proc\(1\)](#)、[ps\(1\)](#)、[coread\(1\)](#)

引用名	pm-updatemanager – 用于更新软件包的应用程序
用法概要	<pre>/usr/bin/pm-updatemanager [options] /usr/bin/pm-updatemanager [-h --help] [-d --debug] [-R dir --image-dir dir]</pre>
描述	<p>pm-updatemanager 检查系统上安装的软件包是否存在可用更新并安装这些更新。</p> <p>注 – 如果 package/pkg、package/pkg/package-manager 或 package/pkg/update-manager 软件包需要更新，则 pm-updatemanager 首先更新这些软件包，然后重新启动以完成其余所有更新。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none">-h--help 显示用法消息。-d--debug 在调试模式下运行 pm-updatemanager。-R <i>dir</i>--image-dir <i>dir</i> 对根目录为 <i>dir</i> 的映像（而不是自动搜索到的映像）执行操作。
示例	<p>示例1 更新当前映像</p> <p>对当前映像调用 pm-updatemanager。这会检查当前映像中安装的软件包的所有可用更新并安装这些更新。</p> <pre>\$ /usr/lib/pm-launch pm-updatemanager</pre> <p>这是桌面菜单选项“系统”>“管理”>“Update Manager”调用的同一命令。</p> <p>示例2 更新指定映像</p> <p>对存储在 /aux0/example_root 中的映像调用 pm-updatemanager。</p> <pre>\$ /usr/lib/pm-launch pm-updatemanager -R /aux0/example_root</pre>
退出状态	<p>将返回以下退出值：</p> <ul style="list-style-type: none">0 一切正常工作。1 出现错误。2 指定的命令行选项无效。

属性 有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	package/pkg/update-manager
接口稳定性	Uncommitted（未确定）

另请参见 [packagemanager\(1\)](#)、[pkg\(1\)](#)、[pkg\(5\)](#)

<http://hub.opensolaris.org/bin/view/Project+pkg/>

附注 对您不具有的映像执行操作时，需要具有足够的特权才能调用 `pm-updatemanager`。在这些情况下，通常使用 `/usr/lib/pm-launch` 调用 `pm-updatemanager`。

引用名	ppgsz - 为栈、堆和/或其他匿名段设置首选页面大小
用法概要	<code>/usr/bin/ppgsz [-F] -o option[,option] cmd -p pid...</code>
描述	ppgsz 实用程序为目标进程（即启动的 <i>cmd</i> 或 <i>pid</i> 列表中的进程）的栈、堆和/或其他匿名段设置首选页面大小。ppgsz 在更改页面大小时会停止目标进程。请参见 memcntl(2) 。
选项	支持以下选项：
-F	强制。为目标进程设置首选页面大小选项（即使目标进程被其他进程所控制）。使用 -F 标志时应谨慎。请参见 proc(1) 。
-o option[,option]	这些选项如下所示：
heap=size	此选项用于为目标进程的堆指定首选页面大小。heap 被定义为 bss（未初始化的数据）及紧随其后的 brk 区域（请参见 brk(2) ）。首选堆页面大小是为现有堆以及将来分配的任何其他堆内存设置的。请参见“附注”部分。
stack=size	此选项用于为目标进程的栈指定首选页面大小。首选栈页面大小是为现有栈以及栈扩展时新分配的栈部分配置的。
anon=size	此选项用于为目标进程的所有现有 MAP_PRIVATE 匿名段（而不是堆和栈）指定首选页面大小，这些匿名段足够大，至少能够容纳一个指定大小的对齐页面。对于足够大的段，首选页面大小是从段中的第一个大小对齐的地址开始设置的。anon 首选页面大小不应用于将来创建的 MAP_PRIVATE 匿名段。请参见 mmap(2) 中的 MAP_ANON。
	匿名内存指的是与没有与某个文件系统中的文件直接关联的 MAP_PRIVATE 页面。ppgsz 命令使用 memcntl(2) 为匿名段设置首选页面大小。请参见 memcntl(2) 中的 MC_HAT_ADVISE。
	必须至少指定以上选项之一。
	size 必须是受支持的页面大小（请参见 pagesize(1) ）或 0，在后一种情况下，将由系统来选择合适的页面大小。请参见 memcntl(2) 。
	size 的缺省单位为字节，可指定为八进制 (0)、十进制或十六进制 (0x) 格式的。可以使用 K、M、G 或 T 对数字值加以限定，以分别

指定千字节、兆字节、千兆字节或兆兆字节。4194304、0x400000、4096K、0x1000K 和 4M 这几种不同的方式指定的都是 4 兆字节。

-p pid

为跟在 **-p** 选项后的进程 ID (*pid*) 列表中的目标进程设置首选页面大小选项。*pid* 列表还可以包含 */proc* 目录中的名称。只允许进程所有者或超级用户设置页面大小。

如果未指定 **-p**，将会解释 *cmd*。ppgsz 将启动 *cmd*，并将页面大小选项应用于新进程。

堆和栈首选页面大小是继承的。在 ppgsz 完成后，从已启动的进程或 *pid* 列表中的目标进程创建的子进程（请参见 [fork\(2\)](#)）将继承首选堆和栈页面大小。在执行 [exec\(2\)](#) 时，所有段的首选页面大小都将重新设置为缺省系统页面大小（请参见 [getpagesize\(3C\)](#)）。已启动的进程或目标进程的子进程不会继承其他所有匿名段的首选页面大小。

示例

示例1 设置首选堆和栈页面大小

下面的示例针对所有以 *ora* 开头的归 *ora* 所有的进程运行命令将首选堆页面大小设置为 4M，将首选栈页面大小设置为 512K：

```
example% ppgsz -o heap=4M,stack=512K -p 'pgrep -u ora '^ora''
```

示例2 设置首选匿名页面大小

下面的示例针对进程 ID 953 将现有合格匿名段的首选页面大小设置为 512k：

```
example% ppgsz -o anon=512k -p 953
```

退出状态

如果指定并成功调用了 *cmd*（请参见 [exec\(2\)](#)），则 ppgsz 的退出状态将是 *cmd* 的退出状态。否则，ppgsz 将以下列值之一退出：

- 0 成功为 *pid* 列表中的进程设置了首选页面大小。
- 125 ppgsz 中出现错误。错误包括：参数无效、指定的页面大小无效，以及无法为 *pid* 列表中的一个或多个进程或 *cmd* 设置首选页面大小。
- 126 找到了 *cmd*，但无法调用。
- 127 找不到 *cmd*。

文件

- /proc/** 进程文件。
- /usr/lib/ld/map.bssalign* 用于对齐 *bss* 的模板链接编辑器 *mapfile*（请参见“附注”部分）。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/extended-system-utilities
接口稳定性	Committed (已确定)

另请参见

[ld\(1\)](#)、[mpss.so.1\(1\)](#)、[pagesize\(1\)](#)、[pgrep\(1\)](#)、[pmap\(1\)](#)、[proc\(1\)](#)、[brk\(2\)](#)、[exec\(2\)](#)、[fork\(2\)](#)、[memcn](#)

《链接程序和库指南》

附注

由于资源限制，首选页面大小的设置不一定可以保证目标进程将获得首选页面大小。可以使用 [pmap\(1\)](#) 查看目标进程的堆和栈页面的实际大小（请参见 [pmap -s](#) 选项）。

在大页面大小的倍数的地址上，需要映射大页面。因为堆通常不是以大页面对齐的，堆的起始部分（位于第一个以大页面对齐的地址下）是以系统内存页面大小映射的。请参见 [getpagesize\(3C\)](#)。

要提供将以大页面大小映射的堆，可使用包含 `bss` 段声明指令的链接编辑器 ([ld\(1\)](#)) `mapfile` 来构建一个应用程序。有关此指令以及 `/usr/lib/ld/map.bssalign` 中提供的 `mapfile` 模板的更多详细信息，请参见《链接程序和库指南》中的“Mapfile 选项”部分。用户需要注意的是，对齐规范可能是特定于计算机的，在不同的硬件平台上可能会失去其作用。未来的发行版中可能会出现更灵活请求最佳底层页面大小的方式。

还可以使用 [mpss.so.1\(1\)](#)（一个可预装入的共享目标文件）来设置首选栈和/或堆页面大小。

引用名	ppriv – 检查或修改进程特权集和属性
用法概要	<pre> /usr/bin/ppriv -e [-D -N] [-M] [-s spec] [-X -r rule] command [arg]... /usr/bin/ppriv [-vn] [-S] [-D -N] [-s spec] [-X -r rule] [pid core]... /usr/bin/ppriv -l [-vn] [privilege-specification extended-policy]... </pre>
描述	<p>ppriv 命令的第一次调用使用根据命令行上的参数修改的特权集和标志运行指定的 <i>command</i>。</p> <p>第二次调用检查或更改正在运行的进程和核心文件的特权状态。</p> <p>第三次调用列出所定义的特权以及关于指定的特权或特权集规范的信息。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -D 为所提供的进程或命令打开特权调试。 -e 将其余参数解释为一个命令行，并使用指定的特权属性和特权集运行该命令行。 -l 在 <code>stdout</code> 上列出当前定义的所有特权。 -M 当系统配置有 Trusted Extensions 时，此选项将启用 <code>NET_MAC_AWARE</code> 和 <code>NET_MAC_AWARE_INHERIT</code> 进程属性。 具有这些属性和 <code>net_mac_aware</code> 特权的进程可与较低级别的远程对等方进行通信。 -n 将端口号和用户显示为数字。通常，ppriv 将端口号和用户显示为符号。此选项仅在显示扩展策略时适用。 -N 为所提供的进程或命令关闭特权调试。 -s <i>spec</i> 根据 <i>spec</i>（以 <code>[AEILP][+ -=]</code><i>privsetspec</i> 格式指定的内容，不包含空格）修改进程的特权集，其中： <ul style="list-style-type: none"> AEILP 表示一个或多个指示要更改的特权集的字母。这些字母不区分大小写，例如，<code>a</code> 或 <code>A</code> 都指示所有特权集。 有关特权集合的单一字母缩写的定义，请参见 privileges(5)。 + -= 表示修饰符，分别用于在 <i>privsetspec</i> 中指定的特权集中添加 (+) 或删除 (-) 所列出的特权，或者将所列出的特权赋予 (=) 该特权集。 <i>privsetspec</i> 表示以逗号分隔的特权集指定内容 (<code>priv1、priv2、</code> 等等)，如 priv_str_to_set(3C) 中所述。

可以使用多个 `-s` 选项修改同一特权集，前提是只指定了对单个特权集的赋值，或者指定了任意数目的添加和删除。也就是说，针对特权集的赋值与添加或删除是互斥的。

`-r rule` 安装扩展策略。请参见 [privileges\(5\)](#)。

可以指定多个规则。新规则将添加到现有策略中。要替换现有策略，请先使用 `-x` 将其删除，然后使用 `-r` 添加新策略。

`-S` 短. 为特权集报告尽可能短的输出字符串。缺省设置是可移植输出。请参见 [priv_str_to_set\(3C\)](#)。

`-X` 禁用扩展策略。

`-v` 详细模式。使用特权名称报告特权集。

用法

`ppriv` 实用程序检查进程和核心文件，并输出或更改其特权集。

`ppriv` 可以在启用或禁用特权调试的情况下执行命令，或者使用比调用进程时更少的特权运行命令。

执行子进程时，可修改的特权集只有 `L` 和 `I`。只能从 `L` 和 `I` 删除特权，因为 `ppriv` 是以 `P=E=I` 设置启动的。

还可使用 `ppriv` 从进程删除特权，或者将特权传递至其他进程。要控制某个进程，`ppriv` 实用程序的有效特权集必须是受控进程的 `E`、`I` 和 `P` 的超集。实用程序的限制特权集必须是目标的限制特权集的超集。如果目标的进程 `uid` 不匹配，则必须在实用程序的有效特权集中声明 `{PRIV_PROC_OWNER}` 特权。如果受控进程具有值为 `0` 的 `uid`，则可能存在更多限制。请参见 [privileges\(5\)](#)。

示例

示例 1 获取当前 Shell 的进程特权

以下示例将获取当前 shell 的进程特权：

```
example$ ppriv $$
387:  -sh
flags = <none>
      E: basic
      I: basic
      P: basic
      L: all
```

示例 2 从 Shell 的可继承的有效特权集中删除一个特权

以下示例将从 shell 的可继承的有效特权集中删除一个特权。

```
example$ ppriv -s EI-proc_session $$
```

子进程仍可检查父 shell，但不能再影响父进程，因为父进程在其允许特权集中的特权比 `ppriv` 子进程多：

示例2 从 Shell 的可继承的有效特权集中删除一个特权 (续)

```
example$ truss -p $$
truss: permission denied: 387
```

```
example$ ppriv $$
387:  -sh
flags = <none>
      E: basic,!proc_session
      I: basic,!proc_session
      P: basic
      L: all
```

示例3 在启用特权调试的情况下运行进程

以下示例在启用特权调试的情况下运行进程：

```
example$ ppriv -e -D cat /etc/shadow
cat[418]: missing privilege "file_dac_read" (euid = 21782),
          needed at ufs_access+0x3c
cat: cannot open /etc/shadow
```

特权调试错误消息将发送到当前进程的控制终端。`needed at` 地址规范是内核实现的人工产物，可在软件更新后的任何时间进行更改。

可使用 `/etc/name_to_sysnum` 将系统调用号映射至系统调用。

示例4 列出当前区域中可用的特权

以下示例将列出当前区域中可用的特权（请参见 [zones\(5\)](#)）。当在全局区域中运行时，会列出已定义的所有特权。

```
example$ ppriv -l zone
... listing of all privileges elided ...
```

示例5 检查特权识别进程

以下示例检查特权识别进程：

```
example$ ppriv -S 'pgrep rpcbind'

928:  /usr/sbin/rpcbind
flags = PRIV_AWARE
      E: net_privaddr,proc_fork,sys_nfs
      I: none
      P: net_privaddr,proc_fork,sys_nfs
      L: none
```

有关标志的解释，请参见 [setpflags\(2\)](#)。

示例6 在扩展策略下运行进程

以下示例在扩展策略下运行进程：

```
example$ ppriv -r '{file_write}:/home/casper/.mozilla/*' \
-r '{file_write}:/tmp/*,{proc_exec}:/usr/*' -e firefox
```

请参见 [privileges\(5\)](#)。

示例7 检查已启动的进程

以下示例检查在示例6中启动的进程：

```
example$ ppriv 101272
101272: /usr/lib/firefox/firefox-bin
flags = PRIV_XPOLICY
Extended policies:
{file_write}:/home/casper/.mozilla/*
{file_write}:/tmp/*
{proc_exec}:/usr/*
E: basic,!file_write,!proc_exec
I: basic,!file_write,!proc_exec
P: basic,!file_write,!proc_exec
L: all
```

退出状态

将返回以下退出值：

0 操作成功。

非零 出现错误。

文件

/proc/* 进程文件

/etc/name_to_sysnum 系统调用名称到系统调用号的映射

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

调用为 Committed（已确定）。输出是 "Uncommitted"（未确定）。

另请参见

[gcore\(1\)](#)、[truss\(1\)](#)、[setpflags\(2\)](#)、[priv_str_to_set\(3C\)](#)、[proc\(4\)](#)、[attributes\(5\)](#)、[privileges\(5\)](#)

引用名

pr – print files

用法概要

```

/usr/bin/pr [+ page] [-column] [-adFmrt] [-e [char] [gap]]
           [-h header] [-i [char] [gap]] [-l lines]
           [-n [char] [width]] [-o offset] [-s [char]]
           [-w width] [-fp] [file]...

/usr/xpg4/bin/pr [+ page] [-column | -c column] [-adFmrt]
                [-e [char] [gap]] [-h header] [-i [char] [gap]]
                [-l lines] [-n [char] [width]] [-o offset]
                [-s [char]] [-w width] [-fp] [file]...

```

描述

The `pr` utility is a printing and pagination filter. If multiple input files are specified, each is read, formatted, and written to standard output. By default, the input is separated into 66-line pages, each with:

- a 5-line header that includes the page number, date, time and the path name of the file
- a 5-line trailer consisting of blank lines

If standard output is associated with a terminal, diagnostic messages will be deferred until the `pr` utility has completed processing.

When options specifying multi-column output are specified, output text columns will be of equal width; input lines that do not fit into a text column will be truncated. By default, text columns are separated with at least one blank character.

选项

The following options are supported. In the following option descriptions, *column*, *lines*, *offset*, *page*, and *width* are positive decimal integers; *gap* is a non-negative decimal integer. Some of the option-arguments are optional, and some of the option-arguments cannot be specified as separate arguments from the preceding option letter. In particular, the `-s` option does not allow the option letter to be separated from its argument, and the options `-e`, `-i`, and `-n` require that both arguments, if present, not be separated from the option letter.

The following options are supported for both `/usr/bin/pr` and `/usr/xpg4/bin/pr`:

- | | |
|----------------------|--|
| <code>+page</code> | Begins output at page number <i>page</i> of the formatted input. |
| <code>-column</code> | Produces multi-column output that is arranged in <i>column</i> columns (default is 1) and is written down each column in the order in which the text is received from the input file. This option should not be used with <code>-m</code> . The <code>-e</code> and <code>-i</code> options will be assumed for multiple text-column output. Whether or not text columns are produced with identical vertical lengths is unspecified, but a text column will never exceed the length of the page (see the <code>-l</code> option). When used with <code>-t</code> , use the minimum number of lines to write the output. |
| <code>-a</code> | Modifies the effect of the <code>-column</code> option so that the columns are filled across the page in a round-robin order (for example, when |

- column* is 2, the first input line heads column 1, the second heads column 2, the third is the second line in column 1, and so forth).
- d Produces output that is double-spaced; append an extra NEWLINE character following every NEWLINE character found in the input.
 - e [*char*] [*gap*] Expands each input TAB character to the next greater column position specified by the formula $n * \textit{gap} + 1$, where n is an integer > 0 . If *gap* is 0 or is omitted, it defaults to 8. All TAB characters in the input will be expanded into the appropriate number of SPACE characters. If any non-digit character, *char*, is specified, it will be used as the input tab character.
 - f Uses a FORMFEED character for new pages, instead of the default behavior that uses a sequence of NEWLINE characters. Pauses before beginning the first page if the standard output is associated with a terminal.
 - h *header* Uses the string *header* to replace the contents of the *file* operand in the page header.
 - l *lines* Overrides the 66-line default and reset the page length to *lines*. If *lines* is not greater than the sum of both the header and trailer depths (in lines), pr will suppress both the header and trailer, as if the -t option were in effect.
 - m Merges files. Standard output will be formatted so pr writes one line from each file specified by *file*, side by side into text columns of equal fixed widths, in terms of the number of column positions. Implementations support merging of at least nine files.
 - n [*char*] [*width*] Provides *width*-digit line numbering (default for *width* is 5). The number will occupy the first *width* column positions of each text column of default output or each line of -m output. If *char* (any non-digit character) is given, it will be appended to the line number to separate it from whatever follows (default for *char* is a TAB character).
 - o *offset* Each line of output will be preceded by offset $\langle \text{space} \rangle s$. If the -o option is not specified, the default offset is 0. The space taken will be in addition to the output line width (see -w option below).
 - p Pauses before beginning each page if the standard output is directed to a terminal (pr will write an ALERT character to standard error and wait for a carriage-return character to be read on /dev/tty).
 - r Writes no diagnostic reports on failure to open files.

- s [*char*] Separates text columns by the single character *char* instead of by the appropriate number of SPACE characters (default for *char* is the TAB character).
- t Writes neither the five-line identifying header nor the five-line trailer usually supplied for each page. Quits writing after the last line of each file without spacing to the end of the page.
- w *width* Sets the width of the line to *width* column positions for multiple text-column output only. If the -w option is not specified and the -s option is not specified, the default width is 72. If the -w option is not specified and the -s option is specified, the default width is 512.

For single column output, input lines will not be truncated.

/usr/bin/pr

The following options are supported for /usr/bin/pr only:

- F Folds the lines of the input file. When used in multi-column mode (with the -a or -m options), lines will be folded to fit the current column's width. Otherwise, they will be folded to fit the current line width (80 columns).
- i [*char*][*gap*] In output, replaces SPACE characters with TAB characters wherever one or more adjacent SPACE characters reach column positions $gap+1$, $2*gap+1$, $3*gap+1$, and so forth. If *gap* is 0 or is omitted, default TAB settings at every eighth column position are assumed. If any non-digit character, *char*, is specified, it will be used as the output TAB character.

/usr/xpg4/bin/pr

The following options are supported for /usr/xpg4/bin/pr only:

- F Uses a FORMFEED character for new pages, instead of the default behavior that uses a sequence of NEWLINE characters.
- i [*char*][*gap*] In output, replaces multiple SPACE characters with TAB characters wherever two or more adjacent SPACE characters reach column positions $gap+1$, $2*gap+1$, $3*gap+1$, and so forth. If *gap* is 0 or is omitted, default TAB settings at every eighth column position are assumed. If any non-digit character, *char*, is specified, it will be used as the output TAB character.

操作数

The following operand is supported:

- file* A path name of a file to be written. If no *file* operands are specified, or if a *file* operand is -, the standard input will be used.

示例

示例 1 Printing a numbered list of all files in the current directory

```
example% ls -a | pr -n -h "Files in $(pwd)."
```

示例 2 Printing files in columns

This example prints file1 and file2 as a double-spaced, three-column listing headed by file list:

```
example% pr -3d -h "file list" file1 file2
```

示例 3 Writing files with expanded column tabs

The following example writes file1 on file2, expanding tabs to columns 10, 19, 28, ...

```
example% pr -e9 -t <file1 >file2
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of pr: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, LC_TIME, TZ, and NLSPATH.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/pr

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

/usr/xpg4/bin/pr

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[expand\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

- 引用名** praliases – display system mail aliases
- 用法概要** praliases [-C *configfile*] [-f *aliasfile*] [*key*]
- 描述** The praliases utility displays system mail aliases. When no key is given, praliases displays the current system aliases, one per line, in no particular order. The form is *key: value*. If a key is given, only that key is looked up and the appropriate *key: value* is displayed if found.
- 选项** The following options are supported:
- C *configfile* Specifies a sendmail configuration file.
 - f *aliasfile* Reads the specified file *aliasfile* instead of the default sendmail system aliases file.
- 操作数** The following operands are supported:
- key* A specific alias key to look up.
- 退出状态** The following exit values are returned:
- 0 Successful operation.
 - >0 An error occurred.
- 文件**
- /etc/mail/aliases Default sendmail system aliases file
 - /etc/mail/aliases.db Database versions of the /etc/mail/aliases file
 - /etc/mail/aliases.dir
 - /etc/mail/aliases.pag Database versions of the /etc/mail/aliases file
 - /etc/mail/sendmail.cf Default sendmail configuration file
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/smtp/sendmail

另请参见 [mailq\(1\)](#), [newaliases\(1M\)](#), [sendmail\(1M\)](#), [attributes\(5\)](#)

引用名	prctl – 获取或设置正在运行的进程、任务和项目的资源控制												
用法概要	prctl [-P] [-t [basic privileged system]] [-n name [-srx] [-v value] [-e -d action] [-p pid]] [-i idtype] id...												
描述	<p>使用 prctl 实用程序，可以检查和修改与系统中的活动进程、任务或项目相关联的资源控制。它允许访问基本限制和特权限制以及当前在指定实体上使用的限制。</p> <p>有关当前发行版的 Solaris 操作系统中支持的资源控制的说明，请参见 resource_controls(5)。</p>												
选项	<p>如果未指定 <code>-s</code>、<code>-r</code>、<code>-x</code>、<code>-v</code>、<code>-d</code> 或 <code>-e</code> 选项中的任何一个，则调用被视为获取操作。否则，它被视为修改操作。</p> <p>支持以下选项：</p> <table border="0" style="margin-left: 2em;"> <tr> <td style="vertical-align: top;"><code>-d -e action</code></td> <td>对 <code>-v</code>、<code>-t</code> 和 <code>-p</code> 指定的资源控制值禁用 (<code>-d</code>) 或启用 (<code>-e</code>) 指定的 <i>action</i>。如果未指定 <code>-v</code>、<code>-t</code> 或 <code>-p</code> 选项中的任何一个，则它们与任何值、特权或接收者 <code>pid</code> 都匹配。例如，仅指定 <code>-v</code> 将修改具有匹配的值且与任意特权和接收者 <code>pid</code> 匹配的 第一个 资源控制。如果未找到匹配的资源控制值，则按指定了 <code>-s</code> 时的方式添加新值。</td> </tr> <tr> <td colspan="2">操作:</td> </tr> <tr> <td style="vertical-align: top;"><code>all</code></td> <td>此操作仅适用于 <code>-d</code>。它将禁用所有操作。对于具有 <code>deny</code> 全局标志的资源控制值，此操作将失败。</td> </tr> <tr> <td style="vertical-align: top;"><code>deny</code></td> <td>表示资源控制尝试拒绝向所请求的资源超出了资源控制值的请求上的进程、任务、项目或区域授予资源。如果资源控制具有 <code>no-deny</code> 全局标志，则无法启用 <code>deny</code> 操作。如果资源控制具有 <code>deny</code> 全局标志，则无法禁用 <code>deny</code> 操作。</td> </tr> <tr> <td style="vertical-align: top;"><code>signal</code></td> <td>此操作仅适用于 <code>-d</code>。它将停用 <code>signal</code> 操作。</td> </tr> <tr> <td style="vertical-align: top;"><code>signal=signalnum</code></td> <td>在 <code>signal=signalnum</code> 操作中，<i>signalnum</i> 是一个信号编号（或信号的字符串表示形式）。对具有 <code>no-local-action</code> 全局标志的资源控制设置</td> </tr> </table>	<code>-d -e action</code>	对 <code>-v</code> 、 <code>-t</code> 和 <code>-p</code> 指定的资源控制值禁用 (<code>-d</code>) 或启用 (<code>-e</code>) 指定的 <i>action</i> 。如果未指定 <code>-v</code> 、 <code>-t</code> 或 <code>-p</code> 选项中的任何一个，则它们与任何值、特权或接收者 <code>pid</code> 都匹配。例如，仅指定 <code>-v</code> 将修改具有匹配的值且与任意特权和接收者 <code>pid</code> 匹配的 第一个 资源控制。如果未找到匹配的资源控制值，则按指定了 <code>-s</code> 时的方式添加新值。	操作:		<code>all</code>	此操作仅适用于 <code>-d</code> 。它将禁用所有操作。对于具有 <code>deny</code> 全局标志的资源控制值，此操作将失败。	<code>deny</code>	表示资源控制尝试拒绝向所请求的资源超出了资源控制值的请求上的进程、任务、项目或区域授予资源。如果资源控制具有 <code>no-deny</code> 全局标志，则无法启用 <code>deny</code> 操作。如果资源控制具有 <code>deny</code> 全局标志，则无法禁用 <code>deny</code> 操作。	<code>signal</code>	此操作仅适用于 <code>-d</code> 。它将停用 <code>signal</code> 操作。	<code>signal=signalnum</code>	在 <code>signal=signalnum</code> 操作中， <i>signalnum</i> 是一个信号编号（或信号的字符串表示形式）。对具有 <code>no-local-action</code> 全局标志的资源控制设置
<code>-d -e action</code>	对 <code>-v</code> 、 <code>-t</code> 和 <code>-p</code> 指定的资源控制值禁用 (<code>-d</code>) 或启用 (<code>-e</code>) 指定的 <i>action</i> 。如果未指定 <code>-v</code> 、 <code>-t</code> 或 <code>-p</code> 选项中的任何一个，则它们与任何值、特权或接收者 <code>pid</code> 都匹配。例如，仅指定 <code>-v</code> 将修改具有匹配的值且与任意特权和接收者 <code>pid</code> 匹配的 第一个 资源控制。如果未找到匹配的资源控制值，则按指定了 <code>-s</code> 时的方式添加新值。												
操作:													
<code>all</code>	此操作仅适用于 <code>-d</code> 。它将禁用所有操作。对于具有 <code>deny</code> 全局标志的资源控制值，此操作将失败。												
<code>deny</code>	表示资源控制尝试拒绝向所请求的资源超出了资源控制值的请求上的进程、任务、项目或区域授予资源。如果资源控制具有 <code>no-deny</code> 全局标志，则无法启用 <code>deny</code> 操作。如果资源控制具有 <code>deny</code> 全局标志，则无法禁用 <code>deny</code> 操作。												
<code>signal</code>	此操作仅适用于 <code>-d</code> 。它将停用 <code>signal</code> 操作。												
<code>signal=signalnum</code>	在 <code>signal=signalnum</code> 操作中， <i>signalnum</i> 是一个信号编号（或信号的字符串表示形式）。对具有 <code>no-local-action</code> 全局标志的资源控制设置												

- signal 操作会失败。可以发送一组有限的信号。有关其他详细信息，请参见“附注”部分。
- i *idtype*** 指定 id 操作数的类型。有效的 *idtype* 为 `process`、`task`、`project` 或 `zone`。还可以为 `pid`、`taskid`、`projid` 和 `zoneid`。如果省略了 `-i` 选项，则缺省 id 类型为 `process`。
- 对于修改操作，id 操作数归属的实体是目标实体。例如，在 `-i process` 上设置一个项目资源控制将会在每个给定进程参数归属的项目上设置该资源控制。
- 对于获取操作，将为 id 操作数归属的所有实体列出资源控制。例如，`-i task taskid` 将为任务以及该任务归属的项目和区域列出任务、项目和区域资源控制。
- n *name*** 指定要获取或设置的资源控制的名称。如果未指定 *name*，将检索所有资源控制。
- p *pid*** 处理基本任务项目或区域资源控制值时（使用 `-s`、`-r`、`-x`、`-d` 或 `-e`），可使用 `-p` 指定接收者 *pid*。在任务、项目或区域上设置一个或多个新的基本资源控制时，如果 `-i idtype` 选项参数不是 `process`，则 `-p` 选项是必需的。
- P** 以空格分隔形式显示资源控制值。
- r** 使用通过 `-v` 选项指定的新值替换第一个资源控制值（与 `-t privilege` 相匹配的）。
- s** 设置一个新的资源控制值。
- 此选项需要 `-v` 选项。
- 如果未指定 `-t` 选项，则使用基本特权。如果要设置基本任务、进程或区域资源控制，则需要 `-p`。如果还指定了 `-e` 或 `-d`，则也将设置对新的资源控制的操作。
- 为了与早期版本兼容，如果指定了 `-v` 且未指定 `-e`、`-d`、`-r` 或 `-x` 中的任何一个，则会隐式包含此选项。

- 有关设置资源控制值时可用于表示大值的单位修饰符和比例因子的说明，请参见 [resource_controls\(5\)](#)。
- t [basic|privileged|system]** 指定要设置的资源控制类型。除非为资源控制设置了 "lowerable" 标志，否则只有其特权等效于 root 的用户（或 setuid 程序）执行的调用可修改特权资源控制。有关 RCTL_GLOBAL_LOWERABLE 标志的说明，请参见 [rctlblk_set_value\(3C\)](#)。如果未指定类型，则采用 **basic**。对于获取操作，如果未指定类型，则将显示所有资源控制类型的值，包括 **system**。
- v value** 为设置操作指定资源控制值。如果未指定 *value*，则将在给定类型的具有最低值的资源控制上执行修改（删除、启用或禁用操作）。
- 有关设置资源控制值时可用于表示大值的单位修饰符和比例因子的说明，请参见 [resource_controls\(5\)](#)。
- x** 删除指定的资源控制值。如果未提供删除选项，则 **prctl** 的缺省操作是修改具有匹配的值和特权的资源控制值，或插入具有给定特权的新值。[setrctl\(2\)](#) 中更完整地讨论了匹配条件。

如果未指定 **-d**、**-e**、**-v** 或 **-x** 选项中的任何一个，则调用被认为是获取操作。

操作数

支持下列操作数：

id 要查询的实体（**process**、**task**、**project** 或 **zone**）的 ID。如果调用方用户的凭证没有特权，且正被查询的实体拥有不同的凭证，则操作将失败。如果未指定 *id*，则将返回一条错误消息。

示例

示例 1 显示当前的资源控制设置

以下示例显示当前 shell 所属任务的当前资源控制设置：

```
example$ ps -o taskid -p $$
TASKID
8
example$ prctl -i task 8
136150: /bin/ksh
NAME      PRIVILEGE      VALUE      FLAG      ACTION      RECIPIENT
task.max-cpu-time
          usage          8s
          system      18.4Es     inf      none      -
task.max-lwps
```

示例1 显示当前的资源控制设置 (续)

```

        usage          39
        system         2.15G      max  deny  -
project.max-contracts
        privileged     10.0K      -   deny  -
project.max-locked-memory
        usage          0B
        privileged     508MB     -   deny  -
project.max-port-ids
        privileged     8.19K      -   deny  -
project.max-shm-memory
        privileged     508MB     -   deny  -
project.max-shm-ids
        privileged     128        -   deny  -
project.max-msg-ids
        privileged     128        -   deny  -
project.max-sem-ids
        privileged     128        -   deny  -
project.max-crypto-memory
        usage          0B
privileged     508MB     -   deny  -
project.max-tasks
        usage          2
        system         2.15G      max  deny  -
project.max-lwps
        usage          39
        system         2.15G      max  deny  -
project.cpu-shares
        usage          1
        privileged     1          -   none  -
zone.max-shm-memory
        system         16.0EB     max  deny  -
zone.max-shm-ids
        system         16.8M      max  deny  -
zone.max-sem-ids
        system         16.8M      max  deny  -
zone.max-msg-ids
        system         16.8M      max  deny  -
zone.max-lwps
        system         2.15G      max  deny  -
zone.cpu-shares
        privileged     1          -   none  -
zone.max-locked-memory
        usage          0B
        privileged     508MB     -   deny  -

```

示例2 显示、替换和验证特定控制的值

以下示例显示、替换和验证某个现有项目上的特定控制的值：

```
example# prctl -n project.cpu-shares -i project group.staff
project: 10: group.staff
NAME    PRIVILEGE      VALUE    FLAG    ACTION          RECIPIENT
project.cpu-shares
  usage          1
  privileged     1        -    none          -
  system        65.5K    max    none          -
```

```
example# prctl -n project.cpu-shares -v 10 -r -i project group.staff
example# prctl -n project.cpu-shares -i project group.staff
project: 10: group.staff
NAME    PRIVILEGE      VALUE    FLAG    ACTION          RECIPIENT
project.cpu-shares
  usage          10
  privileged     10       -    none          -
  system        65.5K    max    none          -
```

示例3 调整资源

以下示例使用了 `project.max-locked-memory` 资源。

首先，使用 `id -p` 找出当前 shell 是哪个项目的成员：

```
/home/garfield> id -p
uid=77880(garfield) gid=10(staff) projid=10(group.staff)
```

使用目标项目，确定更改前的资源限制值：

```
/home/garfield> prctl -n project.max-locked-memory -i project \
group.staff
project 10: group.staff
project.max-locked-memory
  privileged     256MB    -    deny          -
  system        16.0EB  max    deny          -
```

current limit is 256 Megabytes.

然后，将目标项目的 `project.max-locked-memory` 限制调整为 300M 字节：

```
# prctl -n project.max-locked-memory -v 300M -r -i project group.staff
```

更改后的资源限制值将显示新值 300M 字节：

```
# prctl -n project.max-locked-memory -i project group.staff
project 10:group.staff
project.max-locked-memory
  usage          200MG
```

示例3 调整资源 (续)

```

privileged      300MB      -      deny      -
system          16.0EB     max     deny      -

```

示例4 修改项目的CPU上限

prctl 命令可使用 `project.cpu-cap` 资源控制（请参见 [resource_controls\(5\)](#)）设置和修改项目的CPU上限。（可在 `/etc/project` 文件中使用相同的资源控制。请参见 [project\(4\)](#)）。以下命令将修改CPU上限，将 `user.smith` 限制到三个CPU：

```
# prctl -r -t privileged -n project.cpu-cap -v 300 -i project user.smith
```

上面所使用的 `prctl -r` 选项用来动态更改项目或区域的CPU上限。例如，以下命令将上述命令中的上限设置更改为80%：

```
# prctl -r -t privileged -n project.cpu-cap -v 80 -i project user.smith
```

要删除CPU上限，请输入：

```
# prctl -x -n project.cpu-cap $$
```

示例5 修改区域的CPU上限

prctl 命令可使用 `zone.cpu-cap` 资源控制（请参见 [resource_controls\(5\)](#)）设置和修改区域的CPU上限。（可使用 `zonecfg(1M)` 命令操控相同的资源控制。）以下命令将修改CPU上限，将全局区域限制到CPU的80%：

```
# prctl -t privileged -n zone.cpu-cap -v 80 -i zone global
```

可使用以下命令将上限降低至50%：

```
# prctl -r -t privileged -n zone.cpu-cap -v 50 -i zone global
```

退出状态

将返回以下退出值：

- 0 成功。
- 1 遇到致命错误。
- 2 指定的命令行选项无效。

文件

`/proc/pid/*` 进程信息和控制文件

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

命令行语法是 "Committed"（已确定）。人可阅读的输出是 Uncommitted（未确定）。可解析的输出是 "Committed"（已确定）。

另请参见

[rctladm\(1M\)](#)、[zonecfg\(1M\)](#)、[setrctl\(2\)](#)、[rctlblk_get_local_action\(3C\)](#)、[project\(4\)](#)、[attribute](#)

附注

可在允许本地操作的资源控制块上设置的有效信号有

SIGABRT、SIGXRES、SIGHUP、SIGSTOP、SIGTERM 和 SIGKILL。此外，CPU 时间相关控制可发出 SIGXCPU 信号，文件大小相关控制可发送 SIGXFSZ 信号。

引用名	preap – 强制僵尸进程的父进程收割僵尸进程
用法概要	preap [-F] pid...
描述	<p>僵尸进程是其退出状态尚未由其父进程收割的进程。退出状态是经由 wait(3C)、waitid(2) 或 waitpid(3C) 系统调用收割的。在正常的系统运行过程中，可能会出现僵尸进程，但通常是比较短暂的。如果父进程没有收割其部分或所有子进程的退出状态就退出，则会出现这种情况。在这种情况下，这些子进程将重新认 <code>PID 1</code> 为父。请参见 init(1M)，它定期收割此类进程。</p> <p>不可靠的父进程无法花很长的时间退出，这样，就将僵尸进程留在了系统上。因为操作系统在进程成为僵尸进程之前几乎已毁坏了其所有组件，所以这些僵尸进程通常不会影响系统运行。不过，它们的确会消耗少量系统内存。</p> <p>preap 强制 <code>pid</code> 指定的进程的父代 waitid(3C) <code>pid</code>（如果 <code>pid</code> 表示一个僵尸进程）。</p> <p>在下列情况下，preap 会尝试阻止管理员轻率地收割即将由其父进程收割的子进程：</p> <ul style="list-style-type: none"> ▪ 该进程是 init(1M) 的子进程。 ▪ 父进程已停止并且可能会在再次允许它运行时拜访子进程。 ▪ 进程的僵死时间少于一分钟。
选项	<p>支持以下选项：</p> <p><code>-F</code> 强制父进程收割子进程，不管安全检查。</p>
操作数	<p>支持下列操作数：</p> <p><code>pid</code> 进程 ID 列表。</p>
用法	<p>使用 <code>-F</code> 标志时应谨慎。在一个被调试的进程上施加两个控制进程可能会导致混乱。仅当主控制进程（通常是调试器）已停止了被调试的进程，并且在应用 <code>proc</code> 工具的可疑时刻主控制进程未在执行任何操作，才能保证安全。</p>
退出状态	<p>输出被收割的每个目标进程的退出状态的 preap 返回以下退出值：</p> <p><code>0</code> 操作成功。</p> <p>非零 失败，例如没有这样的进程、权限遭拒或选项无效。</p>
属性	<p>有关下列属性的说明，请参见 attributes(5)：</p>

属性类型	属性值
可用性	system/extended-system-utilities

另请参见 [proc\(1\)](#)、[init\(1M\)](#)、[waitid\(2\)](#)、[wait\(3C\)](#)、[waitpid\(3C\)](#)、[proc\(4\)](#)、[attributes\(5\)](#)

警告

应慎用 `preap`，只有在管理员或开发人员确认僵尸进程不会由父进程进行收割时，才可使用。否则，应用 `preap` 可能会对父进程造成不可预测的损害。

引用名

print – shell built-in function to output characters to the screen or window

用法概要

/usr/bin/print print [-CReⁿprsv] [-f *format*] [-u *fd*] [*string*...]

ksh88 print [-Rnprsu [*n*]] [*arg*]...

ksh print [-CReⁿprsv] [-f *format*] [-u *fd*] [*string*...]

描述

ksh88 The shell output mechanism. With no options or if the - option is specified, the arguments that follow are printed on standard output as described by [echo\(1\)](#). If the - option is specified, anything that follows it is processed as an argument, even if it begins with a -.

/usr/bin/print, ksh By default, print writes each string operand to standard output and appends a NEWLINE character.

Unless, the -r, -R, or -f option is specified, each \ character in each string operand is processed specially as follows:

- \a Alert character.
- \b Backspace character.
- \c Terminate output without appending NEWLINE. The remaining string operands are ignored.
- \E Escape character (ASCII octal 033).
- \f FORM FEED character.
- \n NEWLINE character.
- \t Tab character.
- \v Vertical tab character.
- \\ Backslash character.
- \0x The 8-bit character whose ASCII code is the 1-, 2-, or 3-digit octal number *x*.

选项

ksh88 The following options are supported by ksh88:

- n Suppresses new-line from being added to the output.
- r -R Raw mode. Ignore the escape conventions of echo. The -R option prints all subsequent arguments and options other than -n.
- p Cause the arguments to be written onto the pipe of the process spawned with |& instead of standard output.

- s Cause the arguments to be written onto the history file instead of standard output.
- u [*n*] Specify a one digit file descriptor unit number *n* on which the output is placed. The default is 1.

/usr/bin/print, ksh

The following options are supported by /usr/man/print and ksh:

- e Unless -f is specified, process \ sequences in each string operand as described above. This is the default behavior.

If both -e and -r are specified, the last one specified is the one that is used.
- f *format* Write the string arguments using the format string *format* and do not append a NEWLINE. See [printf\(1\)](#) for details on how to specify format.

When the -f option is specified and there are more string operands than format specifiers, the format string is reprocessed from the beginning. If there are fewer string operands than format specifiers, then outputting ends at the first unneeded format specifier.
- n Do not append a NEWLINE character to the output.
- p Write to the current co-process instead of standard output.
- r
- R Do not process \ sequences in each string operand as described above.

If both -e and -r are specified, the last one specified is the one that is used.
- s Write the output as an entry in the shell history file instead of standard output.
- u *fd* Write to file descriptor number *fd* instead of standard output. The default value is 1.
- v Treat each string as a variable name and write the value in %B format. Cannot be used with -f
- C Treat each string as a variable name and write the value in %#B format. Cannot be used with -f.

退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 Output file is not open for writing.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[echo\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [printf\(1\)](#), [attributes\(5\)](#)

引用名 printenv – display environment variables currently set

用法概要 /usr/ucb/printenv [*variable*]

描述 printenv prints out the values of the variables in the environment. If a *variable* is specified, only its value is printed.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [csh\(1\)](#), [echo\(1\)](#), [sh\(1\)](#), [stty\(1\)](#), [tset\(1B\)](#), [attributes\(5\)](#), [environ\(5\)](#)

诊断 If a *variable* is specified and it is not defined in the environment, printenv returns an exit status of 1.

引用名 printf – write formatted output

用法概要

/usr/bin/printf printf *format* [*argument*]. . .

ksh printf *format* [*string*. . .]

描述

/usr/bin/printf The printf utility writes each string operand to standard output using *format* to control the output format.

操作数

/usr/bin/printf The following operands are supported by /usr/bin/printf:

- format* A string describing the format to use to write the remaining operands. The *format* operand is used as the *format* string described on the [formats\(5\)](#) manual page, with the following exceptions:
- A SPACE character in the format string, in any context other than a flag of a conversion specification, is treated as an ordinary character that is copied to the output.
 - A character in the format string is treated as a character, not as a SPACE character.
 - In addition to the escape sequences described on the [formats\(5\)](#) manual page (`\`, `\a`, `\b`, `\f`, `\n`, `\r`, `\t`, `\v`), `\ddd`, where *ddd* is a one-, two- or three-digit octal number, is written as a byte with the numeric value specified by the octal number.
 - The program does not precede or follow output from the `d` or `u` conversion specifications with blank characters not specified by the *format* operand.
 - The program does not precede output from the `o` conversion specification with zeros not specified by the *format* operand.
 - An additional conversion character, `b`, is supported as follows. The argument is taken to be a string that can contain backslash-escape sequences. The following backslash-escape sequences are supported:
 - the escape sequences listed on the [formats\(5\)](#) manual page (`\`, `\a`, `\b`, `\f`, `\n`, `\r`, `\t`, `\v`), which are converted to the characters they represent
 - `\0ddd`, where *ddd* is a zero-, one-, two- or three-digit octal number that is converted to a byte with the numeric value specified by the octal number
 - `\c`, which is written and causes printf to ignore any remaining characters in the string operand containing it, any remaining string operands and any additional characters in the *format* operand.

The interpretation of a backslash followed by any other sequence of characters is unspecified.

Bytes from the converted string are written until the end of the string or the number of bytes indicated by the precision specification is reached. If the precision is omitted, it is taken to be infinite, so all bytes up to the end of the converted string are written. For each specification that consumes an argument, the next argument operand is evaluated and converted to the appropriate type for the conversion as specified below. The *format* operand is reused as often as necessary to satisfy the argument operands. Any extra *c* or *s* conversion specifications are evaluated as if a null string argument were supplied; other extra conversion specifications are evaluated as if a zero argument were supplied. If the *format* operand contains no conversion specifications and *argument* operands are present, the results are unspecified. If a character sequence in the *format* operand begins with a *%* character, but does not form a valid conversion specification, the behavior is unspecified.

argument The strings to be written to standard output, under the control of *format*. The *argument* operands are treated as strings if the corresponding conversion character is *b*, *c* or *s*. Otherwise, it is evaluated as a C constant, as described by the ISO C standard, with the following extensions:

- A leading plus or minus sign is allowed.
- If the leading character is a single- or double-quote, the value is the numeric value in the underlying codeset of the character following the single- or double-quote.

If an argument operand cannot be completely converted into an internal value appropriate to the corresponding conversion specification, a diagnostic message is written to standard error and the utility does not exit with a zero exit status, but continues processing any remaining operands and writes the value accumulated at the time the error was detected to standard output.

ksh The *format* operands support the full range of ANSI C/C99/XPG6 formatting specifiers as well as additional specifiers:

%b Each character in the string operand is processed specially, as follows:

- \a* Alert character.
- \b* Backspace character.
- \c* Terminate output without appending NEWLINE. The remaining string operands are ignored.
- \E* Escape character (ASCII octal 033).
- \f* FORM FEED character.

<code>\n</code>	NEWLINE character.
<code>\t</code>	TAB character.
<code>\v</code>	Vertical tab character.
<code>\\</code>	Backslash character.
<code>\0x</code>	The 8-bit character whose ASCII code is the 1-, 2-, or 3-digit octal number <i>x</i> .
<code>%B</code>	Treat the argument as a variable name and output the value without converting it to a string. This is most useful for variables of type <code>-b</code> .
<code>%H</code>	Output string with characters <code><</code> , <code>&</code> , <code>></code> , <code>"</code> , and non-printable characters, properly escaped for use in HTML and XML documents.
<code>%P</code>	Treat <i>string</i> as an extended regular expression and convert it to a shell pattern.
<code>%q</code>	Output <i>string</i> quoted in a manner that it can be read in by the shell to get back the same string. However, empty strings resulting from missing string operands are not quoted.
<code>%R</code>	Treat <i>string</i> as a shell pattern expression and convert it to an extended regular expression.
<code>%T</code>	Treat <i>string</i> as a date/time string and format it. The <code>T</code> can be preceded by (<i>dformat</i>), where <i>dformat</i> is a date format as defined by the <code>date(1)</code> command.
<code>%Z</code>	Output a byte whose value is <code>0</code> .

When performing conversions of *string* to satisfy a numeric format specifier, if the first character of *string* is `"` or `'`, the value is the numeric value in the underlying code set of the character following the `"` or `'`. Otherwise, *string* is treated like a shell arithmetic expression and evaluated.

If a *string* operand cannot be completely converted into a value appropriate for that format specifier, an error occurs, but remaining *string* operands continue to be processed.

In addition to the format specifier extensions, the following extensions of ANSI C/C99/XPG6 are permitted in format specifiers:

- The escape sequences `\E` and `\e` expand to the escape character which is octal 033 in ASCII.
- The escape sequence `\cx` expands to CTRL-*x*.
- The escape sequence `\C[.name.]` expands to the collating element *name*.
- The escape sequence `\x{hex}` expands to the character corresponding to the hexadecimal value *hex*.
- The format modifier flag `=` can be used to center a field to a specified width. When the output is a terminal, the character width is used rather than the number of bytes.

- Each of the integral format specifiers can have a third modifier after width and precision that specifies the base of the conversion from 2 to 64. In this case, the # modifier causes *base#* to be prepended to the value.
- The # modifier can be used with the d specifier when no base is specified to cause the output to be written in units of 1000 with a suffix of one of k M G T P E.
- The # modifier can be used with the i specifier to cause the output to be written in units of 1024 with a suffix of one of Ki Mi Gi Ti Pi Ei.

If there are more *string* operands than format specifiers, the format string is reprocessed from the beginning. If there are fewer *string* operands than format specifiers, then *string* specifiers are treated as if empty strings were supplied, numeric conversions are treated as if 0 was supplied, and time conversions are treated as if now was supplied.

`/usr/bin/printf` is equivalent to ksh's `printf` built-in and `print -f`, which allows additional options to be specified.

用法

`/usr/bin/printf`

The `printf` utility, like the [printf\(3C\)](#) function on which it is based, makes no special provision for dealing with multi-byte characters when using the %c conversion specification. Applications should be extremely cautious using either of these features when there are multi-byte characters in the character set.

Field widths and precisions cannot be specified as *.

The %b conversion specification is not part of the ISO C standard; it has been added here as a portable way to process backslash escapes expanded in string operands as provided by the `echo` utility. See also the USAGE section of the [echo\(1\)](#) manual page for ways to use `printf` as a replacement for all of the traditional versions of the `echo` utility.

If an argument cannot be parsed correctly for the corresponding conversion specification, the `printf` utility reports an error. Thus, overflow and extraneous characters at the end of an argument being used for a numeric conversion are to be reported as errors.

It is not considered an error if an argument operand is not completely used for a c or s conversion or if a string operand's first or second character is used to get the numeric value of a character.

示例

`/usr/bin/printf`
Examples

示例 1 Printing a Series of Prompts

The following example alerts the user, then prints and reads a series of prompts:

```
example% printf "\aPlease fill in the following: \nName: "  
read name
```

示例 1 Printing a Series of Prompts (续)

```
printf "Phone number: "
read phone
```

示例 2 Printing a Table of Calculations

The following example prints a table of calculations. It reads out a list of right and wrong answers from a file, calculates the percentage correctly, and prints them out. The numbers are right-justified and separated by a single tab character. The percentage is written to one decimal place of accuracy:

```
example% while read right wrong ; do
    percent=$(echo "scale=1;($right*100)/($right+$wrong)" | bc)
    printf "%2d right\t%2d wrong\t(%%s%%)\n" \
        $right $wrong $percent
done < database_file
```

示例 3 Printing number strings

The command:

```
example% printf "%5d%4d\n" 1 21 321 4321 54321
```

produces:

```
    1  21
   3214321
54321  0
```

The *format* operand is used three times to print all of the given strings and that a 0 was supplied by `printf` to satisfy the last `%4d` conversion specification.

示例 4 Tabulating Conversion Errors

The following example tabulates conversion errors.

The `printf` utility tells the user when conversion errors are detected while producing numeric output. These results would be expected on an implementation with 32-bit twos-complement integers when `%d` is specified as the *format* operand:

Arguments	Standard	Diagnostic
5a	5	printf: 5a not completely converted
9999999999	2147483647	printf: 9999999999: Results too large
-9999999999	-2147483648	printf: -9999999999: Results too large
ABC	0	printf: ABC expected numeric value

The value shown on standard output is what would be expected as the return value from the function `strtol(3C)`. A similar correspondence exists between `%u` and `strtoul(3C)`, and `%e`, `%f` and `%g` and `strtod(3C)`.

示例 5 Printing Output for a Specific Locale

The following example prints output for a specific locale. In a locale using the ISO/IEC 646:1991 standard as the underlying codeset, the command:

```
example% printf "%d\n" 3 +3 -3 \'3 \"+3 "'-3"
```

produces:

3	Numeric value of constant 3
3	Numeric value of constant 3
-3	Numeric value of constant -3
51	Numeric value of the character '3' in the ISO/IEC 646:1991 standard codeset
43	Numeric value of the character '+' in the ISO/IEC 646:1991 standard codeset
45	Numeric value of the character '-' in the SO/IEC 646:1991 standard codeset

In a locale with multi-byte characters, the value of a character is intended to be the value of the equivalent of the `wchar_t` representation of the character.

If an argument operand cannot be completely converted into an internal value appropriate to the corresponding conversion specification, a diagnostic message is written to standard error and the utility does exit with a zero exit status, but continues processing any remaining operands and writes the value accumulated at the time the error was detected to standard output.

ksh Examples

The following examples illustrate the use of the ksh93 version of `printf`.

示例 6 Alternative floating point representation 1

The `printf` utility supports an alternative floating point representation (see `printf(3C)` entry for the `"%a"/"%A"`), which allows the output of floating-point values in a format that avoids the usual base16 to base10 rounding errors.

```
example% printf "%a\n" 2 3.1 NaN
```

produces:

```
0x1.00000000000000000000000000000000p+01
0x1.8cccccccccccccccccccccdp+01
nan
```

示例 7 Alternative floating point representation 2

The following example shows two different representations of the same floating-point value.

```
example% x=2 ; printf "%f == %a\n" x x
```

produces:

```
2.000000 == 0x1.00000000000000000000000000000000p+01
```

示例 8 Output of unicode values

The following command will print the EURO unicode symbol (code-point 0x20ac).

```
example% LC_ALL=en_US.UTF-8 printf "\u[20ac]\n"
```

produces:

```
<euro>
```

where <euro> represents the EURO currency symbol character.

示例 9 Convert unicode character to unicode code-point value

The following command will print the hexadecimal value of a given character.

```
example% export LC_ALL=en_US.UTF-8
```

```
example% printf "%x\n" "'<euro>"
```

where <euro> represents the EURO currency symbol character (code-point 0x20ac).

produces:

```
20ac
```

示例 10 Print the numeric value of an ASCII character

```
example% printf "%d\n" "'A"
```

produces:

```
65
```

示例 11 Print the language-independent date and time format

To print the language-independent date and time format, the following statement could be used:

```
example% printf "format" weekday month day hour min
```

For example,

```
$ printf format "Sunday" "July" 3 10 2
```

For American usage, format could be the string:

示例 11 Print the language-independent date and time format (续)

```
"%s, %s %d, %d:%.2d\n"
```

producing the message:

```
Sunday, July 3, 10:02
```

Whereas for EU usage, format could be the string:

```
"%1$s, %3$d. %2$s, %4$d:%5$.2d\n"
```

Note that the '\$' characters must be properly escaped, such as

```
"%1\$s, %3\$d. %2\$s, %4\$d:%5\$.2d\n"
```

 in this case

producing the message:

```
Sunday, 3. July, 10:02
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `printf`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, `LC_NUMERIC`, and `NLSPATH`.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/printf

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

ksh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

另请参见

[awk\(1\)](#), [bc\(1\)](#), [date\(1\)](#), [echo\(1\)](#), [ksh\(1\)](#), [printf\(3C\)](#), [strtod\(3C\)](#), [strtol\(3C\)](#), [strtoul\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [formats\(5\)](#), [standards\(5\)](#)

附注

Using format specifiers (characters following '%') which are not listed in the [printf\(3C\)](#) or this manual page will result in undefined behavior.

Using escape sequences (the character following a backslash ('\')) which are not listed in the [printf\(3C\)](#) or this manual page will result in undefined behavior.

Floating-point values follow C99, XPG6 and IEEE 754 standard behavior and can handle values the same way as the platform's `|long double|` datatype.

Floating-point values handle the sign separately which allows signs for values like NaN (for example, `-nan`), Infinite (for example, `-inf`) and zero (for example, `-0.0`).

引用名 prioctl – 显示或设置指定进程和 LWP 的调度参数

用法概要

```
prioctl -l
prioctl -d [-i idtype] [idlist]
prioctl -s [-c class] [class-specific options]
    [-i idtype] [idlist]
prioctl -e [-c class] [class-specific options] command
    [argument(s)]
```

描述

`prioctl` 命令显示或设置指定进程或 LWP 的调度参数。也可用于显示系统进程调度程序的当前配置信息，或以指定调度参数执行命令。

进程和 LWP 属于完全不同的类，对每个类应用单独的调度策略。当前支持的类包括实时类、分时类、交互式类、公平份额类和固定优先级类。这些类的特征和它们接受的类特定选项将在下面以下标题下的“用法”部分进行说明：**实时类**、**分时类**、**交互式类**、**公平份额类**和**固定优先级类**。拥有适当权限时，`prioctl` 命令可以更改与运行中的进程或 LWP 关联的类和其他调度参数。

缺省配置中，可运行的实时进程或 LWP 在其他所有进程之前运行。因此，对实时进程或 LWP 的不恰当使用会对系统性能有很严重的负面影响。

如果存在 *idlist*，它必须在命令行的最后，且列表中的各个元素必须以空格间隔。如果不存在 *idlist*，`pid`、`ppid`、`pgid`、`sid`、`taskid`、`class`、`uid`、`gid`、`projid` 或 `zoneid` 的 *idtype* 参数会分别指定 `prioctl` 命令本身的进程 ID、父进程 ID、进程组 ID、会话 ID、任务 ID、类、用户 ID、组 ID、项目 ID 或区域 ID。

以下命令

```
prioctl -d [-i idtype] [idlist]
```

显示 *idtype* 和 *idlist* 指定的进程的类和类特定调度参数。

以下命令

```
prioctl -s [-c class] [class-specific options] \
    [-i idtype] [idlist]
```

将指定进程的类和类特定调度参数设置为命令行给出的值。`-c class` 选项指定要设置的类。（有效 *class* 参数对于实时是 `RT`，对于分时是 `TS`，对于交互式是 `IA`，对于公平份额是 `FSS`，对于固定优先级是 `FX`。）

要设置的类特定参数由类特定选项指定，我们将在后面适当的标题下加以解释。如果省略 `-c class` 选项，*idtype* 和 *idlist* 必须指定一组全部位于相同类中的进程或 LWP，否则会出现错误。如果未指定类特定选项，进程的类特定参数设置为由 `-c class` 指定的类的缺省值（或者，如果还省略 `-c class` 选项，则设置为进程的当前类的缺省参数值）。

要使用 `priocntl` 更改进程或 LWP 的调度参数，调用 `priocntl` 的用户的实际或有效用户 ID（各自的 `groupID`）必须与接收进程或 LWP 的实际或有效用户 ID（各自的 `groupID`）相匹配，或者用户的有效用户 ID 必须是超级用户。这些是所有类都要强制执行的最小权限要求。设置类的进程或类特定调度参数时，单个类可以强加其他权限要求。

`idtype` 和 `idlist` 指定一组进程（无论是否有 LWP 列表）时，`priocntl` 会以特定于实现的顺序作用于集中的进程。如果 `priocntl` 遇到有关一个或多个目标进程的错误，它是否能够继续通过进程集合取决于错误的性质。

如果错误与权限有关，`priocntl` 会打印这条错误消息，然后继续通过进程集合，同时重置用户具有适当权限的所有目标进程的参数。如果 `priocntl` 遇到权限之外的错误，则不会继续通过进程集合，而是打印错误消息并立即退出。

存在一个特殊的 `sys` 调度类，目的在于调度某些特别系统进程（如交换程序进程）的执行。不是任何进程的类都可以更改为 `sys`。此外，包括在由 `idtype` 和 `idlist` 指定的进程集合中的 `sys` 类中的任何进程或 LWP 都会被 `priocntl` 忽略。例如，如果 `idtype` 是 `uid`，包含一个零的 `idlist` 将指定 UID 为 0 的所有进程，除了 `sys` 类中的进程和（如果使用 `-s` 选项更改参数）`init` 进程。

`init` 进程（进程 ID 1）是一个特例。为了 `priocntl` 命令可以更改 `init` 进程的类或其他调度参数，`idtype` 必须是 `pid`，`idlist` 必须只包含一个 1。`init` 进程可以分配到系统中配置的任何类，但是分时类几乎永远是最佳选择。非常不建议您选择其他项，有关更多信息，请参见《Oracle Solaris 管理：常见任务》。

以下命令

```
priocntl -e [-c class] [class-specific options] command \
    [argument...]
```

在命令行（`arguments` 是命令的参数）指定的类和调度参数执行特定命令。如果省略 `-c class` 选项，该命令会在用户当前的类中运行。

选项

支持以下选项：

-c class

指定要设置的 `class`。（有效 `class` 参数对于实时是 `RT`，对于分时是 `TS`，对于交互式是 `IA`，对于公平份额是 `FSS`，对于固定优先级是 `FX`。）如果指定的类尚未配置，会自动对其进行配置。

-d

显示与一组进程关联的调度参数。

-e

以与一组进程关联的类和调度参数执行指定命令。

-i idtype

该选项与 `idlist` 参数（如果有）共同指定 `priocntl` 命令要应用到的一个或多个进程或 LWP。对 `idlist` 的解释取决于 `idtype` 的值。如果在使用 `-d` 或 `-s` 选项时省略了 `-i idtype` 选项，将采用 `pid` 的缺省 `idtype`。

有效 *idtype* 参数和对 *idlist* 的相应解释如下：

-i all

prioctl 命令应用于所有的现有进程。不应指定 *idlist*（如果指定，则会忽略）。下面描述的权限限制仍然适用。

-i ctid

idlist 是进程合同 ID 列表。*prioctl* 命令应用于进程合同 ID 等于列表中 ID 的所有进程。

-i class

idlist 包含一个类名称（对于实时是 RT，对于分时是 TS，对于交互式是 IA，对于公平份额是 FSS，对于固定优先级是 FX）。*prioctl* 命令应用于指定类中的所有进程。

-i gid

idlist 是组 ID 的列表。*prioctl* 命令应用于有效组 ID 等于列表中的某个 ID 的所有进程。

-i pgid

idlist 是进程组 ID 的列表。*prioctl* 命令应用于指定进程组中的所有进程。

-i pid[/lwps]

idlist 是进程 ID（每个 ID 可能后跟正斜杠 (/)）的列表，以及以逗号分隔的 LWP ID 的列表。可通过连字符 (-) 分隔范围的第一项和最后一项来指示 LWP ID 范围。

-i ppid

idlist 是父进程 ID 的列表。*prioctl* 命令应用于父进程 ID 位于列表中的所有进程。

-i projid

idlist 是项目 ID 的列表。*prioctl* 命令应用于有效项目 ID 等于列表中的某个 ID 的所有进程。

-i sid

idlist 是会话 ID 的列表。*prioctl* 命令应用于指定会话中的所有进程。

-i taskid

idlist 是任务 ID 的列表。*prioctl* 命令应用于指定任务中的所有进程。

-i uid

idlist 是用户 ID 的列表。*prioctl* 命令应用于有效用户 ID 等于列表中的某个 ID 的所有进程。

-i zoneid

idlist 是区域 ID 的列表。*prioctl* 命令应用于有效区域 ID 等于列表中的某个 ID 的所有进程。

- l
显示系统中当前配置的类的列表，以及有关每个类的类特定信息。类特定信息的显示格式将在“用法”部分进行说明。
- s
设置与一组进程关联的调度参数。
- 用于设置实时参数的有效类特定选项包括：
- p *rtpri*
将指定进程和 LWP 的实时优先级设置为 *rtpri*。
- t *tqntm* [-r *res*]
将指定进程的时间量程设置为 *tqntm*。您可以按照下面的解释有选择地指定精度。
- q *tqsig*
将指定进程和 LWP 的实时时间量程信号设置为 *tqsig*。
- 用于设置分时参数的有效类特定选项包括：
- m *tsuprilim*
将指定进程和 LWP 的用户优先级限制设置为 *tsuprilim*。
- p *tsupri*
将指定进程和 LWP 的用户优先级设置为 *tsupri*。
- 用于设置交互式参数的有效类特定选项包括：
- m *iauprilim*
将指定进程和 LWP 的用户优先级限制设置为 *iauprilim*。
- p *iaupri*
将指定进程和 LWP 的用户优先级设置为 *iaupri*。
- 用于设置公共份额参数的有效类特定选项包括：
- m *fssuprilim*
将指定进程和 LWP 的用户优先级限制设置为 *fssuprilim*。
- p *fssupri*
将指定进程和 LWP 的用户优先级设置为 *fssupri*。
- 用于设置固定优先级参数的有效类特定选项包括：
- m *fxuprilim*
将指定进程和 LWP 的用户优先级限制设置为 *fxuprilim*。
- p *fxupri*
将指定进程和 LWP 的用户优先级设置为 *fxupri*。
- t *tqntm*
[-r *res*] 将指定进程和 LWP 的时间量程设置为 *tqntm*。您可以按照下面的解释有选择地指定精度。

用法

实时类

实时类为那些需要快速和确定响应的进程提供固定优先级优先调度策略，以及对调度优先级的绝对用户/应用程序控制。如果系统中配置了实时类，它会对系统上最高范围的调度优先级具有独占控制。这可以确保在属于任何其他类的任何进程之前为可运行实时进程提供 CPU 服务。

实时类拥有一定范围的实时优先级 (*rtpri*) 值，这些值可以分配给该类中的进程。实时优先级的范围从 0 到 *x*，其中 *x* 值是可配置的，可以通过以下命令为已经配置了实时调度程序的特定安装显示该值：

```
prioctl -l
```

实时调度策略是固定优先级策略。实时进程的调度优先级从不更改，除非用户/应用程序明确要求更改进程的 *rtpri* 值。

对于实时类中的进程，*rtpri* 值实际上相当于进程的调度优先级。*rtpri* 值完全决定实时进程相对应其类内的其他进程的调度优先级。*rtpri* 数值越大，表示优先级越高。因为实时类控制着系统中最高范围的调度优先级，可以保证具有最高 *rtpri* 值的可运行实时进程始终被选择在系统中的任何其他进程之前运行。

除了提供对优先级的控制之外，*prioctl* 还提供对分配给实时类中进程的时间量程长度的控制。时间量程值指定进程可运行的最长时间，假定进程未完成，或进入一个资源或事件等待状态 (*sleep*)。请注意，如果其他进程成为优先级更高的可运行进程，当前运行的进程可以在用完其全部时间量程前被取代。

以下命令

```
prioctl -d [-i idtype] [idlist]
```

显示由 *idtype* 和 *idlist* 指定的集合中每个实时进程的实时优先级、时间量程（毫秒精度）以及时间量程信号值。

-p、*-t* [*-r*] 和 *-q* 选项组合可与 *prioctl -s* 或 *prioctl -e* 共同用于实时类。如果省略某个选项且进程当前是实时的，关联参数不会受到影响。如果将进程的类从某些其他类更改为实时类时省略某个选项，关联参数会设置为缺省值。*rtpri* 的缺省值是 0，时间量程的缺省值取决于 *rtpri* 的值和系统配置；请参见 [rt_dptbl\(4\)](#)。

使用 *-t tqntm* 选项时，您可以选择性地使用 *-r res* 选项指定精度。（如果不指定精度，系统会假定其为毫秒精度。）如果指定 *res*，其必须是 1 到 1,000,000,000（包括）之间的正整数，使用的精度是 *res* 的倒数（以秒为单位）。例如，指定 *-t 10 -r 100* 会将精度设置为百分之一秒，得到的时间量程长度为 10/100 秒（十分之一秒）。尽管可以指定更高的精度（纳秒），但是时间量程长度会由系统向上舍入到系统时钟精度的下一个整数倍。将时间量程设置为零的请求，或量程大于（通常非常大）特定于实现的最大量程，都会导致错误。

实时时间量程信号可以用于通知失控实时进程有关时间量程消耗的情况。那些由实时时间量程信号监视的进程，在时间量程到期时会收到配置的信号。时间量程信号 *tqsig*

的缺省值 (0) 表示不发出任何信号。正值表示发出由值指定的信号。与 `kill(1)` 和其他应用于信号的命令相似，`-q tqsig` 选项也能处理以符号方式命名的信号，如 `XCPU` 或 `KILL`。

为了将进程的类（从任何其他类）更改为实时类，调用 `priocntl` 的用户必须拥有超级用户特权。为了更改实时进程的 `rtpri` 值或时间量程，调用 `priocntl` 的用户必须既是超级用户，或者该用户当前必须在实时类（作为实时进程运行的 `shell`）中，且具有与目标进程的实际或有效用户 ID 匹配的实际或有效用户 ID。

实时优先级、时间量程和时间量程信号将在 `fork(2)` 和 `exec(2)` 系统调用中继承。在 `exec(2)` 系统调用中通过用户定义的信号处理程序使用时间量程信号时，新映像必须在时间量程到期前安装合适的用户定义的信号处理程序。否则会导致不可预期的行为。

分时类

分时调度策略是为在具有各种 CPU 消耗特征的进程中公平有效地分配 CPU 资源而提供的。分时策略的目标是为交互式进程提供快速响应时间，为计算密集型 (CPU-bound) 作业提供很好的吞吐量，同时提供对调度的一定程度的用户/应用程序控制。

分时类拥有一定范围的分时用户优先级 (`tsupri`) 值，这些值可以分配到该类中的进程。用户优先级的范围是从 $-x$ 到 $+x$ ，其中 x 值是可配置的。可通过使用以下命令显示特定安装的范围：

```
priocntl -l
```

用户优先级的目的是提供对分时类中进程调度的一定程度的用户/应用程序控制。增加或降低分时类中进程的 `tsupri` 值会增加或降低进程的调度优先级。但是不保证具有较高 `tsupri` 值的分时进程会在具有较低 `tsupri` 值的进程之前运行。这是因为 `tsupri` 值只是用于决定分时进程调度优先级的一个因素。系统可以根据其他因素（如最近的 CPU 使用情况）来动态调整分时进程的内部调度优先级。

除了在系统范围内对用户优先级（以 `priocntl -l` 显示）作出限制外，还有每个进程的用户优先级限制 (`tsuprilim`)，它指定可为给定进程设置的最大 `tsupri` 值。

以下命令

```
priocntl -d [-i idtype] [idlist]
```

显示 `idtype` 和 `idlist` 指定的集合中每个分时进程的用户优先级和用户优先级限制。

任何分时进程都可降低自己的 `tsuprilim`（或具有相同用户 ID 的其他进程的该值）。只有具有超级用户特权的分时进程可以提高 `tsuprilim`。将进程的类从其他类更改为分时类时，需要超级用户权限才能将初始 `tsuprilim` 设置为一个大于零的值。

任何分时进程都可以将自己的 `tsupri`（或具有相同用户 ID 的其他进程的该值）设置为小于等于进程的 `tsuprilim` 的任何值。尝试将 `tsupri` 设置为高于 `tsuprilim`（和/或将 `tsuprilim` 设置为低于 `tsupri`）会导致 `tsupri` 被设置为等于 `tsuprilim`。

-m 和 -p 选项的任何组合都可以与 `prionctl -s` 或 `prionctl -e` 共同用于分时类。如果省略某个选项且进程当前是分时的，关联参数通常不会受到影响。例外情况是，省略 -p 选项且使用 -m 将 `tsuprilim` 设置为低于当前的 `tsupri` 时。这种情况下，`tsupri` 会设置为等于正在设置的 `tsuprilim`。如果将进程的类从某些其他类更改为实时类时省略某个选项，关联参数会设置为缺省值。`tsuprilim` 的缺省值是 0，`tsupri` 的缺省值是将其设置为等于正在设置的 `tsuprilim`。

分时用户优先级和用户优先级限制将在 `fork(2)` 和 `exec(2)` 系统调用中继承。

交互式类

交互式调度策略是为在具有各种 CPU 消耗特征的进程中公平有效地分配 CPU 资源而提供的，同时为用户交互提供良好的响应速度。交互式策略的目标是为交互式进程提供快速响应时间，为计算密集型 (CPU-bound) 作业提供很好的吞吐量。交互式类中进程优先级的更改方式与分时类相同，但是修改后的优先级可以继续调整，以为用户交互提供快速响应速度。

交互式用户优先级限制 `iaupri` 等效于 `tsupri`。交互式每进程的用户优先级 `iauprilim` 等效于 `tsuprilim`。

将为设置了 `iamode` (“交互模式”) 位的交互式类进程提供优先级增加值 10，在进行计算时 (即，每次调整进程的优先级时) 该值将分解为进程的用户模式优先级。此功能由 X 窗口系统使用，它将为在当前活动窗口内运行的那些进程设置此位，以便为进程提供更高优先级。

公平份额类

公平份额调度策略对项目中的系统 CPU 资源进行公平分配，而不考虑它们拥有的进程数量。每个项目都会得到一定的“份额”，用来控制他们对 CPU 资源的权利。系统会根据时间记录资源使用情况，这样，相对于其他项目，使用多的权利会被减少，而使用少的权利则会被增加。根据进程所有者的权利，在进程中调度 CPU 时间，与各个项目拥有的进程数量无关。

FSS 调度类支持每进程的用户优先级和用户优先级限制的概念，从而与分时调度程序兼容。公平份额调度程序尝试在整个用户优先级范围内提供平均分级效果。`fssupri` 值为负的进程接收时间分片的频率低于正常，而 `fssupri` 值为正的进程接收时间分片的频率高于正常。请注意，用户优先级不会影响份额。也就是说，更改进程的 `fssupri` 值不会影响其项目的整体 CPU 使用情况，后者通常与该项目相对于其他项目分配的份额量有关。

公平份额类中的进程优先级的更改方式与分时类相同。

固定优先级类

对于要求系统不动态调整调度优先级且用户/应用程序可以控制调度优先级的那些进程，固定优先级类为它们提供固定优先级优先调度策略。

缺省情况下，固定优先级类与分时类的调度优先级范围相同。固定优先级类拥有一定范围的固定优先级用户优先级 (`fxupri`) 值，这些值可以分配给类中的进程。用户优先级范围从 0 到 x ，其中 x 值是可配置的。可通过使用以下命令显示特定安装的范围：

```
prionctl -l
```


用户优先级的目的是提供对固定优先级类中进程调度的一定程度的用户/应用程序控制。对于固定优先级类中的进程，*fxupri* 值实际上等效于进程的调度优先级。*fxupri* 值可完全决定固定优先级进程相对于其类中其他进程的调度优先级。*fxupri* 数值越大，表示优先级越高。

除了在系统范围内对用户优先级（以 `prioctl -l` 显示）作出限制外，还有每个进程的用户优先级限制 (*fxuprilim*)，它指定可为给定进程设置的最大 *fxupri* 值。

任何固定优先级进程都可降低自己的 *fxuprilim*（或具有相同用户 ID 的其他进程的该值）。只有具有超级用户权限的进程才可以提高 *fxuprilim*。将进程的类从其他类更改为固定优先级类时，需要超级用户权限才能将初始 *fxuprilim* 设置为大于零的值。

任何固定优先级进程都可以将自己的 *fxupri*（或具有相同用户 ID 的其他进程的该值）设置为小于等于进程 *fxuprilim*。尝试将 *fxupri* 设置为高于 *fxuprilim*（和/或将 *fxuprilim* 设置为低于 *fxupri*），会导致 *fxupri* 设置为等于 *fxuprilim*。

除了提供对优先级的控制之外，`prioctl` 还提供对固定优先级类中分配给进程的时间量程长度的控制。时间量程值指定进程在让出 CPU 之前可运行的最长时间，假定进程未完成，或进入一个资源或事件等待状态 (*sleep*)。请注意，如果其他进程成为优先级更高的可运行进程，当前运行的进程可以在用完其全部时间量程前被取代。

任何 `-m`、`-p` 和 `-t` 选项组合都可以与 `prioctl -s` 或 `prioctl -e` 共同用于固定优先级类。如果省略某个选项且进程当前是固定优先级的，关联参数通常不会受到影响。例外情况是，省略 `-p` 选项且使用 `-m` 将 *fxuprilim* 设置为低于当前的 *fxupri* 时。这种情况下，*fxupri* 会设置为等于正在设置的 *fxuprilim*。如果将进程的类从某些其他类更改为固定优先级类时省略某个选项，关联参数会设置为缺省值。*fxuprilim* 的缺省值为 0。*fxupri* 缺省值是将其设置为等于正在设置的 *fxuprilim* 值。时间量程的缺省值由 *fxupri* 和系统配置决定。请参见 `fx_dptbl(4)`。

固定优先级类中的进程时间量程的更改方式与实时类相同。

固定优先级用户优先级、用户优先级限制和时间量程将在 `fork(2)` 和 `exec(2)` 系统调用中继承。

示例

下面是实时类的示例：

示例1 设置类

以下示例将由 *idtype* 和 *idlist* 选择的任意非实时进程的类设为实时，并将它们的实时优先级设置为缺省值 0。实时类中当前任意进程的实时优先级都不会受到影响。所有指定进程中的时间量程都设置为 1/10 秒。

```
example% prioctl -s -c RT -t 1 -r 10 -i idtype idlist
```

示例2 执行实时类中的命令

以下示例以实时优先级 15 和时间量程 20 毫秒执行实时类中的 *command*。

```
example% prioctl -e -c RT -p 15 -t 20 command
```

示例 3 以指定量程信号执行实时类中的命令

以下示例实时优先级 11、时间量程 250 毫秒和指定实时量程信号 SIGXCPU 执行实时类中的 *command*：

```
example% prioset -e -c RT -p 11 -t 250 -q XCPU command
```

下面是分时类的示例：

示例 4 设置非分时进程的类

以下示例将由 *idtype* 和 *idlist* 选择的任意非分时进程的类设置为分时，并将它们的用户优先级限制和用户优先级设置为 0。已经位于分时类中的进程不受影响。

```
example% prioset -s -c TS -i idtype idlist
```

示例 5 执行分时类中命令

以下示例在分时类中使用参数 *arguments* 执行 *command*，用户优先级限制为 0，用户优先级为 -15：

```
example% prioset -e -c TS -m 0 -p -15 command [arguments]
```

示例 6 执行固定优先级类中的命令

以下示例执行固定优先级类中的命令，用户优先级限制为 20，用户优先级为 10，时间量程为 250 毫秒：

```
example% prioset -e -c FX -m 20 -p 10 -t 250 command
```

示例 7 更改指定 LMP 的优先级

以下示例设置进程 500 中的 LWP 5 的用户优先级限制为 20，用户优先级为 15：

```
example% prioset -s -m 20 -p 15 500/5
```

退出状态

将返回以下退出值：

对于选项 -d、-l 和 -s：

0
操作成功。

1
错误条件。

对于选项 -e：

已执行命令的“退出状态”为“返回”时说明操作成功。否则，

1
不能以指定优先级执行命令。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
CSI	Enabled (已启用)

另请参见

[kill\(1\)](#)、[nice\(1\)](#)、[ps\(1\)](#)、[dispadm\(1M\)](#)、[exec\(2\)](#)、[fork\(2\)](#)、[prioctl\(2\)](#)、[fx_dptbl\(4\)](#)、[pro](#)

《Oracle Solaris 管理：常见任务》

诊断

`prioctl` 打印以下错误消息：

Process(es) not found

不存在指定的进程。

Specified processes from different classes

`-s` 选项正用于设置参数，`-c class` 选项不存在，并且指定了多个类中的进程。

Invalid option or argument

使用了无法识别或无效的选项或选项参数。

引用名 proc, pflags, pcred, pldd, psig, pstack, pfiles, pwdx, pstop, prun, pwait, ptime – proc 工具
用法概要

```
/usr/bin/pflags [-r] pid | core [/lwp] ...
/usr/bin/pcred [pid | core]...
/usr/bin/pcred [-u user/uid] [-g group/gid] [-G grouplist] pid...
/usr/bin/pcred -l login pid...
/usr/bin/pldd [-Fl] [pid | core]...
/usr/bin/psig [-n] pid...
/usr/bin/pstack [-F] pid | core [/lwp] ...
/usr/bin/pfiles [-Fn] pid...
/usr/bin/pwdx pid...
/usr/bin/pstop pid[/lwp] ...
/usr/bin/prun pid[/lwp] ...
/usr/bin/pwait [-v] pid...
/usr/bin/ptime [-Fm] [-p] pid...
/usr/bin/ptime [-m]command [arg]...
```

描述 proc 工具是用于执行 /proc 的功能的实用程序（请参见 [proc\(4\)](#)）。其中大多数工具都接受进程 ID (*pid*) 的列表。确实接受进程 ID 的工具还可以接受 /proc/*nnn* 作为进程 ID，因此可以使用 shell 扩展 /proc/* 来指定系统中的所有进程。

某些 proc 工具还可应用于核心文件（请参见 [core\(4\)](#)）。应用于核心文件的工具可以接受进程 ID 和（或）核心文件的名称的列表。

某些 proc 工具可在各个线程上执行。用户可以仅检查通过将 /*thread-id* 附加到进程 ID 或核心文件名称而选定的线程。可使用 - 和 , 分隔符选择多个线程。例如，/1,2,7-9 将检查线程 1、2、7、8 和 9。

请参见“警告”部分。

- pflags 为每个进程或每个进程中的指定 lwp 输出 /proc 跟踪标志、暂挂的和保留的信号以及其他 /proc 状态信息。
- pcred 输出或设置每个进程的凭证（有效的、实际的、已保存的 UID 和 GID）。
- pldd 列出链接到每个进程的动态库，包括使用 [dlopen\(3C\)](#) 显式连接的共享目标文件。另请参见 [ldd\(1\)](#)。
- psig 列出每个进程的信号操作和处理程序。请参见 [signal.h\(3HEAD\)](#)。
- pstack 为每个进程或每个进程中的指定 lwp 输出十六进制符号栈跟踪。

- pfiles** 报告每个进程中所有打开的文件的 **fstat(2)** 和 **fcntl(2)** 信息。对于网络端点，还会提供本地的（及对等方的，如果已连接）地址信息。对于套接字，还会提供套接字类型、套接字选项以及接收和发送缓冲区大小。此外，如果可从 `/proc/pid/path` 中获取文件路径信息，则还会报告该信息。这不一定是用来打开文件的同一名称。有关更多信息，请参见 **proc(4)**。
- pwdx** 输出每个进程的当前工作目录。
- pstop** 停止每个进程或指定的 **lwp**（**PR_REQUESTED** 停止）。
- prun** 设置正在运行的每个进程或指定的 **lwp**（与 **pstop** 相反）。
- pwait** 等待所有指定的进程终止。
- ptime** 与 **time(1)** 一样，对命令进行计时，但使用微观状态计数以获得可复现的精度。与 **time(1)** 不同的是，不会对命令的子代进行计时。

如果使用 `-p pid` 版本，将显示指定 *pid* 的计时统计信息的快照。

选项

支持以下常规选项：

- F** 强制。抓取目标进程，即使另一进程已掌握了控制权。
- n** （仅限 **psig** 和 **pfiles**）设置非详细模式。**psig** 显示信号处理程序地址，而不是名称。**pfiles** 不显示每个文件描述符的详细信息。相反，**pfiles** 将其输出限制为在进程向其每个文件描述符应用了 **fstat(2)** 的情况下将检索到的信息。
- r** （仅限 **pflags**）如果进程已停止，将显示其计算机寄存器。
- v** （仅限 **pwait**）详细。将结果报告到标准输出。

除了常规选项外，**pcred** 还支持以下选项：

- g group/gid** 将目标进程的实际的、有效的和已保存的组 ID (GID) 设置为指定值。
- G grouplist** 将目标进程的辅助 GID 设置为指定的组列表。应当以组名称 ID 的逗号分隔的列表形式指定辅助组。空列表将清除目标进程的辅助组列表。
- l login** 将目标进程的实际的、有效的和已保存的 UID 设置为指定登录的 UID。将目标进程的实际的、有效的和已保存的 GID 设置为指定登录的 GID。将辅助组列表设置为指定登录的辅助组列表。
- u user/uid** 将目标进程的实际的、有效的和已保存的用户 ID (UID) 设置为指定值。

除了常规选项外，**pldd** 还支持以下选项：

- l** 显示未解析的动态链接程序映射名称。

除了常规选项外，**ptime** 还支持以下选项：

-m 显示整套微观状态计数统计信息。

显示的字段如下所示：

real 挂钟时间。
user 用户级 CPU 时间。
sys 系统调用 CPU 时间。
trap 其他系统陷阱 CPU 时间。
tflt 文本缺页休眠时间。
dflt 数据缺页休眠时间。
kflt 内核缺页休眠时间。
lock 用户锁等待休眠时间。
slp 所有其他休眠时间。
lat CPU 延迟（等待）时间。
stop 停止时间。

-p pid 显示指定 *pid* 的计时统计信息的快照。

要设置其他进程的凭证，某个进程必须具有足够的特权，以将其用户和组 ID 更改为根据 [setuid\(2\)](#) 中制定的规则指定的用户和组 ID，且还必须具有足够的特权以控制目标进程。

用法

以下 `proc` 工具在检查其目标进程并报告结果时会停止目标进程：`pfiles`、`pldd` 和 `pstack`。进程被停止后将无法执行任何操作。因此，如果对 X 服务器进行检查的某个 `proc` 工具是在由 X 服务器控制的窗口中运行，则整个窗口系统将进入死锁状态，因为该 `proc` 工具将试图将其结果输出到一个无法刷新的窗口。在这种情况下，使用 [ssh\(1\)](#) 从其他系统登录并终止违例的 `proc` 工具将会消除死锁。

请参见“警告”部分。

使用 `-F` 标志时应谨慎。在一个被调试的进程上施加两个控制进程可能会导致混乱。仅当主控制进程（通常是调试器）已停止了被调试的进程，并且在应用 `proc` 工具的可疑时刻主控制进程未在执行任何操作，才能保证安全。

某些 `proc` 工具还可应用于核心文件，如上面的概要中所示。核心文件是进程状态的快照，由内核在使用信号终止进程之前生成，或者由 [gcore\(1\)](#) 实用程序生成。某些 `proc` 工具可能需要派生出与对内核进行了转储的进程对应的可执行程序的名称，或者与该进程相关联的共享库的名称。例如，[pstack\(1\)](#) 需要使用这些文件来提供符号表信息。如果该 `proc` 工具无法找到所需的可执行程序或共享库，某些符号信息将无法显

示。同样，如果在一个操作系统发行版上检查另一个不同的操作系统发行版中的核心文件，运行时链接编辑器调试接口 (`librtld_db`) 将无法初始化。在这种情况下，将无法获取共享库的符号信息。

退出状态

将返回以下退出值：

0 操作成功。

非零 出现错误。

文件

`/proc/*` 进程文件

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

人类可读的输出是 "Uncommitted"（未确定）。选项为 "Committed"（已确定）。

另请参见

[gcore\(1\)](#)、[ldd\(1\)](#)、[pargs\(1\)](#)、[pgrep\(1\)](#)、[pkill\(1\)](#)、[plimit\(1\)](#)、[pmap\(1\)](#)、[preap\(1\)](#)、[ps\(1\)](#)、[ptrace\(1\)](#)

警告

以下 `proc` 工具在检查其目标进程并报告结果时将停止目标进程：`pfiles`、`pldd` 和 `pstack`。不过，`pstack` 将停止整个进程，即使它是在单个线程上执行操作。

进程或线程被停止后将无法执行任何操作。在生产环境中停止某个频繁使用的进程或线程（即使仅停止很短时间）可能会导致严重的瓶颈，甚至导致这些进程或线程挂起，使得用户无法使用这些进程。某些数据库可能还会异常终止。因此，当使用上述 `proc` 工具跟踪一个数据库进程或线程时，负载过重的数据库服务器可能会挂起。因此，应避免在生产环境中停止 UNIX 进程或线程。

可通过发出 `/usr/bin/ps -efLL` 并在第一列中查找 "T" 来识别这些工具停止的进程或线程。请注意，某些进程（如 `sched`）缺省情况下大多数时候可显示 "T" 状态。

为网络文件系统上的锁定文件返回的进程 ID 可能没有意义。

引用名 prof – display profile data

用法概要 prof [-ChsVz] [-a | c | n | t] [-o | x] [-g | l] [-m *mdata*]
 [*prog*]

描述 The `prof` command interprets a profile file produced by the `monitor` function. The symbol table in the object file *prog* (`a.out` by default) is read and correlated with a profile file (`mon.out` by default). For each external text symbol the percentage of time spent executing between the address of that symbol and the address of the next is printed, together with the number of times that function was called and the average number of milliseconds per call.

选项 The mutually exclusive options `-a`, `-c`, `-n`, and `-t` determine the type of sorting of the output lines:

- a Sort by increasing symbol address.
- c Sort by decreasing number of calls.
- n Sort lexically by symbol name.
- t Sort by decreasing percentage of total time (default).

The mutually exclusive options `-o` and `-x` specify the printing of the address of each symbol monitored:

- o Print each symbol address (in octal) along with the symbol name.
- x Print each symbol address (in hexadecimal) along with the symbol name.

The mutually exclusive options `-g` and `-l` control the type of symbols to be reported. The `-l` option must be used with care; it applies the time spent in a static function to the preceding (in memory) global function, instead of giving the static function a separate entry in the report. If all static functions are properly located, this feature can be very useful. If not, the resulting report may be misleading.

Assume that `A` and `B` are global functions and only `A` calls static function `S`. If `S` is located immediately after `A` in the source code (that is, if `S` is properly located), then, with the `-l` option, the amount of time spent in `A` can easily be determined, including the time spent in `S`. If, however, both `A` and `B` call `S`, then, if the `-l` option is used, the report will be misleading; the time spent during `B`'s call to `S` will be attributed to `A`, making it appear as if more time had been spent in `A` than really had. In this case, function `S` cannot be properly located.

- g List the time spent in static (non-global) functions separately. The `-g` option function is the opposite of the `-l` function.
- l Suppress printing statically declared functions. If this option is given, time spent executing in a static function is allocated to the closest global function loaded before the static function in the executable. This option is the default. It is the opposite of the `-g` function and should be used with care.

The following options may be used in any combination:

- C Demangle C++ symbol names before printing them out.
- h Suppress the heading normally printed on the report. This is useful if the report is to be processed further.
- m *mdata* Use file *mdata* instead of *mon.out* as the input profile file.
- s Print a summary of several of the monitoring parameters and statistics on the standard error output.
- V Print *prof* version information on the standard error output.
- z Include all symbols in the profile range, even if associated with zero number of calls and zero time.

A single function may be split into subfunctions for profiling by means of the `MARK` macro. See [prof\(5\)](#).

环境变量

`PROFDIR` The name of the file created by a profiled program is controlled by the environment variable `PROFDIR`. If `PROFDIR` is not set, *mon.out* is produced in the directory current when the program terminates. If `PROFDIR=string`, *string/pid.progname* is produced, where *progname* consists of `argv[0]` with any path prefix removed, and *pid* is the process ID of the program. If `PROFDIR` is set, but null, no profiling output is produced.

文件

mon.out default profile file
a.out default namelist (object) file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities

另请参见

[gprof\(1\)](#), [exit\(2\)](#), [pcsample\(2\)](#), [profil\(2\)](#), [malloc\(3C\)](#), [malloc\(3MALLOC\)](#), [monitor\(3C\)](#), [attributes\(5\)](#), [prof\(5\)](#)

附注

If the executable image has been stripped and does not have the `.symtab` symbol table, `gprof` reads the global dynamic symbol tables `.dynsym` and `.SUNW_1dynsym`, if present. The symbols in the dynamic symbol tables are a subset of the symbols that are found in `.symtab`. The `.dynsym` symbol table contains the global symbols used by the runtime linker. `.SUNW_1dynsym` augments the information in `.dynsym` with local function symbols. In the case where `.dynsym` is found and `.SUNW_1dynsym` is not, only the information for the global symbols is available. Without local symbols, the behavior is as described for the `-a` option.

The times reported in successive identical runs may show variances because of varying cache-hit ratios that result from sharing the cache with other processes. Even if a program seems to be the only one using the machine, hidden background or asynchronous processes may blur the data. In rare cases, the clock ticks initiating recording of the program counter may *beat* with loops in a program, grossly distorting measurements. Call counts are always recorded precisely, however.

Only programs that call `exit` or return from `main` are guaranteed to produce a profile file, unless a final call to `monitor` is explicitly coded.

The times for static functions are attributed to the preceding external text symbol if the `-g` option is not used. However, the call counts for the preceding function are still correct; that is, the static function call counts are not added to the call counts of the external function.

If more than one of the options `-t`, `-c`, `-a`, and `-n` is specified, the last option specified is used and the user is warned.

`LD_LIBRARY_PATH` must not contain `/usr/lib` as a component when compiling a program for profiling. If `LD_LIBRARY_PATH` contains `/usr/lib`, the program will not be linked correctly with the profiling versions of the system libraries in `/usr/lib/libp`. See [gprof\(1\)](#).

Functions such as `mcount()`, `_mcount()`, `moncontrol()`, `_moncontrol()`, `monitor()`, and `_monitor()` may appear in the `prof` report. These functions are part of the profiling implementation and thus account for some amount of the runtime overhead. Since these functions are not present in an unprofiled application, time accumulated and call counts for these functions may be ignored when evaluating the performance of an application.

64-bit profiling

64-bit profiling may be used freely with dynamically linked executables, and profiling information is collected for the shared objects if the objects are compiled for profiling. Care must be applied to interpret the profile output, since it is possible for symbols from different shared objects to have the same name. If duplicate names are seen in the profile output, it is better to use the `-s` (summary) option, which prefixes a module id before each symbol that is duplicated. The symbols can then be mapped to appropriate modules by looking at the modules information in the summary.

If the `-a` option is used with a dynamically linked executable, the sorting occurs on a per-shared-object basis. Since there is a high likelihood of symbols from differed shared objects to have the same value, this results in an output that is more understandable. A blank line separates the symbols from different shared objects, if the `-s` option is given.

32-bit profiling

32-bit profiling may be used with dynamically linked executables, but care must be applied. In 32-bit profiling, shared objects cannot be profiled with `prof`. Thus, when a profiled, dynamically linked program is executed, only the *main* portion of the image is sampled. This means that all time spent outside of the *main* object, that is, time spent in a shared object, will not be included in the profile summary; the total time reported for the program may be less than the total time used by the program.

Because the time spent in a shared object cannot be accounted for, the use of shared objects should be minimized whenever a program is profiled with `prof`. If desired, the program should be linked to the profiled version of a library (or to the standard archive version if no profiling version is available), instead of the shared object to get profile information on the functions of a library. Versions of profiled libraries may be supplied with the system in the `/usr/lib/lib` directory. Refer to compiler driver documentation on profiling.

Consider an extreme case. A profiled program dynamically linked with the shared C library spends 100 units of time in some `libc` routine, say, `malloc()`. Suppose `malloc()` is called only from routine B and B consumes only 1 unit of time. Suppose further that routine A consumes 10 units of time, more than any other routine in the *main* (profiled) portion of the image. In this case, `prof` will conclude that most of the time is being spent in A and almost no time is being spent in B. From this it will be almost impossible to tell that the greatest improvement can be made by looking at routine B and not routine A. The value of the profiler in this case is severely degraded; the solution is to use archives as much as possible for profiling.

引用名	profiles – 列出和管理权限配置文件
用法概要	<pre>profiles [-l] [-a user ...] [-S repository] profiles -p profiles [-S repository] profiles -p profiles [-S repository] subcommand profiles -p profiles [-S repository] -f command_file profiles help</pre>
描述	<p>profiles 实用程序在本地文件名称服务或 LDAP 名称服务中的 prof_attr(4) 或 exec_attr(4) 数据库中创建和修改权限配置文件的配置。一个权限配置文件配置由一个配置文件名称和多个属性组成。</p> <p>profiles 子命令的以下概要适用于交互式用法：</p> <pre>profiles -p profile [-S repository] [subcommand]</pre> <p>profiles 命令在标准输出上输出已分配给您（或可选择指定的用户或角色名）的权限配置文件的名称。配置文件是用于枚举执行特定函数所需的命令和授权的一种绑定机制。如果进程是由特权命令解释程序启动的，则随每个可执行程序一起列出的还有进程运行时使用的进程属性，例如有效的用户和组 ID。请参见 pfexec(1) 手册页。配置文件可以包含 prof_attr(4) 中定义的其他配置文件。</p> <p>可以组合多个配置文件来构造合适的访问控制。分配配置文件时，授权将添加到现有的集合中。如果同一命令出现在多个配置文件中，则将第一次出现（根据配置文件的顺序确定）的命令用于进程属性设置。为方便起见，可指定一个通配符以匹配所有命令。</p> <p>特殊配置文件 "Stop"（停止）会截断对后续配置文件的评估。将不会对位于 "Stop"（停止）配置文件后的配置文件进行评估，也不会使用它们来查找其他命令。该配置文件可用来绕过 <code>/etc/security/policy.conf</code> 中以 <code>PROF_GRANTED</code> 键列出的配置文件和以 <code>AUTH_GRANTED</code> 键列出的授权。</p> <p>解释配置文件时，将从 user_attr(4) 装入配置文件列表。如果在 <code>/etc/security/policy.conf</code> 中定义了任何缺省配置文件（请参见 policy.conf(4)），则缺省配置文件的列表将添加到从 user_attr(4) 装入的列表中。prof_attr(4) 中的匹配项提供了授权列表，exec_attr(4) 中的匹配项提供了命令列表。</p>
属性	<p>使用 <code>-p</code> 选项调用时，可以管理指定配置文件的属性及其相关联的可执行文件的属性。不过，为维护系统完整性，此命令不能修改由 Solaris 维护的那些配置文件。这类配置文件只能在系统更新期间通过 pkg(1) 命令进行修改。</p> <p>另外，还可通过 pkg(1) 命令将其他配置文件声明为不可修改的。</p> <p>为防止特权升级，可根据用户的授权来限制属性值。至少要授予管理员 "Rights Management"（权限管理）配置文件。此外，要修改由委托授权控制的安全相关属</p>

性，必须向管理员授予 "Rights Delegation"（权限委托）配置文件。有关详细信息，请参见 [exec_attr\(4\)](#)、[prof_attr\(4\)](#) 及以下摘要。

属性值可以是简单字符串，也可以是简单字符串的逗号分隔列表。包含空格的简单字符串必须括在双引号中。

`profiles` 命令在 `profile` 上下文和 `command` 上下文中运行。`profile` 上下文是初始状态，在此状态下，可管理各种配置文件属性。下表概述了 `profile` 上下文中的属性：

Property Name	Value Type	Required Authorizations
<code>name</code>	simple	none
<code>auths</code>	list of simple	<code>solaris.auth.{assign/delegate}</code>
<code>profiles</code>	list of simple	<code>solaris.profile.{assign/delegate}</code>
<code>privs</code>	list of simple	<code>solaris.privilege.{assign/delegate}</code>
<code>limitpriv</code>	list of simple	<code>solaris.privilege.{assign/delegate}</code>
<code>defaultpriv</code>	list of simple	<code>solaris.privilege.{assign/delegate}</code>
<code>always_audit</code>	list of simple	<code>solaris.audit.assign</code>
<code>never_audit</code>	list of simple	<code>solaris.audit.assign</code>
<code>desc</code>	simple	none
<code>help</code>	simple	none
<code>pam_policy</code>	simple	<code>solaris.account.setpolicy</code>
<code>cmd</code>	simple/new context	none

通过指定 `cmd` 属性可进入 `command` 上下文。在 `command` 上下文中，可管理当前命令的属性。

下表概述了 `command` 上下文中的属性：

Property Name	Value Type	Required Authorizations
<code>id</code>	simple	none
<code>privs</code>	list of simple	<code>solaris.privilege.{assign/delegate}</code>
<code>limitprivs</code>	list of simple	<code>solaris.privilege.{assign/delegate}</code>
<code>euid</code>	simple	<code>solaris.profile.cmd.setuid</code>
<code>uid</code>	simple	<code>solaris.profile.cmd.setuid</code>
<code>egid</code>	simple	<code>solaris.group.{assign/delegate}</code>
<code>gid</code>	simple	<code>solaris.group.{assign/deleg</code>

以下列表介绍了可在 `profile` 上下文属性中指定的值。如以下列表中所指定，在属性与属性值之间需要有一个等号 (=)。

`always_audit`

用来将事件类指定为始终审计的审计标志。在登录和执行 `su` 命令时，只会应用该属性的第一个实例（在用户的 [user_attr\(4\)](#) 条目中或者在所分配的配置文件的有序列表中）。

`auths`

要添加到新配置文件中的一个或多个以逗号分隔的授权。如果在授权名称中使用了通配符 (*)，则名称必须括在双引号 (") 中。

cmd

可执行文件的全限定路径或星号(*)，后者用于指定所有命令。替换路径名中文件名组件的星号用于指示某个特定目录中的所有文件。

这是一个特殊属性，用于进入 `command` 上下文以管理命令的安全属性。

可以将数字 ID 或名称用于这些 ID。

id

此属性初始设置为由前面的 `cmd` 属性指定的值，但是可以修改。与 `select` 子命令一起使用时，可以克隆现有命令的属性以进行后续编辑。

pam_policy

要应用于用户的 PAM 策略。`pam_policy` 必须是 `pam.conf(4)` 格式文件的绝对路径名，或者是位于 `/etc/security/pam_policy` 的 `pam.conf(4)` 格式文件的文件名。有关更多信息，请参见 `pam_user_policy(5)`。

privs

应用于可执行进程的可继承集的特权集。缺省值为 `basic`。

limitprivs

应用于可执行进程的限制集的特权集。缺省值为 `all`。

euid

通过该命令执行的进程的有效用户 ID。

uid

通过该命令执行的进程的实际用户 ID。

egid

通过该命令执行的进程的有效组 ID。

gid

通过该命令执行的进程的实际组 ID。

defaultpriv

分配给用户的进程集的缺省特权集。在登录和执行 `su` 命令时，只会应用该属性的第一个实例（在用户的 `user_attr(4)` 条目中或者在所分配的配置文件的有序列表中）。

desc

新配置文件的说明。文本必须括在引号中。

help

新配置文件的帮助文件名称。帮助文件将被复制到 `/usr/lib/help/profiles/locale/<locale>` 目录中。其中，`<locale>` 是用户的语言环境的值，如果未指定任何内容，则为 `c`。只有在文件系统信息库中才适合指定此属性。

limitpriv

用户或者由用户启动的任何进程（不管是通过 `su(1M)` 还是以任何其他方式）可以获得的最大特权集。在登录和执行 `su` 命令时，只会应用该属性的第一个实例（在用户的 `user_attr(4)` 条目中或者在所分配的配置文件的有序列表中）。

name

配置文件的名称。名称的初始值是在命令行中使用 `-p` 选项指定的。如果名称发生改变，当前的配置文件属性将应用于新命名的配置文件。通过这种方式，可以克隆现有的配置文件以进行后续编辑。名称不得与某个现有的配置文件相同。

never_audit

用来将事件类指定为从不审计的审计标志。在登录和执行 `su` 命令时，只会应用该属性的第一个实例（在用户的 `user_attr(4)` 条目中或者在所分配的配置文件的有序列表中）。

privs

可使用 `pfexec(1)` 命令的 `P` 选项指定的特权集。

配置文件

要添加到新配置文件中的一个或多个以逗号分隔的辅助配置文件。

选项

支持以下选项：

-a

列出指定的系统信息库中的所有配置文件名称。如果未指定系统信息库，它将显示在 `nsswitch.conf(4)` 中为 `prof_attr` 配置的任何配置文件。

-f *command_file*

指定 `profiles` 命令文件的名称。*command_file* 是包含 `profiles` 的子命令的文本文件，一行一个。

-l

提供有关权限配置文件的的信息，并列出具命令及其特殊进程属性，如用户和组 ID。

-p *profile*

指定配置文件名称。

-S *repository*

有效的系统信息库为 `files` 和 `ldap`。*repository* 指定要更新的名称服务。缺省 *repository* 为文件。

子命令

使用 `-p` 选项调用时，可在命令行中或以交互方式提供子命令。可在命令行上指定以分号分隔的多个子命令，并且需要将整个子命令集括在引号中。未提供子命令意味着交互式会话，在此会话期间，可通过使用 `TAB` 键调用子命令的自动完成。

可使用 `add` 和 `select` 子命令选择一个特定的命令，从而使上下文更改为该命令的上下文。在交互式会话期间，`command` 上下文由提示字符串中的命令基名标识。可使用 `end` 和 `cancel` 子命令结束命令指定，从而将上下文恢复为 `profile` 上下文。

可导致破坏性操作或导致工作丢失的子命令具有一个强制执行操作的 **-F** 选项。如果输入来自终端设备，系统会在适当的时候提示用户。如果指定子命令时没有使用 **-F** 选项，这就有可能发生。其他情况下，不允许执行操作，并向标准错误写入一条诊断消息。

属性值可以是一个简单值，对于接受列表的属性来说，则是简单值的列表。支持以下子命令：

add cmd=*pathname*

在 **profile** 上下文中，开始指定给定的命令。上下文将更改为 **command** 类型。

add property-name=*property-value*

将指定值添加到当前属性值。此子命令仅可应用于接受列表的属性。

cancel

结束命令指定，并将上下文重新设置为 **profile**。放弃任何部分指定的资源。**cancel** 仅适用于 **command** 上下文。

clear *property name*

清除属性的值。

commit

将当前配置从内存提交到稳定存储器。必须提交配置才能使更改生效。在提交内存中的配置之前，可以使用 **revert** 子命令删除更改。在 **profiles** 会话完成时，会自动尝试 **commit** 操作。因为配置必须是正确的才能提交，因此该操作将自动执行 **verify**。

delete [-F]

从内存和稳定存储器中删除指定的配置文件。如果该配置文件是同一系统信息库中另一个配置文件的子配置文件，则不允许此操作。但是会提供包含该配置文件的配置文件的列表，用户可在删除该配置文件之前手动将其移除。使用 **-F** 选项强制执行操作：如果允许删除，其操作是即时的，会话将终止。

end

结束命令指定。此子命令仅适用于 **command** 上下文。**profiles** 命令验证是否完整指定了当前命令。如果是，当前命令将添加到内存中的配置（有关将其保存到稳定存储器的信息，请参见 **commit**），且上下文将恢复为 **profile** 上下文。如果指定不完整，它将发出相应的错误消息。

exit [-F]

退出 **profiles** 会话。如果需要，会自动尝试 **commit**。还可使用 EOF 字符退出 **profiles**。可使用 **-F** 选项强制执行操作。

export [-f *output-file*]

将配置输出至标准输出。使用 **-f** 选项可将配置输出至输出文件。此选项以适合在命令文件选项中使用的格式生成输出。

help [*usage*] [*subcommands*] [*properties*] [*<subcommand.>*] [*<properties>*]

输出常规帮助或有关特定主题的帮助。

`info [property-name]`

显示有关当前配置文件或指定属性的信息。

`remove cmd=fullpath`

从配置文件中删除指定的命令。此子命令仅在 `profile` 上下文中有效。

`remove [-F] cmd`

从配置文件中删除所有命令。如果未使用 `-F` 选项，则需要确认。此子命令仅在 `profile` 上下文中有效。

`remove property-name=property-value`

从属性中删除指定的值。这仅可应用于接受列表的属性。

`revert [-F]`

将配置恢复到上次提交时的状态。可使用 `-F` 选项强制执行操作。

`select cmd=fullpath`

选择与给定路径名标准相匹配的命令以进行修改。此子命令仅适用于 `profile` 上下文。

`set property-name=property-value`

将给定属性名称设置为给定值。某些属性（如 `name` 和 `desc`）仅在 `profile` 上下文中有效，而某些属性则仅在 `command` 上下文中有效。此子命令同时适用于 `profile` 和 `command` 上下文。

`verify`

检验当前配置是否正确：

- 是否指定了必需的属性。
- 各个值是否对每个关键字都有效。
- 用户是否有权指定这些值。

示例

示例 1 使用 `profiles` 命令

`profiles` 命令的输出具有以下格式：

```
example% profiles tester01 tester02
tester01 : Audit Management, All Commands
tester02 : Device Management, All Commands
example%
```

示例 2 使用 `list` 选项

```
example% profiles -l tester01 tester02
tester01 :
  Audit Management:
    /usr/sbin/audit          euid=root
    /usr/sbin/auditconfig   euid=root   egid=sys
  All Commands:
    *
tester02 :
```

示例 2 使用 list 选项 (续)

```
Device Management:
  /usr/bin/allocate:      euid=root
  /usr/bin/deallocate:   euid=root
All Commands
*
```

example%

示例 3 创建新的配置文件

以下示例在 LDAP 中创建了一个新的 User Manager 配置文件。新配置文件的说明是 "Manage users and groups"，分配的授权是 solaris.user.manage。分配的辅助配置文件是 Mail Management。帮助文件名是 RtUserMgmt.html。

```
example% profiles -p "User Manager" -S ldap
profiles:User Manager> set desc="Manage users and groups"
profiles:User Manager> set help=RtUserMgmt.html
profiles:User Manager> set auths=solaris.user.manage
profiles:User Manager> set profiles="Mail Management"
profiles:User Manager> exit
```

示例 4 显示有关当前配置的信息

以下命令显示有关 User Manager 配置文件的信息：

```
example% profiles -p "User Manager" -S ldap info
name=User Manager
desc=Manage users and groups
auths=solaris.user.manage
profiles=Mail Management
help=RtUserMgmt.html
```

示例 5 删除配置文件

以下命令从 LDAP 中删除 User Manager 配置文件：

```
example% profiles -p "User Manager" -S ldap delete -F
```

示例 6 修改配置文件

以下示例修改 LDAP 中的 User Manager 配置文件。新配置文件的说明是 "Manage world"，新的授权分配是 solaris.user.* 授权，新的辅助配置文件分配是 All。

```
example% profiles -p "User Manager" -S ldap
profiles:User Manager> set desc="Manage world"
profiles:User Manager> set auths="solaris.user.*"
profiles:User Manager> set profiles=All
profiles:User Manager> exit
```


示例 7 创建 exec_attr 数据库条目

以下命令为 LDAP 中的 User Manager 配置文件创建一个新的 exec_attr 条目。将添加 /usr/bin/cp 条目。此命令的有效用户 ID 为 0，有效组 ID 为 0。

```
example% profiles -p "User Manager" -S ldap
profiles:User Manager> add cmd=/usr/bin/cp
profiles:User Manager:cp> set euid=0
profiles:User Manager:cp> set egid=0
profiles:User Manager:cp> end
profiles:User Manager> exit
example%
```

示例 8 删除 exec_attr 数据库条目

以下示例将从 LDAP 中的 User Manager 配置文件删除一个 exec_attr 数据库条目。将删除为命令 /usr/bin/cp 指定的条目。

```
example% profiles -p "User Manager" -S ldap
profiles:User Manager> remove cmd=/usr/bin/cp
profiles:User Manager> exit
example%
```

示例 9 修改 exec_attr 数据库条目

以下命令为 LDAP 中的 User Manager 配置文件修改 exec_attr 数据库条目的属性。将 /usr/bin/cp 条目修改为以实际用户 ID 0 和实际组 ID 0 执行。

```
example% profiles -p "User Manager" -S ldap
profiles:User Manager> select cmd=/usr/bin/cp
profiles:User Manager:cp> clear euid
profiles:User Manager:cp> clear egid
profiles:User Manager:cp> set uid=0
profiles:User Manager:cp> set gid=0
profiles:User Manager:cp> end
profiles:User Manager> exit
example%
```

退出状态

将返回以下退出值：

0
成功完成。

1
出现错误。

文件

```
/etc/security/exec_attr
/etc/security/prof_attr
/etc/user_attr
/etc/security/policy.conf
```

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[auths\(1\)](#)、[pfexec\(1\)](#)、[pkg\(1\)](#)、[roles\(1\)](#)、[getprofattr\(3C\)](#)、[auth_attr\(4\)](#)、[exec_attr\(4\)](#)、[nsswitch](#)

引用名	projects – 输出用户的项目成员身份
用法概要	projects [-dv] [user] projects -l [projectname [projectname]...]
描述	projects 命令在标准输出上输出调用方用户（或另外指定的用户）所属的项目。每个用户都属于在 project(4) 文件中指定的以及可能在关联的 NIS 映射和项目信息的 LDAP 数据库指定的某组项目。
选项	支持以下选项： -d 仅输出缺省项目。 -l 输出有关每个 projectname 项目的详细信息。如果未给定任何 projectname，则会输出所有对象的信息。 -v 输出项目说明以及项目名称。
操作数	支持下列操作数： projectname 显示指定项目的信息。 user 显示指定用户的项目成员身份。
示例	示例1 显示指定用户的成员身份 example\$ projects paul default beatles wings example\$ projects ringo default beatles example\$ projects -d paul beatles
退出状态	将返回以下退出值： 0 成功完成。 1 执行过程中发生致命错误。 2 指定的命令行选项无效。
文件	/etc/project 包含此计算机的有效项目定义的本地数据库。
属性	有关以下属性的说明，请参见 attributes(5) ：

属性类型	属性值
可用性	system/core-os
稳定性	请参见下文。

调用为 Committed（已确定）。人可阅读的输出是 Uncommitted（未确定）。

另请参见

[getdefaultproj\(3PROJECT\)](#)、[getprojent\(3PROJECT\)](#)、[project\(4\)](#)、[attributes\(5\)](#)

引用名	ps – report process status
用法概要	ps [-aAcdefjHLLPyZ] [-g <i>grplist</i>] [-h <i>lgrplist</i>] [-n <i>namelist</i>] [-o <i>format</i>]... [-p <i>proclist</i>] [-s <i>sidlist</i>] [-t <i>term</i>] [-u <i>uidlist</i>] [-U <i>uidlist</i>] [-G <i>gidlist</i>] [-z <i>zonelist</i>]
描述	<p>The <code>ps</code> command prints information about active processes. Without options, <code>ps</code> prints information about processes that have the same effective user ID and the same controlling terminal as the invoker. The output contains only the process ID, terminal identifier, cumulative execution time, and the command name. Otherwise, the information that is displayed is controlled by the options.</p> <p>Some options accept lists as arguments. Items in a list can be either separated by commas or else enclosed in quotes and separated by commas or spaces. Values for <i>proclist</i> and <i>grplist</i> must be numeric.</p> <p>The <code>ps</code> command tries to determine whether it is called natively or using the command syntax expected by <code>ps(1B)</code>. In the latter case, the <code>ps</code> command behaves exactly as described in <code>ps(1B)</code>.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -a Lists information about all processes most frequently requested: all those except session leaders and processes not associated with a terminal. -A Lists information for all processes. Identical to -e, below. -c Prints information in a format that reflects scheduler properties as described in <code>prIOCNTL(1)</code>. The -c option affects the output of the -f and -l options, as described below. -d Lists information about all processes except session leaders. -e Lists information about every process now running. <p>When the -e option is specified, options -z, -t, -u, -U, -g, -G, -p, -g, -s and -a options have no effect.</p> <ul style="list-style-type: none"> -f Generates a full listing. (See below for significance of columns in a full listing.) -g <i>grplist</i> Lists only process data whose group leader's ID number(s) appears in <i>grplist</i>. (A group leader is a process whose process ID number is identical to its process group ID number.)

- G *gidlist*
Lists information for processes whose real group ID numbers are given in *gidlist*. The *gidlist* must be a single argument in the form of a blank- or comma-separated list.
- h *lgrplist*
Lists only the processes homed to the specified *lgrplist*. Nothing is listed for any invalid group specified in *lgrplist*.
- H
Prints the home lgroup of the process under an additional column header, LGRP.
- j
Prints session ID and process group ID.
- l
Generates a long listing. (See below.)
- L
Prints information about each light weight process (*lwp*) in each selected process. (See below.)
- n *namelist*
Specifies the name of an alternative system *namelist* file in place of the default. This option is accepted for compatibility, but is ignored.
- o *format*
Prints information according to the format specification given in *format*. This is fully described in DISPLAY FORMATS. Multiple -o options can be specified; the format specification is interpreted as the space-character-separated concatenation of all the *format* option-arguments.
- p *proclist*
Lists only process data whose process ID numbers are given in *proclist*.
- P
Prints the number of the processor to which the process or lwp is bound, if any, under an additional column header, PSR.
- s *sidlist*
Lists information on all session leaders whose IDs appear in *sidlist*.
- t *term*
Lists only process data associated with *term*. Terminal identifiers are specified as a device file name, and an identifier. For example, *term/a*, or *pts/0*.
- u *uidlist*
Lists only process data whose effective user ID number or login name is given in *uidlist*. In the listing, the numerical user ID is printed unless you give the -f option, which prints the login name.

-U *uidlist*

Lists information for processes whose real user ID numbers or login names are given in *uidlist*. The *uidlist* must be a single argument in the form of a blank- or comma-separated list.

-y

Under a long listing (`-l`), omits the obsolete `F` and `ADDR` columns and includes an `RSS` column to report the resident set size of the process. Under the `-y` option, both `RSS` and `SZ` (see below) is reported in units of kilobytes instead of pages.

-z *zonelist*

Lists only processes in the specified zones. Zones can be specified either by name or ID. This option is only useful when executed in the global zone.

-Z

Prints the name of the zone with which the process is associated under an additional column header, `ZONE`. The `ZONE` column width is limited to 8 characters. Use `ps -eZ` for a quick way to see information about every process now running along with the associated zone name. Use

```
ps -eo zone,uid,pid,ppid,time,comm,...
```

to see zone names wider than 8 characters.

The following options are used by the `/usr/ucb/ps` command (see [ps\(1B\)](#)). They are supported in `/usr/bin/ps`, allowing the latter to emulate UCB behavior. The UCB options do not use a hyphen. You cannot mix these options with the options described above.

r

Restricts output to running and runnable processes.

S

Displays accumulated CPU time used by this process and all of its reaped children.

v

Displays a version of the output containing virtual memory. This includes fields `SIZE`, `%CPU`, `%MEM`, and `RSS`, described below.

w

Uses a wide output format, that is, 132 columns rather than 80. If the option letter is repeated, that is, `-ww`, this option uses arbitrarily wide output. This information is used to decide how much of long commands to print. *Note:* The wide output option can be viewed only by a superuser or the user who owns the process.

x

Includes processes with no controlling terminal.

num

A process number may be given, in which case the output is restricted to that process. This option must be supplied last.

Many of the options shown are used to select processes to list. If any are specified, the default list is ignored and `ps` selects the processes represented by the inclusive OR of all the selection-criteria options.

Display Formats

Under the `-f` option, `ps` tries to determine the command name and arguments given when the process was created by examining the user block. Failing this, the command name is printed, as it would have appeared without the `-f` option, in square brackets.

The column headings and the meaning of the columns in a `ps` listing are given below; the letters `f` and `l` indicate the option (`full` or `long`, respectively) that causes the corresponding heading to appear; `all` means that the heading always appears. *Note:* These two options determine only what information is provided for a process; they do not determine which processes are listed.

F(l)

Flags (hexadecimal and additive) associated with the process. These flags are available for historical purposes; no meaning should be currently ascribed to them.

S(l)

The state of the process:

O

Process is running on a processor.

S

Sleeping: process is waiting for an event to complete.

R

Runnable: process is on run queue.

T

Process is stopped, either by a job control signal or because it is being traced.

W

Waiting: process is waiting for CPU usage to drop to the CPU-caps enforced limits.

Z

Zombie state: process terminated and parent not waiting.

UID (f,l)

The effective user ID number of the process (the login name is printed under the `-f` option).

PID(all)

The process ID of the process (this datum is necessary in order to kill a process).

PPID(f,l)

The process ID of the parent process.

C(f,l)

Processor utilization for scheduling (obsolete). Not printed when the `-c` option is used.

CLS(f,l)

Scheduling class. Printed only when the `-c` option is used.

PRI(l)

The priority of the process. Without the `-c` option, higher numbers mean lower priority. With the `-c` option, higher numbers mean higher priority.

NI(l)

Nice value, used in priority computation. Not printed when the `-c` option is used. Only processes in the certain scheduling classes have a nice value.

ADDR(l)

The memory address of the process, `0` unless running with all privilege.

SZ(l)

The total size of the process in virtual memory, including all mapped files and devices, in pages. See [pagesize\(1\)](#).

WCHAN(l)

The address of an event for which the process is sleeping. Only visible when running with all privilege, otherwise it is `0`. To determine if a process is sleeping, check the `S` column.

STIME(f)

The starting time of the process, given in hours, minutes, and seconds. (A process begun more than twenty-four hours before the `ps` inquiry is executed is given in months and days.)

TTY(all)

The controlling terminal for the process (the message, `?`, is printed when there is no controlling terminal).

TIME(all)

The cumulative execution time for the process.

LTIME(all)

The execution time for the lwp being reported.

CMD(all)

The command name (the full command name and its arguments, up to a limit of 80 characters, are printed under the `-f` option).

The following two additional columns are printed when the `-j` option is specified:

PGID

The process ID of the process group leader.

SID

The process ID of the session leader.

The following two additional columns are printed when the `-L` option is specified:

LWP

The lwp ID of the lwp being reported.

NLWP

The number of lwps in the process (if `-f` is also specified).

Under the `-L` option, one line is printed for each lwp in the process and the time-reporting fields `STIME` and `LTIME` show the values for the lwp, not the process. A traditional single-threaded process contains only one lwp.

A process that has exited and has a parent, but has not yet been waited for by the parent, is marked `<defunct>`.

-o format

The `-o` option allows the output format to be specified under user control.

The format specification must be a list of names presented as a single argument, blank- or comma-separated. Each variable has a default header. The default header can be overridden by appending an equals sign and the new text of the header. The rest of the characters in the argument is used as the header text. The fields specified are written in the order specified on the command line, and should be arranged in columns in the output. The field widths are selected by the system to be at least as wide as the header text (default or overridden value). If the header text is null, such as `-o user=`, the field width is at least as wide as the default header text. If all header text fields are null, no header line is written.

The following names are recognized in the POSIX locale:

user

The effective user ID of the process. This is the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

ruser

The real user ID of the process. This is the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

group

The effective group ID of the process. This is the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

rgroup

The real group ID of the process. This is the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.

pid

The decimal value of the process ID.

ppid

The decimal value of the parent process ID.

pgid

The decimal value of the process group ID.

pcpu

The ratio of CPU time used recently to CPU time available in the same period, expressed as a percentage. The meaning of “recently” in this context is unspecified. The CPU time available is determined in an unspecified manner.

vsz

The total size of the process in virtual memory, in kilobytes.

nice

The decimal value of the system scheduling priority of the process. See [nice\(1\)](#).

etime

In the POSIX locale, the elapsed time since the process was started, in the form:

[[dd-]hh:]mm:ss

where

dd

is the number of days

hh

is the number of hours

mm

is the number of minutes

ss

is the number of seconds

The *dd* field is a decimal integer. The *hh*, *mm* and *ss* fields is two-digit decimal integers padded on the left with zeros.

time

In the POSIX locale, the cumulative CPU time of the process in the form:

[dd-]hh:mm:ss

The *dd*, *hh*, *mm*, and *ss* fields is as described in the *etime* specifier.

tty

The name of the controlling terminal of the process (if any) in the same format used by the [who\(1\)](#) command.

comm

The name of the command being executed (`argv[0]` value) as a string.

args

The command with all its arguments as a string. The implementation might truncate this value to the field width; it is implementation-dependent whether any further truncation occurs. It is unspecified whether the string represented is a version of the argument list as it

was passed to the command when it started, or is a version of the arguments as they might have been modified by the application. Applications cannot depend on being able to modify their argument list and having that modification be reflected in the output of `ps`. The Solaris implementation limits the string to 80 bytes; the string is the version of the argument list as it was passed to the command when it started.

The following names are recognized in the Solaris implementation:

`f`

Flags (hexadecimal and additive) associated with the process.

`s`

The state of the process.

`c`

Processor utilization for scheduling (obsolete).

`uid`

The effective user ID number of the process as a decimal integer.

`ruid`

The real user ID number of the process as a decimal integer.

`gid`

The effective group ID number of the process as a decimal integer.

`rgid`

The real group ID number of the process as a decimal integer.

`projid`

The project ID number of the process as a decimal integer.

`project`

The project ID of the process as a textual value if that value can be obtained; otherwise, as a decimal integer.

`zoneid`

The zone ID number of the process as a decimal integer.

`zone`

The zone ID of the process as a textual value if that value can be obtained; otherwise, as a decimal integer.

`sid`

The process ID of the session leader.

`taskid`

The task ID of the process.

`class`

The scheduling class of the process.

-
- `pri`
The priority of the process. Higher numbers mean higher priority.
- `opri`
The obsolete priority of the process. Lower numbers mean higher priority.
- `lwp`
The decimal value of the lwp ID. Requesting this formatting option causes one line to be printed for each lwp in the process.
- `nlwp`
The number of lwps in the process.
- `psr`
The number of the processor to which the process or lwp is bound.
- `pset`
The ID of the processor set to which the process or lwp is bound.
- `addr`
The memory address of the process.
- `osz`
The total size of the process in virtual memory, in pages.
- `wchan`
The address of an event for which the process is sleeping (if `-`, the process is running).
- `stime`
The starting time or date of the process, printed with no blanks.
- `rss`
The resident set size of the process, in kilobytes. The `rss` value reported by `ps` is an estimate provided by `proc(4)` that might underestimate the actual resident set size. Users who wish to get more accurate usage information for capacity planning should use `pmap(1) -x` instead.
- `pmem`
The ratio of the process's resident set size to the physical memory on the machine, expressed as a percentage.
- `fname`
The first 8 bytes of the base name of the process's executable file.
- `ctid`
The contract ID of the process contract the process is a member of as a decimal integer.
- `lgrp`
The home lgroup of the process.

Only `comm` and `args` are allowed to contain blank characters; all others, including the Solaris implementation variables, are not.

The following table specifies the default header to be used in the POSIX locale corresponding to each format specifier.

Format Specifier	Default Header	Format Specifier	Default Header
args	COMMAND	ppid	PPID
comm	COMMAND	rgroup	RGROUP
etime	ELAPSED	ruser	RUSER
group	GROUP	time	TIME
nice	NI	tty	TT
pcpu	%CPU	user	USER
pgid	PGID	vsz	VSZ
pid	PID		

The following table lists the Solaris implementation format specifiers and the default header used with each.

Format Specifier	Default Header	Format Specifier	Default Header
addr	ADDR	projid	PROJID
c	C	project	PROJECT
class	CLS	psr	PSR
f	F	rgid	RGID
fname	COMMAND	rss	RSS
gid	GID	ruid	RUID
lgrp	LGRP	s	S
lwp	LWP	sid	SID
nlwp	NLWP	stime	STIME
opri	PRI	taskid	TASKID
osz	SZ	uid	UID
pmem	%MEM	wchan	WCHAN

Format	Default	Format	Default
Specifier	Header	Specifier	Header
pri	PRI	zone	ZONE
ctid	CTID	zoneid	ZONEID

示例

示例 1 Using `ps` Command

The command:

```
example% ps -o user,pid,ppid=MOM -o args
```

writes the following in the POSIX locale:

```
USER  PID  MOM  COMMAND
helene 34   12  ps -o uid,pid,ppid=MOM -o args
```

The contents of the `COMMAND` field need not be the same due to possible truncation.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `ps`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, and `NLSPATH`.

COLUMNS

Override the system-selected horizontal screen size, used to determine the number of text columns to display.

退出状态

The following exit values are returned:

0
Successful completion.

>0
An error occurred.

文件

`/dev/pts/*`
`/dev/term/*`
terminal ("tty") names searcher files
`/etc/passwd`
UID information supplier
`/proc/*`
process control files

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

ATTRIBUTE TYPE	ATTRIBUTE VALUE
CSI	Enabled (see USAGE)
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[kill\(1\)](#), [lgrpinfo\(1\)](#), [nice\(1\)](#), [pagesize\(1\)](#), [pmap\(1\)](#), [priocntl\(1\)](#), [who\(1\)](#), [ps\(1B\)](#), [getty\(1M\)](#), [proc\(4\)](#), [ttysrch\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [resource_controls\(5\)](#), [standards\(5\)](#), [zones\(5\)](#)

附注

Things can change while `ps` is running. The snapshot it gives is true only for a split-second, and it might not be accurate by the time you see it. Some data printed for defunct processes is irrelevant.

If no options to select processes are specified, `ps` reports all processes associated with the controlling terminal. If there is no controlling terminal, there is no report other than the header.

`ps -ef` or `ps -o stime` might not report the actual start of a `tty` login session, but rather an earlier time, when a `getty` was last respawned on the `tty` line.

On prior releases the `ADDR` and `WCHAN` fields might have contained the kernel memory address of the process and/or event it was waiting on. These fields are now always `0` unless requested by a process running with all privilege. The values can still be obtained using the `: : ps` and `: : thread dcmds` within `mdb`.

`ps` is CSI-enabled except for login names (usernames).

引用名	ps – display the status of current processes
用法概要	<code>/usr/ucb/ps [-] [aceglnrSuUvwx] [-t <i>term</i>] [<i>num</i>]</code>
描述	<p>The <code>ps</code> command displays information about processes. Normally, only those processes that are running with your effective user ID and are attached to a controlling terminal (see termio(7I)) are shown. Additional categories of processes can be added to the display using various options. In particular, the <code>-a</code> option allows you to include processes that are not owned by you (that do not have your user ID), and the <code>-x</code> option allows you to include processes without controlling terminals. When you specify both <code>-a</code> and <code>-x</code>, you get processes owned by anyone, with or without a controlling terminal. The <code>-r</code> option restricts the list of processes printed to running and runnable processes.</p> <p><code>ps</code> displays in tabular form the process ID, under PID; the controlling terminal (if any), under TT; the cpu time used by the process so far, including both user and system time, under TIME; the state of the process, under S; and finally, an indication of the COMMAND that is running.</p> <p>The state is given by a single letter from the following:</p> <ul style="list-style-type: none"> O Process is running on a processor. S Sleeping. Process is waiting for an event to complete. R Runnable. Process is on run queue. Z Zombie state. Process terminated and parent not waiting. T Traced. Process stopped by a signal because parent is tracing it.
选项	<p>The following options must all be combined to form the first argument. The <code>ps</code> command accepts the arguments without the leading <code>(-)</code> for historical reasons</p> <ul style="list-style-type: none"> -a Includes information about processes owned by others. -c Displays the command name rather than the command arguments. -e Displays the environment as well as the arguments to the command. -g Displays all processes. Without this option, <code>ps</code> only prints interesting processes. Processes are deemed to be uninteresting if they are process group leaders. This normally eliminates top-level command interpreters and processes waiting for users to login on free terminals. -l Displays a long listing, with fields F, PPID, CP, PRI, NI, SZ, RSS, and WCHAN as described below. -n Produces numerical output for some fields. In a user listing, the USER field is replaced by a UID field. -r Restricts output to running and runnable processes.

- S Displays accumulated CPU time used by this process and all of its reaped children.
 - t *term* Lists only process data associated with the terminal, *term*. Terminal identifiers may be specified in one of two forms: the device's file name (for example, `tty04` or `term/14`) or, if the device's file name starts with `tty`, just the digit identifier (for example, `04`).
 - u Displays user-oriented output. This includes fields `USER`, `%CPU`, `%MEM`, `SZ`, `RSS`, and `START` as described below.
 - U Obsolete. This option no longer has any effect. It causes `ps` to exit without printing the process listing.
 - v Displays a version of the output containing virtual memory. This includes fields `SIZE`, `%CPU`, `%MEM`, and `RSS`, described below.
 - w Uses a wide output format, that is, 132 columns rather than 80. If the option letter is repeated, that is, `-ww`, this option uses arbitrarily wide output. This information is used to decide how much of long commands to print. *Note:* The wide output option can be viewed only by a superuser or the user who owns the process.
 - x Includes processes with no controlling terminal.
- num* A process number may be given, in which case the output is restricted to that process. This option must be supplied last.

Display Formats

Fields that are not common to all output formats:

- `USER` Name of the owner of the process.
- `%CPU` CPU use of the process. This is a decaying average over up to a minute of previous (real) time.
- `NI` Process scheduling increment (see [getpriority\(3C\)](#)).
- `SIZE` The total size of the process in virtual memory, including all mapped files and devices, in kilobyte units.
- `SZ` Same as `SIZE`.
- `RSS` Real memory (resident set) size of the process, in kilobyte units.
- `UID` Numerical user-ID of process owner.
- `PPID` Numerical ID of parent of process.
- `CP` Short-term CPU utilization factor (used in scheduling).
- `PRI` The priority of the process (higher numbers mean lower priority).

- START** The starting time of the process, given in hours, minutes, and seconds. A process begun more than 24 hours before the `ps` inquiry is executed is given in months and days.
- WCHAN** The address of an event for which the process is sleeping (if blank, the process is running).
- %MEM** The ratio of the process's resident set size to the physical memory on the machine, expressed as a percentage.
- F** Flags (hexadecimal and additive) associated with the process. These flags are available for historical purposes; no meaning should be currently ascribed to them.

A process that has exited and has a parent, but has not yet been waited for by the parent, is marked `<defunct>` ; otherwise, `ps` tries to determine the command name and arguments given when the process was created by examining the user block.

文件

`/dev/tty*`

`/etc/passwd` UID information supplier

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	compatibility/ucb

另请参见

[kill\(1\)](#), [ps\(1\)](#), [whodo\(1M\)](#), [getpriority\(3C\)](#), [proc\(4\)](#), [attributes\(5\)](#), [termio\(7I\)](#)

附注

Things can change while `ps` is running. The picture `ps` gives is only a close approximation to the current state. Some data printed for defunct processes is irrelevant.

引用名	ptree – 输出进程树
用法概要	<code>/usr/bin/ptree [-a] [-c] [-z zone] [pid user]...</code>
描述	<code>ptree</code> 实用程序输出包含指定 <i>pid</i> 或 <i>user</i> 的进程树，其中，子进程相对于其各自的父进程缩进排列。如果某个参数全部由数字组成，则会将其视为进程 ID，否则会将其假定为用户登录名。缺省设置为所有进程。
选项	支持以下选项： <ul style="list-style-type: none"> -a 所有。输出所有进程，包括进程 0 的子进程。 -c 合同。输出包括父/子关系在内的进程合同成员身份。请参见 process(4)。此选项隐式指定 -a 选项。 -z zone 区域。仅输出 <i>zone</i> 中指定的进程。每个区域 ID 可以指定为区域名称，也可以指定为数字区域 ID。 <p style="margin-left: 40px;">此选项仅在全局区域中执行时有效。</p>
操作数	支持下列操作数： <p><i>pid</i> 进程 ID 或进程 ID 列表。<code>ptree</code> 还可接受 <code>/proc/nnn</code> 作为进程 ID，因此可使用 shell 扩展 <code>/proc/*</code> 来指定系统中的所有进程。</p> <p><i>user</i> 用户名或用户名列表。显示其有效的用户 ID 与给定 ID 相匹配的进程。</p>
示例	<p>示例 1 使用 <code>ptree</code></p> <p>以下示例输出与命令名 <code>ssh</code> 相匹配的进程的进程树（包括进程 0 的子进程）：</p> <pre>\$ ptree -a 'pgrep ssh' 1 /usr/sbin/init 100909 /usr/lib/ssh/sshd 569150 /usr/lib/ssh/sshd 569157 /usr/lib/ssh/sshd 569159 -ksh 569171 bash 569173 /bin/ksh 569193 bash</pre>
退出状态	将返回以下退出值： <ul style="list-style-type: none"> 0 操作成功。 非零 出现错误。
文件	<code>/proc/*</code> 进程文件
属性	有关下列属性的说明，请参见 attributes(5) ：

属性类型	属性值
可用性	system/core-os
接口稳定性	请参见下文。

人类可阅读的输出是 "Uncommitted"（未确定），选项是 "Committed"（已确定）。

另请参见

[gcore\(1\)](#)、[ltd\(1\)](#)、[pargs\(1\)](#)、[pgrep\(1\)](#)、[pkill\(1\)](#)、[plimit\(1\)](#)、[pmap\(1\)](#)、[preap\(1\)](#)、[proc\(1\)](#)、[p](#)

引用名 pvs – 显示动态目标文件的内部版本信息

用法概要 pvs [-Cdlnorsv] [-I *index-expr*] [-N *name*] *file...*

描述 pvs 实用程序显示 ELF 文件中包含的任何内部版本信息。通常情况下，这些文件是动态可执行文件和共享目标文件，并且有可能是可重定位的目标文件。此版本信息可以是以下两个类别中的一种：

- 版本定义
- 版本依赖项

版本定义描述了可通过 ELF 文件提供的接口。每个版本定义都与该文件提供的一组全局符号相关联。在创建某个文件期间，可以通过链接编辑器使用 `-M` 选项和相关的 `mapfile` 指令将版本定义分配给该文件。有关更多详细信息，请参见《[链接程序和库指南](#)》。

版本依赖项描述了动态目标文件对任何共享目标文件依赖项的版本定义的绑定需求。如果构建的动态目标文件引用了某个共享目标文件，链接编辑器将在动态目标文件内记录相应的信息，指示该共享目标文件是一个依赖项。在运行时必须满足此依赖性。如果共享目标文件也包含**版本定义**，则还会在正在创建的动态目标文件中记录用来满足动态目标文件的全局符号需求的那些版本定义。在进程初始化时，运行时链接程序将**版本依赖项**用作一种验证方法，用以验证用来构建进程的动态目标文件的接口需求。

选项 支持以下选项。如果 `-d` 或 `-r` 选项均未指定，则会同时启用两者。

`-C` 取消改编 C++ 符号名。

`-d` 输出版本定义信息。

`-I index-expr` 通过特定的版本索引或索引范围来限定要检查的版本。例如，可使用以下命令显示某个目标文件中索引为 3 的版本：

```
example% pvs -I 3 filename
```

index-expr 可以是指定一个特定版本的单个非负整数值，如上一示例中所示。另外，*index-expr* 还可以包含两个这样的值，以冒号(:)分隔，指示版本的范围。以下示例显示文件中的版本 3、4 和 5：

```
example% pvs -I 3:5 filename
```

在指定索引范围时，可以省略第二个值以指示文件中的最后一项。例如，以下语句列出从第 10 个到最后的所有版本：

```
example% pvs -I 10: filename
```

有关匹配选项 (`-I`、`-N`) 的更多信息，请参见“[匹配选项](#)”。

`-l` 输出由于版本更新而从全局绑定降级为本机绑定的符号。按照惯例，这些符号条目位于 `.symtab` 部分中，介于表示输出文件的 `FILE` 符号和表示用于生成输出文件的第一个输入文件的 `FILE` 符号之间。这

些降级的符号条目分配有虚构的版本定义 `_LOCAL_`。如果已对文件执行了剥除操作（请参见 `strip(1)`），或者无法确定符号条目约定，则不会输出任何降级的符号。

隐式使用 `-l` 选项会启用 `-s` 选项。

`-n` 标准化版本定义信息。缺省情况下，将显示目标文件内的所有版本定义。不过，版本定义可以继承其他的版本定义。在标准化形式下，只会显示每个继承列表的标题。

`-N name` 与 `-d` 选项一起使用时，`-N` 仅输出给定版本定义 *name* 及其继承的任何版本定义的信息。

与 `-r` 选项一起使用时，`-N` 仅输出给定依赖项文件 *name* 的信息。可以限定依赖项文件中的某个特定版本，方法是将版本放在文件名之后的括号中：

```
example% pvs -N 'dependency (version)' filename
```

有关匹配选项（`-I`、`-N`）的更多信息，请参见“匹配选项”。

`-o` 创建单行版本定义输出。缺省情况下，文件、版本定义和任何符号输出均以缩进形式显示，以便于进行人工检查。此选项在每个输出行前附加文件和版本定义名称，在使用自动工具进行分析时，此选项显得尤为有用。

`-r` 输出版本依赖项性（需求）信息。

`-s` 输出与每个版本定义关联的符号。显示数据项的大小（字节）以及由目标文件定义的版本中的任何数据符号。

`-v` 详细输出。指示任何弱版本定义以及任何版本定义继承。与 `-N` 和 `-d` 选项一起使用时，还会显示基本版本定义的继承。与 `-s` 选项一起使用时，还会显示版本符号定义。

操作数

支持以下操作数。

file 要显示其内部版本信息的 ELF 文件。

用法

匹配选项

`-I` 和 `-N` 选项统称为**匹配选项**。这些选项用于通过索引或名称来缩小要检查的版本范围。

在一个给定的 `pvs` 调用中，可以混合使用任意数量和类型的匹配选项。在这种情况下，`pvs` 显示与所使用的任何匹配选项匹配的所有版本的超集。使用此功能，可以用于指定每个项目的最简便形式来选择复杂的项目分组。

示例

示例1 显示版本定义

以下示例显示 `libelf.so.1` 的版本定义：

```
% pvs -d /lib/libelf.so.1
    libelf.so.1;
    SUNW_1.1
```

示例2 创建单行显示

可以使用 `-n` 和 `-o` 选项创建标准的单行显示，这适用于创建映射文件版本控制指令。

```
% pvs -don /lib/libelf.so.1
/lib/libelf.so.1 -    SUNW_1.1;
```

示例3 显示版本需求

以下示例显示 `ldd` 和 `pvs` 的版本需求：

```
% pvs -r /usr/bin/ldd /usr/bin/pvs
/usr/bin/ldd:
    libelf.so.1 (SUNW_1.1);
    libc.so.1 (SUNW_1.1);
/usr/bin/pvs:
    libelf.so.1 (SUNW_1.1);
    libc.so.1 (SUNW_1.1);
```

示例4 确定依赖项符号版本

以下示例显示 `ldd` 命令预期在运行时从其中找到 `printf` 函数的共享目标文件及其属于的版本：

```
% pvs -ors /usr/bin/ldd | grep ' printf'
/usr/bin/ldd - libc.so.1 (SYSVABI_1.3): printf;
```

示例5 确定特定版本中的所有依赖项符号

可使用 `-N` 选项获取某个依赖项中属于某个特定版本的所有符号的列表。确定 `ldd` 将从 `libc.so.1` 的版本 `SYSVABI_1.3` 中找到的符号：

```
% pvs -s -N 'libc.so.1 (SYSVABI_1.3)' /usr/bin/ldd

libc.so.1 (SYSVABI_1.3):
    _exit;
    strstr;
    printf;
    __fpstart;
    strncmp;
    lseek;
    strcmp;
    getopt;
```


示例5 确定特定版本中的所有依赖项符号 (续)

```

execl;
close;
fflush;
wait;
strerror;
putenv;
sprintf;
getenv;
open;
perror;
fork;
strlen;
geteuid;
access;
setlocale;
atexit;
fprintf;
exit;
read;
malloc;

```

请注意，`ldd` 使用的符号的具体列表在各个 Solaris 发行版之间可能会更改。

示例6 按索引显示所定义的基本版本

按照惯例，由目标文件定义的基本全局版本具有该目标文件的名称。例如，`pvs` 的基本版本的名称为 `'pvs'`。任何目标文件的基本版本始终是版本索引 1。因此，可使用 `-I` 选项显示任何目标文件的基本版本，而不需要指定其名称：

```

% pvs -v -I 1 /usr/bin/pvs
pvs [BASE];

```

退出状态

如果没有找到所请求的版本信息，则返回非零值。否则，将返回 0 值。

如果以下任何内容属实，都将判定为无法找到版本信息：

- 指定了 `-d` 选项，并且没有找到版本定义。
- 指定了 `-r` 选项，并且没有找到版本需求。
- 没有指定 `-d` 和 `-r` 选项，并且没有找到版本定义或版本需求。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	developer/base-developer-utilities

另请参见

[elfdump\(1\)](#)、[ld\(1\)](#)、[ldd\(1\)](#)、[strip\(1\)](#)、[elf\(3ELF\)](#)、[attributes\(5\)](#)

《链接程序和库指南》

- 引用名** pwd – return working directory name
- 用法概要** /usr/bin/pwd
- 描述** The pwd utility writes an absolute path name of the current working directory to standard output.
- Both the Bourne shell, [sh\(1\)](#), and the Korn shells, [ksh\(1\)](#) and [ksh88\(1\)](#), also have a built-in pwd command.
- 环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of pwd: LANG, LC_ALL, LC_MESSAGES, and NLSPATH.
- 退出状态** The following exit values are returned:
- 0 Successful completion.
 - >0 An error occurred.
- If an error is detected, output will not be written to standard output, a diagnostic message will be written to standard error, and the exit status will not be 0.
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [cd\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [shell_builtins\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断 pwd: cannot determine current directory!

Consult your network administrator.

附注 If you move the current directory or one above it, pwd may not give the correct response. Use the [cd\(1\)](#) command with a full path name to correct this situation.

引用名 radadrngen – code generator

用法概要

```
/usr/bin/radadrngen [-N] spec.xml
/usr/bin/radadrngen [-N] -c dir [-r [ -m ] [ -s ]] spec.xml
/usr/bin/radadrngen [-N] -j dir [-i] spec.xml
/usr/bin/radadrngen [-N] {-o docbook | docbook-man | man | rmDOC} spec.xml
/usr/bin/radadrngen -d baseline.xml spec.xml
```

描述 The radadrngen command is the ADR IDL processing tool. Its primary purpose is to generate language bindings for the rad server and the various rad client environments. It can also generate documentation, and audit changes to interfaces for consistency with their versions.

All invocations validate the given `spec.xml` against the ADR schema.

选项 The following options are supported:

Common Options

- N
Do not limit what types can be nullable. This restriction is in place to facilitate mapping to languages where certain types are not naturally nullable. This option relaxes this restriction for uses where it is both permissible and advantageous for all types to be nullable.

C Definition Options

- c *dir*
Generate C definitions for the types and interfaces defined in the input file. Files will be created under *dir*.
- r
Generate C server-side definitions that reference entry points using the prescribed entry-point naming scheme.
- m
Do not generate separate definitions for inherited interfaces. By default, new definitions are generated for inherited interfaces, replicating the definitions in the inherited interface.
- s
Generate C stubs for entry points referenced by the definitions created by the -r option.

Java Definition Options

- j *dir*
Generate Java interfaces for the types and interfaces defined in the input file. The Java source tree will be created under *dir*.
- i
Generate concrete implementation classes for the interfaces created for structured types.

Documentation Generation Options

- o *format*
Generate documentation, in the given format, for the interface definition in `spec.xml`. Documentation can be improved by decorating the various parts of the interface definition with *summary* and *doc* elements.

Valid formats:

docbook
 standard docbook format

docbook-man
 man page content in docbook format, a precursor to man format, below

man
 a man page, in [nt]roff format

rmdoc
 spec.xml, with the *summary* and *doc* elements removed

Interface Comparison
 Options

-d baseline.xml
 Compare the interfaces defined by baseline.xml and spec.xml and verify that the version of the modified interface is consistent with the differences found. An error message is displayed if the versions are inconsistent.

操作数

The following operand must be specified on the command line:

spec.xml
 The path to the API specification for which type and interface definitions must be generated.

文件

/usr/share/lib/xml/rng/radadr.rng.1
 The core ADR RelaxNG schema definition.

/usr/share/lib/xml/rng/radadr-doc.rng.1
 The ADR RelaxNG schema definition for documentation elements.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/management/rad
Interface Stability	Private

另请参见

[rad\(1M\)](#), [attributes\(5\)](#)

引用名 ranlib – convert archives to random libraries

用法概要 ranlib *archive*

描述 The ranlib utility was used in SunOS 4.x to add a table of contents to archive libraries, which converted each archive to a form that could be linked more rapidly. This is no longer needed, as the [ar\(1\)](#) command automatically provides all the functionality ranlib used to provide.

This script is provided as a convenience for software developers who need to maintain Makefiles that are portable across a variety of operating systems.

退出状态 ranlib has exit status 0.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities

另请参见 [ar\(1\)](#), [ar.h\(3HEAD\)](#), [attributes\(5\)](#)

引用名	rcapstat – 报告资源上限执行守护进程统计信息
用法概要	rcapstat [-g] [-p -z] [-T u d] [interval [count]]
描述	<p>rcapstat 命令报告由 rcapd(1M) 限定其上限的项目或区域的相关信息。每个报告都包含关于项目或区域的统计信息以及分页统计信息。分页是指将内存的各部分（称为页）重定位到物理内存或从物理内存重定位的操作。rcapd 对使用频率最低的页执行页出操作。</p> <p>发布的第一个报告中的分页统计信息显示自启动守护进程以来执行的活动。后续报告反映自发布最后一个报告以来执行的活动。</p> <p>以 <i>interval</i> 秒的频率发布由 <i>count</i> 指定的次数的报告，如果未指定 <i>count</i>，则一直发布。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -g 全局统计信息。报告内存上限执行的最小内存使用率（请参见 rcapadm(1M)），并报告当前内存使用占所安装的物理内存的百分比。 -p 报告已限制上限的项目的统计信息。如果没有指定选项，则此选项为缺省选项。 -T u d 显示时间戳。 指定 <i>u</i> 表示时间的内部表示形式的印刷表示形式。请参见 time(2)。指定 <i>d</i> 表示标准日期格式。请参见 date(1)。 -z 报告已限制上限的区域的统计信息。
输出	<p>下面的列表定义了 rcapstat 报告中的列标题，并提供了有关如何解释报告的信息。</p> <ul style="list-style-type: none"> id 已限制上限的项目或区域的项目 ID 或区域 ID。 project 项目名称。 zone 区域名称。 nproc 自上次报告后项目或区域中的进程数。 vm 预留了磁盘或内存交换的所有匿名映射的总和。 rss 项目或区域进程的总驻留集大小（resident set size, RSS），以千字节（K）、兆字节（M）或千兆字节（G）表示。该计数不包括共享页。 cap 项目或区域的 RSS 上限。有关如何指定内存上限的信息，请参见 rcapd(1M)。 at rcapd 试图对其执行页出操作的内存总量。 分页是指将内存的各部分（称为页）重定位到物理内存或从物理内存重定位的操作。rcapd 对使用频率最低的页执行页出操作。

avgat 在每个抽样周期内 **rcapd** 试图对其执行页出操作的内存平均量。可以使用 **rcapadm(1M)** 设置 **rcapd** 对 RSS 进行抽样的速率。

pg **rcapd** 已成功对其执行页出操作的内存总量的估算量。

avgpg 在每个抽样周期内 **rcapd** 成功对其执行页出操作的内存平均量的估算量。使用 **rcapadm** 可以设置 **rcapd** 对进程 RSS 大小进行抽样的速率。

操作数

支持下列操作数：

interval 以秒为单位指定报告间隔。缺省间隔为 5 秒。

count 指定要生成的报告数。缺省情况下，**rcapstat** 会一直报告统计信息，直至收到终止信号或出现 **rcapd** 进程。

示例

示例 1 使用 **rcapstat** 来报告上限和项目信息

为与两个用户相关联的两个项目定义了上限。**user1** 的上限为 50 MB，**user2** 的上限为 10 MB。

以下命令以 5 秒为抽样间隔生成 5 个报告。

```
example# rcapstat 5 5
  id project nproc   vm   rss   cap   at avgat   pg avgpg
112270  user1    24  123M  35M  50M  50M  0K 3312K  0K
 78194  user2     1  2368K 1856K 10M   0K  0K  0K  0K
  id project nproc   vm   rss   cap   at avgat   pg avgpg
112270  user1    24  123M  35M  50M  0K  0K  0K  0K
 78194  user2     1  2368K 1856K 10M   0K  0K  0K  0K
  id project nproc   vm   rss   cap   at avgat   pg avgpg
112270  user1    24  123M  35M  50M  0K  0K  0K  0K
 78194  user2     1  2368K 1928K 10M   0K  0K  0K  0K
  id project nproc   vm   rss   cap   at avgat   pg avgpg
112270  user1    24  123M  35M  50M  0K  0K  0K  0K
 78194  user2     1  2368K 1928K 10M   0K  0K  0K  0K
```

输出的前三行构成了第一个报告，此报告包含自启动 **rcapd** 以来两个项目的上限和项目信息以及换页统计信息。对于 **user1**，**at** 和 **pg** 列中的数字大于零，对于 **user2**，这两列中的数字等于零，这表示在守护进程的历史记录中，有时 **user1** 超过其上限，但 **user2** 却没有。

后续各报告没有显示任何重要的活动。

示例 2 使用 **rcapstat** 来监视项目的 RSS

```
example% rcapstat 5 5
  id project nproc   vm   rss   cap   at avgat   pg avgpg
```


示例2 使用 rcapstat 来监视项目的 RSS (续)

```

376565 user1 57 209M 46M 10M 440M 220M 5528K 2764K
376565 user1 57 209M 44M 10M 394M 131M 4912K 1637K
376565 user1 56 207M 43M 10M 440M 147M 6048K 2016K
376565 user1 56 207M 42M 10M 522M 174M 4368K 1456K
376565 user1 56 207M 44M 10M 482M 161M 3376K 1125K

```

项目 `user1` 具有超出了其物理内存上限的 RSS。`pg` 列中的非零值表示 `rcapd` 在尝试通过降低项目进程的物理内存使用率来满足上限要求时，始终对内存执行页出操作。但是，`rcapd` 未成功，从不断变化却并未真正减小的 `rss` 值可以看出这一点。这表示应用程序的驻留内存一直在被使用，迫使 `rcapd` 影响工作集。在此情况下，系统继续出现高缺页率和关联的 I/O，直到工作集大小 (`working set size, WSS`) 减小、上限提高或应用程序更改其内存访问模式。请注意，必须创建新的页面或者系统必须在交换设备的某页面中进行复制时，便会出现缺页。

示例3 确定项目的工作集大小

此示例是 `Example 1` 的继续，并且使用相同的项目。

```

example% rcapstat 5 5
  id project nproc  vm  rss  cap  at avgat  pg  avgpg
376565 user1 56 207M 44M 10M 381M 191M 15M 7924K
376565 user1 56 207M 46M 10M 479M 160M 2696K 898K
376565 user1 56 207M 46M 10M 424M 141M 7280K 2426K
376565 user1 56 207M 43M 10M 401M 201M 4808K 2404K
376565 user1 56 207M 43M 10M 456M 152M 4800K 1600K
376565 user1 56 207M 44M 10M 486M 162M 4064K 1354K
376565 user1 56 207M 52M 100M 191M 95M 1944K 972K
376565 user1 56 207M 55M 100M 0K 0K 0K 0K
376565 user1 56 207M 56M 100M 0K 0K 0K 0K
376565 user1 56 207M 56M 100M 0K 0K 0K 0K
376565 user1 56 207M 56M 100M 0K 0K 0K 0K
376565 user1 56 207M 56M 100M 0K 0K 0K 0K

```

通过提高项目上限或更改上限执行的最小物理内存使用率来限制上限执行（请参见 `rcapadm(1M)`），驻留集合可变为工作集合。`rss` 列可以稳定地显示项目 WSS，如上面的示例所示。WSS 是使项目的进程在运行时不会始终出现缺页的最小上限值。

退出状态

将返回以下退出值：

- 0 成功完成。
- 1 出现错误。
- 2 指定的命令行选项无效。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/resource-mgmt/resource-caps

另请参见 [rcapadm\(1M\)](#)、[rcapd\(1M\)](#)、[attributes\(5\)](#)

《系统管理指南：资源管理》中的“使用资源限制守护进程进行物理内存控制”

附注 如果为 `rcapstat` 指定的间隔小于为 `rcapd` 指定的报告间隔（使用 `rcapadm(1M)`），则某些间隔的输出可能为零。这是因为 `rcapd` 更新统计信息的频率不高于由 `rcapadm` 指定的间隔确定的频率，并且该间隔独立于 `rcapstat` 使用的抽样间隔，但是精度低于后者。

引用名	rcp – remote file copy
用法概要	<pre>rcp [-p] [-a] [-K] [-x] [-PN -PO] [-k realm] filename1 filename2 rcp [-pr] [-a] [-K] [-x] [-PN -PO] [-k realm] filename... directory</pre>
描述	<p>The rcp command copies files between machines. Each <i>filename</i> or <i>directory</i> argument is either a remote file name of the form:</p> <pre>hostname:path</pre> <p>or a local file name (containing no : (colon) characters, or / (backslash) before any : (colon) characters).</p> <p>The <i>hostname</i> can be an IPv4 or IPv6 address string. See inet(7P) and inet6(7P). Since IPv6 addresses already contain colons, the <i>hostname</i> should be enclosed in a pair of square brackets when an IPv6 address is used. Otherwise, the first occurrence of a colon can be interpreted as the separator between <i>hostname</i> and <i>path</i>. For example,</p> <pre>[1080::8:800:200C:417A]:tmp/file</pre> <p>If a <i>filename</i> is not a full path name, it is interpreted relative to your home directory on <i>hostname</i>. A <i>path</i> on a remote host can be quoted using \ , " , or ' , so that the metacharacters are interpreted remotely. Please notice that the kerberized versions of rcp are not IPv6-enabled.</p> <p>rcp does not prompt for passwords. It either uses Kerberos authentication which is enabled through command-line options or your current local user name must exist on <i>hostname</i> and allow remote command execution by rsh(1).</p> <p>The rcp session can be kerberized using any of the following Kerberos specific options : -a, -PN or -PO, -x, and -k <i>realm</i>. Some of these options (-a, -x and -PN or -PO) can also be specified in the [appdefaults] section of krb5.conf(4). The usage of these options and the expected behavior is discussed in the OPTIONS section below. If Kerberos authentication is used, authorization to the account is controlled by rules in krb5_auth_rules(5). If this authorization fails, fallback to normal rcp using rhosts occurs only if the -PO option is used explicitly on the command line or is specified in krb5.conf(4). If authorization succeeds, remote copy succeeds without any prompting of password. Also notice that the -PN or -PO, -x, and -k <i>realm</i> options are just supersets of the -a option.</p> <p>rcp handles third party copies, where neither source nor target files are on the current machine. Hostnames can also take the form</p> <pre>username@hostname:filename</pre> <p>to use <i>username</i> rather than your current local user name as the user name on the remote host. rcp also supports Internet domain addressing of the remote host, so that:</p> <pre>username@host.domain:filename</pre>

specifies the username to be used, the hostname, and the domain in which that host resides. File names that are not full path names are interpreted relative to the home directory of the user named *username*, on the remote host.

选项

The following options are supported:

- a This option explicitly enables Kerberos authentication and trusts the `.k5login` file for access-control. If the authorization check by `in.rshd(1M)` on the server-side succeeds and if the `.k5login` file permits access, the user is allowed to carry out the `rcp` transfer.
- k *realm* Causes `rcp` to obtain tickets for the remote host in *realm* instead of the remote host's realm as determined by `krb5.conf(4)`.
- K *realm* This option explicitly disables Kerberos authentication. It can be used to override the `autoLogin` variable in `krb5.conf(4)`.
- p Attempts to give each copy the same modification times, access times, modes, and ACLs if applicable as the original file.
- PO
- PN Explicitly requests new (-PN) or old (-PO) version of the Kerberos “rcmd” protocol. The new protocol avoids many security problems prevalent in the old one and is regarded much more secure, but is not interoperable with older (MIT/SEAM) servers. The new protocol is used by default, unless explicitly specified using these options or through `krb5.conf(4)`. If Kerberos authorization fails when using the old “rcmd” protocol, there is fallback to regular, non-kerberized `rcp`. This is not the case when the new, more secure “rcmd” protocol is used.
- r Copies each subtree rooted at *filename*; in this case the destination must be a directory.
- x Causes the information transferred between hosts to be encrypted. Notice that the command is sent unencrypted to the remote system. All subsequent transfers are encrypted.

用法

See `largefile(5)` for the description of the behavior of `rcp` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

The `rcp` command is IPv6-enabled. See `ip6(7P)`. IPv6 is not currently supported with Kerberos V5 authentication.

For the kerberized `rcp` session, each user can have a private authorization list in a file `.k5login` in their home directory. Each line in this file should contain a Kerberos principal name of the form *principal/instance@realm*. If there is a `~/k5login` file, then access is granted to the account if and only if the originator user is authenticated to one of the principals named in the `~/k5login` file. Otherwise, the originating user is granted access to the account if and

only if the authenticated principal name of the user can be mapped to the local account name using the *authenticated-principal-name* → *local-user-name* mapping rules. The `.k5login` file (for access control) comes into play only when Kerberos authentication is being done.

退出状态

The following exit values are returned:

- 0 All files were copied successfully.
- >0 An error occurred.

See the NOTES section for caveats on the exit code.

文件

`$HOME/.profile`

`$HOME/.k5login` File containing Kerberos principals that are allowed access

`/etc/krb5/krb5.conf` Kerberos configuration file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients
CSI	Enabled

另请参见

[cpio\(1\)](#), [ftp\(1\)](#), [rlogin\(1\)](#), [rsh\(1\)](#), [setfacl\(1\)](#), [tar\(1\)](#), [tar\(1\)](#), [in.rshd\(1M\)](#), [hosts.equiv\(4\)](#), [krb5.conf\(4\)](#), [attributes\(5\)](#), [largefile\(5\)](#), [krb5_auth_rules\(5\)](#), [inet\(7P\)](#), [inet6\(7P\)](#), [ip6\(7P\)](#)

附注

`rcp` is meant to copy between different hosts. Attempting to `rcp` a file onto itself, as with:

```
example% rcp tmp/file myhost:/tmp/file
```

results in a severely corrupted file.

`rcp` might not correctly fail when the target of a copy is a file instead of a directory.

`rcp` can become confused by output generated by commands in a `$HOME/.profile` on the remote host.

`rcp` requires that the source host have permission to execute commands on the remote host when doing third-party copies.

`rcp` does not properly handle symbolic links. Use `tar` or `cpio` piped to `rsh` to obtain remote copies of directories containing symbolic links or named pipes. See [tar\(1\)](#) and [cpio\(1\)](#).

If you forget to quote metacharacters intended for the remote host, you get an incomprehensible error message.

`rcp` fails if you copy ACLs to a file system that does not support ACLs.

rcp is CSI-enabled except for the handling of username, hostname, and domain.

When rcp is used to perform third-party copies where either of the remote machines is not running Solaris, the exit code cannot be relied upon. That is, errors could occur when success is reflected in the exit code, or the copy could be completely successful even though an error is reflected in the exit code.

引用名 read – read a line from standard input

用法概要

```
/usr/bin/read            /usr/bin/read [-r] var...
sh                        read name...
csh                       set variable= $<
ksh88                    read [-prsu [n]] [name ? prompt] [name]...
ksh                      read [-ACprs] [-d delim] [-n nsize] [-N nsize] [-t timeout]
                         [-u unit] [vname?prompt] [vname... ]
```

描述

`/usr/bin/read` The read utility reads a single line from standard input.

By default, unless the `-r` option is specified, backslash (`\`) acts as an escape character. If standard input is a terminal device and the invoking shell is interactive, read prompts for a continuation line when:

- The shell reads an input line ending with a backslash, unless the `-r` option is specified.
- A here-document is not terminated after a `NEWLINE` character is entered.

The line is split into fields as in the shell. The first field is assigned to the first variable `var`, the second field to the second variable `var`, and so forth. If there are fewer `var` operands specified than there are fields, the leftover fields and their intervening separators is assigned to the last `var`. If there are fewer fields than `vars`, the remaining `vars` is set to empty strings.

The setting of variables specified by the `var` operands affects the current shell execution environment. If it is called in a sub-shell or separate utility execution environment, such as one of the following:

```
(read foo)
nohup read ...
find . -exec read ... \;
```

It does not affect the shell variables in the caller's environment.

The standard input must be a text file.

`sh` One line is read from the standard input and, using the internal field separator, `IFS` (normally space or tab), to delimit word boundaries, the first word is assigned to the first `name`, the second word to the second `name`, and so on, with leftover words assigned to the last `name`. Lines can be continued using `\newLine`. Characters other than `NEWLINE` can be quoted by preceding them with a backslash. These backslashes are removed before words are assigned to `names`, and no interpretation is done on the character that follows the backslash. The return code is `0`, unless an end-of-file is encountered.

`csh` The notation:

```
set variable = $<
```

loads one line of standard input as the value for *variable*. (See [csh\(1\)](#)).

ksh88 The shell input mechanism. One line is read and is broken up into fields using the characters in IFS as separators. The escape character, (\), is used to remove any special meaning for the next character and for line continuation. In raw mode, the - r, the , and the \ character are not treated specially. The first field is assigned to the first *name*, the second field to the second *name*, and so on, with leftover fields assigned to the last *name*. The - p option causes the input line to be taken from the input pipe of a process spawned by the shell using |&. If the - s flag is present, the input is saved as a command in the history file. The flag - u can be used to specify a one digit file descriptor unit *n* to read from. The file descriptor can be opened with the exec special command. The default value of *n* is 0. If *name* is omitted, REPLY is used as the default *name*. The exit status is 0 unless the input file is not open for reading or an end-of-file is encountered. An end-of-file with the - p option causes cleanup for this process so that another can be spawned. If the first argument contains a ?, the remainder of this word is used as a *prompt* on standard error when the shell is interactive. The exit status is 0 unless an end-of-file is encountered.

ksh read reads a line from standard input and breaks it into fields using the characters in the value of the IFS variable as separators. The escape character, \, is used to remove any special meaning for the next character and for line continuation unless the - r option is specified.

If there are more variables than fields, the remaining variables are set to empty strings. If there are fewer variables than fields, the leftover fields and their intervening separators are assigned to the last variable. If no *var* is specified, the variable REPLY is used.

When *var* has the binary attribute and - n or - N is specified, the bytes that are read are stored directly into *var*.

If you specify *?prompt* after the first *var*, read displays a prompt on standard error when standard input is a terminal or pipe.

选项

`/usr/bin/read`, **ksh88** The following option is supported by `/usr/bin/read` and **ksh88**:

- r Do not treat a backslash character in any special way. Considers each backslash to be part of the input line.

ksh The following options are supported by **ksh**:

- A Unset *var*, and create an indexed array containing each field in the line starting at index 0.

- C Unset *var* and read *var* as a compound variable.

- d *delim* Read until delimiter *delim* instead of to the end of line.

- n *nsize* Read at most *nsize* bytes. Binary field size is in bytes.

- N *nsize* Read exactly *nsize* bytes. Binary field size is in bytes.
- p Read from the current co-process instead of standard input. An end of file causes read to disconnect the co-process so that another can be created.
- r Do not treat \ specially when processing the input line.
- s Save a copy of the input as an entry in the shell history file.
- t *timeout* Specify a *timeout* in seconds when reading from a terminal or pipe.
- u *fd* Read from file descriptor number *fd* instead of standard input. The default value is 0.
- v When reading from a terminal, display the value of the first variable and use it as a default value.

操作数

The following operand is supported:

var The name of an existing or non-existing shell variable.

示例

示例 1 Using the read Command

The following example for `/usr/bin/read` prints a file with the first field of each line moved to the end of the line:

```
example% while read -r xx yy
do
    printf "%s %s\n" "$yy" "$xx"
done < input_file
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of read: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

IFS Determines the internal field separators used to delimit fields.

PS2 Provides the prompt string that an interactive shell writes to standard error when a line ending with a backslash is read and the `-r` option was not specified, or if a here-document is not terminated after a NEWLINE character is entered.

退出状态

The following exit values are returned:

0 Successful completion.

>0 End-of-file was detected or an error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/read, csh,
ksh88, sh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

read(1)

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
Standard	See standards(5) .

ksh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Uncommitted

另请参见

[csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [line\(1\)](#), [set\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名 readonly – shell built-in function to protect the value of the given variable from reassignment

用法概要

```
sh                   readonly [name]...
ksh88               **readonly [name [= value]]...
                    **readonly -p
ksh                 ++readonly [-p] [name [= value]]...
```

描述

sh The given *names* are marked `readonly` and the values of these *names* may not be changed by subsequent assignment. If no arguments are given, a list of all `readonly` names is printed.

ksh88 The given *names* are marked `readonly` and these names cannot be changed by subsequent assignment.

When `-p` is specified, `readonly` writes to the standard output the names and values of all read-only variables, in the following format:

```
"readonly %s=%s\n", name, value
```

if *name* is set, and:

```
"readonly $s\n", name
```

if *name* is unset.

The shell formats the output, including the proper use of quoting, so that it is suitable for reinput to the shell as commands that achieve the same value and `readonly` attribute-setting results in a shell execution environment in which:

1. Variables with values set at the time they were output do not have the `readonly` attribute set.
2. Variables that were unset at the time they were output do not have a value at the time at which the saved output is re-input to the shell.

On this manual page, [ksh88\(1\)](#) commands that are preceded by one or two `**` (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by `**` that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the `=` sign and word splitting and file name generation are not performed.

`ksh` `readonly` sets the `readonly` attribute on each of the variables specified by name which prevents their values from being changed. If `=value` is specified, the variable name is set to `value` before the variable is made `readonly`.

If no names are specified then the names and values of all `readonly` variables are written to standard output.

`readonly` is built-in to the shell as a declaration command so that field splitting and pathname expansion are not performed on the arguments. Tilde expansion occurs on value.

`-p` Causes the output to be in a form of `readonly` commands that can be used as input to the shell to recreate the current set of `readonly` variables.

On this manual page, `ksh(1)` commands that are preceded by one or two `+` symbols are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. They are not valid function names.
5. Words, following a command preceded by `++` that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the `=` sign and field splitting and file name generation are not performed.

退出状态

`ksh` The following exit values are returned:

- `0` Successful completion.
- `>0` An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [typeset\(1\)](#), [attributes\(5\)](#)

引用名	refer – expand and insert references from a bibliographic database
用法概要	refer [-ben] [-ar] [-cstring] [-kx] [-\lm,n] [-p filename] [-skeys] filename...
描述	<p>refer is a preprocessor for <code>nroff(1)</code>, or <code>troff(1)</code>, that finds and formats references. The input files (standard input by default) are copied to the standard output, except for lines between <code>‘ . [’</code> and <code>‘ .]’</code> command lines. Such lines are assumed to contain keywords as for <code>lookbib(1)</code>, and are replaced by information from a bibliographic data base. The user can avoid the search, override fields from it, or add new fields. The reference data, from whatever source, is assigned to a set of <code>troff</code> strings. Macro packages such as <code>ms(5)</code> print the finished reference text from these strings. A flag is placed in the text at the point of reference. By default, the references are indicated by numbers.</p> <p>When <code>refer</code> is used with <code>eqn(1)</code>, <code>neqn</code>, or <code>tbl(1)</code>, <code>refer</code> should be used first in the sequence, to minimize the volume of data passed through pipes.</p>
选项	<p>-b Bare mode — do not put any flags in text (neither numbers or labels).</p> <p>-e Accumulate references instead of leaving the references where encountered, until a sequence of the form:</p> <pre>. [\$LIST\$.]</pre> <p>is encountered, and then write out all references collected so far. Collapse references to the same source.</p> <p>-n Do not search the default file.</p> <p>-ar Reverse the first <i>r</i> author names (Jones, J. A. instead of J. A. Jones). If <i>r</i> is omitted, all author names are reversed.</p> <p>-cstring Capitalize (with SMALL CAPS) the fields whose key-letters are in <i>string</i>.</p> <p>-kx Instead of numbering references, use labels as specified in a reference data line beginning with the characters <code>%x</code>; By default, <i>x</i> is L.</p> <p>-\lm,n Instead of numbering references, use labels from the senior author's last name and the year of publication. Only the first <i>m</i> letters of the last name and the last <i>n</i> digits of the date are used. If either of <i>m</i> or <i>n</i> is omitted, the entire name or date, respectively, is used.</p> <p>-p filename Take the next argument as a file of references to be searched. The default file is searched last.</p> <p>-skeys Sort references by fields whose key-letters are in the <i>keys</i> string, and permute reference numbers in the text accordingly. Using this option implies the <code>-e</code> option. The key-letters in <i>keys</i> may be followed by a number indicating how many such fields are used, with a <code>+</code> sign taken as a very large number. The</p>

default is AD, which sorts on the senior author and date. To sort on all authors and then the date, for instance, use the options '-sA+T'.

文件 /usr/lib/refer directory of programs
 /usr/lib/refer/papers directory of default publication lists and indexes

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

另请参见 [addbib\(1\)](#), [eqn\(1\)](#), [indxbib\(1\)](#), [lookbib\(1\)](#), [nroff\(1\)](#), [roffb\(1\)](#), [sortbib\(1\)](#), [tbl\(1\)](#),
[troff\(1\)](#), [attributes\(5\)](#)

引用名	regcmp – regular expression compile
用法概要	regcmp [-] <i>filename</i> ...
描述	The regcmp command performs a function similar to regcmp and, in most cases, precludes the need for calling regcmp from C programs. Bypassing regcmp saves on both execution time and program size. The command regcmp compiles the regular expressions in <i>filename</i> and places the output in <i>filename.i</i> .
选项	<ul style="list-style-type: none"> – If the <code>--</code> option is used, the output is placed in <i>filename.c</i>. The format of entries in <i>filename</i> is a name (C variable) followed by one or more blanks followed by one or more regular expressions enclosed in double quotes. The output of regcmp is C source code. Compiled regular expressions are represented as extern char vectors. <i>filename.i</i> files may thus be <code>#included</code> in C programs, or <i>filename.c</i> files may be compiled and later loaded. In the C program that uses the regcmp output, <code>regex(abc, line)</code> applies the regular expression named <code>abc</code> to <code>line</code>. Diagnostics are self-explanatory.
示例	<p>示例 1 Using the regcmp command.</p> <pre>name "[A-Za-z][A-Za-z0-9_]*\$0" telno "\({0,1}([2-9][01][1-9])\$0\){0,1}*" "([2-9][0-9]{2})\$1[-]{0,1}" "([0-9]{4})\$2"</pre> <p>The three arguments to <code>telno</code> shown above must all be entered on one line.</p> <p>In the C program that uses the regcmp output,</p> <pre> regex(telno, line, area, exch, rest)</pre> <p>applies the regular expression named <code>telno</code> to <code>line</code>.</p>
环境变量	<p>A general description of the usage of the <code>LC_*</code> environmental variables can be found in environ(5).</p> <p>LC_CTYPE Determines how regcmp handles characters. When <code>LC_CTYPE</code> is set to a valid value, regcmp can display and handle text and filenames containing valid characters for that locale.</p> <p>LC_MESSAGES Determines how diagnostic and informative messages are presented. This includes the language and style of the messages, and the correct form of affirmative and negative responses. In the "C" locale, the messages are presented in the default form found in the program itself (in most cases, U.S. English).</p>

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities
CSI	Enabled

另请参见

[regcmp\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#)

引用名 renice – alter priority of running processes

用法概要

```
renice [-n increment] [-i idtype] ID...
```

```
renice [-n increment] [-g | -p | -u] ID...
```

```
renice priority [-p] pid... [-g gid]... [-p pid]...
      [-u user]...
```

```
renice priority -g gid... [-g gid]... [-p pid]...
      [-u user]...
```

```
renice priority -u user... [-g gid]... [-p pid]...
      [-u user]...
```

描述

The `renice` command alters the scheduling priority of one or more running processes. By default, the processes to be affected are specified by their process IDs.

If the first operand is a number within the valid range of priorities (-20 to 20), `renice` will treat it as a *priority* (as in all but the first synopsis form). Otherwise, `renice` will treat it as an *ID* (as in the first synopsis form).

Altering Process Priority

Users other than the privileged user may only alter the priority of processes they own, and can only monotonically increase their “nice value” within the range 0 to 19 . This prevents overriding administrative fiats. The privileged user may alter the priority of any process and set the priority to any value in the range -20 to 19 . Useful priorities are: 19 (the affected processes will run only when nothing else in the system wants to); 0 (the “base” scheduling priority);, and any negative value (to make things go very fast). 20 is an acceptable nice value, but will be rounded down to 19 .

选项

`renice` supports the following option features:

- The first operand, *priority*, must precede the options and can have the appearance of a multi-digit option.
- The `-g`, `-p`, and `-u` options can each take multiple option-arguments.
- The *pid* option-argument can be used without its `-p` option.
- The `-i` option can be used to specify the *ID* type for the ID list. This is preferred in specifying *ID* type over the use of the `-g | -p | -u` syntax, which is now obsolete. See NOTES.

The following options are supported:

- g Interprets all operands or just the *gid* arguments as unsigned decimal integer process group IDs.
- i This option, together with the *ID* list arguments, specifies a class of processes to which the `renice` command is to apply. The interpretation of the ID list depends on the value of *idtype*. The valid *idtype* arguments are: `pid`, `pgid`, `uid`, `gid`, `sid`, `taskid`, `projid`, and `zoneid`.

- n *increment*** Specifies how the system scheduling priority of the specified process or processes is to be adjusted. The *increment* option-argument is a positive or negative decimal integer that will be used to modify the system scheduling priority of the specified process or processes. Positive *increment* values cause a lower system scheduling priority. Negative *increment* values may require appropriate privileges and will cause a higher system scheduling priority.
- p** Interprets all operands or just the *pid* arguments as unsigned decimal integer process IDs. The -p option is the default if no options are specified.
- u** Interprets all operands or just the *user* argument as users. If a user exists with a user name equal to the operand, then the user ID of that user will be used in further processing. Otherwise, if the operand represents an unsigned decimal integer, it will be used as the numeric user ID of the user.

操作数

The following operands are supported:

- ID*** A process ID, process group ID, or user name/user ID, depending on the option selected.
- priority*** The value specified is taken as the actual system scheduling priority, rather than as an increment to the existing system scheduling priority. Specifying a scheduling priority higher than that of the existing process may require appropriate privileges.

示例

示例 1 Adjusting the scheduling priority of process IDs

Adjust the system scheduling priority so that process IDs 987 and 32 would have a lower scheduling priority:

```
example% renice -n 5 -p 987 32
```

示例 2 Adjusting the scheduling priority of group IDs

Adjust the system scheduling priority so that group IDs 324 and 76 would have a higher scheduling priority, if the user has the appropriate privileges to do so:

```
example% renice -n -4 -g 324 76
```

示例 3 Adjusting the scheduling priority of a user ID and user name

Adjust the system scheduling priority so that numeric user ID 8 and user sas would have a lower scheduling priority:

```
example% renice -n 4 -u 8 sas
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `renice`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLS_PATH`.

- 退出状态** The following exit values are returned:
- 0 Successful completion.
 - >0 An error occurred.
- 文件** /etc/passwd map user names to user IDs
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [nice\(1\)](#), [passwd\(1\)](#), [prioctl\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注 The renice syntax

```
renice [-n increment] [-i idtype] ID ...
```

is preferred over the old syntax

```
renice [-n increment] [-g | -p | -u] ID ...
```

which is now obsolete.

If you make the priority very negative, then the process cannot be interrupted.

To regain control you must make the priority greater than 0.

Users other than the privileged user cannot increase scheduling priorities of their own processes, even if they were the ones that decreased the priorities in the first place.

The `prioctl` command subsumes the function of `renice`.

引用名 rlogin – remote login

用法概要 `rlogin [-8EL] [-ec] [-A] [-K] [-x] [-PN | -PO] [-f | -F] [-a] [-l username] [-k realm] hostname`

描述 The `rlogin` utility establishes a remote login session from your terminal to the remote machine named *hostname*. The user can choose to kerberize the `rlogin` session using Kerberos V5 and also protect the data being transferred.

Hostnames are listed in the *hosts* database, which can be contained in the `/etc/hosts` file, the Network Information Service (NIS) *hosts* map, the Internet domain name server, or a combination of these. Each host has one official name (the first name in the database entry), and optionally one or more nicknames. Either official hostnames or nicknames can be specified in *hostname*.

The user can opt for a secure `rlogin` session which uses Kerberos V5 for authentication. Encryption of the session data is also possible. The `rlogin` session can be kerberized using any of the following Kerberos specific options: `-A`, `-PN` or `-PO`, `-x`, `-f` or `-F`, and `-k realm`. Some of these options (`-A`, `-x`, `-PN` or `-PO`, and `-f` or `-F`) can also be specified in the `[appdefaults]` section of `krb5.conf(4)`. The usage of these options and the expected behavior is discussed in the OPTIONS section below. If Kerberos authentication is used, authorization to the account is controlled through rules in `krb5_auth_rules(5)`. If this authorization fails, fallback to normal `rlogin` using `rhosts` occurs only if the `-PO` option is used explicitly on the command line or is specified in `krb5.conf(4)`. Also notice that the `-PN` or `-PO`, `-x`, `-f` or `-F`, and `-k realm` options are just supersets of the `-A` option.

The remote terminal type is the same as your local terminal type, as given in your environment `TERM` variable. The terminal or window size is also copied to the remote system if the server supports the option. Changes in size are reflected as well. All echoing takes place at the remote site, so that (except for delays) the remote login is transparent. Flow control using Control-S and Control-Q and flushing of input and output on interrupts are handled properly.

选项 The following options are supported:

- `-8` Passes eight-bit data across the net instead of seven-bit data.
- `-a` Forces the remote machine to ask for a password by sending a null local username.
- `-A` Explicitly enables Kerberos authentication and trusts the `.k5login` file for access-control. If the authorization check by `in.rlogind(1M)` on the server-side succeeds and if the `.k5login` file permits access, the user is allowed to login without supplying a password.
- `-ec` Specifies a different escape character, *c*, for the line used to disconnect from the remote host.
- `-E` Stops any character from being recognized as an escape character.

- f Forwards a copy of the local credentials (Kerberos Ticket Granting Ticket) to the remote system. This is a non-forwardable ticket granting ticket. You must forward a ticket granting ticket if you need to authenticate yourself to other Kerberized network services on the remote host. An example is if your home directory on the remote host is NFS mounted via Kerberos V5. If your local credentials are not forwarded in this case, you can not access your home directory. This option is mutually exclusive with the -F option.
- F Forwards a forwardable copy of the local credentials (Kerberos Ticket Granting Ticket) to the remote system. The -F option provides a superset of the functionality offered by the -f option. For example, with the -f option, after you connected to the remote host, any attempt to invoke `/usr/bin/ftp`, `/usr/bin/telnet`, `/usr/bin/rlogin`, or `/usr/bin/rsh` with the -f or -F options would fail. Thus, you would be unable to push your single network sign on trust beyond one system. This option is mutually exclusive with the -f option.
- k *realm* Causes `rlogin` to obtain tickets for the remote host in *realm* instead of the remote host's realm as determined by `krb5.conf(4)`.
- K This option explicitly disables Kerberos authentication. It can be used to override the `autoLogin` variable in `krb5.conf(4)`.
- l *username* Specifies a different *username* for the remote login. If you do not use this option, the remote username used is the same as your local username.
- L Allows the `rlogin` session to be run in "litout" mode.
- PN
- PO Explicitly requests the new (-PN) or old (-PO) version of the Kerberos 'rcmd' protocol. The new protocol avoids many security problems prevalent in the old one and is considered much more secure, but is not interoperable with older (MIT/SEAM) servers. The new protocol is used by default, unless explicitly specified using these options or by using `krb5.conf(4)`. If Kerberos authorization fails when using the old 'rcmd' protocol, there is fallback to regular, non-kerberized `rlogin`. This is not the case when the new, more secure 'rcmd' protocol is used.
- x Turns on DES encryption for all data passed through the `rlogin` session. This reduces response time and increases CPU utilization.

Escape Sequences

Lines that you type which start with the tilde character (~) are "escape sequences." The escape character can be changed using the -e option.

- ~. Disconnects from the remote host. This is not the same as a logout, because the local host breaks the connection with no warning to the remote end.

- ~susp Suspends the login session, but only if you are using a shell with Job Control. `susp` is your “suspend” character, usually Control-Z. See [tty\(1\)](#).
- ~dsusp Suspends the input half of the login, but output is still able to be seen (only if you are using a shell with Job Control). `dsusp` is your “deferred suspend” character, usually Control-Y. See [tty\(1\)](#).

操作数

hostname The remote machine on which *rlogin* establishes the remote login session.

用法

For the kerberized *rlogin* session, each user can have a private authorization list in a file, `.k5login`, in his home directory. Each line in this file should contain a Kerberos principal name of the form *principal/instance@realm*. If there is a `~/k5login` file, access is granted to the account if and only if the originating user is authenticated to one of the principals named in the `~/k5login` file. Otherwise, the originating user is granted access to the account if and only if the authenticated principal name of the user can be mapped to the local account name using the *authenticated-principal-name* → *local-user-name* mapping rules. The `.k5login` file (for access control) comes into play only when Kerberos authentication is being done.

For the non-secure *rlogin* session, each remote machine can have a file named `/etc/hosts.equiv` containing a list of trusted host names with which it shares user names. Users with the same user name on both the local and remote machine can *rlogin* from the machines listed in the remote machine's `/etc/hosts.equiv` file without supplying a password. Individual users can set up a similar private equivalence list with the file `.rhosts` in their home directories. Each line in this file contains two names, that is, a host name and a user name, separated by a space. An entry in a remote user's `.rhosts` file permits the user named *username* who is logged into *hostname* to log in to the remote machine as the remote user without supplying a password. If the name of the local host is not found in the `/etc/hosts.equiv` file on the remote machine, and the local user name and host name are not found in the remote user's `.rhosts` file, then the remote machine prompts for a password. Host names listed in the `/etc/hosts.equiv` and `.rhosts` files must be the official host names listed in the `hosts` database. Nicknames can not be used in either of these files.

For security reasons, the `.rhosts` file must be owned by either the remote user or by root.

文件

- `/etc/passwd` Contains information about users' accounts.
- `/usr/hosts/*` For *hostname* version of the command.
- `/etc/hosts.equiv` List of trusted hostnames with shared user names.
- `/etc/nologin` Message displayed to users attempting to login during machine shutdown.
- `~/HOME/.rhosts` Private list of trusted hostname/username combinations.
- `~/HOME/.k5login` File containing Kerberos principals that are allowed access.
- `/etc/krb5/krb5.conf` Kerberos configuration file.

`/etc/hosts` Hosts database.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

另请参见

[rsh\(1\)](#), [stty\(1\)](#), [tty\(1\)](#), [in.rlogind\(1M\)](#), [hosts\(4\)](#), [hosts.equiv\(4\)](#), [krb5.conf\(4\)](#), [nologin\(4\)](#), [attributes\(5\)](#), [krb5_auth_rules\(5\)](#)

诊断

The following message indicates that the machine is in the process of being shutdown and logins have been disabled:

```
NO LOGINS: System going down in N minutes
```

附注

When a system is listed in `hosts.equiv`, its security must be as good as local security. One insecure system listed in `hosts.equiv` can compromise the security of the entire system.

The Network Information Service (NIS) was formerly known as Sun Yellow Pages (YP.) The functionality of the two remains the same. Only the name has changed.

This implementation can only use the TCP network service.

引用名 rm, rmdir – remove directory entries

用法概要

```
/usr/bin/rm [-f] [-i] file...  
  
/usr/bin/rm -rR [-f] [-i] dirname... [file]...  
  
/usr/xpg4/bin/rm [-fiRr] file...  
  
/usr/bin/rmdir [-ps] dirname...
```

描述

/usr/bin/rm
/usr/xpg4/bin/rm

The `rm` utility removes the directory entry specified by each *file* argument. If a file has no write permission and the standard input is a terminal, the full set of permissions (in octal) for the file are printed followed by a question mark. This is a prompt for confirmation. If the answer is affirmative, the file is deleted, otherwise the file remains.

If *file* is a symbolic link, the link is removed, but the file or directory to which it refers is not deleted. Users do not need write permission to remove a symbolic link, provided they have write permissions in the directory.

If multiple *files* are specified and removal of a *file* fails for any reason, `rm` writes a diagnostic message to standard error, do nothing more to the current *file*, and go on to any remaining *files*.

If the standard input is not a terminal, the utility operates as if the `-f` option is in effect.

/usr/bin/rmdir

The `rmdir` utility removes the directory entry specified by each *dirname* operand, which must refer to an empty directory.

Directories are processed in the order specified. If a directory and a subdirectory of that directory are specified in a single invocation of `rmdir`, the subdirectory must be specified before the parent directory so that the parent directory is empty when `rmdir` tries to remove it.

选项

The following options are supported for `/usr/bin/rm` and `/usr/xpg4/bin/rm`:

`-r` Recursively removes directories and subdirectories in the argument list. The directory is emptied of files and removed. The user is normally prompted for removal of any write-protected files which the directory contains. The write-protected files are removed without prompting, however, if the `-f` option is used, or if the standard input is not a terminal and the `-i` option is not used.

Symbolic links that are encountered with this option is not traversed.

If the removal of a non-empty, write-protected directory is attempted, the utility always fails (even if the `-f` option is used), resulting in an error message.

`-R` Same as `-r` option.

`/usr/bin/rm` The following options are supported for `/usr/bin/rm` only:

- f Removes files (even if write-protected) in a directory without prompting the user. In a write-protected directory, however, files are never removed (whatever their permissions are) and no messages are displayed.
- i Interactive. With this option, `rm` prompts for confirmation before removing any files. It overrides the -f option and remains in effect even if the standard input is not a terminal.

`/usr/xpg4/bin/rm` The following options are supported for `/usr/xpg4/bin/rm` only:

- f Does not prompt for confirmation. Does not write diagnostic messages or modify the exit status in the case of non-existent operands. Any previous occurrences of the -i option is ignored.
- i Prompts for confirmation. Any occurrences of the -f option is ignored.

`/usr/bin/rmdir` The following options are supported for `/usr/bin/rmdir` only:

- p Allows users to remove the directory *dirname* and its parent directories which become empty. A message is printed to standard error if all or part of the path could not be removed.
- s Suppresses the message printed on the standard error when -p is in effect.

操作数

The following operands are supported:

- file* Specifies the pathname of a directory entry to be removed.
- dirname* Specifies the pathname of an empty directory to be removed.

用法

See [largefile\(5\)](#) for the description of the behavior of `rm` and `rmdir` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

The following examples are valid for the commands shown.

`/usr/bin/rm,`
`/usr/xpg4/bin/rm`

示例 1 Removing Directories

The following command removes the directory entries `a.out` and `core`:

```
example% rm a.out core
```

示例 2 Removing a Directory without Prompting

The following command removes the directory `junk` and all its contents, without prompting:

```
example% rm -rf junk
```

/usr/bin/rmdir

示例 3 Removing Empty Directories

If a directory `a` in the current directory is empty, except that it contains a directory `b`, and `a/b` is empty except that it contains a directory `c`, the following command removes all three directories:

```
example% rmdir -p a/b/c
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `rm` and `rmdir`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

Affirmative responses are processed using the extended regular expression defined for the `yesexpr` keyword in the `LC_MESSAGES` category of the user's locale. The locale specified in the `LC_COLLATE` category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for `yesexpr`. The locale specified in `LC_CTYPE` determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the `yesexpr`. See [locale\(5\)](#).

退出状态

The following exit values are returned:

- `0` If the `-f` option was not specified, all the named directory entries were removed; otherwise, all the existing named directory entries were removed.
- `>0` An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/rm,
/usr/bin/rmdir

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

/usr/xpg4/bin/rm

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[rmdir\(2\)](#), [unlink\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

诊断

It is forbidden to remove the files “.” and “..” in order to avoid the consequences of inadvertently doing something like the following:

```
example% rm -r .*
```

It is forbidden to remove the file “/” in order to avoid the consequences of inadvertently doing something like:

```
example% rm -rf $x/$y
```

or

```
example% rm -rf /$y
```

when \$x and \$y expand to empty strings.

附注

A `-` permits the user to mark explicitly the end of any command line options, allowing `rm` to recognize file arguments that begin with a `-`. As an aid to BSD migration, `rm` accepts `--` as a synonym for `-`. This migration aid may disappear in a future release. If a `-` and a `-` both appear on the same command line, the second is interpreted as a file.

引用名	rmformat – 可移除可重写介质格式化实用程序
用法概要	<pre>rmformat [-DeHUv] [-b label] [-c blockno] [-Fquick long force] [-s filename] [devname] rmformat -V read write devname rmformat -l [devname]</pre>
描述	<p>rmformat 实用程序用于对可移除、可重写介质（包括 PCMCIA 内存和 ata 卡）进行格式化、标记和分区，以及对其执行其他杂项功能。对于所有 USB 海量存储设备（包括 USB 硬盘驱动器），也应当使用 rmformat 实用程序。此实用程序还可用于验证和表面分析，并且修复验证过程中发现的坏扇区（如果驱动器或驱动程序支持坏块管理）。</p> <p>格式化之后，rmformat 将写入标签（包括介质的完整容量），作为 PCMCIA 内存卡上的一个分片。可以借助 rmformat 提供的其他选项来更改分区信息。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none">-b <i>label</i> 为介质设置 SUNOS 标签。SUNOS 卷标签名称限制为 8 个字符。对于大小超过 1 TB 的介质，将创建 EFI 标签。对于编写 DOS 卷标签，用户应该使用 <code>mkfs_pcfs(1M)</code>。-c <i>blockno</i> 更正并修复给定的块。此更正和修复选项不一定适用于 rmformat 支持的所有设备，因为某些设备的驱动器可能具有坏块管理功能，而某些设备可能已在驱动程序中实施了此选项。如果驱动器或驱动程序支持坏块管理，则会尽最大的努力来纠正坏块。如果坏块仍然无法纠正，将显示一条消息，指明修复失败。块编号可以采用十进制、八进制或十六进制的格式来提供。 普通的 PCMCIA 内存卡和 ata 卡不支持坏块管理。-e 完成时弹出介质。如果驱动器不支持自动化弹出，则此功能不可用。-F <i>quick long force</i> 格式化介质。 <i>quick</i> 选项启动不进行验证或者只对介质上的某些磁轨进行有限验证的格式化。 <i>long</i> 选项启动完整格式化。对于某些设备，这可能包括连同驱动器自身在内的整个介质的验证。 用于格式化的 <i>force</i> 选项启动长格式化，在启动格式化之前不需要用户进行确认。 在 PCMCIA 内存卡中，所有这些选项都将启动长格式化。-l 列出所有可移除的设备。缺省情况下，如果不使用任何选项，rmformat 也会列出所有可移除的设备。如果给定了 <i>dev_name</i>，则 rmformat 将列出与 <i>dev_name</i> 关联的设备。输出中会显示设备路径名、供应商信息和设备类型。

`-s filename`

使用户可以在 SUNOS 标签中安排分区信息。

用户应提供一个文件作为输入，其中以某种格式包含了关于每个分片的信息，该格式提供了字节偏移量、所需大小、标记和标志，如下所示：

```
slices: n = offset, size [, flags, tags]
```

其中 n 是分片编号，*offset* 是开始分片 n 时的字节偏移量，*size* 是分片 n 所需的大小。*offset* 和 *size* 都必须是 512 字节的倍数。这些数字可以采用十进制、十六进制或八进制数字来提供。不支持浮点数字。有关分片的最大编号的详细信息，请参见《Oracle Solaris 管理：常见任务》。

要以千字节、兆字节或千兆字节指定 *size* 或 *offset*，请分别添加 KB、MB、GB。没有后缀的数字被视为字节偏移量。提供的标志如下所示：

```
wm = read-write, mountable
wu = read-write, unmountable
ru = read-only, unmountable
```

提供的标记如下所

示：unassigned、boot、root、swap、usr、backup、stand、var、home、alternates。

如果不需要对这些值进行更精细的控制，则可以在四元组中省略标记和标志。需要同时省略两者或同时包含两者。如果在某个特定分片的四元组中省略了标记和标志，则会为每个标记和标志假定一个缺省值。标志的缺省值是 *wm*，标记的缺省值是 *unassigned*。

可以提供完整的标记名称，也可以使用标记的缩写。缩写可以是标准标记名称的前两个或前几个字母。在处理所定义的标记和标志时，`rmformat` 区分大小写。

分片的指定按以下格式进行分隔：

例如：

```
slices: 0 = 0, 30MB, "wm", "home" :
        1 = 30MB, 51MB :
        2 = 0, 100MB, "wm", "backup" :
        6 = 81MB, 19MB
```

`rmformat` 会执行必要的检查，检测是否存在任何重叠分区，以及是否有非法请求试图访问超出目标介质的容量的地址。每个分片 n 只可以具有一个分片信息条目。如果为同一个分片 n 提供了多个分片信息条目，则会显示相应的错误消息。分片 2 是备份分片，其中包含整个磁盘的容量。可以使用井号字符 `#` 在输入文件中描述一行注释。如果行以 `#` 开始，则 `rmformat` 会忽略 `#` 之后直到该行结尾的所有字符。

对容量非常小的介质进行分区是允许的，但是在这些设备上使用此选项时要慎重。

-U

在任意文件系统上执行 `umount`，然后格式化。请参见 [mount\(1M\)](#)。此选项卸载所有已挂载的分片，并且对所请求的设备执行长格式化操作。

-V read | write

在格式化后验证介质的每个块。`write` 验证是一种具有破坏性的机制。在开始验证前，会要求用户进行确认。此选项的输出是被识别为坏块的块编号的列表。

`read` 验证仅对块进行验证，并且报告容易出错的块。

可以将所显示的块编号列表与 `-c` 选项配合使用来进行修复。

操作数

支持下列操作数：

devname

devname 可以是绝对设备路径名，或者是设备相对于当前工作目录的相对路径名，或者是别名，如 `cdrom` 或 `rmdisk`。

对于没有运行卷管理的系统，用户也可以提供绝对设备路径名（如 `/dev/rdisk/c?t?d?s?`）或相对于当前工作目录的合适的相对设备路径名。

示例

示例 1 为 PCFS 文件系统格式化可移除介质

以下示例显示了如何创建可选的 `fdisk` 分区：

```
example$ rmformat -F quick /dev/rdisk/c0t4d0s2:c
Formatting will erase all the data on disk.
Do you want to continue? (y/n)y
example$ su
# fdisk /dev/rdisk/c0t4d0s2:c
# mkfs -F pcfs /dev/rdisk/c0t4d0s2:c
Construct a new FAT file system on /dev/rdisk/c0t4d0s2:c: (y/n)? y
#
```

文件*/dev/aliases*

用于使用相应别名称针对在卷管理的控制下的不同介质提供字符设备的符号链接的目录。

/dev/dsk

用于针对 PCMCIA 内存卡和 `ata` 卡以及可移除介质设备提供块设备访问的目录。

/dev/rdisk

用于针对 PCMCIA 内存卡和 `ata` 卡以及可移除介质设备提供字符设备访问的目录。

/dev/aliases/pcmemS

插槽 `S` 中的 PCMCIA 内存卡的字符设备的符号链接，其中 `S` 表示 PCMCIA 插槽编号。

/dev/aliases/rmdisk0

除 CD-ROM、DVD-ROM、PCMCIA 内存卡等等之外的一般可移除介质设备的符号链接。

`/dev/rdisk`

用于针对 PCMCIA 内存卡和 ata 卡以及其他可移除设备提供字符设备访问的目录。

`/dev/dsk`

用于针对 PCMCIA 内存卡和 ata 卡以及其他可移除介质设备提供块设备访问的目录。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/storage/media-volume-manageR

另请参见

[cpio\(1\)](#)、[eject\(1\)](#)、[tar\(1\)](#)、[volcheck\(1\)](#)、[volrmount\(1\)](#)、[format\(1M\)](#)、[mkfs_pcfs\(1M\)](#)、[mou](#)

《Oracle Solaris 管理：常见任务》

附注

在基于 SPARC 的系统上使用 [newfs\(1M\)](#) 创建的包含 `ufs` 文件系统的可写入介质或者 PCMCIA 内存卡或 PCMCIA ata 卡与在基于 x86 的系统上创建的包含 `ufs` 文件系统的可写入介质或者 PCMCIA 内存卡不同。不要在这些平台之间交换包含 `ufs` 的任何可移除介质；请使用 [cpio\(1\)](#) 或 [tar\(1\)](#) 在这些平台之间传输内存卡上的文件。有关可交换的文件系统，请参考 [pcfs\(7FS\)](#) 和 [udfs\(7FS\)](#)。

`rmformat` 可能不会列出虚拟环境中的所有可移除设备。

已知问题

目前，PCMCIA 内存卡不支持坏扇区映射。因此，如果 `rmformat` 发现错误 (`bad sector`)，则内存卡便不可使用。

引用名	rmmount, rmumount – 挂载和卸载可移除介质
用法概要	<pre>rmmount [-u] [-o options] [nickname device] [mount_point] rmmount [-d] [-l] rmumount [nickname mount_point device] rmumount [-d] [-l]</pre>
描述	<p>rmmount 和 rmumount 实用程序挂载和卸载可移除卷或可热插拔卷。可选参数可以通过卷标签、挂载点或块设备路径来标识卷。</p> <p>如果用户具有足够的特权来覆盖缺省的挂载选项，则 rmmount 还可以接受附加的挂载选项。</p> <p>卸载可移除介质不会使介质弹出。使用 eject(1) 可以卸载并弹出介质。</p>
选项	<p>rmmount 和 rmumount 支持以下选项：</p> <ul style="list-style-type: none">-d 显示缺省设备的设备路径。如果未提供参数，则使用此设备。-l 显示可挂载设备的路径和别名。 <p>rmmount 仅支持以下选项：</p> <ul style="list-style-type: none">-o options 显示挂载选项。要使用此选项，用户必须具有可覆盖系统缺省选项的特权。-u 卸载卷，而不是挂载卷。
操作数	<p>支持下列操作数：</p> <ul style="list-style-type: none"><i>device</i> 通过设备在目录 /dev 中显示的名称，指定要挂载或卸载的设备。<i>mount_point</i> 通过设备在目录 /dev 中显示的名称，指定要挂载或卸载的设备。<i>nickname</i> 通过此命令可以识别的设备别名，指定要挂载或卸载的设备。
示例	<p>示例 1 挂载 USB 磁盘</p> <p>以下示例将挂载卷标签为 PHOTOS 的 USB 磁盘：</p> <pre>example% rmmount PHOTOS</pre> <p>示例 2 卸载 pcfs 卷</p> <p>以下示例根据设备路径卸载 pcfs 卷：</p> <pre>example% rmumount /dev/dsk/c4t0d0p0:1</pre>
退出状态	<p>将返回以下退出值：</p> <ul style="list-style-type: none">0 成功完成。

>0 出现错误。

文件

/media 缺省挂载根。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/storage/media-volume-manager
接口稳定性	Uncommitted（未确定）

另请参见

[eject\(1\)](#)、[attributes\(5\)](#)

引用名	roffbib – format and print a bibliographic database
用法概要	roffbib [-e] [-h] [-m <i>filename</i>] [-np] [-olist] [-Q] [-ra <i>N</i>] [-s <i>N</i>] [-T <i>term</i>] [-V] [-x] [<i>filename</i>] ...
描述	roffbib prints out all records in a bibliographic database, in bibliography format rather than as footnotes or endnotes. Generally it is used in conjunction with sortbib(1) : example% sortbib database roffbib
选项	roffbib accepts all options understood by nroff(1) except -i and -q. <ul style="list-style-type: none"> -e Produce equally-spaced words in adjusted lines using full terminal resolution. -h Use output tabs during horizontal spacing to speed output and reduce output character count. TAB settings are assumed to be every 8 nominal character widths. -m <i>filename</i> Prepend the macro file /usr/share/lib/tmac/tmac.<i>name</i> to the input files. There should be a space between the -m and the macro filename. This set of macros will replace the ones defined in /usr/share/lib/tmac/tmac.bib. -np Number first generated page <i>p</i>. -olist Print only page numbers that appear in the comma-separated <i>list</i> of numbers and ranges. A range <i>N–M</i> means pages <i>N</i> through <i>M</i>; an initial -<i>N</i> means from the beginning to page <i>N</i>; a final <i>N–</i> means from page <i>N</i> to end. -Q Queue output for the phototypesetter. Page offset is set to 1 inch. -ra<i>N</i> Set register <i>a</i> (one-character) to <i>N</i>. The command-line argument -r<i>N</i>1 will number the references starting at 1. Four command-line registers control formatting style of the bibliography, much like the number registers of ms(5). The flag -rv2 will double space the bibliography, while -rv1 will double space references but single space annotation paragraphs. The line length can be changed from the default 6.5 inches to 6 inches with the -rL6i argument, and the page offset can be set from the default of 0 to one inch by specifying -r01i (capital O, not zero). -s<i>N</i> Halt prior to every <i>N</i> pages for paper loading or changing (default <i>N</i>=1). To resume, enter NEWLINE or RETURN. -T<i>term</i> Specify <i>term</i> as the terminal type. -V Send output to the Versatec. Page offset is set to 1 inch. -x If abstracts or comments are entered following the %X field key, roffbib will format them into paragraphs for an annotated bibliography. Several %X fields may be given if several annotation paragraphs are desired.

文件 `/usr/share/lib/tmac/tmac.bib` file of macros used by `nroff/troff`

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

另请参见 [addbib\(1\)](#), [indxbib\(1\)](#), [lookbib\(1\)](#), [nroff\(1\)](#) [refer\(1\)](#), [sortbib\(1\)](#), [troff\(1\)](#), [attributes\(5\)](#)

已知问题 Users have to rewrite macros to create customized formats.

引用名 roles – 输出授予用户的角色

用法概要 roles [*user*]...

描述 roles 命令在标准输出上输出授予您或另外指定的用户的角色。角色是特殊帐户，对应于某个功能性职责而不是对应于实际的人员（称为普通用户）。

每个用户可以有零个或多个角色。角色具有普通用户的大多数属性，并且像普通用户一样是在 [passwd\(4\)](#) 和 [shadow\(4\)](#) 中标识的。每个角色在 [user_attr\(4\)](#) 文件中都必须有一个将其标识为角色的条目。角色可以有自己的授权和配置文件。有关下列属性的描述，请参见 [auths\(1\)](#) 和 [profiles\(1\)](#)。

不允许角色作为主用户登录到系统中。相反，用户必须以其自己的身份登录并担任该角色。角色的操作被视为系普通用户所为。当启用了审计时，角色的被审计事件包含担任该角色的原始用户的审计 ID。

角色不能担任其自身或任何其他角色。角色没有层次。不过，权限配置文件（请参见 [prof_attr\(4\)](#)）是分层次的，并且可用于实现与分层角色相同的效果。

可以使用 [su\(1M\)](#)、[ssh\(1\)](#) 或某个支持 PAM_AUSER 变量的其他服务来执行角色担任。成功的担任需要角色验证和成员身份。角色验证可能需要用户口令或角色口令，具体取决于角色的 [user_attr\(4\)](#) 条目中 roleauth 属性的设置。缺省情况下，需要角色的口令。通常向角色分配配置文件 shell。按照惯例，在标准 shell 的名称前添加 pf 来指定配置文件 shell，例如，pfbash。角色分配是在 [user_attr\(4\)](#) 中指定的。

示例 示例 1 样例输出

roles 命令的输出具有以下格式：

```
example% roles tester01 tester02
tester01 : admin
tester02 : secadmin, root
example%
```

退出状态 将返回以下退出值：

- 0 成功完成。
- 1 出现错误。

文件 /etc/user_attr
/etc/security/auth_attr
/etc/security/prof_attr

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[auths\(1\)](#)、[pfexec\(1\)](#)、[profiles\(1\)](#)、[rlogin\(1\)](#)、[ssh\(1\)](#)、[su\(1M\)](#)、[auth_attr\(4\)](#)、[passwd\(4\)](#)、[p](#)

引用名 rpcgen – an RPC protocol compiler

用法概要 rpcgen *infile*

```
rpcgen [-a] [-A] [-b] [-C] [-D name [= value]] [-i size]
       [-I [-K seconds]] [-L] [-M] [-N] [-T] [-v]
       [-Y pathname] infile
```

```
rpcgen [-c | -h | -l | -m | -t | -Sc | -Ss | -Sm]
       [-o outfile] [infile]
```

```
rpcgen [-s nettype] [-o outfile] [infile]
```

```
rpcgen [-n netid] [-o outfile] [infile]
```

描述 The `rpcgen` utility is a tool that generates C code to implement an RPC protocol. The input to `rpcgen` is a language similar to C known as RPC Language (Remote Procedure Call Language).

The `rpcgen` utility is normally used as in the first synopsis where it takes an input file and generates three output files. If the *infile* is named `proto.x`, then `rpcgen` generates a header in `proto.h`, XDR routines in `proto_xdr.c`, server-side stubs in `proto_svc.c`, and client-side stubs in `proto_clnt.c`. With the `-T` option, it also generates the RPC dispatch table in `proto_tbl.i`.

`rpcgen` can also generate sample client and server files that can be customized to suit a particular application. The `-Sc`, `-Ss`, and `-Sm` options generate sample client, server and makefile, respectively. The `-a` option generates all files, including sample files. If the *infile* is `proto.x`, then the client side sample file is written to `proto_client.c`, the server side sample file to `proto_server.c` and the sample makefile to `makefile.proto`.

The server created can be started both by the port monitors (for example, `inetd`) or by itself. When it is started by a port monitor, it creates servers only for the transport for which the file descriptor `0` was passed. The name of the transport must be specified by setting up the environment variable `PM_TRANSPORT`. When the server generated by `rpcgen` is executed, it creates server handles for all the transports specified in the `NETPATH` environment variable, or if it is unset, it creates server handles for all the visible transports from the `/etc/netconfig` file. Note: the transports are chosen at run time and not at compile time. When the server is self-started, it backgrounds itself by default. A special define symbol `RPC_SVC_FG` can be used to run the server process in foreground.

The second synopsis provides special features which allow for the creation of more sophisticated RPC servers. These features include support for user-provided `#defines` and RPC dispatch tables. The entries in the RPC dispatch table contain:

- pointers to the service routine corresponding to that procedure
- a pointer to the input and output arguments
- the size of these routines

A server can use the dispatch table to check authorization and then to execute the service routine. A client library can use the dispatch table to deal with the details of storage management and XDR data conversion.

The other three synopses shown above are used when one does not want to generate all the output files, but only a particular one. See the EXAMPLES section below for examples of rpcgen usage. When rpcgen is executed with the `-s` option, it creates servers for that particular class of transports. When executed with the `-n` option, it creates a server for the transport specified by *netid*. If *infile* is not specified, rpcgen accepts the standard input.

All the options mentioned in the second synopsis can be used with the other three synopses, but the changes are made only to the specified output file.

The C preprocessor `cc -E` is run on the input file before it is actually interpreted by rpcgen. For each type of output file, rpcgen defines a special preprocessor symbol for use by the rpcgen programmer:

RPC_HDR	defined when compiling into headers
RPC_XDR	defined when compiling into XDR routines
RPC_SVC	defined when compiling into server-side stubs
RPC_CLNT	defined when compiling into client-side stubs
RPC_TBL	defined when compiling into RPC dispatch tables

Any line beginning with “%” is passed directly into the output file, uninterpreted by rpcgen, except that the leading “%” is stripped off. To specify the path name of the C preprocessor, use the `-Y` flag.

For every data type referred to in *infile*, rpcgen assumes that there exists a routine with the string `xdr_` prepended to the name of the data type. If this routine does not exist in the RPC/XDR library, it must be provided. Providing an undefined data type allows customization of XDR routines.

Server Error Reporting By default, errors detected by `proto_svc.c` is reported to standard error and/or the system log.

This behavior can be overridden by compiling the file with a definition of `RPC_MSGOUT`, for example, `-DRPC_MSGOUT=mymsgfunc`. The function specified is called to report errors. It must conform to the following `printf`-like signature:

```
extern void RPC_MSGOUT(const char *fmt, ...);
```

选项

The following options are supported:

`-a` Generates all files, including sample files.

- A Enables the Automatic MT mode in the server main program. In this mode, the RPC library automatically creates threads to service client requests. This option generates multithread-safe stubs by implicitly turning on the -M option. Server multithreading modes and parameters can be set using the `rpc_control(3NSL)` call. rpcgen generated code does not change the default values for the Automatic MT mode.
- b Backward compatibility mode. Generates transport-specific RPC code for older versions of the operating system.
- c Compiles into XDR routines.
- C Generates header and stub files which can be used with ANSIC compilers. Headers generated with this flag can also be used with C++ programs.
- Dname[=value] Defines a symbol *name*. Equivalent to the `#define` directive in the source. If no *value* is given, *value* is defined as 1. This option can be specified more than once.
- h Compiles into C data-definitions (a header). The -T option can be used in conjunction to produce a header which supports RPC dispatch tables.
- i size Size at which to start generating inline code. This option is useful for optimization. The default *size* is 5.
- I Compiles support for `inetd(1M)` in the server side stubs. Such servers can be self-started or can be started by `inetd`. When the server is self-started, it backgrounds itself by default. A special define symbol `RPC_SVC_FG` can be used to run the server process in foreground, or the user can simply compile without the -I option.
- If there are no pending client requests, the `inetd` servers exit after 120 seconds (default). The default can be changed with the -K option. All of the error messages for `inetd` servers are always logged with `syslog(3C)`.
- Note:* This option is supported for backward compatibility only. It should always be used in conjunction with the -b option which generates backward compatibility code. By default (that is, when -b is not specified), rpcgen generates servers that can be invoked through portmonitors.
- K seconds By default, services created using rpcgen and invoked through port monitors wait 120 seconds after servicing a request before exiting. That interval can be changed using the -K flag. To create a server that exits immediately upon servicing a request, use -K 0. To create a server that never exits, the appropriate argument is -K -1.

- When monitoring for a server, some portmonitors, *always* spawn a new process in response to a service request. If it is known that a server are used with such a monitor, the server should exit immediately on completion. For such servers, rpcgen should be used with `-K 0`.
- l Compiles into client-side stubs.
 - L When the servers are started in foreground, uses `syslog(3C)` to log the server errors instead of printing them on the standard error.
 - m Compiles into server-side stubs, but do not generate a “main” routine. This option is useful for doing callback-routines and for users who need to write their own “main” routine to do initialization.
 - M Generates multithread-safe stubs for passing arguments and results between rpcgen-generated code and user written code. This option is useful for users who want to use threads in their code.
 - N This option allows procedures to have multiple arguments. It also uses the style of parameter passing that closely resembles C. So, when passing an argument to a remote procedure, you do not have to pass a pointer to the argument, but can pass the argument itself. This behavior is different from the old style of rpcgen-generated code. To maintain backward compatibility, this option is not the default.
 - n *netid* Compiles into server-side stubs for the transport specified by *netid*. There should be an entry for *netid* in the `netconfig` database. This option can be specified more than once, so as to compile a server that serves multiple transports.
 - o *outfile* Specifies the name of the output file. If none is specified, standard output is used (`-c`, `-h`, `-l`, `-m`, `-n`, `-s`, `-Sc`, `-Sm`, `-Ss`, and `-t` modes only).
 - s *nettype* Compiles into server-side stubs for all the transports belonging to the class *nettype*. The supported classes are `netpath`, `visible`, `circuit_n`, `circuit_v`, `datagram_n`, `datagram_v`, `tcp`, and `udp` (see `rpc(3NSL)` for the meanings associated with these classes). This option can be specified more than once. *Note:* The transports are chosen at run time and not at compile time.
 - Sc Generates sample client code that uses remote procedure calls.
 - Sm Generates a sample Makefile which can be used for compiling the application.
 - Ss Generates sample server code that uses remote procedure calls.
 - t Compiles into RPC dispatch table.
 - T Generates the code to support RPC dispatch tables.

The options `-c`, `-h`, `-l`, `-m`, `-s`, `-Sc`, `-Sm`, `-Ss`, and `-t` are used exclusively to generate a particular type of file, while the options `-D` and `-T` are global and can be used with the other options.

`-v` Displays the version number.

`-Y pathname` Gives the name of the directory where `rpcgen` starts looking for the C preprocessor.

操作数

The following operand is supported:

infile input file

示例

示例 1 Generating the output files and dispatch table

The following entry

```
example% rpcgen -T prot.x
```

generates all the five files: `prot.h`, `prot_clnt.c`, `prot_svc.c`, `prot_xdr.c`, and `prot_tbl.i`.

示例 2 Sending headers to standard output

The following example sends the C data-definitions (header) to the standard output:

```
example% rpcgen -h prot.x
```

示例 3 Sending a test version

To send the test version of the `-DTEST`, server side stubs for all the transport belonging to the class `datagram_n` to standard output, use:

```
example% rpcgen -s datagram_n -DTEST prot.x
```

示例 4 Creating server side stubs

To create the server side stubs for the transport indicated by `netid` `tcp`, use:

```
example% rpcgen -n tcp -o prot_svc.c prot.x
```

退出状态

`0` Successful operation.

`>0` An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer/utilities

另请参见

`inetd(1M)`, `rpc(3NSL)`, `rpc_control(3NSL)`, `rpc_svc_calls(3NSL)`, `syslog(3C)`,
`netconfig(4)`, `attributes(5)`

The `rpcgen` chapter in the 《[ONC+ Developer's Guide](#)》 manual.

引用名 rpm2cpio – 将 Red Hat 软件包 (Red Hat Package, RPM) 转换成 cpio 归档

用法概要 rpm2cpio [*file.rpm*]

描述 rpm2cpio 实用程序将指定为其唯一参数的 .rpm 文件在标准输出上转换为 cpio 归档。(请参见“附注”部分。) 如果未给定参数, 则从标准输入读取 rpm 流。在上述两种情况下, 如果标准输出是一个终端, 则 rpm2cpio 都将失败并输出一条用法消息。因此, 输出通常重定向到某个文件或通过 [cpio\(1\)](#) 实用程序进行管道传输。

示例 示例1 转换 rpm 文件

```
example% rpm2cpio Device3Dfx-1.1-2.src.rpm | cpio -itv
CPIO archive found!
-rw-r--r-- 1 root root 2635 Sep 13 16:39 1998, 3dfx.gif
-rw-r--r-- 1 root root 11339 Sep 27 16:03 1998, Dev3Dfx.tar.gz
-rw-r--r-- 1 root root 1387 Sep 27 16:04 1998, Device3Dfx-1.1-2.spec
31 blocks
```

示例2 从标准输入进行转换

```
example% rpm2cpio < Device3Dfx-1.1-2.src.rpm | cpio -itv
CPIO archive found!
-rw-r--r-- 1 root root 2635 Sep 13 16:39 1998, 3dfx.gif
-rw-r--r-- 1 root root 11339 Sep 27 16:03 1998, Dev3Dfx.tar.gz
-rw-r--r-- 1 root root 1387 Sep 27 16:04 1998, Device3Dfx-1.1-2.spec
31 blocks
```

属性 有关下列属性的说明, 请参见 [attributes\(5\)](#):

属性类型	属性值
可用性	package/rpm

另请参见 [cpio\(1\)](#)、[attributes\(5\)](#)

附注 rpm2cpio 处理版本 3 和 4 的 RPM。

引用名 rsh, remsh, remote_shell – remote shell

用法概要

```
rsh [-n] [-a] [-K] [-PN | -PO] [-x] [-f | -F] [-l username]
      [-k realm] hostname command
```

```
rsh hostname [-n] [-a] [-K] [-PN | -PO] [-x] [-f | -F]
      [-l username] [-k realm] command
```

```
remsh [-n] [-a] [-K] [-PN | -PO] [-x] [-f | -F] [-l username]
      [-k realm] hostname command
```

```
remsh hostname [-n] [-a] [-K] [-PN | -PO] [-x] [-f | -F]
      [-l username] [-k realm] command
```

```
hostname [-n] [-a] [-PN | -PO] [-x] [-f | -F]
      [-l username] [-k realm] command
```

描述

The rsh utility connects to the specified *hostname* and executes the specified *command*. rsh copies its standard input to the remote command, the standard output of the remote command to its standard output, and the standard error of the remote command to its standard error. Interrupt, quit, and terminate signals are propagated to the remote command. rsh normally terminates when the remote command does.

The user can opt for a secure session of rsh which uses Kerberos V5 for authentication. Encryption of the network session traffic is also possible. The rsh session can be kerberized using any of the following Kerberos specific options: -a, -PN or -PO, -x, -f or -F, and -k *realm*. Some of these options (-a, -x, -PN or -PO, and -f or -F) can also be specified in the [appdefaults] section of [krb5.conf\(4\)](#). The usage of these options and the expected behavior is discussed in the OPTIONS section below. If Kerberos authentication is used, authorization to the account is controlled by rules in [krb5_auth_rules\(5\)](#). If this authorization fails, fallback to normal rsh using rhosts occurs only if the -PO option is used explicitly on the command line or is specified in [krb5.conf\(4\)](#). Also, the -PN or -PO, -x, -f or -F, and -k *realm* options are just supersets of the -a option.

If you omit *command*, instead of executing a single command, rsh logs you in on the remote host using [rlogin\(1\)](#).

rsh does not return the exit status code of *command*.

Shell metacharacters which are not quoted are interpreted on the local machine, while quoted metacharacters are interpreted on the remote machine. See EXAMPLES.

If there is no locale setting in the initialization file of the login shell (.cshrc, ...) for a particular user, rsh always executes the command in the “C” locale instead of using the default locale of the remote machine.

The command is sent unencrypted to the remote system. All subsequent network session traffic is encrypted. See -x.

选项

The following options are supported:

- a Explicitly enable Kerberos authentication and trusts the `.k5login` file for access-control. If the authorization check by `in.rshd(1M)` on the server-side succeeds and if the `.k5login` file permits access, the user is allowed to carry out the command.
- f Forward a copy of the local credentials (Kerberos Ticket Granting Ticket) to the remote system. This is a non-forwardable ticket granting ticket. Forward a ticket granting ticket if you need to authenticate yourself to other Kerberized network services on the remote host. An example would be if your home directory on the remote host is NFS mounted by way of Kerberos V5. If your local credentials are not forwarded in this case, you cannot access your home directory. This option is mutually exclusive with the `-F` option.
- F Forward a forwardable copy of the local credentials (Kerberos Ticket Granting Ticket) to the remote system. The `-F` option provides a superset of the functionality offered by the `-f` option. For example, with the `-f` option, if after you connected to the remote host, your remote command attempted to invoke `/usr/bin/ftp`, `/usr/bin/telnet`, `/usr/bin/rlogin`, or `/usr/bin/rsh`, with the `-f` or `-F` options, the attempt would fail. Thus, you would be unable to push your single network sign on trust beyond one system. This option is mutually exclusive with the `-f` option.
- k *realm* Causes `rsh` to obtain tickets for the remote host in *realm* instead of the remote host's realm as determined by `krb5.conf(4)`.
- K This option explicitly disables Kerberos authentication. It can be used to override the `autoLogin` variable in `krb5.conf(4)`.
- l *username* Uses *username* as the remote username instead of your local username. In the absence of this option, the remote username is the same as your local username.
- n Redirect the input of `rsh` to `/dev/null`. You sometimes need this option to avoid unfortunate interactions between `rsh` and the shell which invokes it. For example, if you are running `rsh` and invoke a `rsh` in the background without redirecting its input away from the terminal, it blocks even if no reads are posted by the remote command. The `-n` option prevents this.
- PO
- PN Explicitly request new (`-PN`) or old (`-PO`) version of the Kerberos “rcmd” protocol. The new protocol avoids many security problems prevalent in the old one and is regarded much more secure, but is not interoperable with older (MIT/SEAM) servers. The new protocol is used by default, unless explicitly specified using these options or through `krb5.conf(4)`. If Kerberos

authorization fails when using the old “rcmd” protocol, there is fallback to regular, non-kerberized rsh. This is not the case when the new, more secure “rcmd” protocol is used.

-x Cause the network session traffic to be encrypted. See DESCRIPTION.

The type of remote shell (sh, rsh, or other) is determined by the user's entry in the file `/etc/passwd` on the remote system.

操作数

The following operand is supported:

command The command to be executed on the specified *hostname*.

用法

See [largefile\(5\)](#) for the description of the behavior of rsh and remsh when encountering files greater than or equal to 2 Gbyte (2³¹ bytes).

The rsh and remsh commands are IPv6-enabled. See [ip6\(7P\)](#). IPv6 is not currently supported with Kerberos V5 authentication.

Hostnames are given in the *hosts* database, which can be contained in the `/etc/hosts` file, the Internet domain name database, or both. Each host has one official name (the first name in the database entry) and optionally one or more nicknames. Official hostnames or nicknames can be given as *hostname*.

If the name of the file from which rsh is executed is anything other than rsh, rsh takes this name as its *hostname* argument. This allows you to create a symbolic link to rsh in the name of a host which, when executed, invokes a remote shell on that host. By creating a directory and populating it with symbolic links in the names of commonly used hosts, then including the directory in your shell's search path, you can run rsh by typing *hostname* to your shell.

If rsh is invoked with the basename remsh, rsh checks for the existence of the file `/usr/bin/remsh`. If this file exists, rsh behaves as if remsh is an alias for rsh. If `/usr/bin/remsh` does not exist, rsh behaves as if remsh is a host name.

For the kerberized rsh session, each user can have a private authorization list in a file `.k5login` in their home directory. Each line in this file should contain a Kerberos principal name of the form *principal/instance@realm*. If there is a `~/k5login` file, then access is granted to the account if and only if the originater user is authenticated to one of the principals named in the `~/k5login` file. Otherwise, the originating user is granted access to the account if and only if the authenticated principal name of the user can be mapped to the local account name using the *authenticated-principal-name* → *local-user-name* mapping rules. The `.k5login` file (for access control) comes into play only when Kerberos authentication is being done.

For the non-secure rsh session, each remote machine can have a file named `/etc/hosts.equiv` containing a list of trusted hostnames with which it shares usernames. Users with the same username on both the local and remote machine can run rsh from the machines listed in the remote machine's `/etc/hosts.equiv` file. Individual users can set up a

similar private equivalence list with the file `.rhosts` in their home directories. Each line in this file contains two names: a hostname and a username separated by a space. The entry permits the user named `username` who is logged into `hostname` to use `rsh` to access the remote machine as the remote user. If the name of the local host is not found in the `/etc/hosts.equiv` file on the remote machine, and the local username and hostname are not found in the remote user's `.rhosts` file, then the access is denied. The hostnames listed in the `/etc/hosts.equiv` and `.rhosts` files must be the official hostnames listed in the `hosts` database; nicknames can not be used in either of these files.

You cannot log in using `rsh` as a trusted user from a trusted hostname if the trusted user account is locked.

`rsh` does not prompt for a password if access is denied on the remote machine unless the *command* argument is omitted.

示例

示例 1 Using `rsh` to Append Files

The following command appends the remote file `lizard.file` from the machine called `lizard` to the file called `example.file` on the machine called `example`:

```
example% rsh lizard cat lizard.file >> example.file
```

The following command appends the file `lizard.file` on the machine called `lizard` to the file `lizard.file2` which also resides on the machine called `lizard`:

```
example% rsh lizard cat lizard.file ">>" lizard.file2
```

退出状态

The following exit values are returned:

0 Successful completion.

1 An error occurred.

文件

<code>/etc/hosts</code>	Internet host table
<code>/etc/hosts.equiv</code>	Trusted remote hosts and users
<code>/etc/passwd</code>	System password file
<code>\$HOME/.k5login</code>	File containing Kerberos principals that are allowed access
<code>/etc/krb5/krb5.conf</code>	Kerberos configuration file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients
CSI	Enabled

另请参见

[on\(1\)](#), [rlogin\(1\)](#), [ssh\(1\)](#), [telnet\(1\)](#), [vi\(1\)](#), [in.rshd\(1M\)](#), [hosts\(4\)](#), [hosts.equiv\(4\)](#), [krb5.conf\(4\)](#), [attributes\(5\)](#), [krb5_auth_rules\(5\)](#), [largefile\(5\)](#), [ip6\(7P\)](#)

附注

When a system is listed in `hosts.equiv`, its security must be as good as local security. One insecure system listed in `hosts.equiv` can compromise the security of the entire system.

You cannot run an interactive command (such as [vi\(1\)](#)). Use `rlogin` if you wish to do this.

Stop signals stop the local `rsh` process only. This is arguably wrong, but currently hard to fix for reasons too complicated to explain here.

The current local environment is not passed to the remote shell.

Sometimes the `-n` option is needed for reasons that are less than obvious. For example, the command:

```
example% rsh somehost dd if=/dev/nrmt0 bs=20b | tar xvpBf -
```

puts your shell into a strange state. Evidently, the `tar` process terminates before the `rsh` process. The `rsh` command then tries to write into the "broken pipe" and, instead of terminating neatly, proceeds to compete with your shell for its standard input. Invoking `rsh` with the `-n` option avoids such incidents.

This bug occurs only when `rsh` is at the beginning of a pipeline and is not reading standard input. Do not use the `-n` option if `rsh` actually needs to read standard input. For example:

```
example% tar cf - . | rsh sundial dd of=/dev/rmt0 obs=20b
```

does not produce the bug. If you were to use the `-n` option in a case like this, `rsh` would incorrectly read from `/dev/null` instead of from the pipe.

For most purposes, [ssh\(1\)](#) is preferred over `rsh`.

引用名 runat - 在扩展属性名称空间中执行命令

用法概要 /usr/bin/runat *file* [*command*]

描述 runat 实用程序用于在文件的隐藏属性目录中执行 shell 命令。实际上，此实用程序将当前工作目录更改为与文件参数关联的隐藏属性目录，然后在 bourne shell (/bin/sh) 中执行指定的命令。如果未提供 *command* 参数，则会派生一个交互式 shell。环境变量 \$SHELL 定义了要派生的 shell。如果未定义该变量，则将使用缺省 shell (/bin/sh)。

file 参数可以是能够支持扩展属性的任何文件（包括目录）。在调用 runat 命令之前，不需要此文件具有任何属性，也不需要以任何方式准备此文件。

操作数 支持下列操作数：

file 可以支持扩展属性的任何文件（包括目录）。

command 要在属性目录中执行的命令。

错误 如果 runat 无法访问 *file* 参数，或者 *file* 参数不支持扩展属性，将返回一个非零值退出状态。

用法 有关扩展文件属性的详细说明，请参见 [fsattr\(5\)](#)。

runat 命令创建的进程上下文将其当前工作目录设置为包含文件扩展属性的隐藏目录。该目录的父代（“..”项）始终是指在命令行上提供的文件。因此，它可能不是目录。因此，依赖于格式正确的父代（即目录）的命令（如 `pwd`）可能会失败。

缺少 *command* 参数时，runat 将派生一个新的交互式 shell，其当前工作目录设置为给定文件的隐藏属性目录。请注意，如上所述，当目录的父代不是目录时，某些 shell（如 `zsh` 和 `tcsh`）将不能正常运行。这些 shell 不应该与 runat 一起使用。

示例 示例1 使用 runat 列出文件上的扩展属性

```
example% runat file.1 ls -l
example% runat file.1 ls
```

示例2 创建扩展属性

```
example% runat file.2 cp /tmp/attrdata attr.1
example% runat file.2 cat /tmp/attrdata > attr.1
```

示例3 将属性从一个文件复制到另一个文件

```
example% runat file.2 cat attr.1 | runat file.1 "cat > attr.1"
```

示例4 使用 runat 派生交互式 shell

```
example% runat file.3 /bin/sh
```

示例 4 使用 runat 派生交互式 shell (续)

这将在 `file.3` 的属性目录中派生一个新的 shell。请注意，该 shell 无法确定您的当前目录是什么。要离开该属性目录，请退出派生的 shell，或者使用绝对路径更改目录 (`cd`)。

用于执行基本属性操作的推荐方法：

display (显示)	<code>runat file ls [options]</code>
read	<code>runat file cat attribute</code>
create/modify (创建/修改)	<code>runat file cp absolute-file-path attribute</code>
delete	<code>runat file rm attribute</code>
permission changes (权限更改)	<code>runat file chmod mode attribute</code> <code>runat file chgrp group attribute</code> <code>runat file chown owner attribute</code>

interactive shell (交互式 shell)

```
runat file /bin/sh
或将您的 $SHELL 设置为 /bin/sh 并运行
runat file
```

以上列表包括了已知的可与 `runat` 配合使用的命令。虽然许多其他命令也可使用，但是不能保证此列表范围外的任何命令都能够使用。任何要求能够确定当前工作目录的命令都可能会失败。此类命令的示例如下所示：

示例 5 在属性目录中使用 `man`

```
example% runat file.1 man runat
>getcwd: Not a directory
```

示例 6 在属性目录中派生 `tcsh` shell

```
example% runat file.3 /usr/bin/tcsh
tcsh: Not a directory
tcsh: Trying to start from "/home/user"
```

已派生了一个新的 `tcsh` shell，其当前工作目录设置为用户的起始目录。

示例 7 在属性目录中派生 `zsh` shell

```
example% runat file.3 /usr/bin/zsh
example%
```

示例7 在属性目录中派生 zsh shell (续)

虽然命令看上去已运行，但是 zsh 实际上仅仅是将当前工作目录更改为了“/”。这可以通过使用 `/bin/pwd` 来查看：

```
example% /bin/pwd
/
```

环境变量 SHELL 指定 runat 要调用的命令 shell。

退出状态 将返回以下退出值：

125 *file* 参数引用的文件的属性目录不可访问。

126 无法执行所提供的 *command* 参数。

其他情况下，返回的退出状态是被调用来执行给定命令的 shell 的退出状态。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
CSI	Enabled (已启用)
接口稳定性	Committed (已确定)

另请参见 [open\(2\)](#)、[attributes\(5\)](#)、[fsattr\(5\)](#)

附注 对于无法确定当前工作目录时，命令在 runat 中为什么会失败，原因并不总是很明显的。错误结果可能是令人困惑的或是含糊的（请参见上述的 tcsh 和 zsh 示例）。

引用名 `rup` – show host status of remote machines (RPC version)

用法概要 `rup [-hlt]`
`rup [host]...`

描述 `rup` gives a status similar to `uptime` for remote machines. It broadcasts on the local network, and displays the responses it receives.

Normally, the listing is in the order that responses are received, but this order can be changed by specifying one of the options listed below.

When *host* arguments are given, rather than broadcasting `rup` will only query the list of specified hosts.

A remote host will only respond if it is running the `rstatd` daemon, which is normally started up from [inetd\(1M\)](#).

In the absence of a name service, such as LDAP or NIS, `rup` displays host names as numeric IP addresses.

选项 `-h` Sort the display alphabetically by host name.
`-l` Sort the display by load average.
`-t` Sort the display by up time.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

另请参见 [ruptime\(1\)](#), [inetd\(1M\)](#), [attributes\(5\)](#)

《安装 Oracle Solaris 11.1 系统》

已知问题 Broadcasting does not work through gateways.

引用名 `rup` – show host status of remote machines (RPC version)

用法概要 `rup [-hlt]`
`rup [host] . . .`

描述 `rup` gives a status similar to `uptime` for remote machines. It broadcasts on the local network, and displays the responses it receives.

Normally, the listing is in the order that responses are received, but this order can be changed by specifying one of the options listed below.

When *host* arguments are given, rather than broadcasting `rup` only queries the list of specified hosts.

A remote host will only respond if it is running the `rstatd` daemon, which is normally started up from [inetd\(1M\)](#).

选项 `-h` Sort the display alphabetically by host name.
`-l` Sort the display by load average.
`-t` Sort the display by up time.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/extended-system-utilities

另请参见 [ruptime\(1\)](#), [inetd\(1M\)](#), [attributes\(5\)](#)

已知问题 Broadcasting does not work through gateways.

引用名 ruptime – show host status of local machines

用法概要 ruptime [-ar] [-l | -t | -u]

描述 The ruptime utility gives a status line like uptime (see [uptime\(1\)](#)) for each machine on the local network; these are formed from packets broadcast by each host on the network approximately every three minutes.

Machines for which no status report has been received for 11 minutes are shown as being down.

Normally, the listing is sorted by host name, but this order can be changed by specifying one of the options listed below.

选项 The following options are supported:

- a Counts even those users who have been idle for an hour or more.
- r Reverses the sorting order.
- l | -t | -u These options are mutually exclusive. The use of one overrides the previous one(s).
 - l Sorts the display by load average.
 - t Sorts the display by up time.
 - u Sorts the display by number of users.

文件 /var/spool/rwho/whod.* data files

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

另请参见 [uptime\(1\)](#), [rwho\(1\)](#), [in.rwhod\(1M\)](#), [attributes\(5\)](#)

引用名 `rusage` – print resource usage for a command

用法概要 `/usr/ucb/rusage command`

描述 The `rusage` command is similar to [time\(1\)](#). It runs the given *command*, which must be specified; that is, *command* is not optional as it is in the C shell's timing facility. When the command is complete, `rusage` displays the real (wall clock), the system CPU, and the user CPU times which elapsed during execution of the command, plus other fields in the `rusage` structure, all on one long line. Times are reported in seconds and hundredths of a second.

示例 示例 1 The format of `rusage` output

The example below shows the format of `rusage` output.

```
example% rusage wc /usr/share/man/man1/csh (1)
3045 13423 78071 /usr/share/man/man1/csh (1)
2.26 real 0.80 user 0.36 sys 11 pf 38 pr 0 sw 11 rb 0 wb 16 vcx 37
icx 24 mx 0 ix 1230 id 9 is
example%
```

Each of the fields identified corresponds to an element of the `rusage` structure, as described in [getrusage\(3C\)](#), as follows:

<code>real</code>		elapsed real time
<code>user</code>	<code>ru_utime</code>	user time used
<code>sys</code>	<code>ru_stime</code>	system time used
<code>pf</code>	<code>ru_majflt</code>	page faults requiring physical I/O
<code>pr</code>	<code>ru_minflt</code>	page faults not requiring physical I/O
<code>sw</code>	<code>ru_nswap</code>	swaps
<code>rb</code>	<code>ru_inblock</code>	block input operations
<code>wb</code>	<code>ru_oublock</code>	block output operations
<code>vcx</code>	<code>ru_nvcsw</code>	voluntary context switches
<code>icx</code>	<code>ru_nivcsw</code>	involuntary context switches
<code>mx</code>	<code>ru_maxrss</code>	maximum resident set size
<code>ix</code>	<code>ru_ixrss</code>	currently 0
<code>id</code>	<code>ru_idrss</code>	integral resident set size
<code>is</code>	<code>ru_isrss</code>	currently 0

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见

[csh\(1\)](#), [time\(1\)](#), [getrusage\(3C\)](#), [attributes\(5\)](#)

已知问题

When the command being timed is interrupted, the timing values displayed may be inaccurate.

引用名 rusers – who is logged in on remote machines

用法概要 rusers [-ahilu] host...

描述 The rusers command produces output similar to [who\(1\)](#), but for remote machines. The listing is in the order that responses are received, but this order can be changed by specifying one of the options listed below.

The default is to print out the names of the users logged in. When the `-l` flag is given, additional information is printed for each user:

userid hostname:terminal login_date login_time idle_time login_host

If *hostname* and *login host* are the same value, the *login_host* field is not displayed. Likewise, if *hostname* is not idle, the *idle_time* is not displayed.

A remote host will only respond if it is running the rusersd daemon, which may be started up from [inetd\(1M\)](#).

In the absence of a name service, such as LDAP or NIS, rusers displays host names as numeric IP addresses.

选项

- a Give a report for a machine even if no users are logged on.
- h Sort alphabetically by host name.
- i Sort by idle time.
- l Give a longer listing in the style of [who\(1\)](#).
- u Sort by number of users.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

另请参见 [who\(1\)](#), [inetd\(1M\)](#), [attributes\(5\)](#)

引用名	rwho – who is logged in on local machines
用法概要	rwho [-a]
描述	<p>The <code>rwho</code> command produces output similar to who(1), but for all machines on your network. If no report has been received from a machine for 5 minutes, <code>rwho</code> assumes the machine is down, and does not report users last known to be logged into that machine.</p> <p>If a user has not typed to the system for a minute or more, <code>rwho</code> reports this idle time. If a user has not typed to the system for an hour or more, the user is omitted from the output of <code>rwho</code> unless the <code>-a</code> flag is given.</p>
选项	-a Report all users whether or not they have typed to the system in the past hour.
文件	<code>/var/spool/rwho/whod.*</code> information about other machines
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-servers

另请参见 [finger\(1\)](#), [ruptime\(1\)](#), [who\(1\)](#), [in.rwhod\(1M\)](#), [attributes\(5\)](#)

附注 `rwho` does not work through gateways.

The directory `/var/spool/rwho` must exist on the host from which `rwho` is run.

This service takes up progressively more network bandwidth as the number of hosts on the local net increases. For large networks, the cost becomes prohibitive.

The `rwho` service daemon, [in.rwhod\(1M\)](#), must be enabled for this command to return useful results.

引用名 sar – 系统活动报告程序

用法概要

```
sar [-aAbcdgkmpqruvwy] [-o filename] t [n]
sar [-aAbcdgkmpqruvwy] [-e time] [-f filename] [-i sec]
    [-s time]
```

描述 在第一个实例中，sar 实用程序在操作系统中以 *n* 次间隔（每个间隔 *t* 秒）对累积活动计数器进行抽样，其中 *t* 应该为 5 或者更大。如果使用多个选项指定 *t*，则所有标题会一起打印，输出会难以辨认。（如果抽样间隔小于 5，则 sar 自身的活动会影响样例。）如果指定了 -o 选项，则它以二进制格式在 *filename* 中保存样例。*n* 的缺省值为 1。

在第二个实例中，没有指定任何抽样间隔。sar 从之前记录的 *filename* 文件中提取数据，该文件可以是由 -f 选项指定的文件，也可以是缺省情况下当前日期 *dd* 的标准系统活动每日数据文件 `/var/adm/sa/sa dd`。报告的开始时间和结束时间可以使用 -e 和 -s 参数进行绑定，其中 *time* 使用 `hh[:mm[:ss]]` 格式指定。-i 选项以 *sec* 秒间隔选择记录。否则，将报告数据文件中找到的所有间隔。

选项 以下选项修改 sar 报告的信息子集。

- a 报告文件访问系统例程的使用：iget/s、namei/s、dirblk/s
- A 报告所有数据。等效于 -abcdgkmpqruvwy。
- b 报告缓冲区活动：
 - bread/s、bwrit/s 每秒在系统缓冲区和磁盘或其他块设备之间的数据传输。
 - lread/s、lwrit/s 系统缓冲区的访问。
 - %rcache、%wcache 高速缓存命中率，即 $(1 - \text{bread}/\text{lread})$ ，百分比形式。
 - pread/s、pwrit/s 使用原始（物理）设备机制进行传输。

如果在非全局区域中运行，并且池设备处于活动状态，则这些值反映向区域绑定到的池的处理器集合中处理器上的活动。
- c 报告系统调用：
 - scall/s 所有类型的系统调用。
 - sread/s、swrit/s、fork/s、exec/s 特定的系统调用。
 - rchar/s、wchar/s 由读取和写入系统调用传输的字符。没有报告任何传入或传出 `exec(2)` 和 `fork(2)` 调用。

如果在非全局区域中运行，并且池设备处于活动状态，则这些值反映向区域绑定到的池的处理器集合中处理器上的活动。

- d** 报告每个块设备（例如，磁盘或磁带机）的活动，除了 XDC 磁盘和磁带机。显示数据时，设备规范 *dsk-* 通常用于表示磁盘驱动器。用于表示磁带机的设备规范是与计算机有关的。报告的活动数据是：
- | | |
|--|--------------------------------------|
| <code>%busy</code> 、 <code>avque</code> | 设备忙于处理传输请求而花费的时间，在此时间内未解决的平均请求数。 |
| <code>read/s</code> 、 <code>write/s</code> 、 <code>blks/s</code> | 从设备或到设备的读取/写入传输数目，以 512 字节为单元传输的字节数。 |
| <code>avwait</code> | 以毫秒为单位的平均等待时间。 |
| <code>avserv</code> | 以毫秒为单位的平均服务时间。 |
- 有关更多常规系统统计信息，请使用 `iostat(1M)`、`sar(1M)` 或 `vmstat(1M)`。
- 有关磁盘的命名约定，请参见《[System Administration Guide: Advanced Administration](#)》。
- e time** 选择到 `time` 为止的数据。缺省值是 `18:00`。
- f filename** 将 `filename` 用作 `sar` 的数据源。缺省文件为当前每日数据文件 `/var/adm/sa/sadd`。
- g** 报告分页活动：
- | | |
|-----------------------|--|
| <code>pgout/s</code> | 每秒的页出请求。 |
| <code>ppgout/s</code> | 每秒进行页出操作的页。 |
| <code>pgfree/s</code> | 每秒由页窃取守护进程置于空闲表中的页。 |
| <code>pgscan/s</code> | 每秒由页窃取守护进程扫描的页。 |
| <code>%ufs_ipf</code> | 具有关联的可重用页的 <code>iget</code> 从空闲表中取消的 UFS inode 的百分比。这些页面被刷新，并且不能由进程回收。因此，这是具有页面刷新的 <code>igets</code> 的百分比。 |
- 如果在非全局区域中运行，并且池设备处于活动状态，则这些值反映向区域绑定到的池的处理器集合中处理器上的活动。
- i sec** 以尽可能接近 `sec` 秒的间隔选择数据。
- k** 报告内核内存分配 (KMA) 活动：
- | | |
|---|---|
| <code>sml_mem</code> 、 <code>alloc</code> 、 <code>fail</code> | 有关内存池保留和为小型请求分配空间的信息： <code>KMA</code> 针对小型池具有的内存量（以字节表示）、用于满足少量内存请求的已分配字节数，以及未满足的（失败的）少量内存请求数。 |
|---|---|

- lg_mem、alloc、fail 有关大型内存池的信息（类似于有关小型内存池的信息）。
- ovsz_alloc、fail 为过大请求分配的内存量以及无法满足的过大请求的数目（因为过大内存是动态分配的，所以没有池）。
- m 报告消息和信号活动：
msg/s、sema/s 每秒的基元。
如果在非全局区域中运行，并且池设备处于活动状态，则这些值反映向区域绑定到的池的处理器集合中处理器上的活动。
- o *filename* 将样例以二进制格式保存在文件 *filename* 中。
- p 报告分页活动：
at/s 每秒通过回收当前在内存中的页来满足的缺页（每秒附加数）。
pgin/s 每秒的页入请求。
ppgin/s 每秒进行页入操作的页。
pflt/s 每秒由保护错误引起的缺页（非法访问页面）或者“写复制”。
vflt/s 每秒的地址转换缺页（内存中不存在有效页）。
slock/s 每秒由需要物理 I/O 的软件锁定请求导致的错误。
如果在非全局区域中运行，并且池设备处于活动状态，则这些值反映向区域绑定到的池的处理器集合中处理器上的活动。
- q 报告占用时的平均队列长度以及占用的时间百分比：
runq-sz、%runocc 在内存中运行内核线程队列并且可运行
swpq-sz、%swpocc 交换进程队列
- r 报告未使用的内存页和磁盘块：
freemem 可用于用户进程的平均页数。
freeswap 可用于页交换的磁盘块。
- s *time* 选择晚于 *time*、格式为 *hh[:mm]* 的数据。缺省值是 08:00。
- u 报告 CPU 使用率（缺省）：
%usr、%sys、%wio、%idle 以用户模式运行、以系统模式运行、某些进程等待块 I/O 的空闲状态以及其他空闲状态的时间。

- 如果在非全局区域中运行，并且池设备处于活动状态，则这些值反映向区域绑定到的池的处理器集合中处理器上的活动。
- v** 报告进程状态、i-node、文件表：
 proc-sz、inod-sz、file-sz、ock-sz 每个表的项数/大小，在抽样点处计算一次。
 ov 在每个表的抽样点之间发生的溢出。
- w** 报告系统交换和切换活动：
 swpin/s、swpot/s、bswin/s、bswot/s 传输的数目以及为换入和换出传输的 512 字节单元的数目（包括初始装载某些程序）。
 pswch/s 进程切换。
- 如果在非全局区域中运行，并且池设备处于活动状态，则这些值反映向区域绑定到的池的处理器集合中处理器上的活动。
- y** 报告 TTY 设备活动：
 rawch/s、canch/s、outch/s 输入字符率、由 canon 处理的输入字符率、输出字符率。
 rcvin/s、xmtin/s、mdmin/s 接收、传输和调制解调器中断率。
- 如果在非全局区域中运行，并且池设备处于活动状态，则这些值反映向区域绑定到的池的处理器集合中处理器上的活动。

示例

示例1 查看系统活动

以下示例显示今天到目前为止的 CPU 活动：

```
example% sar
```

示例2 观察系统活动发展

要观察 CPU 活动在 10 分钟内的的发展并保存数据：

```
example% sar -o temp 60 10
```

示例3 查看磁盘和磁带活动

要稍后查看此时间段内的磁盘和磁带活动：

```
example% sar -d -f temp
```

文件 `/var/adm/sa/sadd` 每日数据文件，其中 *dd* 是用于表示月份日期的数字

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/accounting/legacy-accounting

另请参见 [iostat\(1M\)](#)、[sar\(1M\)](#)、[vmstat\(1M\)](#)、[exec\(2\)](#)、[fork\(2\)](#)、[attributes\(5\)](#)

《Oracle Solaris 管理：常见任务》

附注 由于百分比图生成过程中的舍入误差，CPU 使用率总量可能与 100 稍有不同。

引用名	sccs – front end for the Source Code Control System (SCCS)
用法概要	<pre> /usr/bin/sccs [-r] [-drootprefix] [-psubdir] subcommand [option]... [file]... /usr/xpg4/bin/sccs [-r] [-d rootprefix] [-p subdir] subcommand [option]... [file]... </pre>
描述	<p>The <code>sccs</code> command is a comprehensive, straightforward front end to the various utility programs of the Source Code Control System (SCCS).</p> <p><code>sccs</code> applies the indicated <i>subcommand</i> to the history file associated with each of the indicated files.</p> <p>The name of an SCCS history file is derived by prepending the 's.' prefix to the filename of a working copy. The <code>sccs</code> command normally expects these 's.' files to reside in an SCCS subdirectory. Thus, when you supply <code>sccs</code> with a <i>file</i> argument, it normally applies the subcommand to a file named <code>s.file</code> in the SCCS subdirectory. If <i>file</i> is a path name, <code>sccs</code> looks for the history file in the SCCS subdirectory of that file's parent directory. If <i>file</i> is a directory, however, <code>sccs</code> applies the subcommand to every <code>s.</code> file file it contains. Thus, the command:</p> <pre>example% sccs get program.c</pre> <p>would apply the <code>get</code> subcommand to a history file named <code>SCCS/s.program.c</code>, while the command:</p> <pre>example% sccs get SCCS</pre> <p>would apply it to every <code>s.</code> file in the SCCS subdirectory.</p> <p>Options for the <code>sccs</code> command itself must appear before the <i>subcommand</i> argument. Options for a given subcommand must appear after the <i>subcommand</i> argument. These options are specific to each subcommand, and are described along with the subcommands themselves (see Subcommands below).</p>
Running Setuid	The <code>sccs</code> command also includes the capability to run "setuid" to provide additional protection. However, this does not apply to subcommands such as <code>sccs-admin(1)</code> , since this would allow anyone to change the authorizations of the history file. Commands that would do so always run as the real user.
选项	The following options are supported:
<code>/usr/bin/sccs</code>	<code>-drootprefix</code>
<code>/usr/xpg4/bin/sccs</code>	<code>-d rootprefix</code> Defines the root portion of the path name for SCCS history files. The default root portion is the current directory. <i>rootprefix</i> is prepended to the entire <i>file</i> argument, even if <i>file</i> is an absolute path name. <code>-d</code> overrides any directory specified by the <code>PROJECTDIR</code> environment variable (see <code>ENVIRONMENT VARIABLES</code> below).
<code>/usr/bin/sccs</code>	<code>-psubdir</code>

`/usr/xpg4/bin/sccs` `-psubdir`
Defines the (sub)directory within which a history file is expected to reside. SCCS is the default. (See EXAMPLES below).

`-r`
Runs `sccs` with the real user ID, rather than set to the effective user ID.

操作数 The following operands are supported:

file
a file passed to *subcommand*

option
an option or option-argument passed to *subcommand*

subcommand
one of the subcommands listed in Usage

用法 The usage for `sccs` is described below.

Subcommands Many of the following `sccs` subcommands invoke programs that reside in `/usr/bin`. Many of these subcommands accept additional arguments that are documented in the reference page for the utility program the subcommand invokes.

`admin`
Modify the flags or checksum of an SCCS history file. Refer to [sccs-admin\(1\)](#) for more information about the `admin` utility. While `admin` can be used to initialize a history file, you might find that the `create` subcommand is simpler to use for this purpose.

`/usr/bin/sccs` `cdc -rsid [-y[comment]]`

`/usr/xpg4/bin/sccs` `cdc -rsid | -rsid [-y[comment]]`

Annotate (change) the delta commentary. Refer to [sccs-cdc\(1\)](#). The `fix` subcommand can be used to replace the delta, rather than merely annotating the existing commentary.

`-r sid | -rsid`
Specify the SCCS delta ID (SID) to which the change notation is to be added. The SID for a given delta is a number, in Dewey decimal format, composed of two or four fields: the *release* and *level* fields, and for branch deltas, the *branch* and *sequence* fields. For instance, the SID for the initial delta is normally 1.1.

`-y“[comment]”`
Specify the comment with which to annotate the delta commentary. If `-y` is omitted, `sccs` prompts for a comment. A null *comment* results in an empty annotation.

`/usr/bin/sccs` `check [-b] [-u[username]]`

/usr/xpg4/bin/sccs

check [-b] [-u *username*] [-U]

Check for files currently being edited. Like `info` and `tell`, but returns an exit code, rather than producing a listing of files. `check` returns a non-zero exit status if anything is being edited.

-b

Ignore branches.

-u[*username*] | -u [*username*] | -U

Check only files being edited by you. When *username* is specified, check only files being edited by that user. For `/usr/xpg4/bin/sccs`, the `-U` option is equivalent to `-u <current_user>`.

clean [-b]

Remove everything in the current directory that can be retrieved from an SCCS history. Does not remove files that are being edited.

-b

Do not check branches to see if they are being edited. '`clean -b`' is dangerous when branch versions are kept in the same directory.

comb

Generate scripts to combine deltas. Refer to [sccs-comb\(1\)](#).

create

Create (initialize) history files. `create` performs the following steps:

- Renames the original source file to `,program.c` in the current directory.
- Create the history file called `s.program.c` in the SCCS subdirectory.
- Performs an '`sccs get`' on `program.c` to retrieve a read-only copy of the initial version.

deledit [-s] [-y[*comment*]]

Equivalent to an '`sccs delta`' and then an '`sccs edit`'. `deledit` checks in a delta, and checks the file back out again, but leaves the current working copy of the file intact.

-s

Silent. Do not report delta numbers or statistics.

-y[*comment*]

Supply a comment for the delta commentary. If `-y` is omitted, `delta` prompts for a comment. A null *comment* results in an empty comment field for the delta.

delget [-s] [-y[*comment*]]

Perform an '`sccs delta`' and then an '`sccs get`' to check in a delta and retrieve read-only copies of the resulting new version. See the `deledit` subcommand for a description of `-s` and `-y`. `sccs` performs a `delta` on all the files specified in the argument list, and then a `get` on all the files. If an error occurs during the `delta`, the `get` is not performed.

delta [-s] [-y[*comment*]]

Check in pending changes. Records the line-by-line changes introduced while the file was checked out. The effective user ID must be the same as the ID of the person who has the file checked out. Refer to [sccs-delta\(1\)](#). See the `deledit` subcommand for a description of `-s`

and `-y`.

`/usr/bin/sccs diff [-C] [-I] [-c date-time] [-rsid] diff-options`

`/usr/xpg4/bin/sccs diff [-C] [-I] [-c date-time | -c date-time] [-r sid | -rsid] diff-options`

Compare (in [diff\(1\)](#) format) the working copy of a file that is checked out for editing, with a version from the SCCS history. Use the most recent checked-in version by default. The `diffs` subcommand accepts the same options as `diff`.

Any `-r`, `-c`, `-i`, `-x`, and `-t` options are passed to subcommand `get`. A `-C` option is passed to `diff` as `-c`. An `-I` option is passed to `diff` as `-i`.

`-c date-time | -c date-time`

Use the most recent version checked in before the indicated date and time for comparison. *date-time* takes the form: `yy[mm[dd[hh[mm[ss]]]]]`. Omitted units default to their maximum possible values; that is `-c7502` is equivalent to `-c750228235959`.

`-r sid | -rsid`

Use the version corresponding to the indicated delta for comparison.

`edit`

Retrieve a version of the file for editing. '`sccs edit`' extracts a version of the file that is writable by you, and creates a `p` file in the SCCS subdirectory as lock on the history, so that no one else can check that version in or out. ID keywords are retrieved in unexpanded form. `edit` accepts the same options as `get`, below. Refer to [sccs-get\(1\)](#) for a list of ID keywords and their definitions.

`enter`

Similar to `create`, but omits the final '`sccs get`'. This can be used if an '`sccs edit`' is to be performed immediately after the history file is initialized.

`/usr/bin/sccs fix -rsid`

`/usr/xpg4/bin/sccs fix -r sid | -rsid`

Revise a (leaf) delta. Remove the indicated delta from the SCCS history, but leave a working copy of the current version in the directory. This is useful for incorporating trivial updates for which no audit record is needed, or for revising the delta commentary. `fix` must be followed by a `-r` option, to specify the SID of the delta to remove. The indicated delta must be the most recent (leaf) delta in its branch. Use `fix` with caution since it does not leave an audit trail of differences (although the previous commentary is retained within the history file).

`/usr/bin/sccs get [-ekmps] [-Gnewname] [-c date-time] [-r [sid]]`

`/usr/xpg4/bin/sccs` `get [-ekmps] [-G newname | -Gnewname]`
`[-c date-time | -cdate-time] [-r sid | -rsid]`
 Retrieve a version from the SCCS history. By default, this is a read-only working copy of the most recent version. ID keywords are in expanded form. Refer to [sccs-get\(1\)](#), which includes a list of ID keywords and their definitions.

`-c date-time | -cdate-time`
 Retrieve the latest version checked in prior to the date and time indicated by the *date-time* argument. *date-time* takes the form: *yy[mm[dd[hh[mm[ss]]]]]*.

`-e`
 Retrieve a version for editing. Same as `sccs edit`.

`-G newname | -Gnewname`
 Use *newname* as the name of the retrieved version.

`-k`
 Retrieve a writable copy but do not check out the file. ID keywords are unexpanded.

`-m`
 Precede each line with the SID of the delta in which it was added.

`-p`
 Produce the retrieved version on the standard output. Reports that would normally go to the standard output (delta IDs and statistics) are directed to the standard error.

`-r sid | -rsid`
 Retrieve the version corresponding to the indicated SID. For `/usr/bin/sccs`, if no *sid* is specified, the latest *sid* for the specified file is retrieved.

`-s`
 Silent. Do not report version numbers or statistics.

`sccs -help message-code|sccs-command`
`sccs -help stuck`
 Supply more information about SCCS diagnostics. `sccs -help` displays a brief explanation of the error when you supply the code displayed by an SCCS diagnostic message. If you supply the name of an SCCS command, it prints a usage line. `sccs -help` also recognizes the keyword `stuck`. Refer to [sccs-help\(1\)](#).

`/usr/bin/sccs` `info [-b] [-u[username]]`
`/usr/xpg4/bin/sccs` `info [-b] [-u [username]] -U]`
 Display a list of files being edited, including the version number checked out, the version to be checked in, the name of the user who holds the lock, and the date and time the file was checked out.

`-b`
 Ignore branches.

`-u[username] | -u [username] | -U`

List only files checked out by you. When *username* is specified, list only files checked out by that user. For `/usr/xpg4/bin/sccs`, the `-U` option is equivalent to `-u <current_user>`.

`print`

Print the entire history of each named file. Equivalent to an `'sccs prs -e'` followed by an `'sccs get -p -m'`.

`/usr/bin/sccs`

`prs [-el] [-cdate-time] [-rsid]`

`/usr/xpg4/bin/sccs`

`prs [-el] [-c date-time | -cdate-time] [-r sid | -rsid]`

Peruse (display) the delta table, or other portion of an `s.` file. Refer to [sccs-prs\(1\)](#).

`-c date-time | -cdate-time`

Specify the latest delta checked in before the indicated date and time. The *date-time* argument takes the form: `yy[mm[dd[hh[mm[ss]]]]]`.

`-e`

Display delta table information for all deltas earlier than the one specified with `-r` (or all deltas if none is specified).

`-l`

Display information for all deltas later than, and including, that specified by `-c` or `-r`.

`-r sid | -rsid`

Specify a given delta by SID.

`prt [-y]` Display the delta table, but omit the MR field (see [sccsfile\(4\)](#) for more information on this field). Refer to [sccs-prt\(1\)](#).

`-y` Display the most recent delta table entry. The format is a single output line for each file argument, which is convenient for use in a pipeline with [awk\(1\)](#) or [sed\(1\)](#).

`/usr/bin/sccs`

`rmDEL -rsid`

`/usr/xpg4/bin/sccs`

`rmDEL -r sid`

Remove the indicated delta from the history file. That delta must be the most recent (leaf) delta in its branch. Refer to [sccs-rmDEL\(1\)](#).

`sact`

Show editing activity status of an SCCS file. Refer to [sccs-sact\(1\)](#).

`sccsdiff -r old-sid -r new-sid diff-options`

Compare two versions corresponding to the indicated SIDs (deltas) using `diff`. Refer to [sccs-sccsdiff\(1\)](#).

`/usr/bin/sccs`

`tell [-b] [-u[username]]`

`/usr/xpg4/bin/sccs` `tell [-b] [-u username] [-U]`
 Display the list of files that are currently checked out, one file per line.

-b
 Ignore branches.

-u*username* | **-u** *username* | **-U**
 List only files checked out to you. When *username* is specified, list only files checked out to that user. For `/usr/xpg4/bin/sccs`, the `-U` option is equivalent to `-u <current_user>`.

`unedit`
 “Undo” the last edit or ‘get -e’, and return the working copy to its previous condition. `unedit` backs out all pending changes made since the file was checked out.

`unget`
 Same as `unedit`. Refer to [sccs-unget\(1\)](#).

`val`
 Validate the history file. Refer to [sccs-val\(1\)](#).

`what`
 Display any expanded ID keyword strings contained in a binary (object) or text file. Refer to [what\(1\)](#) for more information.

示例

示例 1 Checking out, editing, and checking in a file

To check out a copy of `program.c` for editing, edit it, and then check it back in:

```
example% sccs edit program.c
```

```
1.1
new delta 1.2
14 lines
```

```
example% vi program.c
```

your editing session

```
example% sccs delget program.c
```

```
comments? clarified cryptic diagnostic
1.2
3 inserted
2 deleted
12 unchanged
1.2
15 lines
```

示例 2 Defining the root portion of the command pathname

`sccs` converts the command:

```
example% sccs -d/usr/src/include get stdio.h
```

to:

示例 2 Defining the root portion of the command pathname (续)

```
/usr/bin/get /usr/src/include/SCCS/s.stdio.h
```

示例 3 Defining the resident subdirectory

The command:

```
example% sccs -pprivate get include/stdio.h
```

becomes:

```
/usr/bin/get include/private/s.stdio.h
```

示例 4 Initializing a history file

To initialize the history file for a source file named `program.c`, make the SCCS subdirectory, and then use 'sccs create':

```
example% mkdir SCCS  
example% sccs create program.c  
program.c:  
1.1  
14 lines
```

After verifying the working copy, you can remove the backup file that starts with a comma:

```
example% diff program.c ,program.c  
example% rm ,program.c
```

示例 5 Retrieving a file from another directory

To retrieve a file from another directory into the current directory:

```
example% sccs get /usr/src/sccs/cc.c
```

or:

```
example% sccs -p/usr/src/sccs/ get cc.c
```

示例 6 Checking out all files

To check out all files under SCCS in the current directory:

```
example% sccs edit SCCS
```

示例 7 Checking in all files

To check in all files currently checked out to you:

```
example% sccs delta 'sccs tell -u'
```


示例 8 Entering multiple lines of comments

If using `-y` to enter a comment, for most shells, enclose the comment in single or double quotes. In the following example, `Myfile` is checked in with a two-line comment:

```
example% sccs deledit Myfile -y"Entering a
multi-line comment"
No id keywords (cm7)
1.2
2 inserted
0 deleted
14 unchanged
1.2
new delta 1.3
```

Displaying the SCCS history of `Myfile`:

```
example% sccs prt Myfile

SCCS/s.Myfile:

D 1.2  01/04/20  16:37:07  me 2 1   00002/00000/00014
Entering a
multi-line comment

D 1.1  01/04/15  13:23:32  me 1 0   00014/00000/00000
date and time created 01/04/15 13:23:32 by me
```

If `-y` is not used and `sccs` prompts for a comment, the newlines must be escaped using the backslash character (`\`):

```
example% sccs deledit Myfile
comments? Entering a \
multi-line comment
No id keywords (cm7)
1.2
0 inserted
0 deleted
14 unchanged
1.2
new delta 1.3
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `sccs`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

PROJECTDIR If contains an absolute path name (beginning with a slash), `sccs` searches for SCCS history files in the directory given by that variable.

If `PROJECTDIR` does not begin with a slash, it is taken as the name of a user, and `sccs` searches the `src` or source subdirectory of that user's home directory for history files. If such a directory is found, it is used. Otherwise,

the value is used as a relative path name.

退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

文件

- SCCS SCCS subdirectory
- SCCS/d.*file* temporary file of differences
- SCCS/p.*file* lock (permissions) file for checked-out versions
- SCCS/q.*file* temporary file
- SCCS/s.*file* SCCS history file
- SCCS/x.*file* temporary copy of the s.*file*
- SCCS/z.*file* temporary lock file
- /usr/bin/* SCCS utility programs

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/sccs

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

/usr/xpg4/bin/sccs

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[awk\(1\)](#), [diff\(1\)](#), [sccs-admin\(1\)](#), [sccs-cdc\(1\)](#), [sccs-comb\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-rmdel\(1\)](#), [sccs-sact\(1\)](#), [sccs-sccsdiff\(1\)](#), [sccs-unget\(1\)](#), [sccs-val\(1\)](#), [sed\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名	sccs-admin, admin – create and administer SCCS history files
用法概要	<pre> /usr/bin/admin [-bhnz] [-a <i>username</i> <i>groupid</i>]... [-d <i>flag</i>] ... [-e <i>username</i> <i>groupid</i>]... [-f <i>flag</i> [<i>value</i>]] ... [-i [<i>filename</i>]] [-m <i>mr-list</i>] [-rrelease] [-t [<i>description-file</i>]] [-y [<i>comment</i>]] <i>s.filename</i>...</pre>
描述	<p>The <code>admin</code> command creates or modifies the flags and other parameters of SCCS history files. Filenames of SCCS history files begin with the <code>s.</code> prefix, and are referred to as <code>s.</code> files, or history files.</p> <p>The named <code>s.</code> file is created if it does not exist already. Its parameters are initialized or modified according to the options you specify. Parameters not specified are given default values when the file is initialized, otherwise they remain unchanged.</p> <p>If a directory name is used in place of the <code>s.filename</code> argument, the <code>admin</code> command applies to all <code>s.</code> files in that directory. Unreadable <code>s.</code> files produce an error. The use of <code>'-'</code> as the <code>s.filename</code> argument indicates that the names of files are to be read from the standard input, one <code>s.</code> file per line.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> <code>-a <i>username</i> <i>groupid</i></code> Adds a user name, or a numerical group ID, to the list of users who may check deltas in or out. If the list is empty, any user is allowed to do so. <code>-b</code> Forces encoding of binary data. Files that contain ASCII NUL or other control characters, or that do not end with a NEWLINE, are recognized as binary data files. The contents of such files are stored in the history file in encoded form. See uuencode(1C) for details about the encoding. This option is normally used in conjunction with <code>-i</code> to force <code>admin</code> to encode initial versions not recognized as containing binary data. <code>-d <i>flag</i></code> Deletes the indicated <i>flag</i> from the SCCS file. The <code>-d</code> option may be specified only for existing <code>s.</code> files. See <code>-f</code> for the list of recognized flags. <code>-e <i>username</i> <i>groupid</i></code> Erases a user name or group ID from the list of users allowed to make deltas. <code>-f <i>flag</i> [<i>value</i>]</code> Sets the indicated <i>flag</i> to the (optional) <i>value</i> specified. The following flags are recognized: <ul style="list-style-type: none"> <code>b</code> Enables branch deltas. When <code>b</code> is set, branches can be created using the <code>-b</code> option of the SCCS <code>get</code> command (see sccs-get(1)).

cceil

Sets a ceiling on the releases that can be checked out. *ceil* is a number less than or equal to 9999. If *c* is not set, the ceiling is 9999.

dsid

Specifies the default delta number, or SID, to be used by an SCCS get command.

ffloor

Sets a floor on the releases that can be checked out. The floor is a number greater than 0 but less than 9999. If *f* is not set, the floor is 1.

i

Treats the 'No id keywords (ge6)' message issued by an SCCS get or delta command as an error rather than a warning.

j

Allows concurrent updates.

*la**l release[, release...]*

Locks the indicated list of releases against deltas. If *a* is used, this flag locks out deltas to all releases. An SCCS 'get -e' command fails when applied against a locked release.

mmodule

Supplies a value for the module name to which the `sccs-admin.1` keyword is to expand. If the *m* flag is not specified, the value assigned is the name of the SCCS file with the leading `s.` removed.

n

Creates empty releases when releases are skipped. These null (empty) deltas serve as anchor points for branch deltas.

qvalue

Supplies a *value* to which the keyword is to expand when a read-only version is retrieved with the SCCS get command.

snumber

Specifies how many lines of code are scanned for the SCCS keyword.

ttype

Supplies a value for the module type to which the keyword is to expand.

- v*[*program*]
 Specifies a validation *program* for the MR numbers associated with a new delta. The optional *program* specifies the name of an MR number validity checking *program*. If this flag is set when creating an SCCS file, the *-m* option must also be used, in which case the list of MRs may be empty.
- y*[*value*, [*value*]]
 Specifies the SCCS keywords to be expanded. If no *value* is specified, no keywords will be expanded.
- h* Checks the structure of an existing *s*. file (see [sccsfile\(4\)](#)), and compares a newly computed check-sum with one stored in the first line of that file. *-h* inhibits writing on the file and so nullifies the effect of any other options.
- i* [*filename*]
 Initializes the history file with text from the indicated file. This text constitutes the initial delta, or set of checked-in changes. If *filename* is omitted, the initial text is obtained from the standard input. Omitting the *-i* option altogether creates an empty *s*. file. You can only initialize one *s*. file with text using *-i*. This option implies the *-n* option.
- m mr-list*
 Inserts the indicated Modification Request (MR) numbers into the commentary for the initial version. When specifying more than one MR number on the command line, *mr-list* takes the form of a quoted, space-separated list. A warning results if the *v* flag is not set or the MR validation fails.
- n* Creates a new SCCS history file.
- r release*
 Specifies the release for the initial delta. *-r* may be used only in conjunction with *-i*. The initial delta is inserted into release 1 if this option is omitted. The level of the initial delta is always 1. Initial deltas are named 1.1 by default.
- t* [*description-file*]
 Inserts descriptive text from the file *description-file*. When *-t* is used in conjunction with *-n*, or *-i* to initialize a new *s*.file, the *description-file* must be supplied. When modifying the description for an existing file: a *-t* option without a *description-file* removes the descriptive text, if any; a *-t* option with a *description-file* replaces the existing text.
- y* [*comment*]
 Inserts the indicated *comment* in the "Comments:" field for the initial delta. Valid only in conjunction with *-i* or *-n*. If *-y* option is omitted, a default comment line is inserted that notes the date and time the history file was created.

- z Recomputes the file check-sum and stores it in the first line of the `s` file. *Caution:* It is important to verify the contents of the history file (see [sccs-val\(1\)](#), and the `print` subcommand in [sccs\(1\)](#)), since using `-z` on a truly corrupted file may prevent detection of the error.

示例

示例 1 Preventing SCCS keyword expansion

In the following example, 10 lines of `file` will be scanned and only the `W`, `Y`, `X` keywords will be interpreted:

```
example% sccs admin -fs10 file
example% sccs admin -fyW,Y,X file
example% get file
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `alias` and `unalias`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLS_PATH`.

退出状态

The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred.

文件

`s.*` history file
 SCCS/`s.*` history file in SCCS subdirectory
`z.*` temporary lock file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[sccs\(1\)](#), [sccs-cdc\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-rmdel\(1\)](#), [sccs-val\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断

Use the `sccs-help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

警告

The last component of all SCCS filenames must have the `'s.'` prefix. New SCCS files are given mode 444 (see [chmod\(1\)](#)). All writing done by `admin` is to a temporary file with an `x.` prefix, created with mode 444 for a new SCCS file, or with the same mode as an existing SCCS file. After successful execution of `admin`, the existing `s.` file is removed and replaced with the `x.` file. This ensures that changes are made to the SCCS file only when no errors have occurred.

It is recommended that directories containing SCCS files have permission mode 755, and that the `s` files themselves have mode 444. The mode for directories allows only the owner to modify the SCCS files contained in the directories, while the mode of the `s` files prevents all modifications except those performed using SCCS commands.

If it should be necessary to patch an SCCS file for any reason, the mode may be changed to 644 by the owner to allow use of a text editor. However, extreme care must be taken when doing this. The edited file should *always* be processed by an `'admin -h'` command to check for corruption, followed by an `'admin -z'` command to generate a proper check-sum. Another `'admin -h'` command is recommended to ensure that the resulting `s` file is valid.

`admin` also uses a temporary lock `s` file, starting with the `'z.'` prefix, to prevent simultaneous updates to the `s` file. See [sccs-get\(1\)](#) for further information about the `'z.'` file'.

引用名 `scs-cdc, cdc` – change the delta commentary of an SCCS delta

用法概要 `cdc -rsid [-mmr-list] [-y [comment]] s.filename...`

描述 `cdc` annotates the delta commentary for the SCCS delta ID (SID) specified by the `-r` option in each named `s`. file.

If the `v` flag is set in the `s`. file, you can also use `cdc` to update the Modification Request (MR) list.

If you checked in the delta, or, if you own the file and directory and have write permission, you can use `cdc` to annotate the commentary.

Rather than replacing the existing commentary, `cdc` inserts the new comment you supply, followed by a line of the form:

```
*** CHANGED *** yy/mm/dd hh/mm/ss username
```

above the existing commentary.

If a directory is named as the `s.filename` argument, the `cdc` command applies to all `s`. files in that directory. Unreadable `s`. files produce an error; processing continues with the next file (if any). If `'-'` is given as the `s.filename` argument, each line of the standard input is taken as the name of an SCCS history file to be processed, and the `-m` and `-y` options must be used.

选项

<code>-rsid</code>	Specify the SID of the delta to change.
<code>-mmr-list</code>	Specify one or more MR numbers to add or delete. When specifying more than one MR on the command line, <code>mmr-list</code> takes the form of a quoted, space-separated list. To delete an MR number, precede it with a <code>!</code> character (an empty MR list has no effect). A list of deleted MRs is placed in the comment section of the delta commentary. If <code>-m</code> is not used and the standard input is a terminal, <code>cdc</code> prompts with <code>MRs?</code> for the list (before issuing the <code>comment s?</code> prompt). <code>-m</code> is only useful when the <code>v</code> flag is set in the <code>s</code> . file. If that flag has a value, it is taken to be the name of a program to validate the MR numbers. If that validation program returns a non-zero exit status, <code>cdc</code> terminates and the delta commentary remains unchanged.
<code>-y[comment]</code>	Use <code>comment</code> as the annotation in the delta commentary. The previous comments are retained; the <code>comment</code> is added along with a notation that the commentary was changed. A null <code>comment</code> leaves the commentary unaffected. If <code>-y</code> is not specified and the standard input is a terminal, <code>cdc</code> prompts with <code>comment s?</code> for the text of the notation to be added. An unescaped NEWLINE character terminates the annotation text.

示例

示例 1 Changing the annotated commentary

The following command:

```
example% cdc -r1.6 -y"corrected commentary" s.program.c
```

produces the following annotated commentary for delta 1.6 in s.program.c:

```
D 1.6 88/07/05 23:21:07 username 9 0 00001/00000/00000
```

```
MRs:
```

```
COMMENTS:
```

```
corrected commentary
```

```
*** CHANGED *** 88/07/07 14:09:41 username
```

```
performance enhancements in main()
```

文件

z.file temporary lock file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	developer/build/make

另请参见

[sccs\(1\)](#), [sccs-admin\(1\)](#), [sccs-comb\(1\)](#), [sccs-delta\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [sccs-rmdel\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#)

诊断

Use the `sccs -help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

引用名 sccs-comb, comb – combine SCCS deltas

用法概要 comb [-os] [-csid-list] [-psid] *s.filename*...

描述 comb generates a shell script (see [sh\(1\)](#)) that you can use to reconstruct the indicated *s*. files. This script is written to the standard output.

If a directory name is used in place of the *s.filename* argument, the comb command applies to all *s*. files in that directory. Unreadable *s*. files produce an error; processing continues with the next file (if any). The use of – as the *s.filename* argument indicates that the names of files are to be read from the standard input, one *s*. file per line.

If no options are specified, comb preserves only the most recent (leaf) delta in a branch, and the minimal number of ancestors needed to preserve the history.

选项 The following options are supported:

-o For each get -e generated, access the reconstructed file at the release of the delta to be created. Otherwise, the reconstructed file is accessed at the most recent ancestor. The use of -o can decrease the size of the reconstructed *s*. file. It can also alter the shape of the delta tree of the original file.

-s Generate scripts to gather statistics, rather than combining deltas. When run, the shell scripts report: the file name, size (in blocks) after combining, original size (also in blocks), and the percentage size change, computed by the formula:

$$100 * (original - combined) / original$$

This option can be used to calculate the space that is saved, before actually doing the combining.

-csid-list Include the indicated list of deltas. All other deltas are omitted. *sid-list* is a comma-separated list of SCCS delta IDs (SIDs). To specify a range of deltas, use a – separator instead of a comma, between two SIDs in the list.

-pSID The SID of the oldest delta to be preserved.

文件 *s*. COMB reconstructed SCCS file

comb????? temporary file

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

另请参见 [sccs\(1\)](#), [sccs-admin\(1\)](#), [sccs-cdc\(1\)](#), [sccs-delta\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [sccs-rmdel\(1\)](#), [sccs-sccsdiff\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#)

诊断

Use the `sccs -help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

已知问题

`comb` might rearrange the shape of the tree of deltas. It might not save any space; in fact, it is possible for the reconstructed file to actually be larger than the original.

引用名 sccs-delta, delta – make a delta to an SCCS file

用法概要

```
/usr/bin/delta [-dnps] [-g sid-list | -gsid-list]
               [-m mr-list | -mmr-list] [-r sid | -rsid]
               [-y [comment]] s.filename...
```

```
/usr/xpg4/bin/delta [-dnps] [-g sid-list | -gsid-list]
                    [-m mr-list | -mmr-list] [-r sid | -rsid]
                    [-y [comment]] s.filename...
```

描述

The `delta` utility checks in a record of the line-by-line differences made to a checked-out version of a file under SCCS control. These changes are taken from the writable working copy that was retrieved using the SCCS `get` command (see [sccs-get\(1\)](#)). This working copy does not have the 's.' prefix, and is also referred to as a g-file.

If a directory name is used in place of the *s.filename* argument, the `delta` command applies to all s. files in that directory. Unreadable s. files produce an error; processing continues with the next file (if any). The use of '-' as the *s.filename* argument indicates that the names of files are to be read from the standard input, one s. file per line (requires -y, and in some cases, -m).

`delta` can issue prompts on the standard output depending upon the options specified and the flags that are set in the s. file (see [sccs-admin\(1\)](#), and the -m and -y options below, for details).

`/usr/xpg4/bin/delta` The SID of the delta is not echoed to stdout.

选项

The following options are supported:

- d Use command [diff\(1\)](#) instead of [bdiff\(1\)](#). Returns exit status 2 if *s.filename* argument is not specified.
- n Retain the edited g-file, which is normally removed at the completion of processing.
- p Display line-by-line differences (in [diff\(1\)](#) format) on the standard output.
- s Silent. Do not display warning or confirmation messages. Do not suppress error messages (which are written to standard error).
- g *sid-list* | -gsid-list Specify a list of deltas to omit when the file is accessed at the SCCS version ID (SID) created by this delta. *sid-list* is a comma-separated list of SIDs. To specify a range of deltas, use a '-' separator instead of a comma, between two SIDs in the list.
- m *mr-list* | -mmr-list If the SCCS file has the v flag set (see [sccs-admin\(1\)](#)), you must supply one or more Modification Request (MR) numbers for the new delta. When specifying more than one MR number on the command line, *mr-list* takes the form of a quoted, space-separated list. If -m is not used and the standard input is a terminal, `delta` prompts with

MRs? for the list (before issuing the comments? prompt). If the v flag in the s . file has a value, it is taken to be the name of a program to validate the MR numbers. If that validation program returns a non-zero exit status, de l t a terminates without checking in the changes.

- r *sid* | -r*sid* When two or more versions are checked out, specify the version to check in. This SID value can be either the SID specified on the get command line, or the SID of the new version to be checked in as reported by get. A diagnostic results if the specified SID is ambiguous, or if one is required but not supplied.
- y[*comment*] Supply a comment for the delta table (version log). A null comment is accepted, and produces an empty commentary in the log. If -y is not specified and the standard input is a terminal, de l t a prompts with 'comments?'. An unescaped NEWLINE terminates the comment.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of de l t a: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 Successful completion.
- 1 An error occurred and the -d option had not been specified.
- 2 An error occurred, the -d option had been specified, and the *s.filename* argument was not specified.

文件

- d . file temporary file of differences
- p . file lock file for a checked-out version
- q . file temporary file
- s . file SCCS history file
- x . file temporary copy of the s . file
- z . file temporary file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/delta

ATTRIBUTE	ATTRIBUTEVALUE
Availability	developer/build/make

/usr/xpg4/bin/delta

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[bdiff\(1\)](#), [diff\(1\)](#), [sccs-admin\(1\)](#), [sccs-cdc\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [sccs-rmdel\(1\)](#), [sccs-sccsdiff\(1\)](#), [sccs-unget\(1\)](#), [sccs\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断

Use the `SCCS help` command for explanations (see [sccs-help\(1\)](#)).

警告

Lines beginning with an ASCII SOH character (binary 001) cannot be placed in the SCCS file unless the SOH is escaped. This character has special meaning to SCCS (see [sccsfile\(4\)](#)) and produces an error.

引用名	sccs-get, get – retrieve a version of an SCCS file
用法概要	<pre> /usr/bin/get [-begkmpst] [-l [p]] [-asequence] [-c <i>date-time</i> -c<i>date-time</i>] [-Gg-file] [-i <i>sid-list</i> -i<i>sid-list</i>] [-r [<i>sid</i>]] [-x <i>sid-list</i> -x<i>sid-list</i>] <i>s.filename...</i> /usr/xpg4/bin/get [-begkmpst] [-l [p]] [-asequence] [-c <i>date-time</i> -c<i>date-time</i>] [-Gg-file] [-i <i>sid-list</i> -i<i>sid-list</i>] [-r <i>sid</i> -r<i>sid</i>] [-x <i>sid-list</i> -x<i>sid-list</i>] <i>s.filename...</i> </pre>
描述	<p>The get utility retrieves a working copy from the SCCS history file, according to the specified options.</p> <p>For each <i>s.filename</i> argument, get displays the SCCS delta ID (SID) and number of lines retrieved.</p> <p>If a directory name is used in place of the <i>s.filename</i> argument, the get command applies to all <i>s.</i> files in that directory. Unreadable <i>s.</i> files produce an error; processing continues with the next file (if any). The use of '-' as the <i>s.filename</i> argument indicates that the names of files are to be read from the standard input, one <i>s.</i> file per line.</p> <p>The retrieved file normally has the same filename base as the <i>s.</i> file, less the prefix, and is referred to as the <i>g</i>-file.</p> <p>For each file processed, get responds (on the standard output) with the SID being accessed, and with the number of lines retrieved from the <i>s.</i> file.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -<i>asequence</i> Retrieves the version corresponding to the indicated delta sequence number. This option is used primarily by the SCCS comb command (see sccs-comb(1)). For users, -r is an easier way to specify a version. The -a option supersedes the -r option when both are used. -b Creates a new branch. Used with the -e option to indicate that the new delta should have a SID in a new branch. Instead of incrementing the level for version to be checked in, get indicates in the <i>p.</i> file that the delta to be checked in should either initialize a new branch and sequence (if there is no existing branch at the current level), or increment the branch component of the SID. If the b flag is not set in the <i>s.</i> file, this option is ignored. -c <i>date-time</i> -c<i>date-time</i> Retrieves the latest version checked in prior to the date and time indicated by the <i>date-time</i> argument. <i>date-time</i> takes the form:

yy[mm[dd[hh[mm[ss]]]]]

Units omitted from the indicated date and time default to their maximum possible values; that is -c7502 is equivalent to -c750228235959. Values of *yy* in the range 69–99 refer to the twentieth century. Values in the range 00–68 refer to the twenty-first century. Any number of non-numeric characters can separate the various 2 digit components. If white-space characters occur, the *date-time* specification must be quoted.

- e Retrieves a version for editing. With this option, *get* places a lock on the *s*.file, so that no one else can check in changes to the version you have checked out. If the *j* flag is set in the *s*.file, the lock is advisory: *get* issues a warning message. Concurrent use of '*get -e*' for different SIDs is allowed. However, *get* does not check out a version of the file if a writable version is present in the directory. All SCCS file protections stored in the *s*.file, including the release ceiling, floor, and authorized user list, are honored by '*get -e*'.
- g Gets the SCCS version ID, without retrieving the version itself. Used to verify the existence of a particular SID.
- Gnewname* Uses *newname* as the name of the retrieved version.
- i *sid-list* | -i*sid-list* Specifies a list of deltas to include in the retrieved version. The included deltas are noted in the standard output message. *sid-list* is a comma-separated list of SIDs. To specify a range of deltas, use a '-' separator instead of a comma, between two SIDs in the list.
- k Suppresses expansion of ID keywords. -k is implied by the -e.
- l [*p*] Retrieves a summary of the delta table (version log) and write it to a listing file, with the 'l.' prefix (called 'l.file'). When -lp is used, write the summary onto the standard output.
- m Precedes each retrieved line with the SID of the delta in which it was added to the file. The SID is separated from the line with a TAB.
- n Precedes each line with the %M% ID keyword and a TAB. When both the -m and -n options are used, the ID keyword precedes the SID, and the line of text.
- p Writes the text of the retrieved version to the standard output. All messages that normally go to the standard output are written to the standard error instead.

- s Suppresses all output normally written on the standard output. However, fatal error messages (which always go to the standard error) remain unaffected.
- t Retrieves the most recently created (top) delta in a given release (for example: -r1).

/usr/bin/get

- r[*sid*] Retrieves the version corresponding to the indicated SID (delta).

The SID for a given delta is a number, in Dewey decimal format, composed of two or four fields: the *release* and *level* fields, and for branch deltas, the *branch* and *sequence* fields. For instance, if 1.2 is the SID, 1 is the release, and 2 is the level number. If 1.2.3.4 is the SID, 3 is the branch and 4 is the sequence number.

You need not specify the entire SID to retrieve a version with get. When you omit -r altogether, or when you omit both release and level, get normally retrieves the highest release and level. If the d flag is set to an SID in the s . file and you omit the SID, get retrieves the default version indicated by that flag.

When you specify a release but omit the level, get retrieves the highest level in that release. If that release does not exist, get retrieves highest level from the next-highest existing release.

Similarly with branches, if you specify a release, level and branch, get retrieves the highest sequence in that branch.

/usr/xpg4/bin/get

- r *sid* | -r*sid* Same as for /usr/bin/get except that SID is mandatory.
- x *sid-list* | -x*sid-list* Excludes the indicated deltas from the retrieved version. The excluded deltas are noted in the standard output message. *sid-list* is a comma-separated list of SIDs. To specify a range of deltas, use a '-' separator instead of a comma, between two SIDs in the list.

Output

/usr/bin/get

The output format for /usr/bin/get is as follows:

```
"%s\n%d lines\n", <SID>, <number of lines>
```

/usr/xpg4/bin/get

The output format for /usr/xpg4/bin/get is as follows:

```
"%s\n%d\n", <SID>, <number of lines>
```

用法

Usage guidelines are as follows:

ID Keywords

In the absence of -e or -k, get expands the following ID keywords by replacing them with the indicated values in the text of the retrieved source.

<i>Keyword</i>	<i>Value</i>
%A%	Shorthand notation for an ID line with data for <code>what(1)</code> : %Z%Y% %M% %I%Z%
%B%	SID branch component
%C%	Current line number. Intended for identifying messages output by the program such as ‘ <i>this shouldn't have happened</i> ’ type errors. It is <i>not</i> intended to be used on every line to provide sequence numbers.
%D%	Current date: <i>yy/mm/dd</i>
%E%	Date newest applied delta was created: <i>yy/mm/dd</i>
%F%	SCCS s . file name
%G%	Date newest applied delta was created: <i>mm/dd/yy</i>
%H%	Current date: <i>mm/dd/yy</i>
%I%	SID of the retrieved version: %R%. %L%. %B%. %S%
%%	SID level component
%M%	Module name: either the value of the <code>m</code> flag in the <code>s . file</code> (see <code>scs-admin(1)</code>), or the name of the <code>s . file</code> less the prefix
%P%	Fully qualified <code>s . file</code> name
%Q%	Value of the <code>q</code> flag in the <code>s . file</code>
%R%	SID Release component
%S%	SID Sequence component
%T%	Current time: <i>hh:mm:ss</i>
%U%	Time the newest applied delta was created: <i>hh:mm:ss</i>
%W%	Shorthand notation for an ID line with data for what: %Z%&;%I%
%Y%	Module type: value of the <code>t</code> flag in the <code>s . file</code>
%Z%	4-character string: ‘@(#)’, recognized by <code>what</code>

ID String

The table below explains how the SCCS identification string is determined for retrieving and creating deltas.

Determination of SCCS Identification String				
SID (1) Specified	-b Option Used (2)	Other Conditions	SID Retrieved	SID of Delta to be Created
none (3)	no	R defaults to mR	mR.mL	mR.(mL+1)
none (3)	yes	R defaults to mR	mR.mL	mR.mL.(mB+1).1

Determination of SCCS Identification String				
SID (1) Specified	-b Option Used (2)	Other Conditions	SID Retrieved	SID of Delta to be Created
R	no	R > mR	mR.mL	R.1 (4)
R	no	R = mR	mR.mL	mR.(mL+1)
R	yes	R > mR	mR.mL	mR.mL.(mB+1).1
R	yes	R = mR	mR.mL	mR.mL.(mB+1).1
R	–	R < mR and R does <i>not</i> exist	hR.mL (5)	hR.mL.(mB+1).1
R	–	Trunk succ. (6) in release > R and R exists	R.mL	R.mL.(mB+1).1
R.L	no	No trunk succ.	R.L	R.(L+1)
R.L	yes	No trunk succ.	R.L	R.L.(mB+1).1
R.L	–	Trunk succ. in release ≥ R	R.L	R.L.(mB+1).1
R.L.B	no	No branch succ.	R.L.B.mS	R.L.B.(mS+1)
R.L.B	yes	No branch succ.	R.L.B.mS	R.L.(mB+1).1
R.L.B.S	no	No branch succ.	R.L.B.S	R.L.B.(S+1)
R.L.B.S	yes	No branch succ.	R.L.B.S	R.L.(mB+1).1
R.L.B.S	–	Branch succ.	R.L.B.S	R.L.(mB+1).1

- (1) 'R', 'L', 'B', and 'S' are the 'release', 'level', 'branch', and 'sequence' components of the SID, respectively; 'm' means 'maximum'. Thus, for example, 'R.mL' means 'the maximum level number within release R'; 'R.L.(mB+1).1' means 'the first sequence number on the *new* branch (that is, maximum branch number plus one) of level L within release R'. *Note:* If the SID specified is of the form 'R.L', 'R.L.B', or 'R.L.B.S', each of the specified components *must* exist.
- (2) The -b option is effective only if the b flag is present in the file. An entry of '–' means 'irrelevant'.
- (3) This case applies if the d (default SID) flag is *not* present in the file. If the d flag is present in the file, the SID obtained from the d flag is interpreted as if it had been specified on the command line. Thus, one of the other cases in this table applies.
- (4) Forces creation of the *first* delta in a *new* release.
- (5) 'hR' is the highest *existing* release that is lower than the specified, *nonexistent*, release R.
- (6) Successor.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of get: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

文件

- “g-file” version retrieved by get
- l.*file* file containing extracted delta table info
- p.*file* permissions (lock) file
- z.*file* temporary copy of s.*file*

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/get

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

/usr/xpg4/bin/get

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [sccs\(1\)](#), [sccs-admin\(1\)](#), [sccs-delta\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [sccs-sact\(1\)](#), [sccs-unget\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断 Use the `sccs-help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

已知问题 If the effective user has write permission (either explicitly or implicitly) in the directory containing the SCCS files, but the real user does not, only one file can be named when using `-e`.

引用名 sccs-help, sccshelp – ask for help regarding SCCS error or warning messages

用法概要 /usr/bin/sccshelp [*argument*]...

描述 The sccs-help utility retrieves information to further explain errors messages and warnings from SCCS commands. It also provides some information about SCCS command usage. If no arguments are given, sccs-help prompts for one.

An *argument* may be a message number (which normally appears in parentheses following each SCCS error or warning message), or an SCCS command name. sccs-help responds with an explanation of the message or a usage line for the command.

When all else fails, try /usr/bin/sccshelp stuck.

文件 /usr/lib/help directory containing files of message text

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

另请参见 [sccs\(1\)](#), [sccs-admin\(1\)](#), [sccs-cdc\(1\)](#), [sccs-comb\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [sccs-rmdel\(1\)](#), [sccs-sact\(1\)](#), [sccs-sccsdiff\(1\)](#), [sccs-unget\(1\)](#), [sccs-val\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#)

引用名	sccs-prs, prs – display selected portions of an SCCS history
用法概要	prs [-ael] [-cdate-time] [-ddataspec] [-rsid] s.filename...
描述	<p>The prs utility displays part or all of the SCCS file (see sccsfile(4)) in a user supplied format.</p> <p>If a directory name is used in place of the <i>s.filename</i> argument, the prs command applies to all <i>s.</i> files in that directory. Unreadable <i>s.</i> files produce an error; processing continues with the next file (if any). The use of '-' as the <i>s.filename</i> argument indicates that the names of files are to be read from the standard input, one <i>s.</i> file per line.</p>
选项	<p>In the absence of options, prs displays the delta table (version log). In the absence of -d, or -l, prs displays the entry for each delta indicated by the other options.</p> <p>-a Includes all deltas, including those marked as removed (see sccs-rmdel(1)).</p> <p>-e Requests information for all deltas created <i>earlier</i> than, and including, the delta indicated with -r or -c.</p> <p>-l Requests information for all deltas created <i>later</i> than, and including, the delta indicated with -r or -c.</p> <p>-cdate-time Either options -e or -l must be used with this option. -cdate-time displays information on the deltas checked in either prior to and including the date and time indicated by the <i>date-time</i> argument (option -e); or later than and including the date and time indicated (option -l). <i>date-time</i> takes the form:</p> <pre>yy[mm[dd[hh[mm[ss]]]]]]]</pre> <p>Units omitted from the indicated date and time default to their maximum possible values; that is -c7502 is equivalent to -c750228235959. Any number of non-numeric characters may separate the various 2 digit components. If white-space characters occur, the <i>date-time</i> specification must be quoted. Values of <i>yy</i> in the range 69–99 refer to the twentieth century. Values in the range of 00–68 refer to the twenty-first century.</p> <p>-ddataspec Produce a report according to the indicated data specification. <i>dataspec</i> consists of a (quoted) text string that includes embedded data keywords of the form: ':key:' (see Data Keywords, below). prs expands these keywords in the output it produces. To specify a TAB character in the output, use \t; to specify a NEWLINE in the output, use \n.</p> <p>-rsid Specifies the SCCS delta ID (SID) of the delta for which information is desired. If no SID is specified, the most recently created delta is used.</p>
用法	Usage of prs is described below.

Data Keywords

Data keywords specify which parts of an SCCS file are to be retrieved. All parts of an SCCS file (see [sccsfile\(4\)](#)) have an associated data keyword. A data keyword may appear any number of times in a data specification argument to `-d`. These data keywords are listed in the table below:

<i>Keyword</i>	<i>Data Item</i>	<i>File Section*</i>	<i>Value</i>	<i>Format**</i>
:A:	a format for the what string:	N/A	:Z::Y: :M: :I::Z:	S
:B:	branch number	D	<i>nnnn</i>	S
:BD:	body	B	<i>text</i>	M
:BF:	branch flag	F	yes or no	S
:CB:	ceiling boundary	F	:R:	S
:C:	comments for delta	D	<i>text</i>	M
:D:	date delta created	D	:Dy:/:Dm:/:Dd:	S
:Dd:	day delta created	D	<i>nn</i>	S
:Dg:	deltas ignored (seq #)	D	:DS: :DS: . . .	S
:DI:	seq-no. of deltas included, excluded, ignored	D	:Dn:/:Dx:/:Dg:	S
:DL:	delta line statistics	D	:Li:/:Ld:/:Lu:	S
:Dm:	month delta created	D	<i>nn</i>	S
:Dn:	deltas included (seq #)	D	:DS: :DS: . . .	S
:DP:	predecessor delta seq-no.	D	<i>nnnn</i>	S
:Ds:	default SID	F	:I:	S
:DS:	delta sequence number	D	<i>nnnn</i>	S
:Dt:	delta information	D	:DT: :I: :D: :T: :P: :DS: :DP:	S
:DT:	delta type	D	D or R	S
:Dx:	deltas excluded (seq #)	D	:DS: . . .	S
:Dy:	year delta created	D	<i>nn</i>	S
:F:	s. file name	N/A	<i>text</i>	S
:FB:	floor boundary	F	:R:	S
:FD:	file descriptive text	C	<i>text</i>	M

<i>Keyword</i>	<i>Data Item</i>	<i>File Section*</i>	<i>Value</i>	<i>Format**</i>
:FL:	flag list	F	<i>text</i>	M
:GB:	gotten body	B	<i>text</i>	M
:I:	SCCS delta ID (SID)	D	:R: . : L: . : B: . : S:	S
:J:	joint edit flag	F	yes or no	S
:KF:	keyword error/warning flag	F	yes or no	S
:L:	level number	D	<i>nnnn</i>	S
:Ld:	lines deleted by delta	D	<i>nnnnn</i>	S
:Li:	lines inserted by delta	D	<i>nnnnn</i>	S
:LK:	locked releases	F	:R: . . .	S
:Lu:	lines unchanged by delta	D	<i>nnnnn</i>	S
:M:	module name	F	<i>text</i>	S
:MF:	MR validation flag	F	yes or no	S
:MP:	MR validation program	F	<i>text</i>	S
:MR:	MR numbers for delta	D	<i>text</i>	M
:ND:	null delta flag	F	yes or no	S
:Q:	user defined keyword	F	<i>text</i>	S
:P:	user who created delta	D	<i>username</i>	S
:PN:	s . file's pathname	N/A	<i>text</i>	S
:R:	release number	D	<i>nnnn</i>	S
:S:	sequence number	D	<i>nnnn</i>	S
:T:	time delta created	D	:Th: . : Tm: . : Ts:	S
:Th:	hour delta created	D	<i>nn</i>	S
:Tm:	minutes delta created	D	<i>nn</i>	S
:Ts:	seconds delta created	D	<i>nn</i>	S
:UN:	user names	U	<i>text</i>	M
:W:	a form of what string	N/A	:Z: . :M: \t: I:	S
:Y:	module type flag	F	<i>text</i>	S
:Z:	what string delimiter	N/A	@(#)	S

*B = body, D = delta table, F = flags, U = user names

**S = simple format, M = multi-line format

示例

示例 1 Displaying delta entries

The following command displays delta entries:

```
example% prs -e -d":I:\t:P:" program.c
```

produces:

```
1.6  username
1.5  username...
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of prs: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

文件

/tmp/pr????? temporary file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[sccs\(1\)](#), [sccs-cdc\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-prt\(1\)](#), [sccs-sact\(1\)](#), [sccs-sccsdiff\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断

Use the `sccs-help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

引用名 `scs-prt, prt` – display delta table information from an SCCS file

用法概要 `prt [-abdefistu] [-cdate-time] [-rdate-time]`
 `[-ysid] s.filename...`

描述 `prt` prints selected portions of an SCCS file. By default, it prints the delta table (version log).
 If a directory name is used in place of the *s.filename* argument, the `prt` command applies to all *s.* files in that directory. Unreadable *s.* files produce an error; processing continues with the next file (if any). The use of ‘-’ as the *s.filename* argument indicates that the names of files are to be read from the standard input, one *s.* file per line.

选项 If any option other than `-y`, `-c`, or `-r` is supplied, the name of each file being processed (preceded by one NEWLINE and followed by two NEWLINE characters) appears above its contents.

If none of the `-u`, `-f`, `-t`, or `-b` options are used, `-d` is assumed. `-s`, `-i` are mutually exclusive, as are `-c` and `-r`.

- `-a` Display log entries for all deltas, including those marked as removed.
- `-b` Print the body of the *s.* file.
- `-d` Print delta table entries. This is the default.
- `-e` Everything. This option implies `-d`, `-i`, `-u`, `-f`, and `-t`.
- `-f` Print the flags of each named *s.* file.
- `-i` Print the serial numbers of included, excluded, and ignored deltas.
- `-s` Print only the first line of the delta table entries; that is, only up to the statistics.
- `-t` Print the descriptive text contained in the *s.* file.
- `-u` Print the user-names and/or numerical group IDs of users allowed to make deltas.
- `-cdate-time` Exclude delta table entries that are specified cutoff date and time. Each entry is printed as a single line, preceded by the name of the SCCS file. This format (also produced by `-r`, and `-y`) makes it easy to sort multiple delta tables in chronological order. When both `-y` and `-c`, or `-y` and `-r` are supplied, `prt` stops printing when the first of the two conditions is met.
- `-rdate-time` Exclude delta table entries that are newer than the specified cutoff date and time.
- `-ysid` Exclude delta table entries made prior to the SID specified. If no delta in the table has the specified SID, the entire table is printed. If no SID is specified, the most recent delta is printed.

用法

Output Format

The following format is used to print those portions of the `s` file that are specified by the various options.

- NEWLINE
- Type of delta (D or R)
- SPACE
- SCCS delta ID (SID)
- TAB
- Date and time of creation in the form: *yy/mm/dd hh/mm/ss*
- SPACE
- Username the delta's creator
- TAB
- Serial number of the delta
- SPACE
- Predecessor delta's serial number
- TAB
- Line-by-line change statistics in the form: *inserted/deleted/unchanged*
- NEWLINE
- List of included deltas, followed by a NEWLINE (only if there were any such deltas and the `-i` options was used)
- List of excluded deltas, followed by a NEWLINE (only if there were any such deltas and the `-i` options was used)
- List of ignored deltas, followed by a NEWLINE (only if there were any such deltas and the `-i` options was used)
- List of modification requests (MRs), followed by a NEWLINE (only if any MR numbers were supplied).
- Lines of the delta commentary (if any), followed by a NEWLINE.

示例

示例 1 Producing a Display of the Delta Table

The following command produces a one-line display of the delta table entry for the most recent version:

```
example% prt -y program.c
s.program.c: D 1.6 88/07/06 21:39:39 username 5 4 00159/00080/00636
```

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

另请参见 [sccs\(1\)](#), [sccs-cdc\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-sact\(1\)](#), [sccs-sccsdiff\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#)

诊断 Use the `sccs-help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

- 引用名** sccs-rmdel, rmdel – remove a delta from an SCCS file
- 用法概要** rmdel -rsid *s.filename*...
- 描述** The rmdel utility removes the delta specified by the SCCS delta ID (SID) supplied with -r. The delta to be removed must be the most recent (leaf) delta in its branch. In addition, the SID must *not* be that of a version checked out for editing: it must not appear in any entry of the version lock file (p.file).
- If you created the delta, or, if you own the file and directory and have write permission, you can remove it with rmdel.
- If a directory name is used in place of the *s.filename* argument, the rmdel command applies to all s.files in that directory. Unreadable s.files produce an error; processing continues with the next file (if any). The use of '-' as the *s.filename* argument indicates that the names of files are to be read from the standard input, one s.file per line.
- 选项** The following option is supported:
- rsid Remove the version corresponding to the indicated SID (delta).
- 环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of rmdel: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.
- 文件**
- p.file permissions file
 - s.file history file
 - z.file temporary copy of the s.file
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:
- | ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|---------------------|------------------------------------|
| Availability | developer/build/make |
| Interface Stability | Committed |
| Standard | See standards(5) . |
- 另请参见** [sccs\(1\)](#), [sccs-admin\(1\)](#), [sccs-cdc\(1\)](#), [sccs-comb\(1\)](#), [sccs-delta\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [sccs-sccsdiff\(1\)](#), [sccs-unget\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)
- 诊断** Use the SCCS help command for explanations (see [sccs-help\(1\)](#)).

引用名 sccs-sact, sact – show editing activity status of an SCCS file

用法概要 sact *s.filename*...

描述 The sact utility informs the user of any SCCS files that are checked out for editing.

The output for each named file consists of five fields separated by SPACE characters.

- SID of a delta that currently exists in the SCCS file, to which changes are made to make the new delta
- SID for the new delta to be created
- Username of the person who has the file checked out for editing.
- Date that the version was checked out.
- Time that the version was checked out.

If a directory name is used in place of the *s.filename* argument, the sact command applies to all *s.* files in that directory. Unreadable *s.* files produce an error; processing continues with the next file (if any). The use of '-' as the *s.filename* argument indicates that the names of files are to be read from the standard input, one *s.* file per line.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of sact: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [sccs\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断 Use the `sccs-help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

- 引用名** sccs-sccsdiff, sccsdiff – compare two versions of an SCCS file
- 用法概要** sccsdiff [-p] -rsid -rsid [diff-options] s.filename
- 描述** sccsdiff compares two versions of an SCCS file and displays the differences between the two versions. Any number of SCCS files can be specified. The options specified apply to all named s . files.
- 选项** The following options are supported:
- p Pipe output for each file through [pr\(1\)](#).
 - rsid Specify a version corresponding to the indicated SCCS delta ID (SID) for comparison. Versions are passed to [diff\(1\)](#) in the order given.
 - diff-options Pass options to [diff\(1\)](#), including: -b, -c, -e, -f, -h, -u, -C number, -U number, and -D string.
- 文件** /tmp/get????? temporary files
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

- 另请参见** [diff\(1\)](#), [sccs\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#)
- 诊断** *filename*: No differences If the two versions are the same.
- Use the [sccs-help](#) command for explanations of SCCS commands. See [sccs-help\(1\)](#).

- 引用名** sccs-unget, unget – undo a previous get of an SCCS file
- 用法概要** unget [-ns] [-rsid] *s.filename*...
- 描述** The unget utility undoes the effect of a get -e command executed before the creation of the pending delta.
- If a directory name is used in place of the *s.filename* argument, the unget command applies to all *s.* files in that directory. Unreadable *s.* files produce an error; processing continues with the next file (if any). The use of '-' as the *s.filename* argument indicates that the names of files are to be read from the standard input, one *s.* file per line.
- 选项** The following options are supported:
- n Retains the retrieved version, which is otherwise removed.
 - s Suppress display of the SCCS delta ID (SID).
 - rsid When multiple versions are checked out, this option specifies which pending delta to abort. A diagnostic results if the specified SID is ambiguous, or if it is necessary but omitted from the command line.
- 环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of unget: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLS_PATH.
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:
- | ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|---------------------|------------------------------------|
| Availability | developer/build/make |
| Interface Stability | Committed |
| Standard | See standards(5) . |
- 另请参见** [sccs\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [sccs-rmdel\(1\)](#), [sccs-sact\(1\)](#), [sccs-sccsdiff\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)
- 诊断** Use the `sccs-help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

引用名	sccs-val, val – validate an SCCS file
用法概要	val [-s] [-m <i>name</i>] [-rsid] [-y <i>type</i>] <i>s.filename...</i>
描述	<p>The val utility determines if the specified <i>s</i>. files meet the characteristics specified by the indicated arguments. val can process up to 50 files on a single command line.</p> <p>val has a special argument, '-', which reads the standard input until the end-of-file condition is detected. Each line read is independently processed as if it were a command line argument list.</p> <p>val generates diagnostic messages on the standard output for each command line and file processed and also returns a single 8-bit code upon exit as described below.</p> <p>The 8-bit code returned by val is a disjunction of the possible errors, that is, it can be interpreted as a bit string where (moving from left to right) the bits set are interpreted as follows:</p> <ul style="list-style-type: none"> bit 0 = missing file argument bit 1 = unknown or duplicate option bit 2 = corrupted <i>s</i>.file bit 3 = can not open file or file not in <i>s</i>.file format bit 4 = the SCCS delta ID (SID) is invalid or ambiguous bit 5 = the SID does not exist bit 6 = mismatch between <i>Y%</i> and <i>-y</i> argument bit 7 = mismatch between sccs-val.1 and <i>-m</i> argument <p>val can process two or more files on a given command line, and in turn can process multiple command lines (when reading the standard input). In these cases, an aggregate code is returned which is the logical OR of the codes generated for each command line and file processed.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -s Silent. Suppresses the normal error or warning messages. -m <i>name</i> Compares <i>name</i> with the %M% ID keyword in the <i>s</i>.file. -rsid Checks to see if the indicated SID is ambiguous, invalid, or absent from the <i>s</i>.file. -y <i>type</i> Compares <i>type</i> with the %Y% ID keyword.
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of val: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[sccs\(1\)](#), [sccs-admin\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [what\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断

Use the `sccs-help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

引用名	scp – secure copy (remote file copy program)
用法概要	<pre> scp [-pqrVBC46] [-F ssh_config] [-S program] [-P port] [-c cipher] [-i identity_file] [-o ssh_option] [[user@]host1:]file1 [...] [[user@]host2:]file2 </pre>
描述	<p>The scp utility copies files between hosts on a network. It uses ssh(1) for data transfer, and uses the same authentication and provides the same security as ssh(1). Unlike rcp(1), scp will ask for passwords or passphrases if they are needed for authentication.</p> <p>Any file name may contain a host and user specification to indicate that the file is to be copied to/from that host. Copies between two remote hosts are permitted.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -4 Forces scp to use IPv4 addresses only. -6 Forces scp to use IPv6 addresses only. -B Selects batch mode. (Prevents asking for passwords or passphrases.) -c <i>cipher</i> Selects the cipher to use for encrypting the data transfer. This option is directly passed to ssh(1). -C Compression enable. Passes the -C flag to ssh(1) to enable compression. -F <i>ssh_config</i> Specifies an alternative per-user configuration file for ssh(1). -i <i>identity_file</i> Selects the file from which the identity (private key) for RSA authentication is read. This option is directly passed to ssh(1). -o <i>ssh_option</i> The given option is directly passed to ssh(1). -p Preserves modification times, access times, and modes from the original file. -P <i>port</i> Specifies the port to connect to on the remote host. Notice that this option is written with a capital 'P', because -p is already reserved for preserving the times and modes of the file in rcp(1). -q Disables the progress meter. -r Recursively copies entire directories. -S <i>program</i> Specifies the name of the program to use for the encrypted connection. The program must understand ssh(1) options. -v Verbose mode. Causes scp and ssh(1) to print debugging messages about their progress. This is helpful in debugging connection, authentication, and configuration problems.

操作数 The following operands are supported:
host1, host2,... The name(s) of the host from or to which the file is to be copied.
file1, file2,... The file(s) to be copied.

退出状态 The following exit values are returned:
0 Successful completion.
1 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/ssh
Interface Stability	Committed

另请参见 [rcp\(1\)](#), [ssh\(1\)](#), [ssh-add\(1\)](#), [ssh-agent\(1\)](#), [ssh-keygen\(1\)](#), [sshd\(1M\)](#), [ssh_config\(4\)](#),
[attributes\(5\)](#)

附注 Generally, use of scp with password or keyboard-interactive authentication method and two remote hosts does not work. It does work with either the pubkey, hostbased or gssapi-keyex authentication method. For the pubkey authentication method, either private keys not protected by a passphrase, or an explicit ssh agent forwarding have to be used. The gssapi-keyex authentication method works with the kerberos_v5 GSS-API mechanism, but only if the GSSAPIDelegateCredentials option is enabled.

引用名 script – make record of a terminal session

用法概要 script [-a] [*filename*]

描述 The `script` utility makes a record of everything printed on your screen. The record is written to *filename*. If no file name is given, the record is saved in the file `typescript`. See WARNINGS.

The `script` command forks and creates a sub-shell, according to the value of `$SHELL`, and records the text from this session. The script ends when the forked shell exits or when Control-d is typed.

选项 The following option is supported:

-a Appends the session record to *filename*, rather than overwriting it.

附注 `script` places everything that appears on the screen in *filename*, including prompts.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

另请参见 [attributes\(5\)](#)

警告 `script` can pose a security risk when used in directories that are writable by other users (for example, `/tmp`), especially when run by a privileged user, that is, root. Be sure that `typescript` is not a link before running `script`.

引用名	<code>sdiff</code> – print differences between two files side-by-side
用法概要	<code>sdiff [-l] [-s] [-o <i>output</i>] [-w <i>n</i>] <i>filename1 filename2</i></code>
描述	<code>sdiff</code> uses the output of the <code>diff</code> command to produce a side-by-side listing of two files indicating lines that are different. Lines of the two files are printed with a blank gutter between them if the lines are identical, a < in the gutter if the line appears only in <i>filename1</i> , a > in the gutter if the line appears only in <i>filename2</i> , and a for lines that are different. (See the EXAMPLES section below.)
选项	<p><code>-l</code> Print only the left side of any lines that are identical.to</p> <p><code>-s</code> Do not print identical lines.</p> <p><code>-o <i>output</i></code> Use the argument <i>output</i> as the name of a third file that is created as a user-controlled merge of <i>filename1</i> and <i>filename2</i>. Identical lines of <i>filename1</i> and <i>filename2</i> are copied to <i>output</i>. Sets of differences, as produced by <code>diff</code>, are printed; where a set of differences share a common gutter character. After printing each set of differences, <code>sdiff</code> prompts the user with a % and waits for one of the following user-typed commands:</p> <ul style="list-style-type: none"><code>l</code> Append the left column to the output file.<code>r</code> Append the right column to the output file.<code>s</code> Turn on silent mode; do not print identical lines.<code>v</code> Turn off silent mode.<code>e l</code> Call the editor with the left column.<code>e r</code> Call the editor with the right column.<code>e b</code> Call the editor with the concatenation of left and right.<code>e</code> Call the editor with a zero length file.<code>q</code> Exit from the program. <p>On exit from the editor, the resulting file is concatenated to the end of the <i>output</i> file.</p> <p><code>-w <i>n</i></code> Use the argument <i>n</i> as the width of the output line. The default line length is 130 characters.</p>
用法	See largefile(5) for the description of the behavior of <code>sdiff</code> when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).
示例	<p>示例 1 An example of the <code>sdiff</code> command.</p> <p>A sample output of <code>sdiff</code> follows.</p>

示例 1 An example of the `sdiff` command. (续)

```
x | y
a   a
b <
c <
d   d
   > c
```

环境变量

If any of the `LC_*` variables (`LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, `LC_COLLATE`, `LC_NUMERIC`, and `LC_MONETARY`) (see [environ\(5\)](#)) are not set in the environment, the operational behavior of `sdiff` for each corresponding locale category is determined by the value of the `LANG` environment variable. If `LC_ALL` is set, its contents are used to override both the `LANG` and the other `LC_*` variables. If none of the above variables is set in the environment, the "C" locale determines how `sdiff` behaves.

`LC_CTYPE` Determines how `sdiff` handles characters. When `LC_CTYPE` is set to a valid value, `sdiff` can display and handle text and filenames containing valid characters for that locale.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

另请参见

[diff\(1\)](#), [ed\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#)

引用名 sed – stream editor

用法概要

```
/usr/bin/sed [-n] script [file]...  
/usr/bin/sed [-n] [-e script]... [-f script_file]...  
          [file]...  
/usr/xpg4/bin/sed [-n] script [file]...  
/usr/xpg4/bin/sed [-n] [-e script]... [-f script_file]...  
          [file]...
```

描述

The sed utility is a stream editor that reads one or more text files, makes editing changes according to a script of editing commands, and writes the results to standard output. The script is obtained from either the *script* operand string, or a combination of the option-arguments from the *-e script* and *-f script_file* options.

The sed utility is a text editor. It cannot edit binary files or files containing ASCII NUL (\0) characters or very long lines.

选项

The following options are supported:

-e script *script* is an edit command for sed. See USAGE below for more information on the format of *script*. If there is just one *-e* option and no *-f* options, the flag *-e* may be omitted.

-f script_file Takes the script from *script_file*. *script_file* consists of editing commands, one per line.

-n Suppresses the default output.

Multiple *-e* and *-f* options may be specified. All commands are added to the script in the order specified, regardless of their origin.

操作数

The following operands are supported:

file A path name of a file whose contents will be read and edited. If multiple *file* operands are specified, the named files will be read in the order specified and the concatenation will be edited. If no *file* operands are specified, the standard input will be used.

script A string to be used as the script of editing commands. The application must not present a *script* that violates the restrictions of a text file except that the final character need not be a NEWLINE character.

用法

A script consists of editing commands, one per line, of the following form:

```
[ address [ , address ] ] command [ arguments ]
```

Zero or more blank characters are accepted before the first address and before *command*. Any number of semicolons are accepted before the first address.

In normal operation, sed cyclically copies a line of input (less its terminating NEWLINE character) into a *pattern space* (unless there is something left after a D command), applies in sequence all commands whose *addresses* select that pattern space, and copies the resulting pattern space to the standard output (except under -n) and deletes the pattern space. Whenever the pattern space is written to standard output or a named file, sed will immediately follow it with a NEWLINE character.

Some of the commands use a *hold space* to save all or part of the *pattern space* for subsequent retrieval. The *pattern* and *hold spaces* will each be able to hold at least 8192 bytes.

sed Addresses

An *address* is either empty, a decimal number that counts input lines cumulatively across files, a \$ that addresses the last line of input, or a context address, which consists of a */regular expression/* as described on the [regexp\(5\)](#) manual page.

A command line with no addresses selects every pattern space.

A command line with one address selects each pattern space that matches the address.

A command line with two addresses selects the inclusive range from the first pattern space that matches the first address through the next pattern space that matches the second address. Thereafter the process is repeated, looking again for the first address. (If the second address is a number less than or equal to the line number selected by the first address, only the line corresponding to the first address is selected.)

Typically, address are separated from each other by a comma (,). They may also be separated by a semicolon (;).

sed Regular Expressions

sed supports the basic regular expressions described on the [regexp\(5\)](#) manual page, with the following additions:

`\cREc` In a context address, the construction `\cREc`, where *c* is any character other than a backslash or NEWLINE character, is identical to `/RE/`. If the character designated by *c* appears following a backslash, then it is considered to be that literal character, which does not terminate the RE. For example, in the context address `\xabc\xdefx`, the second *x* stands for itself, so that the regular expression is `abcxdef`.

`\n` The escape sequence `\n` matches a NEWLINE character embedded in the pattern space. A literal NEWLINE character must not be used in the regular expression of a context address or in the substitute command.

Editing commands can be applied only to non-selected pattern spaces by use of the negation command ! (described below).

sed Editing Commands

In the following list of functions the maximum number of permissible addresses for each function is indicated.

The `r` and `w` commands take an optional *rfile* (or *wfile*) parameter, separated from the command letter by one or more blank characters.

Multiple commands can be specified by separating them with a semicolon (;) on the same command line.

The *text* argument consists of one or more lines, all but the last of which end with \ to hide the NEWLINE. Each embedded NEWLINE character in the text must be preceded by a backslash. Other backslashes in text are removed and the following character is treated literally. Backslashes in text are treated like backslashes in the replacement string of an `s` command, and may be used to protect initial blanks and tabs against the stripping that is done on every script line. The *rfile* or *wfile* argument must terminate the command line and must be preceded by exactly one blank. The use of the *wfile* parameter causes that file to be initially created, if it does not exist, or will replace the contents of an existing file. There can be at most 10 distinct *wfile* arguments.

Regular expressions match entire strings, not just individual lines, but a NEWLINE character is matched by \n in a sed RE. A NEWLINE character is not allowed in an RE. Also notice that \n cannot be used to match a NEWLINE character at the end of an input line; NEWLINE characters appear in the pattern space as a result of the `N` editing command.

Two of the commands take a *command-list*, which is a list of sed commands separated by NEWLINE characters, as follows:

```
{ command
  command
}
```

The { can be preceded with blank characters and can be followed with white space. The *commands* can be preceded by white space. The terminating } must be preceded by a NEWLINE character and can be preceded or followed by <blank>s. The braces may be preceded or followed by <blank>s. The command may be preceded by <blank>s, but may not be followed by <blank>s.

The following table lists the functions, with the maximum number of permissible addresses.

Max Address	Command	Description
1	a\ <i>text</i>	Append by executing <code>N</code> command or beginning a new cycle. Place <i>text</i> on the output before reading the next input line.
2	b <i>label</i>	Branch to the : command bearing the <i>label</i> . If <i>label</i> is empty, branch to the end of the script. Labels are recognized unique up to eight characters.
2	c\ <i>text</i>	Change. Delete the pattern space. Place <i>text</i> on the output. Start the next cycle.

Max Address	Command	Description
2	d	Delete the pattern space. Start the next cycle.
2	D	Delete the initial segment of the pattern space through the first new-line. Start the next cycle. (See the N command below.)
2	g	Replace the contents of the pattern space by the contents of the hold space.
2	G	Append the contents of the hold space to the pattern space.
2	h	Replace the contents of the hold space by the contents of the pattern space.
2	H	Append the contents of the pattern space to the hold space.
1	i\ <i>text</i>	Insert. Place <i>text</i> on the standard output.
2	l	<code>/usr/bin/sed</code> : List the pattern space on the standard output in an unambiguous form. Non-printable characters are displayed in octal notation and long lines are folded.
		<code>/usr/xpg4/bin/sed</code> : List the pattern space on the standard output in an unambiguous form. Non-printable characters are displayed in octal notation and long lines are folded. The characters (\, \a, \b, \f, \r, \t, and \v) are written as the corresponding escape sequences. Non-printable characters not in that table will be written as one three-digit octal number (with a preceding backslash character) for each byte in the character (most significant byte first). If the size of a byte on the system is greater than nine bits, the format used for non-printable characters is implementation dependent. Long lines are folded, with the point of folding indicated by writing a backslash followed by a NEWLINE; the length at which folding occurs is unspecified, but should be appropriate for the output device. The end of each line is marked with a \$.
2	n	Copy the pattern space to the standard output if default output is not suppressed. Replace the pattern space with the next line of input.
2	N	Append the next line of input to the pattern space with an embedded new-line. (The current line number changes.) If no next line of input is available, the N command verb shall branch to the end of the script and quit without starting a new cycle and without writing the pattern space.
2	p	Print. Copy the pattern space to the standard output.

Max Address	Command	Description
2	P	Copy the initial segment of the pattern space through the first new-line to the standard output.
1	q	Quit. Branch to the end of the script. Do not start a new cycle.
2	r <i>rfile</i>	Read the contents of <i>rfile</i> . Place them on the output before reading the next input line. If <i>rfile</i> does not exist or cannot be read, it is treated as if it were an empty file, causing no error condition.
2	t <i>label</i>	Test. Branch to the : command bearing the <i>label</i> if any substitutions have been made since the most recent reading of an input line or execution of a t. If <i>label</i> is empty, branch to the end of the script.
2	w <i>wfile</i>	Write. Append the pattern space to <i>wfile</i> . The first occurrence of w will cause <i>wfile</i> to be cleared. Subsequent invocations of w will append. Each time the sed command is used, <i>wfile</i> is overwritten.
2	x	Exchange the contents of the pattern and hold spaces.
2	! <i>command</i>	Don't. Apply the <i>command</i> (or group, if <i>command</i> is { }) only to lines <i>not</i> selected by the address(es).
0	: <i>label</i>	This command does nothing; it bears a <i>label</i> for b and t commands to branch to.
1	=	Place the current line number on the standard output as a line.
2	{ <i>command-list</i> }	Execute <i>command-list</i> only when the pattern space is selected.
0		An empty command is ignored.
0	#	If a # appears as the first character on a line of a script file, then that entire line is treated as a comment, with one exception: if a # appears on the first line and the character after the # is an n, then the default output will be suppressed. The rest of the line after #n is also ignored. A script file must contain at least one non-comment line.

Max Addr	Command (Using <i>strings</i>) and Description
2	s/ <i>regular expression</i> / <i>replacement</i> / <i>flags</i>

Max Addr	Command (Using <i>strings</i>) and Description
	<p>Substitute the <i>replacement</i> string for instances of the <i>regular expression</i> in the pattern space. Any character other than backslash or newline can be used instead of a slash to delimit the RE and the replacement. Within the RE and the replacement, the RE delimiter itself can be used as a literal character if it is preceded by a backslash.</p> <p>An ampersand (&) appearing in the <i>replacement</i> will be replaced by the string matching the RE. The special meaning of & in this context can be suppressed by preceding it by backslash. The characters <code>\n</code>, where <i>n</i> is a digit, will be replaced by the text matched by the corresponding backreference expression. For each backslash (\) encountered in scanning <i>replacement</i> from beginning to end, the following character loses its special meaning (if any). It is unspecified what special meaning is given to any character other than &, \ or digits.</p> <p>A line can be split by substituting a NEWLINE character into it. The application must escape the NEWLINE character in the <i>replacement</i> by preceding it with backslash. A substitution is considered to have been performed even if the replacement string is identical to the string that it replaces.</p> <p><i>flags</i> is zero or more of:</p> <p><i>n n</i>= 1 - 512. Substitute for just the <i>n</i>th occurrence of the <i>regular expression</i>.</p> <p><i>g</i> Global. Substitute for all nonoverlapping instances of the <i>regular expression</i> rather than just the first one. If both <i>g</i> and <i>n</i> are specified, the results are unspecified.</p>
	<p><i>p</i> Print the pattern space if a replacement was made.</p> <p><i>P</i> Copy the initial segment of the pattern space through the first new-line to the standard output.</p> <p><i>w wfile</i> Write. Append the pattern space to <i>wfile</i> if a replacement was made. The first occurrence of <i>w</i> will cause <i>wfile</i> to be cleared. Subsequent invocations of <i>w</i> will append. Each time the sed command is used, <i>wfile</i> is overwritten.</p>
2	<p><i>y/ string1 / string2 /</i></p> <p>Transform. Replace all occurrences of characters in <i>string1</i> with the corresponding characters in <i>string2</i>. <i>string1</i> and <i>string2</i> must have the same number of characters, or if any of the characters in <i>string1</i> appear more than once, the results are undefined. Any character other than backslash or NEWLINE can be used instead of slash to delimit the strings. Within <i>string1</i> and <i>string2</i>, the delimiter itself can be used as a literal character if it is preceded by a backslash. For example, <i>y/abc/ABC/</i> replaces a with A, b with B, and c with C.</p>

See [largefile\(5\)](#) for the description of the behavior of sed when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

示例 1 An example sed script

This sed script simulates the BSD `cat -s` command, squeezing excess blank lines from standard input.

```
sed -n '
# Write non-empty lines.
./ {
    p
    d
}
# Write a single empty line, then look for more empty lines.
/^\$/ p
# Get next line, discard the held <newline> (empty line),
# and look for more empty lines.
:Empty
/^\$/ {
    N
    s/./ /
    b Empty
}
# Write the non-empty line before going back to search
# for the first in a set of empty lines.
p
,
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of sed: LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/sed

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Not enabled

/usr/xpg4/bin/sed

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed

ATTRIBUTETYPE	ATTRIBUTEVALUE
Standard	See standards(5) .

另请参见

[awk\(1\)](#), [ed\(1\)](#), [grep\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#), [standards\(5\)](#)

引用名 sed – stream editor

用法概要 sed [-n] [-e *script*] [-f *sfilename*] [*filename*]. . .

描述 The sed utility copies the *filenames* (standard input default) to the standard output, edited according to a script of commands.

选项 The following options are supported:

- n Suppresses the default output.
- e *script* *script* is an edit command for sed. If there is just one -e option and no -f options, the -e flag may be omitted.
- f *sfilename* Takes the script from *sfilename*.

用法

sed scripts consist of editing commands, one per line, of the following form:

```
[ address [, address ] ] function [ arguments ]
```

In normal operation, sed cyclically copies a line of input into a *pattern space* (unless there is something left after a D command), sequentially applies all commands with *addresses* matching that pattern space until reaching the end of the script, copies the pattern space to the standard output (except under -n), and finally, deletes the pattern space.

Some commands use a *hold space* to save all or part of the pattern space for subsequent retrieval.

An *address* is either:

- a decimal number linecount, which is cumulative across input files;
- a \$, which addresses the last input line;
- or a context address, which is a */regular expression/* as described on the [regexp\(5\)](#) manual page, with the following exceptions:

\?RE? In a context address, the construction \?*regular expression*?, where ? is any character, is identical to */regular expression/*. Note: in the context address \xabc\xdefx, the second x stands for itself, so that the regular expression is abcxdef.

\n Matches a NEWLINE embedded in the pattern space.

. Matches any character except the NEWLINE ending the pattern space.

null A command line with no address selects every pattern space.

address Selects each pattern space that matches.

address1 , *address2* Selects the inclusive range from the first pattern space matching *address1* to the first pattern space matching *address2*. Selects only one line if *address1* is greater than or equal to *address2*.

Comments If the first nonwhite character in a line is a '#' (pound sign), sed treats that line as a comment, and ignores it. If, however, the first such line is of the form:

#n

sed runs as if the -n flag were specified.

Functions The maximum number of permissible addresses for each function is indicated in parentheses in the list below.

An argument denoted *text* consists of one or more lines, all but the last of which end with \ to hide the NEWLINE. Backslashes in text are treated like backslashes in the replacement string of an s command, and may be used to protect initial SPACE and TAB characters against the stripping that is done on every script line.

An argument denoted *rfilename* or *wfilename* must terminate the command line and must be preceded by exactly one SPACE. Each *wfilename* is created before processing begins. There can be at most 10 distinct *wfilename* arguments.

(1)a\

text Append: place *text* on the output before reading the next input line.

(2)b *label*

Branch to the ':' command bearing the *label*. Branch to the end of the script if *label* is empty.

(2)c\

text Change: delete the pattern space. With 0 or 1 address or at the end of a 2 address range, place *text* on the output. Start the next cycle.

(2)d

Delete the pattern space. Start the next cycle.

(2)D

Delete the initial segment of the pattern space through the first NEWLINE. Start the next cycle.

(2)g

Replace the contents of the pattern space by the contents of the hold space.

(2)G

Append the contents of the hold space to the pattern space.

(2)h

Replace the contents of the hold space by the contents of the pattern space.

(2)H

Append the contents of the pattern space to the hold space.

(1)i\

text Insert: place *text* on the standard output.

- (2)l List the pattern space on the standard output in an unambiguous form. Non-printing characters are spelled in two digit ASCII and long lines are folded.
- (2)n Copy the pattern space to the standard output. Replace the pattern space with the next line of input.
- (2)N Append the next line of input to the pattern space with an embedded newline. (The current line number changes.)
- (2)p Print: copy the pattern space to the standard output.
- (2)P Copy the initial segment of the pattern space through the first NEWLINE to the standard output.
- (1)q Quit: branch to the end of the script. Do not start a new cycle.
- (2)r *rfilename* Read the contents of *rfilename*. Place them on the output before reading the next input line.
- (2)s/*regular expression*/*replacement*/*flags* Substitute the *replacement* string for instances of the *regular expression* in the pattern space. Any character may be used instead of '/'. For a fuller description see [regexp\(5\)](#). *flags* is zero or more of:
- n* *n*= 1 – 512. Substitute for just the *n*th occurrence of the *regular expression*.
- g Global: substitute for all nonoverlapping instances of the *regular expression* rather than just the first one.
- p Print the pattern space if a replacement was made.
- w *wfilename* Write: append the pattern space to *wfilename* if a replacement was made.
- (2)t *label* Test: branch to the ':' command bearing the *label* if any substitutions have been made since the most recent reading of an input line or execution of a t. If *label* is empty, branch to the end of the script.
- (2)w *wfilename* Write: append the pattern space to *wfilename*.
- (2)x Exchange the contents of the pattern and hold spaces.

(2)y/string1/string2/	Transform: replace all occurrences of characters in <i>string1</i> with the corresponding character in <i>string2</i> . The lengths of <i>string1</i> and <i>string2</i> must be equal.
(2)! function	Do not: apply the function (or group, if function is '{') only to lines <i>not</i> selected by the address(es).
(0): label	This command does nothing. It bears a <i>label</i> for b and t commands to branch to. <i>Note:</i> The maximum length of <i>label</i> is seven characters.
(1)=	Place the current line number on the standard output as a line.
(2){	Execute the following commands through a matching '}' only when the pattern space is selected. Commands are separated by ';'.
(0)	An empty command is ignored.

Large Files

See [largefile\(5\)](#) for the description of the behavior of sed when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

诊断

Too many commands	The command list contained more than 200 commands.
Too much command text	The command list was too big for sed to handle. Text in the a, c, and i commands, text read in by r commands, addresses, regular expressions and replacement strings in s commands, and translation tables in y commands all require sed to store data internally.
Command line too long	A command line was longer than 4000 characters.
Too many line numbers	More than 256 decimal number linecounts were specified as addresses in the command list.
Too many files in w commands	More than 10 different files were specified in w commands or w options for s commands in the command list.
Too many labels	More than 50 labels were specified in the command list.
Unrecognized command	A command was not one of the ones recognized by sed.
Extra text at end of command	A command had extra text after the end.

Illegal line number	An address was neither a decimal number linecount, a \$, nor a context address.
Space missing before filename	There was no space between an r or w command, or the w option for a s command, and the filename specified for that command.
Too many { 's	There were more { than } in the list of commands to be executed.
Too many } 's	There were more } than { in the list of commands to be executed.
No addresses allowed	A command that takes no addresses had an address specified.
Only one address allowed	A command that takes one address had two addresses specified.
"\digit" out of range	The number in a \n item in a regular expression or a replacement string in ans command was greater than 9.
Bad number	One of the endpoints in a range item in a regular expression (that is, an item of the form {n} or {n,m}) was not a number.
Range endpoint too large	One of the endpoints in a range item in a regular expression was greater than 255.
More than 2 numbers given in \{ \}	More than two endpoints were given in a range expression.
} expected after \	A \ appeared in a range expression and was not followed by a }.
First number exceeds second in \{ \}	The first endpoint in a range expression was greater than the second.
Illegal or missing delimiter	The delimiter at the end of a regular expression was absent.
\(\) imbalance	There were more \(than \), or more \) than \(, in a regular expression.
[] imbalance	There were more [than], or more] than [, in a regular expression.

First RE may not be null	The first regular expression in an address or in a <code>s</code> command was null (empty).
Ending delimiter missing on substitution	The ending delimiter in a <code>s</code> command was absent.
Ending delimiter missing on string	The ending delimiter in a <code>y</code> command was absent.
Transform strings not the same size	The two strings in a <code>y</code> command were not the same size.
Suffix too large - 512 max	The suffix in a <code>s</code> command, specifying which occurrence of the regular expression should be replaced, was greater than 512.
Label too long	A label in a command was longer than 8 characters.
Duplicate labels	The same label was specified by more than one <code>:</code> command.
File name too long	The filename specified in a <code>r</code> or <code>w</code> command, or in the <code>w</code> option for a <code>s</code> command, was longer than 1024 characters.
Output line too long	An output line was longer than 4000 characters long.
Too many appends or reads after line <i>n</i>	More than 20 <code>a</code> or <code>r</code> commands were to be executed for line <i>n</i> .
Hold space overflowed.	More than 4000 characters were to be stored in the <i>hold space</i> .

文件

usr/ucb/sed BSD sed

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见

[awk\(1\)](#), [grep\(1\)](#), [lex\(1\)](#), [attributes\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#)

已知问题

There is a combined limit of 200 `-e` and `-f` arguments. In addition, there are various internal size limits which, in rare cases, may overflow. To overcome these limitations, either combine or break out scripts, or use a pipeline of `sed` commands.

引用名 set, unset, setenv, unsetenv, export – shell built-in functions to determine the characteristics for environmental variables of the current shell and its descendents

用法概要

```
sh      set [--aefhkntuvx argument...]...
        unset [name]...
        export [name]...

csh     set [var [= value]]
        set var [n] = word
        unset pattern
        setenv [VAR [word]]
        unsetenv variable

ksh88  set [ $\pm$ abCefhkmnopstuvx] [ $\pm$ o option]... [ $\pm$ A name]
        [arg]...
        unset [-f] name...
        **export [name [=value]]...
        **export [-p]

ksh     +set [ $\pm$ abCefGhkmnoprstuvx] [ $\pm$ o option]... [ $\pm$ A vname]
        [arg]...
        +unset [-fnv] vname...
        ++export [-p] [name[=value]]...
```

描述

```
sh      The set built-in command has the following options:

--      Does not change any of the flags. This option is useful in setting $1 to -.
-a      Marks variables which are modified or created for export.
-e      Exits immediately if a command exits with a non-zero exit status.
-f      Disables file name generation.
-h      Locates and remembers function commands as functions are defined. Function
        commands are normally located when the function is executed.
-k      All keyword arguments are placed in the environment for a command, not just those
        that precede the command name.
-n      Reads commands but does not execute them.
-t      Exits after reading and executing one command.
```

- u Treats unset variables as an error when substituting.
- v Prints shell input lines as they are read.
- x Prints commands and their arguments as they are executed.

Using + rather than – causes these flags to be turned off. These flags can also be used upon invocation of the shell. The current set of flags can be found in \$-. The remaining *arguments* are positional parameters and are assigned, in order, to \$1, \$2, If no *arguments* are specified the values of all names are printed.

For each *name*, unset removes the corresponding variable or function value. The variables PATH, PS1, PS2, MAILCHECK, and IF cannot be unset.

With the `export` built-in, the specified *names* are marked for automatic export to the *environment* of subsequently executed commands. If no arguments are specified, variable names that have been marked for export during the current shell's execution are listed. Function names are *not* exported.

csh

With no arguments, `set` displays the values of all shell variables. Multiword values are displayed as a parenthesized list. With the *var* argument alone, `set` assigns an empty (null) value to the variable *var*. With arguments of the form *var* = *value* `set` assigns *value* to *var*, where *value* is one of:

- word* A single word (or quoted string).
- (*wordlist*) A space-separated list of words enclosed in parentheses.

Values are command and filename expanded before being assigned. The form `set var[n]=word` replaces the *n*'th word in a multiword value with *word*.

`unset` removes variables whose names match (filename substitution) *pattern*. All variables are removed by `'unset *'`.

With no arguments, `setenv` displays all environment variables. With the *VAR* argument, `setenv` sets the environment variable *VAR* to an empty (null) value. (By convention, environment variables are normally specified upper-case names.) With both *VAR* and *word* arguments specified, `setenv` sets *VAR* to *word*, which must be either a single word or a quoted string. The *PATH* variable can take multiple *word* arguments, separated by colons (see *EXAMPLES*). The most commonly used environment variables, *USER*, *TERM*, and *PATH*, are automatically imported to and exported from the *csh* variables *user*, *term*, and *path*. Use `setenv` if you need to change these variables. In addition, the shell sets the *PWD* environment variable from the *csh* variable *cwd* whenever the latter changes.

The environment variables *LC_CTYPE*, *LC_MESSAGES*, *LC_TIME*, *LC_COLLATE*, *LC_NUMERIC*, and *LC_MONETARY* take immediate effect when changed within the C shell. See [environ\(5\)](#) for descriptions of these environment variables.

unsetenv removes *variable* from the environment. As with unset, pattern matching is not performed.

ksh88

The flags for the set built-in have meaning as follows:

- A Array assignment. Unsets the variable *name* and assigns values sequentially from the list *arg*. If +A is used, the variable *name* is not unset first.
- a All subsequent variables that are defined are automatically exported.
- b Causes the shell to notify the user asynchronously of background job completions.
- C Prevents existing files from being overwritten by the shell's > redirection operator. The >| redirection operator overrides this noclobber option for an individual file.
- e If a command has a non-zero exit status, executes the ERR trap, if set, and exits. This mode is disabled while reading profiles.
- f Disables file name generation.
- h Each command becomes a tracked alias when first encountered.
- k All variable assignment arguments are placed in the environment for a command, not just those that precede the command name.
- m Background jobs run in a separate process group and a line prints upon completion. The exit status of background jobs is reported in a completion message. On systems with job control, this flag is turned on automatically for interactive shells.
- n Reads commands and checks them for syntax errors, but does not execute them. Ignored for interactive shells.
- +o Writes the current option settings to standard output in a format that is suitable for reinput to the shell as commands that achieve the same option settings.
- o *option* The *option* argument can be one of the following option names:
 - allexport Same as -a.
 - errexit Same as -e.
 - bgnice All background jobs are run at a lower priority. This is the default mode. emacs Puts you in an emacs style in-line editor for command entry.
 - gmacs Puts you in a gmacs style in-line editor for command entry.
 - ignoreeof The shell does not exit on end-of-file. The command `exit` must be used.

keyword	Same as -k.
markdirs	All directory names resulting from file name generation have a trailing / appended.
monitor	Same as -m.
noclobber	Prevents redirection operator > from truncating existing files. Requires the > operator to truncate a file when turned on. Same as -C.
noexec	Same as -n.
noglob	Same as -f.
nolog	Does not save function definitions in history file.
notify	Same as -b.
nounset	Same as -u.
privileged	Same as -p.
verbose	Same as -v.
trackall	Same as -h.
vi	Puts you in insert mode of a vi style in-line editor until you hit escape character 033. This puts you in control mode. A return sends the line.
viraw	Each character is processed as it is typed in vi mode.
xtrace	Same as -x.

If no option name is supplied then the current option settings are printed.

- p Disables processing of the \$HOME/.profile file and uses the file /etc/suid_profile instead of the ENV file. This mode is on whenever the effective uid is not equal to the real uid, or when the effective gid is not equal to the real gid. Turning this off causes the effective uid and gid to be set to the real uid and gid.
- s Sorts the positional parameters lexicographically.
- t Exits after reading and executing one command.
- u Treats unset parameters as an error when substituting.
- v Prints shell input lines as they are read.
- x Prints commands and their arguments as they are executed.
- Turns off -x and -v flags and stops examining arguments for flags.

- Does not change any of the flags. This option is useful in setting \$1 to a value beginning with -. If no arguments follow this flag then the positional parameters are unset.

Using + rather than - causes these flags to be turned off. These flags can also be used upon invocation of the shell. The current set of flags can be found in \$-. Unless -A is specified, the remaining arguments are positional parameters and are assigned, in order, to \$1 \$2 If no arguments are specified then the names and values of all variables are printed on the standard output.

The variables specified by the list of *names* are unassigned, that is, their values and attributes are erased. `readonly` variables cannot be unset. If the -f flag is set, then the names refer to function names. Unsetting `ERRNO`, `LINENO`, `MAILCHECK`, `OPTARG`, `OPTIND`, `RANDOM`, `SECONDS`, `TMOU`, and `_` removes their special meaning even if they are subsequently assigned.

When using `unset`, the variables specified by the list of *names* are unassigned, i.e., their values and attributes are erased. `readonly` variables cannot be unset. If the -f flag is set, then the names refer to function names. Unsetting `ERRNO`, `LINENO`, `MAILCHECK`, `OPTARG`, `OPTIND`, `RANDOM`, `SECONDS`, `TMOU`, and `_` removes their special meaning even if they are subsequently assigned.

With the `export` built-in, the specified *names* are marked for automatic export to the environment of subsequently-executed commands.

When -p is specified, `export` writes to the standard output the names and values of all exported variables in the following format:

```
"export %s=%s\n", name, value
```

if *name* is set, and:

```
"export %s\n", name
```

if *name* is unset.

The shell formats the output, including the proper use of quoting, so that it is suitable for reinput to the shell as commands that achieve the same exporting results, except for the following:

1. Read-only variables with values cannot be reset.
2. Variables that were unset at the time they were output are not reset to the unset state if a value is assigned to the variable between the time the state was saved and the time at which the saved output is reinput to the shell.

On this manual page, [ksh88\(1\)](#) commands that are preceded by one or two * (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.

2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by ****** that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.

ksh

set sets or unsets options and positional parameters. Options that are specified with a - cause the options to be set. Options that are specified with a + cause the option to be unset.

set without any options or arguments displays the names and values of all shell variables in the order of the collation sequence in the current locale. The values are quoted so that they are suitable for input again to the shell.

If no arguments are specified, not even the end of options argument --, the positional parameters are unchanged. Otherwise, unless the -A option has been specified, the positional parameters are replaced by the list of arguments. A first argument of -- is ignored when setting positional parameters.

For backwards compatibility, a set command without any options specified, whose first argument is - turns off the -v and -x options. If any additional arguments are specified, they replace the positional parameters.

The options for set in ksh are:

- a Set the export attribute for each variable whose name does not contain a . that you assign a value in the current shell environment.
- A *name* Assign the arguments sequentially to the array named by *name* starting at subscript 0 rather than to the positional parameters.
- b The shell writes a message to standard error as soon it detects that a background job completes rather than waiting until the next prompt.
- B Enable { . . . } group expansion. On by default.
- C Prevents existing regular files from being overwritten using the > redirection operator. The >| redirection overrides this noclobber option.
- e A simple command that has a non-zero exit status causes the shell to exit unless the simple command is:
 - contained in an && or || list
 - the command immediately following if, while, or until
 - contained in the pipeline following !
- f Pathname expansion is disabled.

- G Causes ** by itself to also match all sub-directories during pathname expansion.
- h Obsolete. Causes each command whose name has the syntax of an alias to become a tracked alias when it is first encountered.
- H Enable !-style history expansion similar to csh.
- k This is obsolete. All arguments of the form *name=value* are removed and placed in the variable assignment list for the command. Ordinarily, variable assignments must precede command arguments.
- m When enabled, the shell runs background jobs in a separate process group and displays a line upon completion. This mode is enabled by default for interactive shells on systems that support job control.
- n The shell reads commands and checks for syntax errors, but does not execute the command. Usually specified on command invocation.
- o [*option*] If option is not specified, the list of options and their current settings is written to standard output. When invoked with a + the options are written in a format that can be input again to the shell to restore the settings. This option can be repeated to enable or disable multiple options.

The value of *option* must be one of the following:

allexport	Same as -a.
bgnice	All background jobs are run at lower priorities.
braceexpand	Same as -B.
emacs	Enables or disables emacs editing mode.
errexit	Same as -e.
globstar	Equivalent to -G.
gmacs	Enables or disables gmacs. gmacs editing mode is the same as emacs editing mode, except for the handling of CTRL-T.
histexpand	Same as -H.
ignoreeof	The interactive shell does not exit on end-of-file.
keyword	Same as -k.
markdirs	All directory names resulting from file name generation have a trailing / appended.
monitor	Same as -m.

<code>multiline</code>	Use multiple lines when editing lines that are longer than the window width.
<code>noclobber</code>	Same as <code>-C</code> .
<code>noexec</code>	Same as <code>-n</code> .
<code>noglob</code>	Same as <code>-f</code> .
<code>nolog</code>	This has no effect. It is provided for backward compatibility.
<code>notify</code>	Same as <code>-b</code> .
<code>nounset</code>	Same as <code>-u</code> .
<code>pipefail</code>	A pipeline does not complete until all components of the pipeline have completed, and the exit status of the pipeline is the value of the last command to exit with non-zero exit status, or is zero if all commands return zero exit status.
<code>privileged</code>	Same as <code>-p</code> .
<code>showme</code>	Simple commands preceded by a ; are traced as if <code>-x</code> were enabled but not executed.
<code>trackall</code>	Same as <code>-h</code> .
<code>verbose</code>	Same as <code>-v</code> .
<code>vi</code>	Enables or disables <code>vi</code> editing mode.
<code>viraw</code>	Does not use canonical input mode when using <code>vi</code> edit mode
<code>xtrace</code>	Same as <code>-x</code> .
<code>-p</code>	Privileged mode. Disabling <code>-p</code> sets the effective user id to the real user id, and the effective group id to the real group id. Enabling <code>-p</code> restores the effective user and group ids to their values when the shell was invoked. The <code>-p</code> option is on whenever the real and effective user id is not equal or the real and effective group id is not equal. User profiles are not processed when <code>-p</code> is enabled.
<code>-r</code>	Restricted. Enables restricted shell. This option cannot be unset once enabled.
<code>-s</code>	Sort the positional parameters
<code>-t</code>	Obsolete. The shell reads one command and then exits.
<code>-u</code>	If enabled, the shell displays an error message when it tries to expand a variable that is unset.
<code>-v</code>	Verbose. The shell displays its input onto standard error as it reads it.
<code>-x</code>	Execution trace. The shell displays each command after all expansion and before execution preceded by the expanded value of the <code>PS4</code> parameter.

The following exit values are returned by `set` in `ksh`:

- 0 Successful completion.
- >0 An error occurred.

For each *name* specified, `unset` unsets the variable, or function if `-f` is specified, from the current shell execution environment. Read-only variables cannot be unset.

The options for `unset` in `ksh` are:

- `-f` Where *name* refers to a function name, the shell unsets the function definition.
- `-n` If *name* refers to variable that is a reference, the variable *name* is unset rather than the variable it references. Otherwise, this option is equivalent to the `-v` option.
- `-v` Where *name* refers to a variable name, the shell unsets it and removes it from the environment. This is the default behavior.

The following exit values are returned by `unset` in `ksh`:

- 0 Successful completion. All names were successfully unset.
- >0 An error occurred, or one or more *name* operands could not be unset

`export` sets the export attribute on each of the variables specified by name which causes them to be in the environment of subsequently executed commands. If `=value` is specified, the variable *name* is set to *value*.

If no *name* is specified, the names and values of all exported variables are written to standard output.

`export` is built-in to the shell as a declaration command so that field splitting and pathname expansion are not performed on the arguments. Tilde expansion occurs on value.

The options for `export` in `ksh` are:

- `-p` Causes the output to be in the form of `export` commands that can be used as input to the shell to recreate the current exports.

The following exit values are returned by `export` in `ksh`:

- 0 Successful completion.
- >0 An error occurred.

On this manual page, `ksh(1)` commands that are preceded by one or two `+` are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.

2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. They are not valid function names.
5. Words, following a command preceded by ++ that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and field splitting and file name generation are not performed.

示例

csH

The following example sets the PATH variable to search for files in the /bin, /usr/bin, /usr/sbin, and /usr/ucb/bin directories, in that order:

```
setenv PATH "/bin:/usr/bin:/usr/sbin:usr/ucb/bin"
```

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [read\(1\)](#), [sh\(1\)](#), [typeset\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#)

引用名 setfacl – 修改一个或多个文件的访问控制列表 (Access Control List, ACL)

用法概要 setfacl [-r] -s *acl_entries* *file*

setfacl [-r] -md *acl_entries* *file*

setfacl [-r] -f *acl_file* *file*

描述 对于指定的每个文件，setfacl 将替换其整个 ACL（包括目录上的缺省 ACL），或者将添加、修改或删除一个或多个 ACL 条目（包括目录上的缺省条目）。

使用 setfacl 命令时，会导致对文件权限位发生更改。文件所有者的用户 ACL 条目发生更改时，将会修改文件所有者类权限位。文件组类的组 ACL 条目发生更改时，将会修改文件组类权限位。其他 ACL 条目发生更改时，将会修改文件的其他类权限位。

如果您使用 chmod(1) 命令更改含有 ACL 条目的文件的文件组所有者权限，则会同时将文件组所有者权限和 ACL 掩码更改为新的权限。请注意，对于文件中有其 ACL 条目的附加用户和组，新的 ACL 掩码权限可能会更改其有效权限。

目录可以包含缺省 ACL 条目。如果某个文件或目录是在包含缺省 ACL 条目的目录中创建的，则该新创建的文件具有根据缺省 ACL 条目的交集生成的权限以及创建时请求的权限。如果目录包含缺省 ACL 条目，则不会应用 umask(1)。如果为某个特定用户（或多个用户）指定了缺省 ACL，该文件会创建一个常规 ACL。否则，将根据上面所述的交集初始化模式位。缺省 ACL 应当被看作可以授予的最大自主访问权限。

可使用 setfacl 命令对 UFS 文件系统中的文件设置 ACL，UFS 文件系统支持 POSIX 式 ACL（或 aclent_t 式 ACL）。可使用 chmod 命令对 ZFS 文件系统中的文件设置 ACL，ZFS 文件系统支持 NFSv4 式 ACL（或 ace_t 式 ACL）。

acl_entries 语法 对于 -m 和 -s 选项，*acl_entries* 是一个或多个以逗号分隔的 ACL 条目。

ACL 条目包含以下以冒号分隔的字段：

entry_type 设置文件权限时所基于的 ACL 条目的类型。例如，*entry_type* 可以是 *user*（文件的所有者）或 *mask*（ACL 掩码）。

uid 或 *gid* 用户名或用户标识号。或者，组名或组标识号。

perms 表示对 *entry_type* 设置的权限。*perms* 可以由符号字符 *rwX* 或数字指示（随 chmod 命令使用的相同权限编号）。

下表列出了有效的 ACL 条目（只能为目录指定缺省条目）：

ACL 条目	说明
u[ser]::perms	文件所有者权限。
g[roup]::perms	文件组所有者权限。
o[ther]::perms	文件所有者或文件组所有者成员以外的其他用户的权限。

ACL 条目	说明
<code>m[ask]:perms</code>	ACL 掩码。掩码条目指示允许供用户（所有者除外）和组使用的最大权限。可以通过掩码快速更改所有用户和组的权限。
<code>u[ser]:uid:perms</code>	特定用户的权限。对于 <code>uid</code> ，可以指定用户名或数字 UID。
<code>g[roup]:gid:perms</code>	特定组的权限。对于 <code>gid</code> ，可以指定组名或数字 GID。
<code>d[efault]:u[ser]::perms</code>	缺省文件所有者权限。
<code>d[efault]:g[roup]::perms</code>	缺省文件组所有者权限。
<code>d[efault]:o[ther]:perms</code>	文件所有者或文件组所有者成员以外的其他用户的缺省权限。
<code>d[efault]:m[ask]:perms</code>	缺省 ACL 掩码。
<code>d[efault]:u[ser]:uid:perms</code>	特定用户的缺省权限。对于 <code>uid</code> ，可以指定用户名或数字 UID。
<code>d[efault]:g[roup]:gid:perms</code>	特定组的缺省权限。对于 <code>gid</code> ，可以指定组名或数字 GID。

对于 `-d` 选项，`acl_entries` 是一个或多个以逗号分隔的没有权限的 ACL 条目。请注意，无法删除文件所有者、文件组所有者、ACL 掩码和其他用户的条目。

选项

这些选项的含义如下所示：

- `-d acl_entries` 从文件中删除一个或多个条目。无法从 ACL 中删除文件所有者、文件组所有者和其他用户的条目。请注意，删除条目不一定与从条目中删除所有权限具有相同的效果。
- `-f acl_file` 使用名为 `acl_file` 的文件中包含的 ACL 条目设置文件的 ACL。对指定条目所具有的约束与 `-s` 选项相同。不要求条目在文件中采用任何特定顺序。此外，如果为 `acl_file` 指定了短划线 (-)，则会使用标准输入来设置文件的 ACL。

`acl_file` 中的字符 `#` 可用于指示注释。以 `#` 开头的所有字符（直至行尾）都将被忽略。请注意，如果 `acl_file` 已作为 `getfacl(1)` 命令的输出而创建，则会忽略 `#` 之后的任何有效权限。
- `-m acl_entries` 将一个或多个新 ACL 条目添加到文件，和/或修改文件中的一个或多个现有 ACL 条目。如果对于指定的 `uid` 或 `gid` 已经存在一个条目，则指定的权限会替换当前权限。如果对于指定的 `uid` 或 `gid` 不存在条目，则会创建一个条目。使用 `-m` 选项修改缺省 ACL 时，第一次必须指定完整的缺省 ACL（用户、组、其他、掩码和任何附加条目）。

- r 重新计算 ACL 掩码条目的权限。在 ACL 掩码条目中指定的权限将被忽略，并由对 ACL 中所有其他用户、文件组所有者和其他组条目授予访问权限所需的最大权限替换。其他用户、文件组所有者和其他组条目中的权限保持不变。
- s *acl_entries* 设置文件的 ACL。所有旧 ACL 条目将被删除，并替换为新指定的 ACL。相应条目不需要采用任何特定顺序。在应用于文件之前，将由该命令对其进行排序。

必需的条目：

- 只为文件所有者指定一个 `user` 条目。
- 只为文件组所有者指定一个 `group` 条目。
- 只指定一个 `other` 条目。

如果存在其他用户和组条目：

- 只为指示用户（所有者除外）和组允许使用的最大权限的 ACL 掩码指定一个 `mask` 条目。
- 不得存在 `uid` 相同的重复 `user` 条目。
- 不得存在 `gid` 相同的重复 `group` 条目。

如果 *file* 是一个目录，则可以指定以下缺省 ACL 条目：

- 只为文件所有者指定一个 `default user` 条目。
- 只为文件组所有者指定一个 `default group` 条目。
- 只为 ACL 掩码指定一个 `default mask` 条目。
- 只指定一个 `default other` 条目。

可以指定其他 `default user` 条目和其他 `default group` 条目，但不能指定 `uid` 相同的重复其他 `default user` 条目，或 `gid` 相同的重复 `default group` 条目。

示例

示例1 只添加读取权限

以下示例将一个 ACL 条目添加到文件 `abc`，这将只为用户 `shea` 授予读取权限。

```
setfacl -m user:shea:r— abc
```

示例2 替换文件的整个 ACL

以下示例替换文件 `abc` 的整个 ACL，这将授予 `shea` 读取访问权限，授予文件所有者所有访问权限，授予文件组所有者只读访问权限，授予 ACL 掩码只读访问权限，而不授予其他用户任何访问权限。

```
setfacl -s user:shea:rx, user::rx, group::rw-, mask:r--, other:--- abc
```

请注意，执行此命令后，文件权限位是 `rxr-----`。虽然为文件组所有者设置了读取/写入权限，但 ACL 掩码条目将其限制为只具有读取权限。该掩码条目还指定可用于所有其他用户和组 ACL 条目的最大权限。同样，虽然为用户 `shea` 设置了所有访问

示例 2 替换文件的整个 ACL (续)

权限，但掩码会将其限制为只具有读取权限。通过 ACL 掩码条目，可以快速限制或开放对 ACL 中所有用户和组条目的访问权限。例如，通过将掩码条目更改为读取/写入，文件组所有者和用户 shea 均将被授予读取/写入访问权限。

示例 3 对两个文件设置相同的 ACL

以下示例对文件 abc 设置与文件 xyz 相同的 ACL。

```
getfacl xyz | setfacl -f - abc
```

文件

```
/etc/passwd  口令文件
```

```
/etc/group    组文件
```

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[chmod\(1\)](#)、[getfacl\(1\)](#)、[umask\(1\)](#)、[aclcheck\(3SEC\)](#)、[aclsort\(3SEC\)](#)、[group\(4\)](#)、[passwd\(4\)](#)、[at](#)

引用名 setlabel – 更改文件的有效敏感标签

用法概要 /usr/bin/setlabel newlabel filename...

描述 大多数情况下，setlabel 将文件移动到其标签与 newlabel 相对应的区域。调整旧的文件路径名，使其相对于新区域的根路径名。如果文件父目录的旧路径名在新区域中没有作为一个目录而存在，则不移动该文件。一旦移动，可能再也无法在当前区域中访问该文件。

但是，对于位于多级别 ZFS 文件系统上的文件和目录，setlabel 的行为方式不同。请参见 [zfs\(1M\)](#)。在这种情况下，不会移动文件，但是会就地重新设置标签，因为多级别文件系统支持每文件标签。

除非已指定 newlabel 和 filename，否则不设置任何标签。

标签由站点的安全管理员定义。系统始终以大写字母形式显示标签。用户可以大写字母和小写字母的任何组合形式输入标签。支持对标签的增量更改。

有关满足该命令所需的条件和执行该命令所需的特权的完整说明，请参见 [setlabel\(3TSOL\)](#)。

退出状态 setlabel 在以下值之一时退出：

- 0 成功完成。
- 1 用法错误。
- 2 获取、设置或转换标签时出错。

用法 在命令行中，给标签加上双引号，除非标签只有一个字。如果没有引号，则被空格分隔开的另一个字或字符会被解释为另一个参数。

```
% setlabel SECRET somefile
% setlabel "TOP SECRET" somefile
```

使用大写字母和小写字母的任何组合。可在标签中通过空白、制表符、逗号或斜杠 (/) 来分隔各项。不要使用任何其他标点。

```
% setlabel "ts a b" somefile
% setlabel "ts,a,b" somefile
% setlabel "ts/a b" somefile
% setlabel " TOP SECRET A B " somefile
```

示例 示例1 设置标签。

要将 somefile 的标签设置为 SECRET A：

```
example% setlabel "Secret a" somefile
```

示例2 启用区间。

可使用加号和减号修改现有标签。加号可以启用 *somefile* 的标签的指定区间。

```
example% setlabel +b somefile
```

示例3 禁用区间。

减号可禁用与分类相关联的区间。要禁用区间 A（位于 *somefile* 标签中）：

```
example% setlabel -A somefile
```

如果已对现有标签进行增量更改，且标签的第一个字符是连字符 (-)，则前面需要双连字符 (--).

要禁用区间 -A（位于 *somefile* 的标签中）：

```
example% setlabel -- -A somefile
```

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/trusted
接口稳定性	Committed（已确定）

另请参见

[zfs\(1M\)](#)、[setlabel\(3TSOL\)](#)、[label_encodings\(4\)](#)、[attributes\(5\)](#)

附注

仅当系统配置有 Trusted Extensions 时，本手册页中介绍的功能才可用。

这种设置标签实现对国防情报局 (Defense Intelligence Agency, DIA) 的强制访问控制 (Mandatory Access Control, MAC) 策略非常有意义。有关更多信息，请参见 [label_encodings\(4\)](#)。

- 引用名** setpgrp – 设置进程组 ID
- 用法概要** setpgrp *command* [*arg*]...
- 描述** 如果当前进程尚不是会话引导者，则 setpgrp 实用程序会将进程组 ID 和会话 ID 设置为当前的进程 ID，并且会执行 *command* 的 exec() 及其参数（如果存在）。
- 操作数** 支持下列操作数：
command 要调用的命令名称。
arg 用于 *command* 的选项或参数。
- 退出状态** 将返回以下退出值：
1 执行 setpgrp 实用程序或在 *command* 的 exec() 期间出错。
否则，退出状态会是 *command* 的状态。
- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [exec\(2\)](#)、[setpgrp\(2\)](#)、[attributes\(5\)](#)

引用名 sftp – secure file transfer program

用法概要

```
sftp [-lCv] [-B buffer_size] [-b batchfile] [-F ssh_config]
      [-o ssh_option] [-P sftp_server_path] [-R num_requests]
      [-S program] [-s subsystem | sftp_server] host
```

```
sftp [[user@]host[:file [file]]]
```

```
sftp [[user@]host[:dir[/]]]
```

```
sftp -b batchfile [user@]host
```

描述 The sftp utility is an interactive file transfer program with a user interface similar to ftp(1) that uses the ssh(1) command to create a secure connection to the server.

sftp implements the SSH File Transfer Protocol as defined in IETF draft-ietf-secsh-filexfer. There is no relationship between the protocol used by sftp and the FTP protocol (RFC 959) provided by ftp(1).

The first usage format causes sftp to connect to the specified host and enter an interactive mode. If a username was provided then sftp tries to log in as the specified user. If a directory is provided then sftp tries to change the current directory on the server to the specified directory before entering the interactive mode.

The second usage format retrieves the specified file from the server and copies it to the specified target file or directory on the client. If a username is specified sftp tries to log in as the specified user.

选项 The following options are supported:

- b *batchfile* Batch mode reads a series of commands from an input *batchfile* instead of stdin. Since it lacks user interaction, it should be used in conjunction with non-interactive authentication. A batchfile of - can be used to indicate standard input. sftp aborts if any of the following commands fail: get, put, rm, rename, ln, rm, mkdir, chdir, ls, lchdir, chmod, chown, chgrp, lpwd, and lmkdir. Termination on error can be suppressed on a command by command basis by prefixing the command with a - character (for example, -rm /tmp/blah*).
- B *buffer_size* Specifies the size of the buffer that sftp uses when transferring files. Larger buffers require fewer round trips at the cost of higher memory consumption. The default is 32768 bytes.
- C Enables compression, using the -C flag in ssh(1).
- F *ssh_config* Specifies an alternative per-user configuration file for ssh. This option is directly passed to ssh(1).
- o *ssh_option* Specifies an option to be directly passed to ssh(1).

<code>-P sftp_server path</code>	Executes the specified path as an <i>sftp-server</i> and uses a pipe, rather than an ssh connection, to communicate with it. This option can be useful in debugging the sftp client and server. When the <code>-P</code> is specified, the <code>-S</code> option is ignored.
<code>-R num_requests</code>	Specifies how many requests can be outstanding at any one time. Increasing this can slightly improve file transfer speed but increases memory usage. The default is 64 outstanding requests.
<code>-s subsystem sftp_server</code>	Specifies the SSH2 subsystem or the path for an sftp server on the remote host. A path is useful for using sftp over protocol version 1, or when the remote sshd does not have an sftp subsystem configured.
<code>-S ssh_program path</code>	Uses the specified program instead of <code>ssh(1)</code> to connect to the sftp server. When the <code>-P</code> option is specified, the <code>-S</code> option is ignored. The program must understand <code>ssh(1)</code> options.
<code>-v</code>	Raises logging level. This option is also passed to <code>ssh(1)</code> .
<code>-1</code>	Specifies the use of protocol version 1.

操作数

The following operands are supported:

`hostname | user@hostname` The name of the host to which sftp connects and logs into.

Interactive Commands

Once in interactive mode, sftp understands a set of commands similar to those of `ftp(1)`. Commands are case insensitive and path names can be enclosed in quotes if they contain spaces.

<code>bye</code>	Quits sftp.
<code>cd path</code>	Changes remote directory to <i>path</i> .
<code>chgrp grp path</code>	Changes group of file <i>path</i> to <i>grp</i> . <i>grp</i> must be a numeric GID.
<code>chmod mode path</code>	Changes permissions of file <i>path</i> to <i>mode</i> .
<code>chown own path</code>	Changes owner of file <i>path</i> to <i>own</i> . <i>own</i> must be a numeric UID.
<code>exit</code>	Quits sftp.
<code>get [flags] remote-path [local-path]</code>	Retrieves the <i>remote-path</i> and stores it on the local machine. If the local path name is not specified, it is specified the same name it has on the remote machine. If the <code>-P</code> flag is specified, then the file's full permission and access time are copied too.
<code>help</code>	Displays help text.

	Identical to the ? command.
<code>lcd path</code>	Changes local directory to <i>path</i> .
<code>lls [ls-options] [path]</code>	Displays local directory listing of either <i>path</i> or current directory if <i>path</i> is not specified.
<code>lmkdir path</code>	Creates local directory specified by <i>path</i> .
<code>ln oldpath newpath</code>	Creates a link from <i>oldpath</i> to <i>newpath</i> .
<code>lpwd</code>	Prints local working directory.
<code>ls [-laflnrSt] [path]</code>	Displays remote directory listing of either <i>path</i> or current directory if <i>path</i> is not specified. <i>path</i> can contain wildcards.
	The <code>ls</code> supports the following options:
	-a Lists files beginning with a dot (.).
	-f Does not sort the listing. The default sort order is lexicographical.
	-l Displays additional details including permissions and ownership information.
	-n Produces a long listing with user and group information presented numerically.
	-r Reverses the sort order of the listing.
	-S Sorts the listing by file size.
	-t Sorts the listing by last modification time.
	-1 Produces single column output.
<code>lumask umask</code>	Sets local umask to <i>umask</i> .
<code>mkdir path</code>	Creates remote directory specified by <i>path</i> .
<code>put [flags] local-path [local-path]</code>	Uploads <i>local-path</i> and stores it on the remote machine. If the remote path name is not specified, it is specified the same name it has on the local machine. If the <code>-P</code> flag is specified, then the file's full permission and access time are copied too.
<code>pwd</code>	Displays remote working directory.
<code>quit</code>	Quits <code>sftp</code> .
<code>rename oldpath newpath</code>	Renames remote file from <i>oldpath</i> to <i>newpath</i> .

<code>rm path</code>	Deletes remote file specified by <i>path</i> .
<code>rmdir path</code>	Removes remote directory specified by <i>path</i> .
<code>symlink oldpath newpath</code>	Creates a symbolic link from <i>oldpath</i> to <i>newpath</i> .
<code>version</code>	Displays the <code>sftp</code> protocol version.
<code># [comment]</code>	Include a comment. This is useful in batch files.
<code>! [command]</code>	If <i>command</i> is not specified, escapes to the local shell. If <i>command</i> is specified, executes <i>command</i> in the local shell.
<code>?</code>	Displays help text. Identical to the <code>help</code> command.

退出状态

The following exit values are returned:

- `0` Successful completion.
- `>0` An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/ssh
Interface Stability	Committed

另请参见

[ftp\(1\)](#), [scp\(1\)](#), [ssh\(1\)](#), [ssh-add\(1\)](#), [ssh-keygen\(1\)](#), [sshd\(1M\)](#), [attributes\(5\)](#)

引用名	sh, jsh – standard and job control shell and command interpreter
用法概要	<pre> /usr/sunos/bin/sh [-acefhiknrstuvx] [argument]... /usr/xpg4/bin/sh [± abCefhikmnrstuvx] [± o option]... [-c string] [arg]... /usr/sunos/bin/jsh [-acefhiknrstuvx] [argument]... </pre>
描述	<p>The <code>/usr/sunos/bin/sh</code> utility is a command programming language that executes commands read from a terminal or a file.</p> <p>The <code>/usr/xpg4/bin/sh</code> utility is a standards compliant shell. This utility provides all the functionality of ksh88(1), except in cases discussed in ksh88(1) where differences in behavior exist.</p> <p>The <code>jsh</code> utility is an interface to the shell that provides all of the functionality of <code>sh</code> and enables job control (see Job Control section below).</p> <p>注 – <code>/usr/bin/sh</code> and <code>/usr/bin/jsh</code> are links to ksh93. See ksh93(1).</p> <p>Arguments to the shell are listed in the Invocation section below.</p>
Definitions	<p>A <i>blank</i> is a tab or a space. A <i>name</i> is a sequence of ASCII letters, digits, or underscores, beginning with a letter or an underscore. A <i>parameter</i> is a name, a digit, or any of the characters <code>*</code>, <code>@</code>, <code>#</code>, <code>?</code>, <code>-</code>, <code>\$</code>, and <code>!</code>.</p>
用法	
Commands	<p>A <i>simple-command</i> is a sequence of non-blank <i>words</i> separated by <i>blanks</i>. The first <i>word</i> specifies the name of the command to be executed. Except as specified below, the remaining <i>words</i> are passed as arguments to the invoked command. The command name is passed as argument 0 (see exec(2)). The <i>value</i> of a <i>simple-command</i> is its exit status if it terminates normally, or (octal) <code>200+status</code> if it terminates abnormally. See signal.h(3HEAD) for a list of status values.</p> <p>A <i>pipeline</i> is a sequence of one or more <i>commands</i> separated by <code> </code>. The standard output of each <i>command</i> but the last is connected by a pipe(2) to the standard input of the next <i>command</i>. Each <i>command</i> is run as a separate process. The shell waits for the last <i>command</i> to terminate. The exit status of a <i>pipeline</i> is the exit status of the last command in the <i>pipeline</i>.</p> <p>A <i>list</i> is a sequence of one or more <i>pipelines</i> separated by <code>;</code>, <code>&</code>, <code>&&</code>, or <code> </code>, and optionally terminated by <code>;</code> or <code>&</code>. Of these four symbols, <code>;</code> and <code>&</code> have equal precedence, which is lower than that of <code>&&</code> and <code> </code>. The symbols <code>&&</code> and <code> </code> also have equal precedence. A semicolon (<code>;</code>) causes sequential execution of the preceding <i>pipeline</i>, that is, the shell waits for the <i>pipeline</i> to finish before executing any commands following the semicolon. An ampersand (<code>&</code>) causes asynchronous execution of the preceding pipeline, that is, the shell does <i>not</i> wait for that pipeline to finish. The symbol <code>&&</code> (<code> </code>) causes the <i>list</i> following it to be executed only if the preceding pipeline returns a zero (non-zero) exit status. An arbitrary number of newlines can appear in a <i>list</i>, instead of semicolons, to delimit commands.</p>

A *command* is either a *simple-command* or one of the following. Unless otherwise stated, the value returned by a command is that of the last *simple-command* executed in the command.

`for name [in word . . .] do list done` Each time a `for` command is executed, *name* is set to the next *word* taken from the `in word` list. If `in word . . .` is omitted, then the `for` command executes the `do list` once for each positional parameter that is set (see Parameter Substitution section below). Execution ends when there are no more words in the list.

`case word in [pattern [| pattern]) list ; ;] . . . esac`

A `case` command executes the *list* associated with the first *pattern* that matches *word*. The form of the patterns is the same as that used for file-name generation (see File Name Generation section), except that a slash, a leading dot, or a dot immediately following a slash need not be matched explicitly.

`if list ; then list elif list ; then list ;] . . . [else list ;] fi`

The *list* following `if` is executed and, if it returns a zero exit status, the *list* following the first `then` is executed. Otherwise, the *list* following `elif` is executed and, if its value is zero, the *list* following the next `then` is executed. Failing that, the `else list` is executed. If no `else list` or `then list` is executed, then the `if` command returns a zero exit status.

`while list do list done` A `while` command repeatedly executes the `while list` and, if the exit status of the last command in the list is zero, executes the `do list`; otherwise the loop terminates. If no commands in the `do list` are executed, then the `while` command returns a zero exit status; `until` can be used in place of `while` to negate the loop termination test.

`(list)` Execute *list* in a sub-shell.

`{ list ; }` *list* is executed in the current (that is, parent) shell. The `{` must be followed by a space.

`name () { list ; }` Define a function which is referenced by *name*. The body of the function is the *list* of commands between `{` and `}`. The `{` must be followed by a space. Execution of functions is described below (see Execution section). The `{` and `}` are unnecessary if the body of the function is a *command* as defined above, under Commands.

The following words are only recognized as the first word of a command and when not quoted:

`if then else elif fi case esac for while until do done { }`

Comments Lines

A word beginning with `#` causes that word and all the following characters up to a newline to be ignored.

Command Substitution The shell reads commands from the string between two grave accents (“”) and the standard output from these commands can be used as all or part of a word. Trailing newlines from the standard output are removed.

No interpretation is done on the string before the string is read, except to remove backslashes (\) used to escape other characters. Backslashes can be used to escape a grave accent (‘) or another backslash (\) and are removed before the command string is read. Escaping grave accents allows nested command substitution. If the command substitution lies within a pair of double quotes (“ . . . ‘ . . . ‘ . . . ”), a backslash used to escape a double quote (\”) is removed. Otherwise, it is left intact.

If a backslash is used to escape a newline character (\newline), both the backslash and the newline are removed (see the later section on Quoting). In addition, backslashes used to escape dollar signs (\\$) are removed. Since no parameter substitution is done on the command string before it is read, inserting a backslash to escape a dollar sign has no effect. Backslashes that precede characters other than \, ‘, “, newline, and \$ are left intact when the command string is read.

Parameter Substitution The character \$ is used to introduce substitutable *parameters*. There are two types of parameters, positional and keyword. If *parameter* is a digit, it is a positional parameter. Positional parameters can be assigned values by set. Keyword parameters (also known as variables) can be assigned values by writing:

```
name=value [ name=value ] . . .
```

Pattern-matching is not performed on *value*. There cannot be a function and a variable with the same *name*.

<code>\${parameter}</code>	The value, if any, of the parameter is substituted. The braces are required only when <i>parameter</i> is followed by a letter, digit, or underscore that is not to be interpreted as part of its name. If <i>parameter</i> is * or @, all the positional parameters, starting with \$1, are substituted (separated by spaces). Parameter \$0 is set from argument zero when the shell is invoked.
<code>\${parameter:-word}</code>	Use Default Values. If <i>parameter</i> is unset or null, the expansion of <i>word</i> is substituted; otherwise, the value of <i>parameter</i> is substituted.
<code>\${parameter:=word}</code>	Assign Default Values. If <i>parameter</i> is unset or null, the expansion of <i>word</i> is assigned to <i>parameter</i> . In all cases, the final value of <i>parameter</i> is substituted. Only variables, not positional parameters or special parameters, can be assigned in this way.
<code>\${parameter:?word}</code>	If <i>parameter</i> is set and is non-null, substitute its value; otherwise, print <i>word</i> and exit from the shell. If <i>word</i> is omitted, the message “parameter null or not set” is printed.

`${parameter:+word}` If *parameter* is set and is non-null, substitute *word*; otherwise substitute nothing.

In the above, *word* is not evaluated unless it is to be used as the substituted string, so that, in the following example, `pwd` is executed only if `d` is not set or is null:

```
echo ${d:-'pwd'}
```

If the colon (`:`) is omitted from the above expressions, the shell only checks whether *parameter* is set or not.

The following parameters are automatically set by the shell.

- # The number of positional parameters in decimal.
- Flags supplied to the shell on invocation or by the `set` command.
- ? The decimal value returned by the last synchronously executed command.
- \$ The process number of this shell.
- ! The process number of the last background command invoked.

The following parameters are used by the shell. The parameters in this section are also referred to as environment variables.

HOME	The default argument (home directory) for the <code>cd</code> command, set to the user's login directory by <code>login(1)</code> from the password file (see <code>passwd(4)</code>).
PATH	The search path for commands (see Execution section below).
CDPATH	The search path for the <code>cd</code> command.
MAIL	If this parameter is set to the name of a mail file <i>and</i> the <code>MAILPATH</code> parameter is not set, the shell informs the user of the arrival of mail in the specified file.
MAILCHECK	This parameter specifies how often (in seconds) the shell checks for the arrival of mail in the files specified by the <code>MAILPATH</code> or <code>MAIL</code> parameters. The default value is <code>600</code> seconds (10 minutes). If set to <code>0</code> , the shell checks before each prompt.
MAILPATH	A colon-separated list of file names. If this parameter is set, the shell informs the user of the arrival of mail in any of the specified files. Each file name can be followed by <code>%</code> and a message that is e printed when the modification time changes. The default message is, you have mail.
PS1	Primary prompt string, by default “ \$ ”.
PS2	Secondary prompt string, by default “ > ”.
IFS	Internal field separators, normally space, tab, and newline (see Blank Interpretation section).

SHACCT	If this parameter is set to the name of a file writable by the user, the shell writes an accounting record in the file for each shell procedure executed.
SHELL	When the shell is invoked, it scans the environment (see <code>Environment</code> section below) for this name.

See `environ(5)` for descriptions of the following environment variables that affect the execution of `sh`: `LC_CTYPE` and `LC_MESSAGES`.

The shell gives default values to `PATH`, `PS1`, `PS2`, `MAILCHECK`, and `IFS`. Default values for `HOME` and `MAIL` are set by `login(1)`.

Blank Interpretation

After parameter and command substitution, the results of substitution are scanned for internal field separator characters (those found in `IFS`) and split into distinct arguments where such characters are found. Explicit null arguments ("" or '') are retained. Implicit null arguments (those resulting from *parameters* that have no values) are removed.

Input/Output Redirection

A command's input and output can be redirected using a special notation interpreted by the shell. The following can appear anywhere in a *simple-command* or can precede or follow a *command* and are *not* passed on as arguments to the invoked command. *Note*: Parameter and command substitution occurs before *word* or *digit* is used.

< <i>word</i>	Use file <i>word</i> as standard input (file descriptor 0).
> <i>word</i>	Use file <i>word</i> as standard output (file descriptor 1). If the file does not exist, it is created; otherwise, it is truncated to zero length.
>> <i>word</i>	Use file <i>word</i> as standard output. If the file exists, output is appended to it by first seeking to the EOF. Otherwise, the file is created.
< > <i>word</i>	Open file <i>word</i> for reading and writing as standard input.
<<[-] <i>word</i>	After parameter and command substitution is done on <i>word</i> , the shell input is read up to the first line that literally matches the resulting <i>word</i> , or to an EOF. If, however, the hyphen (-) is appended to <<: <ol style="list-style-type: none"> 1. leading tabs are stripped from <i>word</i> before the shell input is read (but after parameter and command substitution is done on <i>word</i>); 2. leading tabs are stripped from the shell input as it is read and before each line is compared with <i>word</i>; and 3. shell input is read up to the first line that literally matches the resulting <i>word</i>, or to an EOF.

If any character of *word* is quoted (see `Quoting` section later), no additional processing is done to the shell input. If no characters of *word* are quoted:

1. parameter and command substitution occurs;
2. (escaped) `\newLines` are removed; and

3. \ must be used to quote the characters \, \$, and '.

The resulting document becomes the standard input.

<&digit Use the file associated with file descriptor *digit* as standard input. Similarly for the standard output using >&digit.

<&- The standard input is closed. Similarly for the standard output using >&-.

If any of the above is preceded by a digit, the file descriptor which is associated with the file is that specified by the digit (instead of the default 0 or 1). For example:

```
... 2>&1
```

associates file descriptor 2 with the file currently associated with file descriptor 1.

The order in which redirections are specified is significant. The shell evaluates redirections left-to-right. For example:

```
... 1>xxx 2>&1
```

first associates file descriptor 1 with file *xxx*. It associates file descriptor 2 with the file associated with file descriptor 1 (that is, *xxx*). If the order of redirections were reversed, file descriptor 2 would be associated with the terminal (assuming file descriptor 1 had been) and file descriptor 1 would be associated with file *xxx*.

Using the terminology introduced on the first page, under *Commands*, if a *command* is composed of several *simple commands*, redirection is evaluated for the entire *command* before it is evaluated for each *simple command*. That is, the shell evaluates redirection for the entire *list*, then each *pipeline* within the *list*, then each *command* within each *pipeline*, then each *list* within each *command*.

If a command is followed by &, the default standard input for the command is the empty file, /dev/null. Otherwise, the environment for the execution of a command contains the file descriptors of the invoking shell as modified by input/output specifications.

File Name Generation Before a command is executed, each command *word* is scanned for the characters *, ?, and [. If one of these characters appears the word is regarded as a *pattern*. The word is replaced with alphabetically sorted file names that match the pattern. If no file name is found that matches the pattern, the word is left unchanged. The character . at the start of a file name or immediately following a /, as well as the character / itself, must be matched explicitly.

* Matches any string, including the null string.

? Matches any single character.

[...] Matches any one of the enclosed characters. A pair of characters separated by - matches any character lexically between the pair, inclusive. If the first character following the opening [is a !, any character not enclosed is matched.

Notice that all quoted characters (see below) must be matched explicitly in a filename.

Quoting

The following characters have a special meaning to the shell and cause termination of a word unless quoted:

```
; & ( ) | ^ < > newline space tab
```

A character can be *quoted* (that is, made to stand for itself) by preceding it with a backslash (\) or inserting it between a pair of quote marks (' ' or ""). During processing, the shell can quote certain characters to prevent them from taking on a special meaning. Backslashes used to quote a single character are removed from the word before the command is executed. The pair `\newline` is removed from a word before command and parameter substitution.

All characters enclosed between a pair of single quote marks (' '), except a single quote, are quoted by the shell. Backslash has no special meaning inside a pair of single quotes. A single quote can be quoted inside a pair of double quote marks (for example, " ' "), but a single quote can not be quoted inside a pair of single quotes.

Inside a pair of double quote marks (""), parameter and command substitution occurs and the shell quotes the results to avoid blank interpretation and file name generation. If `$*` is within a pair of double quotes, the positional parameters are substituted and quoted, separated by quoted spaces (`"$1 $2 ..."`). However, if `$@` is within a pair of double quotes, the positional parameters are substituted and quoted, separated by unquoted spaces (`"$1" "$2" ...`). \ quotes the characters \, ' , (comma), and \$. The pair `\newline` is removed before parameter and command substitution. If a backslash precedes characters other than \, ' , (comma), \$, and newline, then the backslash itself is quoted by the shell.

Prompting

When used interactively, the shell prompts with the value of `PS1` before reading a command. If at any time a newline is typed and further input is needed to complete a command, the secondary prompt (that is, the value of `PS2`) is issued.

Environment

The *environment* (see [environ\(5\)](#)) is a list of name-value pairs that is passed to an executed program in the same way as a normal argument list. The shell interacts with the environment in several ways. On invocation, the shell scans the environment and creates a parameter for each name found, giving it the corresponding value. If the user modifies the value of any of these parameters or creates new parameters, none of these affects the environment unless the `export` command is used to bind the shell's parameter to the environment (see also `set -a`). A parameter can be removed from the environment with the `unset` command. The environment seen by any executed command is thus composed of any unmodified name-value pairs originally inherited by the shell, minus any pairs removed by `unset`, plus any modifications or additions, all of which must be noted in `export` commands.

The environment for any *simple-command* can be augmented by prefixing it with one or more assignments to parameters. Thus:

```
TERM=450 command
```

and

```
(export TERM; TERM=450;  command
```

are equivalent as far as the execution of *command* is concerned if *command* is not a Special Command. If *command* is a Special Command, then

```
TERM=450  command
```

modifies the TERM variable in the current shell.

If the -k flag is set, *all* keyword arguments are placed in the environment, even if they occur after the command name. The following example first prints a=b c and c:

```
echo a=b c
```

```
a=b c
```

```
set -k
```

```
echo a=b c
```

```
c
```

Signals

The INTERRUPT and QUIT signals for an invoked command are ignored if the command is followed by &. Otherwise, signals have the values inherited by the shell from its parent, with the exception of signal 11 (but see also the trap command below).

Execution

Each time a command is executed, the command substitution, parameter substitution, blank interpretation, input/output redirection, and filename generation listed above are carried out. If the command name matches the name of a defined function, the function is executed in the shell process (note how this differs from the execution of shell script files, which require a sub-shell for invocation). If the command name does not match the name of a defined function, but matches one of the Special Commands listed below, it is executed in the shell process.

The positional parameters \$1, \$2, . . . are set to the arguments of the function. If the command name matches neither a Special Command nor the name of a defined function, a new process is created and an attempt is made to execute the command via [exec\(2\)](#).

The shell parameter PATH defines the search path for the directory containing the command. Alternative directory names are separated by a colon (:). The path to sh is /usr/sunos/bin. The current directory is specified by a null path name, which can appear immediately after the equal sign, between two colon delimiters anywhere in the path list, or at the end of the path list. If the command name contains a / the search path is not used. Otherwise, each directory in the path is searched for an executable file. If the file has execute permission but is not an a.out file, it is assumed to be a file containing shell commands. A sub-shell is spawned to read it. A parenthesized command is also executed in a sub-shell.

The location in the search path where a command was found is remembered by the shell (to help avoid unnecessary *execs* later). If the command was found in a relative directory, its location must be re-determined whenever the current directory changes. The shell forgets all remembered locations whenever the PATH variable is changed or the hash -r command is executed (see below).

Special Commands	Input/output redirection is now permitted for these commands. File descriptor 1 is the default output location. When Job Control is enabled, additional Special Commands are added to the shell's environment (see Job Control section below).
:	No effect; the command does nothing. A zero exit code is returned.
. <i>filename</i>	Read and execute commands from <i>filename</i> and return. The search path specified by PATH is used to find the directory containing <i>filename</i> .
bg [% <i>jobid</i> . . .]	When Job Control is enabled, the bg command is added to the user's environment to manipulate jobs. Resumes the execution of a stopped job in the background. If % <i>jobid</i> is omitted the current job is assumed. (See Job Control section below for more detail.)
break [<i>n</i>]	Exit from the enclosing for or while loop, if any. If <i>n</i> is specified, break <i>n</i> levels.
cd [<i>argument</i>]	Change the current directory to <i>argument</i> . The shell parameter HOME is the default <i>argument</i> . The shell parameter CDPATH defines the search path for the directory containing <i>argument</i> . Alternative directory names are separated by a colon (:). The default path is <null> (specifying the current directory). <i>Note:</i> The current directory is specified by a null path name, which can appear immediately after the equal sign or between the colon delimiters anywhere else in the path list. If <i>argument</i> begins with a / the search

`chdir [dir]`

path is not used. Otherwise, each directory in the path is searched for *argument*.

`chdir` changes the shell's working directory to directory *dir*. If no argument is given, change to the home directory of the user. If *dir* is a relative pathname not found in the current directory, check for it in those directories listed in the `CDPATH` variable. If *dir* is the name of a shell variable whose value starts with a `/`, change to the directory named by that value.

`continue [n]`

Resume the next iteration of the enclosing `for` or `while` loop. If *n* is specified, resume at the *n*-th enclosing loop.

`echo [arguments ...]`

The words in *arguments* are written to the shell's standard output, separated by space characters. See [echo\(1\)](#) for fuller usage and description.

`eval [argument ...]`

The arguments are read as input to the shell and the resulting command(s) executed.

`exec [argument ...]`

The command specified by the arguments is executed in place of this shell without creating a new process. Input/output arguments can appear and, if no other arguments are given, cause the shell input/output to be modified.

`exit [n]`

Causes the calling shell or shell script to exit with the exit status specified by *n*. If *n* is omitted the exit status is that of the last command executed (an EOF also causes the shell to exit.)

`export [name ...]`

The given *names* are marked for automatic export to the *environment* of subsequently executed commands.

`fg [%jobid...]`

If no arguments are given, variable names that have been marked for export during the current shell's execution are listed. (Variable names exported from a parent shell are listed only if they have been exported again during the current shell's execution.) Function names are *not* exported.

When Job Control is enabled, the `fg` command is added to the user's environment to manipulate jobs. This command resumes the execution of a stopped job in the foreground and also moves an executing background job into the foreground. If `%jobid` is omitted, the current job is assumed. (See Job Control section below for more detail.)

`getopts`

Use in shell scripts to support command syntax standards (see [Intro\(1\)](#)). This command parses positional parameters and checks for legal options. See [getoptcvt\(1\)](#) for usage and description.

`hash [-r] [name...]`

For each *name*, the location in the search path of the command specified by *name* is determined and remembered by the shell. The `-r` option causes the shell to forget all remembered locations. If no arguments are given, information about remembered commands is presented. *Hits* is the number of times a command has been invoked by the shell process. *Cost* is a measure of the work required to locate a command in the search path. If a command is found in a “relative” directory in the search path, after changing to that directory, the stored location of that command is recalculated. Commands for which this are done are indicated

jobs [-p|-l] [%jobid ...]
jobs -x *command* [*arguments*]

by an asterisk (*) adjacent to the *hits* information. *Cost* is incremented when the recalculation is done.

Reports all jobs that are stopped or executing in the background. If %*jobid* is omitted, all jobs that are stopped or running in the background are reported. (See Job Control section below for more detail.)

kill [-sig] %job ...
kill -l

Sends either the TERM (terminate) signal or the specified signal to the specified jobs or processes. Signals are either given by number or by names (as given in [signal.h\(3HEAD\)](#) stripped of the prefix “SIG” with the exception that SIGCHD is named CHLD). If the signal being sent is TERM (terminate) or HUP (hangup), then the job or process is sent a CONT (continue) signal if it is stopped. The argument *job* can be the process id of a process that is not a member of one of the active jobs. See Job Control section below for a description of the format of *job*. In the second form, kill -l, the signal numbers and names are listed. (See [kill\(1\)](#)).

login [*argument* ...]

Equivalent to ‘exec login *argument*. . .’. See [login\(1\)](#) for usage and description.

newgrp [*argument*]

Equivalent to exec newgrp *argument*. See [newgrp\(1\)](#) for usage and description.

pwd

Print the current working directory. See [pwd\(1\)](#) for usage and description.

read *name* ...

One line is read from the standard input and, using the internal field separator, IFS (normally space or tab),

to delimit word boundaries, the first word is assigned to the first *name*, the second word to the second *name*, and so forth, with leftover words assigned to the last *name*. Lines can be continued using `\newLine`. Characters other than `\newLine` can be quoted by preceding them with a backslash. These backslashes are removed before words are assigned to *names*, and no interpretation is done on the character that follows the backslash. The return code is 0, unless an EOF is encountered.

`readonly [name...]`

The given *names* are marked `readonly` and the values of these *names* can not be changed by subsequent assignment. If no arguments are given, a list of all `readonly` names is printed.

`return [n]`

Causes a function to exit with the return value specified by *n*. If *n* is omitted, the return status is that of the last command executed.

`set [-aefhkntuvx [argument...]]`

- a Mark variables which are modified or created for export.
- e Exit immediately if a command exits with a non-zero exit status.
- f Disable file name generation.
- h Locate and remember function commands as functions are defined (function commands are normally located when the function is executed).
- k All keyword arguments are placed in the environment for a

command, not just those that precede the command name.

- n Read commands but do not execute them.
- t Exit after reading and executing one command.
- u Treat unset variables as an error when substituting.
- v Print shell input lines as they are read.
- x Print commands and their arguments as they are executed.
- Do not change any of the flags; useful in setting \$1 to –.

Using + rather than – causes these flags to be turned off. These flags can also be used upon invocation of the shell. The current set of flags can be found in \$-. The remaining arguments are positional parameters and are assigned, in order, to \$1, \$2, ... If no arguments are given, the values of all names are printed.

`shift [n]`

The positional parameters from \$*n*+1 ... are renamed \$1 ... If *n* is not given, it is assumed to be 1.

`stop pid ...`

Halt execution of the process number *pid*. (see [ps\(1\)](#)).

`suspend`

Stops the execution of the current shell (but not if it is the login shell).

`test`

Evaluate conditional expressions. See [test\(1\)](#) for usage and description.

`times`

Print the accumulated user and system times for processes run from the shell.

`trap [argument n [n2 . . .]]`

The command *argument* is to be read and executed when the shell receives numeric or symbolic signal(s) (*n*). (*Note: argument* is scanned once when the trap is set and once when the trap is taken.) Trap commands are executed in order of signal number or corresponding symbolic names. Any attempt to set a trap on a signal that was ignored on entry to the current shell is ineffective. An attempt to trap on signal 11 (memory fault) produces an error. If *argument* is absent, all trap(s) *n* are reset to their original values. If *argument* is the null string, this signal is ignored by the shell and by the commands it invokes. If *n* is 0, the command *argument* is executed on exit from the shell. The `trap` command with no arguments prints a list of commands associated with each signal number.

`type [name . . .]`

For each *name*, indicate how it would be interpreted if used as a command name.

`ulimit [[-HS] [-a | -cdfnstv]]`

`ulimit [[-HS] [-c | -d | -f | -n | -s | -t | -v]] limit`

`ulimit` prints or sets hard or soft resource limits. These limits are described in [getrlimit\(2\)](#).

If *limit* is not present, `ulimit` prints the specified limits. Any number of limits can be printed at one time. The `-a` option prints all limits.

If *limit* is present, `ulimit` sets the specified limit to *limit*. The string `unlimited` requests that the current limit, if any, be removed. Any user can set a soft limit to any value less than or equal to the hard limit. Any user can lower a hard limit. Only a user with

appropriate privileges can raise or remove a hard limit. See [getrlimit\(2\)](#).

The `-H` option specifies a hard limit. The `-S` option specifies a soft limit. If neither option is specified, `ulimit` sets both limits and print the soft limit.

The following options specify the resource whose limits are to be printed or set. If no option is specified, the file size limit is printed or set.

- `-c` maximum core file size (in 512-byte blocks)
- `-d` maximum size of data segment or heap (in kbytes)
- `-f` maximum file size (in 512-byte blocks)
- `-n` maximum file descriptor plus 1
- `-s` maximum size of stack segment (in kbytes)
- `-t` maximum CPU time (in seconds)
- `-v` maximum size of virtual memory (in kbytes)

Run the [sysdef\(1M\)](#) command to obtain the maximum possible limits for your system. The values reported are in hexadecimal, but can be translated into decimal numbers using the [bc\(1\)](#) utility. See [swap\(1M\)](#).)

As an example of `ulimit`, to limit the size of a core file dump to 0 Megabytes, type the following:

```
ulimit -c 0
```

<code>umask [<i>nnn</i>]</code>	The user file-creation mask is set to <i>nnn</i> (see <code>umask(1)</code>). If <i>nnn</i> is omitted, the current value of the mask is printed.
<code>unset [<i>name</i> . . .]</code>	For each <i>name</i> , remove the corresponding variable or function value. The variables <code>PATH</code> , <code>PS1</code> , <code>PS2</code> , <code>MAILCHECK</code> , and <code>IFS</code> cannot be unset.
<code>wait [<i>n</i>]</code>	Wait for your background process whose process id is <i>n</i> and report its termination status. If <i>n</i> is omitted, all your shell's currently active background processes are waited for and the return code is zero.

Invocation

If the shell is invoked through `exec(2)` and the first character of argument zero is `-`, commands are initially read from `/etc/profile` and from `$HOME/.profile`, if such files exist. Thereafter, commands are read as described below, which is also the case when the shell is invoked as `/usr/sunos/bin/sh`. The flags below are interpreted by the shell on invocation only. *Note:* Unless the `-c` or `-s` flag is specified, the first argument is assumed to be the name of a file containing commands, and the remaining arguments are passed as positional parameters to that command file:

- `-c string` If the `-c` flag is present commands are read from *string*.
- `-i` If the `-i` flag is present or if the shell input and output are attached to a terminal, this shell is *interactive*. In this case, `TERMINATE` is ignored (so that `kill 0` does not kill an interactive shell) and `INTERRUPT` is caught and ignored (so that `wait` is interruptible). In all cases, `QUIT` is ignored by the shell.
- `-p` If the `-p` flag is present, the shell does not set the effective user and group IDs to the real user and group IDs.
- `-r` If the `-r` flag is present the shell is a restricted shell (see `rsh(1M)`).
- `-s` If the `-s` flag is present or if no arguments remain, commands are read from the standard input. Any remaining arguments specify the positional parameters. Shell output (except for Special Commands) is written to file descriptor 2.

The remaining flags and arguments are described under the `set` command above.

Job Control (jsh)

When the shell is invoked as `jsh`, Job Control is enabled in addition to all of the functionality described previously for `sh`. Typically, Job Control is enabled for the interactive shell only. Non-interactive shells typically do not benefit from the added functionality of Job Control.

With Job Control enabled, every command or pipeline the user enters at the terminal is called a *job*. All jobs exist in one of the following states: foreground, background, or stopped. These terms are defined as follows:

1. A job in the foreground has read and write access to the controlling terminal.
2. A job in the background is denied read access and has conditional write access to the controlling terminal (see [stty\(1\)](#)).
3. A stopped job is a job that has been placed in a suspended state, usually as a result of a SIGTSTP signal (see [signal.h\(3HEAD\)](#)).

Every job that the shell starts is assigned a positive integer, called a *job number* which is tracked by the shell and is used as an identifier to indicate a specific job. Additionally, the shell keeps track of the *current* and *previous* jobs. The *current job* is the most recent job to be started or restarted. The *previous job* is the first non-current job.

The acceptable syntax for a Job Identifier is of the form:

%jobid

where *jobid* can be specified in any of the following formats:

- | | |
|--------------------|---|
| % or + | For the current job. |
| - | For the previous job. |
| ?< <i>string</i> > | Specify the job for which the command line uniquely contains <i>string</i> . |
| <i>n</i> | For job number <i>n</i> . |
| <i>pref</i> | Where <i>pref</i> is a unique prefix of the command name. For example, if the command <code>ls -l <i>name</i></code> were running in the background, it could be referred to as <code>%ls</code> . <i>pref</i> cannot contain blanks unless it is quoted. |

When Job Control is enabled, the following commands are added to the user's environment to manipulate jobs:

- | | |
|--|---|
| <code>bg [%jobid . . .]</code> | Resumes the execution of a stopped job in the background. If <i>%jobid</i> is omitted the current job is assumed. |
| <code>fg [%jobid . . .]</code> | Resumes the execution of a stopped job in the foreground, also moves an executing background job into the foreground. If <i>%jobid</i> is omitted the current job is assumed. |
| <code>jobs [-p -l] [%jobid . . .]</code> | |
| <code>jobs -x command</code>
<code>[arguments]</code> | Reports all jobs that are stopped or executing in the background. If <i>%jobid</i> is omitted, all jobs that are stopped or running in the background is reported. The following options modify/enhance the output of <code>jobs</code> : |

	-l	Report the process group ID and working directory of the jobs.
	-p	Report only the process group ID of the jobs.
	-x	Replace any <i>jobid</i> found in <i>command</i> or <i>arguments</i> with the corresponding process group ID, and then execute <i>command</i> passing it <i>arguments</i> .
	<code>kill [-signal] %jobid</code>	Builtin version of <code>kill</code> to provide the functionality of the <code>kill</code> command for processes identified with a <i>jobid</i> .
	<code>stop %jobid ...</code>	Stops the execution of a background job(s).
	<code>suspend</code>	Stops the execution of the current shell (but not if it is the login shell).
	<code>wait [%jobid ...]</code>	<code>wait</code> builtin accepts a job identifier. If <i>%jobid</i> is omitted <code>wait</code> behaves as described above under Special Commands.
Large File Behavior		See largefile(5) for the description of the behavior of <code>sh</code> and <code>jsh</code> when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).
退出状态		Errors detected by the shell, such as syntax errors, cause the shell to return a non-zero exit status. If the shell is being used non-interactively execution of the shell file is abandoned. Otherwise, the shell returns the exit status of the last command executed (see also the <code>exit</code> command above).
jsh Only		If the shell is invoked as <code>jsh</code> and an attempt is made to exit the shell while there are stopped jobs, the shell issues one warning: There are stopped jobs. This is the only message. If another exit attempt is made, and there are still stopped jobs they are sent a SIGHUP signal from the kernel and the shell is exited.
文件		<code>\$HOME/.profile</code> <code>/dev/null</code> <code>/etc/profile</code> <code>/tmp/sh*</code>
属性		See attributes(5) for descriptions of the following attributes:

/usr/sunos/bin/sh,
/usr/sunos/bin/jsh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

/usr/xpg4/bin/sh

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

另请参见

[Intro\(1\)](#), [bc\(1\)](#), [echo\(1\)](#), [getoptcvt\(1\)](#), [kill\(1\)](#), [ksh88\(1\)](#), [ksh93\(1\)](#), [login\(1\)](#), [newgrp\(1\)](#), [pfsh\(1\)](#), [pfexec\(1\)](#), [ps\(1\)](#), [pwd\(1\)](#), [set\(1\)](#), [shell_builtins\(1\)](#), [stty\(1\)](#), [test\(1\)](#), [umask\(1\)](#), [wait\(1\)](#), [rsh\(1M\)](#), [su\(1M\)](#), [swap\(1M\)](#), [sysdef\(1M\)](#), [dup\(2\)](#), [exec\(2\)](#), [fork\(2\)](#), [getrlimit\(2\)](#), [pipe\(2\)](#), [ulimit\(2\)](#), [setlocale\(3C\)](#), [signal.h\(3HEAD\)](#), [passwd\(4\)](#), [profile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [XPG4\(5\)](#)

警告

The use of setuid shell scripts is *strongly* discouraged.

附注

Words used for filenames in input/output redirection are not interpreted for filename generation (see File Name Generation section above). For example, `cat file1 >a*` creates a file named `a*`.

Because commands in pipelines are run as separate processes, variables set in a pipeline have no effect on the parent shell.

If the input or the output of a `while` or `until` loop is redirected, the commands in the loop are run in a sub-shell, and variables set or changed there have no effect on the parent process:

```
lastline=
while read line
do

    lastline=$line
done < /etc/passwd
echo "lastline=$lastline"      # lastline is empty!
```

In these cases, the input or output can be redirected by using `exec`, as in the following example:

```
# Save standard input (file descriptor 0) as file
# descriptor 3, and redirect standard input from the file
/etc/passwd:

exec 3<&0                # save standard input as fd 3
exec </etc/passwd       # redirect input from file

lastline=
```

```
while read line
do
    lastline=$line
done

exec 0<&3          # restore standard input
exec 3<&-         # close file descriptor 3
echo "$lastline" # lastline
```

If you get the error message, “cannot fork, too many processes”, try using the [wait\(1\)](#) command to clean up your background processes. If this doesn't help, the system process table is probably full or you have too many active foreground processes. There is a limit to the number of process ids associated with your login, and to the number the system can keep track of.

Only the last process in a pipeline can be waited for.

If a command is executed, and a command with the same name is installed in a directory in the search path before the directory where the original command was found, the shell continues to exec the original command. Use the [hash](#) command to correct this situation.

The Bourne shell has a limitation on the effective UID for a process. If this UID is less than 100 (and not equal to the real UID of the process), then the UID is reset to the real UID of the process.

Because the shell implements both foreground and background jobs in the same process group, they all receive the same signals, which can lead to unexpected behavior. It is, therefore, recommended that other job control shells be used, especially in an interactive environment.

When the shell executes a shell script that attempts to execute a non-existent command interpreter, the shell returns an erroneous diagnostic message that the shell script file does not exist.

引用名 shcomp – 编译 ksh shell 脚本

用法概要 shcomp [-nv] [*infile* [*outfile*]]

shcomp -D [*infile* [*outfile*]]

描述 如果未指定 -D 选项，shcomp 会采用 shell 脚本 *infile*，并且会创建二进制格式的文件 *outfile*，ksh 读取和执行该文件的效果与使用原始脚本的效果相同。

读取脚本时处理别名。如果别名定义的值需要变量扩展，该定义将不能正常工作。

选项 支持以下选项：

-D

--dictionary 生成需要放置在消息目录中以实现国际化的字符串列表。

使用该选项，会打印字符串前有 \$ 的所有双引号括起来的字符串，每行一个字符串型。字符串型 "\$foo" 会在输出中打印出 "foo"。这些消息需要转换为国际化的语言环境特定版本。

-n

--noexec 显示有关过时或尚未确认的结构警告信息。

-v

--verbose 读取时将来自 *infile* 的输入显示为标准错误。

操作数 支持下列操作数：

infile 指定包含要用作输入的 shell 脚本的文件名称。

如果省略 *infile*，将从标准输入读取 shell 脚本。

outfile 指定输出文件的名称。

如果省略 *outfile*，则这两种模式都会将其结果写入标准输出。

退出状态 将返回以下退出值：

0 成功完成。

>0 出现错误。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
CSI	Enabled（已启用）
接口稳定性	请参见下文。

命令行界面和系统变量是 "Committed"（已确定）。编译的 shell 代码格式是 "Private"（专用）。-D 选项的输出是 "Volatile"（可变）。

另请参见

[ksh\(1\)](#)、[attributes\(5\)](#)

引用名 shell_builtins, case, for, foreach, function, if, repeat, select, switch, until, while – shell 命令解释程序内置命令

描述 shell 命令解释程序 `cs(1)`、`ksh(1)`、`ksh88(1)` 和 `sh(1)` 具有特殊的内置命令。命令 `case`、`for`、`foreach`、`function`、`if`、`repeat`、`select`、`switch`、`until` 和 `while` 是可被 shell 识别的语法中的命令。这些命令在各自 shell 手册页中的 `Commands` 一节中进行了介绍。缺省情况下，在 `ksh(1)` 中，`fc`、`hash`、`stop`、`suspend`、`times` 和 `type` 为别名。

下表中列出的其余命令出于效率或在命令调用之间共享数据等原因而内置到 shell 中。这些命令在各自的手册页中进行了介绍。

命令	Shell
alarm	ksh
++**alias	cs(1)、ksh88(1)、ksh(1)
bg	cs(1)、ksh88(1)、ksh(1)、sh(1)
+*break	cs(1)、ksh88(1)、ksh(1)、sh(1)
builtin	ksh
case	cs(1)、ksh88(1)、ksh(1)、sh(1)
cd	cs(1)、ksh88(1)、ksh(1)、sh(1)
chdir	cs(1)、sh(1)
command	ksh
+*continue	cs(1)、ksh88(1)、ksh(1)、sh(1)
dirs	cs(1)
disown	ksh
echo	cs(1)、ksh88(1)、ksh(1)、sh(1)
enum	ksh
+*eval	cs(1)、ksh88(1)、ksh(1)、sh(1)
+*exec	cs(1)、ksh88(1)、ksh(1)、sh(1)
+*exit	cs(1)、ksh88(1)、ksh(1)、sh(1)
++**export	ksh88(1)、ksh(1)、sh(1)
false	ksh88(1)、ksh(1)
fc	ksh88(1)、ksh(1)

命令	Shell
fg	csH、ksh88、ksh、sh
for	ksh88、ksh、sh
foreach	csH
function	ksh88、ksh
getopts	ksh88、ksh、sh
glob	csH
goto	csH
hash (散列)	ksh88、ksh、sh
hashstat	csH
hist	ksh
history	csH
if	csH、ksh88、ksh、sh
jobs	csH、ksh88、ksh、sh
kill	csH、ksh88、ksh、sh
let	ksh88、ksh
limit	csH
login	csH、ksh88、ksh、sh
logout	csH
nice	csH
+*newgrp	ksh88、ksh、sh
nohup	csH
notify	csH
onintr	csH
popd	csH
print	ksh88、ksh
printf	ksh
pushd	csH
pwd	ksh88、ksh、sh

命令	Shell
read	ksh88、ksh、sh
++**readonly	ksh88、ksh、sh
rehash	csH
repeat	csH
+*return	ksh88、ksh、sh
select	ksh88、ksh
+set	csH、ksh88、ksh、sh
setenv	csH
*shift	csH、ksh88、ksh、sh
source (源)	csH
stop	csH、ksh88、ksh、sh
suspend	csH、ksh88、sh
switch	csH
test	ksh88、ksh、sh
time	csH
*times	ksh88、ksh、sh
*+trap	ksh88、ksh、sh
true	ksh88、ksh
type	ksh88、ksh、sh
++**typeset	ksh88、ksh
ulimit	ksh88、ksh、sh
umask	csH、ksh88、ksh、sh
+unalias	csH、ksh88、ksh
unhash	csH
unlimit	csH
+unset	csH、ksh88、ksh、sh
unsetenv	csH
until	ksh88、ksh、sh

命令	Shell
vmap	ksh
vpath	ksh
*wait	csh、ksh88、ksh、sh
whence	ksh88、ksh
while	csh、ksh88、ksh、sh

- Bourne Shell、sh、特殊命令** 允许对这些命令执行输入/输出重定向。文件描述符 1 是缺省输出位置。启用 "Job Control"（作业控制）后，将向 shell 的环境添加其他**特殊命令**。
- 除了这些内置保留命令字，sh 还使用：
- ： 无影响；命令不执行任何操作。返回零退出代码。
 - . 文件名称** 读取和执行 *filename* 中的命令并返回。PATH 指定的搜索路径用于查找包含 *filename* 的目录。
- Cshell、csh** 内置命令在 C shell 内执行。如果内置命令是流水线除最后一个组件之外的任意组件，则在子 shell 中执行该命令。除了这些内置保留命令字，csh 还使用：
- ： 空命令。此命令将被解释，但不执行任何操作。
- Korn Shell、ksh88、特殊命令** 允许输入/输出重定向。除非有指示，否则输出写在文件描述符 1 上，并且如果没有语法错误，退出状态为零。
- 前面有一个或两个 *（星号）的命令专门通过以下方式进行处理：
1. 命令完成时，该命令前面的变量赋值列表仍然有效。
 2. 在变量赋值后进行 I/O 重定向。
 3. 脚本中包含的错误会导致其中止。
 4. 如果字的后面是前面加有 ** 的命令且字为变量赋值的格式，则使用与变量赋值相同的规则扩展这些字。这表示在 = 符号之后执行波浪号替换，并且不执行字拆分和文件名生成。
- 除了这些内置保留命令字，ksh88 还使用：
- * : [*arg...*] 该命令仅扩展参数。
 - * .*file* [*arg...*] 读取完整的 *file*，然后执行命令。在当前 shell 环境中执行命令。PATH 指定的搜索路径用于查找包含 *file* 的目录。如果指定了任何参数 *arg*，则它们将成为位置参数。否则，将不会更改位置参数。退出状态是最后执行的命令的退出状态。循环终止测试。

Korn Shell、ksh、特殊命令

允许输入/输出重定向。除非有指示，否则输出写在文件描述符 1 上，并且如果没有语法错误，退出状态为零。

除了 `:`、`true`、`false`、`echo`、`newgrp` 和 `login`，所有内置命令均可用 `--` 来表示选项结束。它们还将选项 `--man` 解释为要显示有关标准错误的手册页的请求，将 `-?` 解释为打印有关标准错误的使用消息的请求。

前面有一个或两个 `+` 的命令专门通过以下方式进行处理：

1. 命令完成时，该命令前面的变量赋值列表仍然有效。
2. 在变量赋值后进行 I/O 重定向。
3. 脚本中包含的错误会导致其中止。
4. 它们不是有效的函数名称。
5. 如果字的后面是前面加有 `++` 的命令且字为变量赋值的格式，则使用与变量赋值相同的规则扩展这些字。这意味着将在 `=` 符号之后执行波浪号替换，并且不执行字段分割和文件名生成。

除了这些内置保留命令字，ksh 还使用：

`:[arg...]` 该命令仅扩展参数。

`.name [arg...]` 如果 *name* 是使用函数 *name* 保留字语法定义的函数，则在当前环境中执行该函数（如同使用 `name()` 语法定义了该函数一样）。否则，如果 *name* 表示一个文件，则会读取整个文件并在当前的 shell 环境中执行命令。`PATH` 指定的搜索路径用于查找包含该文件的目录。如果指定了任何参数 *arg*，则在处理 `.` 命令时，这些参数将成为位置参数，并且原始位置参数会在完成后恢复。否则，位置参数保持不变。退出状态是最后执行的命令的退出状态。

另请参见

[Intro\(1\)](#)、[alias\(1\)](#)、[break\(1\)](#)、[builtin\(1\)](#)、[cd\(1\)](#)、[chmod\(1\)](#)、[csh\(1\)](#)、[disown\(1\)](#)、[echo\(1\)](#)、[exec\(1\)](#)

引用名	shift – shell built-in function to traverse either a shell's argument list or a list of field-separated words
用法概要	
sh	shift [<i>n</i>]
csh	shift [<i>variable</i>]
ksh88	*shift [<i>n</i>]
ksh88	+shift [<i>n</i>]
描述	
sh	The positional parameters from $\$n+1$. . . are renamed $\$1$ If <i>n</i> is not specified, it is assumed to be 1.
csh	The components of <i>argv</i> , or <i>variable</i> , if supplied, are shifted to the left, discarding the first component. It is an error for the variable not to be set or to have a null value.
ksh88	<p>The positional parameters from $\\$n+1$ $\\$n+1$. . . are renamed $\\$1$. . . , default <i>n</i> is 1. The parameter <i>n</i> can be any arithmetic expression that evaluates to a non-negative number less than or equal to $\\$#$.</p> <p>On this manual page, ksh88(1) commands that are preceded by one or two * (asterisks) are treated specially in the following ways:</p> <ol style="list-style-type: none"> 1. Variable assignment lists preceding the command remain in effect when the command completes. 2. I/O redirections are processed after variable assignments. 3. Errors cause a script that contains them to abort. 4. Words, following a command preceded by ** that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.
ksh	<p>shift is a shell special built-in that shifts the positional parameters to the left by the number of places defined by <i>n</i>, or 1 if <i>n</i> is omitted. The number of positional parameters remaining is reduced by the number of places that are shifted.</p> <p>If <i>n</i> is specified, it is evaluated as an arithmetic expression to determine the number of places to shift. It is an error to shift more than the number of positional parameters or a negative number of places.</p> <p>The following exit values are returned by shift in ksh:</p> <p>0 Successful completion. The positional parameters were successfully shifted.</p>

>0 An error occurred.

On this manual page, [ksh\(1\)](#) commands that are preceded by one or two + are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. They are not valid function names.
5. Words, following a command preceded by ++ that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and field splitting and file name generation are not performed.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见

[csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#)

- 引用名** shutdown – close down the system at a given time
- 用法概要** /usr/ucb/shutdown [-fhknr] *time* [*warning-message*]. . .
- 描述** shutdown provides an automated procedure to notify users when the system is to be shut down. *time* specifies when shutdown will bring the system down; it may be the word *now* (indicating an immediate shutdown), or it may specify a future time in one of two formats: *+number* and *hour: min*. The first form brings the system down in *number* minutes, and the second brings the system down at the time of day indicated in 24-hour notation.
- At intervals that get closer as the apocalypse approaches, warning messages are displayed at terminals of all logged-in users, and of users who have remote mounts on that machine.
- At shutdown time a message is written to the system log daemon, [syslogd\(1M\)](#), containing the time of shutdown, the instigator of the shutdown, and the reason. Then a terminate signal is sent to *init*, which brings the system down to single-user mode.
- 选项** As an alternative to the above procedure, these options can be specified:
- f Arrange, in the manner of [fastboot\(1B\)](#), that when the system is rebooted, the file systems will not be checked.
 - h Execute [halt\(1M\)](#).
 - k Simulate shutdown of the system. Do not actually shut down the system.
 - n Prevent the normal [sync\(2\)](#) before stopping.
 - r Execute [reboot\(1M\)](#).
- 文件** /etc/rmtab remote mounted file system table
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [fastboot\(1B\)](#), [login\(1\)](#), [halt\(1M\)](#), [reboot\(1M\)](#), [syslogd\(1M\)](#), [sync\(2\)](#), [rmtab\(4\)](#), [attributes\(5\)](#)

附注 Only allows you to bring the system down between now and 23:59 if you use the absolute time for shutdown.

引用名

size – print section sizes in bytes of object files

用法概要

size [-f] [-F] [-n] [-o] [-V] [-x] *filename...*

描述

The `size` command produces segment or section size information in bytes for each loaded section in ELF object files. `size` prints out the size of the text, data, and bss (uninitialized data) segments (or sections) and their total.

`size` processes ELF object files entered on the command line. If an archive file is input to the `size` command, the information for each object file in the archive is displayed.

When calculating segment information, the `size` command prints out the total file size of the non-writable segments, the total file size of the writable segments, and the total memory size of the writable segments minus the total file size of the writable segments.

If it cannot calculate segment information, `size` calculates section information. When calculating section information, it prints out the total size of sections that are allocatable, non-writable, and not NOBITS, the total size of the sections that are allocatable, writable, and not NOBITS, and the total size of the writable sections of type NOBITS. NOBITS sections do not actually take up space in the *filename*.

If `size` cannot calculate either segment or section information, it prints an error message and stops processing the file.

选项

The following options are supported:

- f Prints out the size of each allocatable section, the name of the section, and the total of the section sizes. If there is no section data, `size` prints out an error message and stops processing the file.
- F Prints out the size of each loadable segment, the permission flags of the segment, then the total of the loadable segment sizes. If there is no segment data, `size` prints an error message and stops processing the file.
- n Prints out non-loadable segment or non-allocatable section sizes. If segment data exists, `size` prints out the memory size of each loadable segment or file size of each non-loadable segment, the permission flags, and the total size of the segments. If there is no segment data, `size` prints out, for each allocatable and non-allocatable section, the memory size, the section name, and the total size of the sections. If there is no segment or section data, `size` prints an error message and stops processing.
- o Prints numbers in octal, not decimal.
- V Prints the version information for the `size` command on the standard error output.
- x Prints numbers in hexadecimal, not decimal.

示例

The examples below are typical `size` output.

示例 1 Producing size information

```
example% size filename
2724 + 88 + 0 = 2812
```

示例 2 Producing allocatable section size information

```
example% size -f filename
26(.text) + 5(.init) + 5(.fini) = 36
```

示例 3 Producing loadable segment size information

```
example% size -F filename
2724(r-x) + 88(rwx) + 0(rwx) = 2812 ... (If statically linked)
```

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities

另请参见

[as\(1\)](#), [ld\(1\)](#), [ar.h\(3HEAD\)](#), [a.out\(4\)](#), [attributes\(5\)](#)

附注

Since the size of bss sections is not known until link-edit time, the `size` command will not give the true total size of pre-linked objects.

引用名	sleep – suspend execution for an interval
用法概要	<code>/usr/bin/sleep interval[d h m s]...</code>
描述	sleep suspends execution for at least the time in seconds specified by <i>seconds</i> or until a SIGALRM signal is received. The <i>seconds</i> operand can be specified as a floating point number but the actual granularity normally depends on the underlying system.
操作数	<i>interval</i> A floating-point number specifying the time for which to suspend execution. The floating-point number may be specified in all formats required by C99/XPG6, including constants such as <code>Inf</code> or <code>infinite</code> . One of four suffixes may optionally be specified, indicating the number specified is days (d), hours (h), minutes (m), or seconds (s). With no suffix, the interval is assumed to be seconds. If multiple intervals are specified they are summed together. Individual intervals may be negative but the sum must be greater than or equal to zero.

示例

示例 1 Suspending Command Execution

The following example executes a command after a certain amount of time:

```
example% (sleep 105; command)&
```

示例 2 Executing a Command Every So Often

The following example executes a command every so often:

```
example% while true
do
    command
    sleep 37
done
```

示例 3 Suspending Command Execution Forever

The following example suspends command execution forever or until a SIGALRM signal is received:

```
example% sleep Inf
```

示例 4 Suspending Command Execution for 0.5 Seconds

Suspending command execution for 0.5 seconds using an alternative floating-point representation for the value 0.5:

```
example% printf "%a\n" 0.5
0x1.00000000000000000000000000000000p-01
example% sleep 0x1.00000000000000000000000000000000p-01
```

示例 5 Suspending Execution for 23 Hours

The following example suspends execution for twenty three hours using a letter suffixes:

```
example% sleep 1d -1h
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `sleep`: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 The execution was successfully suspended for at least *time* seconds, or a SIGALRM signal was received (see NOTES).
- >0 An error has occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[wait\(1\)](#), [alarm\(2\)](#), [sleep\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注

If the `sleep` utility receives a SIGALRM signal, one of the following actions is taken:

- Terminate normally with a zero exit status.
- Effectively ignore the signal.

The `sleep` utility takes the standard action for all other signals.

引用名 soelim – resolve and eliminate .so requests from nroff or troff input

用法概要 soelim [*filename*]. . .

描述 soelim reads the specified files or the standard input and performs the textual inclusion implied by the [nroff\(1\)](#) directives of the form:

```
.so somefile
```

when they appear at the beginning of input lines.

This is useful as programs such as [tbl\(1\)](#) do not normally do this. It allows the placement of individual tables in separate files to be run as a part of a large document.

An argument consisting of – is taken to be a file name corresponding to the standard input.

Inclusion can be suppressed by using a single quote (') instead of a dot (.) that is,

```
' so /usr/share/lib/tmac/tmac.s
```

示例 示例1 Using the soelim Command

The following is an example of the soelim command:

```
example% soelim exum?.n | tbl | nroff -ms | col | lpr
```

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

另请参见 [more\(1\)](#), [nroff\(1\)](#), [tbl\(1\)](#), [attributes\(5\)](#)

引用名	sort – sort, merge, or sequence check text files
用法概要	<pre> /usr/bin/sort [-bcdfimMru] [-k <i>keydef</i>] [-o <i>output</i>] [-S <i>kmem</i>] [-t <i>char</i>] [-T <i>directory</i>] [+<i>pos1</i> [-<i>pos2</i>]] [<i>file</i>]... /usr/xpg4/bin/sort [-bcdfimMru] [-k <i>keydef</i>] [-o <i>output</i>] [-S <i>kmem</i>] [-t <i>char</i>] [-T <i>directory</i>] [-y [<i>kmem</i>]] [-z <i>recsz</i>] [+<i>pos1</i> [-<i>pos2</i>]] [<i>file</i>]... </pre>
描述	<p>The <code>sort</code> command sorts lines of all the named files together and writes the result on the standard output.</p> <p>Comparisons are based on one or more sort keys extracted from each line of input. By default, there is one sort key, the entire input line. Lines are ordered according to the collating sequence of the current locale.</p>
选项	The following options alter the default behavior:
<code>/usr/bin/sort</code>	<p><code>-c</code> Checks that the single input file is ordered as specified by the arguments and the collating sequence of the current locale. The exit code is set and no output is produced unless the file is out of sort.</p>
<code>/usr/xpg4/bin/sort</code>	<p><code>-c</code> Same as <code>/usr/bin/sort</code> except no output is produced under any circumstances.</p> <p><code>-y <i>kmem</i></code> (obsolete). This option was used to specify the amount of main memory initially used by <code>sort</code>. Its functionality is not appropriate for a virtual memory system; memory usage for <code>sort</code> is now specified using the <code>-S</code> option.</p> <p><code>-z <i>recsz</i></code> (obsolete). This option was used to prevent abnormal termination when lines longer than the system-dependent default buffer size are encountered. Because <code>sort</code> automatically allocates buffers large enough to hold the longest line, this option has no effect.</p>
<code>/usr/bin/sort</code> and <code>/usr/xpg4/bin/sort</code>	<p><code>-m</code> Merges only. The input files are assumed to be already sorted.</p> <p><code>-o <i>output</i></code> Specifies the name of an output file to be used instead of the standard output. This file can be the same as one of the input files.</p> <p><code>-S <i>kmem</i></code> Specifies the maximum amount of swap-based memory used for sorting, in kilobytes (the default unit). <i>kmem</i> can also be specified directly as a number of bytes (b), kilobytes (k), megabytes (m), gigabytes (g), or terabytes (t); or as a percentage (%) of the installed physical memory.</p> <p><code>-T <i>directory</i></code> Specifies the <i>directory</i> in which to place temporary files.</p> <p><code>-u</code> Unique: suppresses all but one in each set of lines having equal keys. If used with the <code>-c</code> option, checks that there are no lines with duplicate keys in addition to checking that the input file is sorted.</p>

Ordering Options The default sort order depends on the value of `LC_COLLATE`. If `LC_COLLATE` is set to `C`, sorting is in ASCII order. If `LC_COLLATE` is set to `en_US`, sorting is case insensitive except when the two strings are otherwise equal and one has an uppercase letter earlier than the other. Other locales have other sort orders.

The following options override the default ordering rules. When ordering options appear independent of any key field specifications, the requested field ordering rules are applied globally to all sort keys. When attached to a specific key (see `Sort Key Options`), the specified ordering options override all global ordering options for that key. In the obsolescent forms, if one or more of these options follows a `+pos1` option, it affects only the key field specified by that preceding option.

- d Dictionary order: only letters, digits, and blanks (spaces and tabs) are significant in comparisons.
- f Folds lower-case letters into upper case.
- i Ignores non-printable characters.
- M Compares as months. The first three non-blank characters of the field are folded to upper case and compared. For example, in English the sorting order is "JAN" < "FEB" < . . . < "DEC". Invalid fields compare low to "JAN". The `-M` option implies the `-b` option (see below).
- n Restricts the sort key to an initial numeric string, consisting of optional blank characters, optional minus sign, and zero or more digits with an optional radix character and thousands separators (as defined in the current locale), which is sorted by arithmetic value. An empty digit string is treated as zero. Leading zeros and signs on zeros do not affect ordering.
- r Reverses the sense of comparisons.

Field Separator Options The treatment of field separators can be altered using the following options:

- b Ignores leading blank characters when determining the starting and ending positions of a restricted sort key. If the `-b` option is specified before the first sort key option, it is applied to all sort key options. Otherwise, the `-b` option can be attached independently to each `-k field_start,field_end`, or `+pos1` or `-pos2` option-argument (see below).
- t *char* Use *char* as the field separator character. *char* is not considered to be part of a field (although it can be included in a sort key). Each occurrence of *char* is significant (for example, `<char><char>` delimits an empty field). If `-t` is not specified, blank characters are used as default field separators; each maximal non-empty sequence of blank characters that follows a non-blank character is a field separator.

Sort Key Options Sort keys can be specified using the options:

-k *keydef*

The *keydef* argument is a restricted sort key field definition. The format of this definition is:

```
-k field_start [type] [,field_end [type] ]
```

where:

field_start and *field_end*

define a key field restricted to a portion of the line.

type

is a modifier from the list of characters `bdfiMnr`. The `b` modifier behaves like the `-b` option, but applies only to the *field_start* or *field_end* to which it is attached and characters within a field are counted from the first non-blank character in the field. (This applies separately to *first_character* and *last_character*.) The other modifiers behave like the corresponding options, but apply only to the key field to which they are attached. They have this effect if specified with *field_start*, *field_end* or both. If any modifier is attached to a *field_start* or to a *field_end*, no option applies to either.

When there are multiple key fields, later keys are compared only after all earlier keys compare equal. Except when the `-u` option is specified, lines that otherwise compare equal are ordered as if none of the options `-d`, `-f`, `-i`, `-n` or `-k` were present (but with `-r` still in effect, if it was specified) and with all bytes in the lines significant to the comparison.

The notation:

```
-k field_start[type][,field_end[type]]
```

defines a key field that begins at *field_start* and ends at *field_end* inclusive, unless *field_start* falls beyond the end of the line or after *field_end*, in which case the key field is empty. A missing *field_end* means the last character of the line.

A field comprises a maximal sequence of non-separating characters and, in the absence of option `-t`, any preceding field separator.

The *field_start* portion of the *keydef* option-argument has the form:

```
field_number[.first_character]
```

Fields and characters within fields are numbered starting with 1. *field_number* and *first_character*, interpreted as positive decimal integers, specify the first character to be used as part of a sort key. If *first_character* is omitted, it refers to the first character of the field.

The *field_end* portion of the *keydef* option-argument has the form:

```
field_number[.last_character]
```

The *field_number* is as described above for *field_start*. *last_character*, interpreted as a non-negative decimal integer, specifies the last character to be used as part of the sort key. If *last_character* evaluates to zero or *.last_character* is omitted, it refers to the last character of the field specified by *field_number*.

If the *-b* option or *b* type modifier is in effect, characters within a field are counted from the first non-blank character in the field. (This applies separately to *first_character* and *last_character*.)

[*+pos1* [*-pos2*]] (obsolete). Provide functionality equivalent to the *-keydef* option.

pos1 and *pos2* each have the form *m.n* optionally followed by one or more of the flags *bdfiMnr*. A starting position specified by *+m.n* is interpreted to mean the *n*+1st character in the *m*+1st field. A missing *.n* means *.0*, indicating the first character of the *m*+1st field. If the *b* flag is in effect *n* is counted from the first non-blank in the *m*+1st field; *+m.0b* refers to the first non-blank character in the *m*+1st field.

A last position specified by *-m.n* is interpreted to mean the *n*th character (including separators) after the last character of the *m*th field. A missing *.n* means *.0*, indicating the last character of the *m*th field. If the *b* flag is in effect *n* is counted from the last leading blank in the *m*+1st field; *-m.1b* refers to the first non-blank in the *m*+1st field.

The fully specified *+pos1 -pos2* form with type modifiers *T* and *U*:

```
+w.xT -y.zU
```

is equivalent to:

```
undefined (z==0 & U contains b & -t is present)
-k w+1.x+1T,y.0U      (z==0 otherwise)
-k w+1.x+1T,y+1.zU   (z > 0)
```

Implementations support at least nine occurrences of the sort keys (the *-k* option and obsolescent *+pos1* and *-pos2*) which are significant in command line order. If no sort key is specified, a default sort key of the entire line is used.

操作数

The following operand is supported:

file A path name of a file to be sorted, merged or checked. If no *file* operands are specified, or if a *file* operand is *-*, the standard input is used.

用法

See [largefile\(5\)](#) for the description of the behavior of `sort` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

示例

In the following examples, first the preferred and then the obsolete way of specifying sort keys are given as an aid to understanding the relationship between the two forms.

示例 1 Sorting with the Second Field as a sort Key

Either of the following commands sorts the contents of `infile` with the second field as the sort key:

```
example% sort -k 2,2 infile
example% sort +1 -2 infile
```

示例 2 Sorting in Reverse Order

Either of the following commands sorts, in reverse order, the contents of `infile1` and `infile2`, placing the output in `outfile` and using the second character of the second field as the sort key (assuming that the first character of the second field is the field separator):

```
example% sort -r -o outfile -k 2.2,2.2 infile1 infile2
example% sort -r -o outfile +1.1 -1.2 infile1 infile2
```

示例 3 Sorting Using a Specified Character in One of the Files

Either of the following commands sorts the contents of `infile1` and `infile2` using the second non-blank character of the second field as the sort key:

```
example% sort -k 2.2b,2.2b infile1 infile2
example% sort +1.1b -1.2b infile1 infile2
```

示例 4 Sorting by Numeric User ID

Either of the following commands prints the `passwd(4)` file (user database) sorted by the numeric user ID (the third colon-separated field):

```
example% sort -t : -k 3,3n /etc/passwd
example% sort -t : +2 -3n /etc/passwd
```

示例 5 Printing Sorted Lines Excluding Lines that Duplicate a Field

Either of the following commands prints the lines of the already sorted file `infile`, suppressing all but one occurrence of lines having the same third field:

```
example% sort -um -k 3.1,3.0 infile
example% sort -um +2.0 -3.0 infile
```

示例 6 Sorting by Host IP Address

Either of the following commands prints the `hosts(4)` file (IPv4 hosts database), sorted by the numeric IP address (the first four numeric fields):

```
example$ sort -t . -k 1,1n -k 2,2n -k 3,3n -k 4,4n /etc/hosts
example$ sort -t . +0 -1n +1 -2n +2 -3n +3 -4n /etc/hosts
```

示例 6 Sorting by Host IP Address (续)

Since '.' is both the field delimiter and, in many locales, the decimal separator, failure to specify both ends of the field leads to results where the second field is interpreted as a fractional portion of the first, and so forth.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of sort: LANG, LC_ALL, LC_COLLATE, LC_MESSAGES, and NLSPATH.

LC_CTYPE Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single- versus multi-byte characters in arguments and input files) and the behavior of character classification for the -b, -d, -f, -i and -n options.

LC_NUMERIC Determine the locale for the definition of the radix character and thousands separator for the -n option.

退出状态

The following exit values are returned:

- 0 All input files were output successfully, or -c was specified and the input file was correctly sorted.
- 1 Under the -c option, the file was not ordered as specified, or if the -c and -u options were both specified, two input lines were found with equal keys.
- >1 An error occurred.

文件

/var/tmp/stm??? Temporary files

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/sort

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled

/usr/xpg4/bin/sort

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见	comm(1) , join(1) , uniq(1) , nl_langinfo(3C) , strftime(3C) , hosts(4) , passwd(4) , attributes(5) , environ(5) , largefile(5) , standards(5)
诊断	Comments and exits with non-zero status for various trouble conditions (for example, when input lines are too long), and for disorders discovered under the <code>-c</code> option.
附注	<p>When the last line of an input file is missing a new-line character, <code>sort</code> appends one, prints a warning message, and continues.</p> <p><code>sort</code> does not guarantee preservation of relative line ordering on equal keys.</p> <p>One can tune <code>sort</code> performance for a specific scenario using the <code>-S</code> option. However, one should note in particular that <code>sort</code> has greater knowledge of how to use a finite amount of memory for sorting than the virtual memory system. Thus, a <code>sort</code> invoked to request an extremely large amount of memory via the <code>-S</code> option could perform extremely poorly.</p> <p>As noted, certain of the field modifiers (such as <code>-M</code> and <code>-d</code>) cause the interpretation of input data to be done with reference to locale-specific settings. The results of this interpretation can be unexpected if one's expectations are not aligned with the conventions established by the locale. In the case of the month keys, <code>sort</code> does not attempt to compensate for approximate month abbreviations. The precise month abbreviations from nl_langinfo(3C) or strftime(3C) are the only ones recognized. For printable or dictionary order, if these concepts are not well-defined by the locale, an empty sort key might be the result, leading to the next key being the significant one for determining the appropriate ordering.</p>

引用名 sortbib – sort a bibliographic database

用法概要 sortbib [-s KEYS] *database*...

描述 sortbib sorts files of records containing refer key-letters by user-specified keys. Records may be separated by blank lines, or by '[' and '.' delimiters, but the two styles may not be mixed together. This program reads through each *database* and pulls out key fields, which are sorted separately. The sorted key fields contain the file pointer, byte offset, and length of corresponding records. These records are delivered using disk seeks and reads, so sortbib may not be used in a pipeline to read standard input.

The most common key-letters and their meanings are given below.

%A	Author's name
%B	Book containing article referenced
%C	City (place of publication)
%D	Date of publication
%E	Editor of book containing article referenced
%F	Footnote number or label (supplied by refer)
%G	Government order number
%H	Header commentary, printed before reference
%I	Issuer (publisher)
%J	Journal containing article
%K	Keywords to use in locating reference
%L	Label field used by -k option of refer
%M	Bell Labs Memorandum (undefined)
%N	Number within volume
%O	Other commentary, printed at end of reference
%P	Page number(s)
%Q	Corporate or Foreign Author (unreversed)
%R	Report, paper, or thesis (unpublished)
%S	Series title
%T	Title of article or book
%V	Volume number
%X	Abstract — used by roffbib, not by refer

%Y,Z Ignored by refer

By default, `sortbib` alphabetizes by the first %A and the %D fields, which contain the senior author and date.

`sortbib` sorts on the last word on the %A line, which is assumed to be the author's last name. A word in the final position, such as 'jr.' or 'ed.', will be ignored if the name beforehand ends with a comma. Authors with two-word last names or unusual constructions can be sorted correctly by using the `nr off` convention '\0' in place of a blank. A %Q field is considered to be the same as %A, except sorting begins with the first, not the last, word. `sortbib` sorts on the last word of the %D line, usually the year. It also ignores leading articles (like 'A' or 'The') when sorting by titles in the %T or %J fields; it will ignore articles of any modern European language. If a sort-significant field is absent from a record, `sortbib` places that record before other records containing that field.

No more than 16 databases may be sorted together at one time. Records longer than 4096 characters will be truncated.

选项

-s*KEYS* Specify new *KEYS*. For instance, -sATD will sort by author, title, and date, while -sA+D will sort by all authors, and date. Sort keys past the fourth are not meaningful.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools

另请参见

[addbib\(1\)](#), [indxbib\(1\)](#), [lookbib\(1\)](#), [refer\(1\)](#), [roffbib\(1\)](#), [attributes\(5\)](#)

已知问题

Records with missing author fields should probably be sorted by title.

引用名	sotruss – 跟踪共享库过程调用
用法概要	<code>/usr/bin/sotruss [-f] [-F <i>bindfromlist</i>] [-T <i>bindtolist</i>] [-o <i>outputfile</i>] executable [<i>executable arguments...</i>]</code>
描述	sotruss 执行指定的命令并对其执行的库调用生成跟踪记录。每行跟踪输出会报告在执行每个过程调用时动态目标文件之间出现的绑定。sotruss 通过 <i>Procedure Linkage Table</i> 跟踪动态目标文件之间发生的所有过程调用，所以将只跟踪通过 <i>Procedure Linkage Table</i> 绑定的那些过程调用。请参见 《链接程序和库指南》
选项	<p><code>-F <i>bindfromlist</i></code> 要跟踪的库的冒号分隔列表。仅会跟踪来自这些库的调用。缺省值为仅跟踪从主可执行对象发出的调用。</p> <p><code>-T <i>bindtolist</i></code> 要跟踪的库的冒号分隔列表。仅会跟踪到这些库的调用。缺省设置是跟踪所有调用。</p> <p><code>-o <i>outputfile</i></code> sotruss 输出会定向到 <i>outputfile</i>。如果该选项与 <code>-f</code> 选项结合，会将执行程序 <i>pid</i> 放置在文件名的结尾。缺省情况下，sotruss 输出置于 <code>stderr</code> 中。</p> <p><code>-f</code> 跟随 <code>fork()</code> 创建的所有子进程，并在每个子进程中打印 <code>truss</code> 输出。该选项还会导致在每个 <code>truss</code> 输出行中输出 <i>pid</i>。</p>

示例

示例1 sotruss 示例。

以下是显示跟踪简单 `ls` 命令的简单示例：

```
% sotruss ls | more
ls      ->   libc.so.1:*atexit(0xef7d7d1c, 0x23c00, 0x0)
ls      ->   libc.so.1:*atexit(0x1392c, 0xef7d7d1c, 0xef621bb0)
ls      ->   libc.so.1:*setlocale(0x6, 0x1396c, 0xef621ba8)
ls      ->   libc.so.1:*textdomain(0x13970, 0x1396c, 0xef621ba8)
ls      ->   libc.so.1:*time(0x0, 0xef61f6fc, 0xef621ba8)
ls      ->   libc.so.1:*isatty(0x1, 0xef61f6fc, 0x0)
ls      ->   libc.so.1:*getopt(0x1, 0xfffff8fc, 0x13980)
ls      ->   libc.so.1:*malloc(0x100, 0x0, 0x0)
ls      ->   libc.so.1:*malloc(0x9000, 0x0, 0x0)
ls      ->   libc.so.1:*lstat64(0x23ee8, 0xfffff7a0, 0x0)
...
ls      ->   libc.so.1:*printf(0x13a64, 0x26208, 0x23ef0)
ls      ->   libc.so.1:*printf(0x13a64, 0x26448, 0x23ef0)
ls      ->   libc.so.1:*exit(0x0, 0x24220, 0x2421c)
```

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	developer/base-developer-utilities

另请参见

[ld.so.1\(1\)](#)、[truss\(1\)](#)、[whocalls\(1\)](#)、[fork\(2\)](#)、[attributes\(5\)](#)

《链接程序和库指南》

引用名 spell, hashmake, spellin, hashcheck – report spelling errors

用法概要 spell [-bilvx] [+ *local_file*] [*file*] ...

/usr/lib/spell/hashmake

/usr/lib/spell/spellin *n*

/usr/lib/spell/hashcheck *spelling_list*

描述 The spell command collects words from the named files and looks them up in a spelling list. Words that neither occur among nor are derivable (by applying certain inflections, prefixes, or suffixes) from words in the spelling list are written to the standard output.

If there are no file arguments, words to check are collected from the standard input. spell ignores most troff(1), tbl(1), and eqn(1) constructs. Copies of all output words are accumulated in the history file (spellhist), and a stop list filters out misspellings (for example, their=thy-y+ier) that would otherwise pass.

By default, spell (like deroff(1)) follows chains of included files (.so and .nx troff(1) requests), unless the names of such included files begin with /usr/lib.

The standard spelling list is based on many sources, and while more haphazard than an ordinary dictionary, is also more effective in respect to proper names and popular technical words. Coverage of the specialized vocabularies of biology, medicine and chemistry is light.

Three programs help maintain and check the hash lists used by spell:

hashmake Reads a list of words from the standard input and writes the corresponding nine-digit hash code on the standard output.

spellin Reads *n* hash codes from the standard input and writes a compressed spelling list on the standard output.

hashcheck Reads a compressed *spelling_list* and recreates the nine-digit hash codes for all the words in it. It writes these codes on the standard output.

选项 The following options are supported:

-b Check British spelling. Besides preferring *centre*, *colour*, *programme*, *speciality*, *travelled*, and so forth, this option insists upon *-ise* in words like *standardise*.

-i Cause deroff(1) to ignore .so and .nx commands. If deroff(1) is not present on the system, then this option is ignored.

-l Follow the chains of *all* included files.

-v Print all words not literally in the spelling list, as well as plausible derivations from the words in the spelling list.

-x Print every plausible stem, one per line, with = preceding each word.

+local_file Specify a set of words that are correct spellings (in addition to `spell`'s own spelling list) for each job. *local_file* is the name of a user-provided file that contains a sorted list of words, one per line. Words found in *local_file* are removed from `spell`'s output. Use [sort\(1\)](#) to order *local_file* in ASCII collating sequence. If this ordering is not followed, some entries in *local_file* might be ignored.

操作数

The following operands are supported:

`file` A path name of a text file to check for spelling errors. If no files are named, words are collected from the standard input.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `spell`: `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

文件

`D_SPELL=/usr/lib/spell/hlist[ab]` hashed spelling lists, American & British

`S_SPELL=/usr/lib/spell/hstop` hashed stop list

`H_SPELL=$HOME/.spellhist` history file

`/usr/share/lib/dict/words` master dictionary

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/spelling-utilities

另请参见

[deroff\(1\)](#), [eqn\(1\)](#), [sort\(1\)](#), [tbl\(1\)](#), [troff\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注

`spell` works only on English words defined in the U.S. ASCII codeset.

已知问题

The spelling list's coverage is uneven. New installations might wish to monitor the output for several months to gather local additions.

British spelling was done by an American.

Misspelled words can be monitored by default. To do so, set the `H_SPELL` environment variable to the name of a file which is writable to the `spell` process. If `H_SPELL` is not set, `$HOME/.spellhist` is used as the history file. If no monitoring is desired, one can create the appropriate `spell` history file with write permission disabled.

引用名 split – split a file into pieces

用法概要 split [-linecount | -l *linecount*] [-a *suffixlength*]
[*file* [*name*]]
split [-b *n* | *nk* | *nm*] [-a *suffixlength*] [*file* [*name*]]

描述 The `split` utility reads *file* and writes it in *linecount*-line pieces into a set of output-files. The name of the first output-file is *name* with `aa` appended, and so on lexicographically, up to `zz` (a maximum of 676 files). The maximum length of *name* is 2 characters less than the maximum filename length allowed by the filesystem. See [statvfs\(2\)](#). If no output name is given, `x` is used as the default (output-files will be called `xaa`, `xab`, and so forth).

选项 The following options are supported:

`-linecount` | `-l linecount` Number of lines in each piece. Defaults to 1000 lines.

`-a suffixlength` Uses *suffixlength* letters to form the suffix portion of the filenames of the split file. If `-a` is not specified, the default suffix length is 2. If the sum of the *name* operand and the *suffixlength* option-argument would create a filename exceeding `NAME_MAX` bytes, an error will result; `split` will exit with a diagnostic message and no files will be created.

`-b n` Splits a file into pieces *n* bytes in size.

`-b nk` Splits a file into pieces *n**1024 bytes in size.

`-b nm` Splits a file into pieces *n**1 048 576 bytes in size.

操作数 The following operands are supported:

file The path name of the ordinary file to be split. If no input file is given or *file* is `-`, the standard input will be used.

name The prefix to be used for each of the files resulting from the `split` operation. If no *name* argument is given, `x` will be used as the prefix of the output files. The combined length of the basename of *prefix* and *suffixlength* cannot exceed `NAME_MAX` bytes. See `OPTIONS`.

用法 See [largefile\(5\)](#) for the description of the behavior of `split` when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `split`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态 The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[csplit\(1\)](#), [statvfs\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

引用名 srchtxt – display contents of, or search for a text string in, message data bases

用法概要 srchtxt [-s] [-l *locale*] [-m *msgfile* , ...] [*text*]

描述 The srchtxt utility is used to display all the text strings in message data bases, or to search for a text string in message data bases (see [mkmsgs\(1\)](#)). These data bases are files in the directory `/usr/lib/locale/locale/LC_MESSAGES` (see [setlocale\(3C\)](#)), unless a file name given with the `-m` option contains a `/`. The directory *locale* can be viewed as the name of the language in which the text strings are written. If the `-l` option is not specified, the files accessed will be determined by the value of the environment variable `LC_MESSAGES`. If `LC_MESSAGES` is not set, the files accessed will be determined by the value of the environment variable `LANG`. If `LANG` is not set, the files accessed will be in the directory `/usr/lib/locale//C/LC_MESSAGES`, which contains default strings.

If no *text* argument is present, then all the text strings in the files accessed will be displayed.

If the `-s` option is not specified, the displayed text is prefixed by message sequence numbers. The message sequence numbers are enclosed in angle brackets: `<msgfile:msgnum>`.

msgfile name of the file where the displayed text occurred

msgnum sequence number in *msgfile* where the displayed text occurred

This display is in the format used by [gettext\(1\)](#) and [gettext\(3C\)](#).

选项

- `-s` Suppress printing of the message sequence numbers of the messages being displayed.
- `-l locale` Access files in the directory `/usr/lib/locale/locale/LC_MESSAGES`. If `-m msgfile` is also supplied, `LOCALE` is ignored for *msgfiles* containing a `/`.
- `-m msgfile` Access files specified by one or more *msgfiles*. If *msgfile* contains a `/` character, then *msgfile* is interpreted as a pathname; otherwise, it will be assumed to be in the directory determined as described above. To specify more than one *msgfile*, separate the file names using commas.
- text* Search for the text string specified by *text* and display each one that matches. *text* can take the form of a regular expression; see [regex\(5\)](#).

示例 示例 1 Using srchtxt

If message files have been installed in a locale named `french` by using [mkmsgs\(1\)](#), then you could display the entire set of text strings in the `french` locale (`/usr/lib/locale/french/LC_MESSAGES/*`) by typing:

```
example% srchtxt -l french
```

示例2 Using srchtxt

If a set of error messages associated with the operating system have been installed in the file UX in the french locale (`/usr/lib/locale/french/LC_MESSAGES/UX`), then, using the value of the LANG environment variable to determine the locale to be searched, you could search that file in that locale for all error messages dealing with files by typing:

```
example% setenv LANG=french; export LANG
example% srchtxt -m UX "[Ff]ichier"
```

If `/usr/lib/locale/french/LC_MESSAGES/UX` contained the following strings:

```
Erreur E/S\n
Liste d'arguments trop longue\n
Fichier inexistant\n
Argument invalide\n
Trop de fichiers ouverts\n
Fichier trop long\n
Trop de liens\n
Argument hors du domaine\n
Identificateur supprim\n
Etreinte fatale\n
.\n
.\n
.
```

then the following strings would be displayed:

```
<UX:3>Fichier inexistant\n
<UX:5>Trop de fichiers ouverts\n
<UX:6>Fichier trop long\n
```

示例3 Using srchtxt

If a set of error messages associated with the operating system have been installed in the file UX and a set of error messages associated with the INGRESS data base product have been installed in the file `ingress`, both in the german locale, then you could search for the pattern `[Dd]atei` in both the files UX and `ingress` in the german locale by typing:

```
example% srchtxt -l german -m UX,ingress "[Dd]atei"
```

环境变量

See [environ\(5\)](#) for a description of the LC_CTYPE environment variable that affects the execution of `srchtxt`.

文件

<code>/usr/lib/locale/C/LC_MESSAGES/*</code>	default files created by mkmsgs(1)
<code>/usr/lib/locale/locale/LC_MESSAGES/*</code>	message files created by mkmsgs(1)

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale

另请参见

[exstr\(1\)](#), [gettxt\(1\)](#), [locale\(1\)](#), [mkmsgs\(1\)](#), [gettxt\(3C\)](#), [setlocale\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [locale\(5\)](#), [regex\(5\)](#)

诊断

The error messages produced by `srchtxt` are intended to be self-explanatory. They indicate an error in the command line or errors encountered while searching for a particular locale and/or message file.

引用名

ssh – secure shell client (remote login program)

用法概要

```
ssh [-l login_name] hostname | user@hostname [ command]

ssh [-afgknqstvxACNTX1246] [-b bind_address] [-m mac_spec]
    [-c cipher_spec] [-e escape_char] [-i identity_file]
    [-i PKCS#11-URI]
    [-l login_name] [-F configfile] [-o option] [-p port]
    [-L [bind_address:]port:host:hostport]
    [-R [bind_address:]port:host:hostport]
    [-D [bind_address:]port] hostname | user@hostname [ command]
```

描述

ssh (Secure Shell) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to replace rlogin and rsh, and to provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel.

This implementation of ssh supports both SSH protocol versions 1 and 2 simultaneously. Because of security weaknesses in the v1 protocol, only v2 should be run, if possible.

Support for v1 is provided to help sites with existing ssh v1 servers to transition to v2. v1 might not be supported in a future release.

ssh connects and logs into the specified hostname. The user must prove his or her identity to the remote machine using one of several methods depending on the protocol version used:

SSH Protocol Version 1

First, if the machine the user logs in from is listed in `/etc/hosts.equiv` or `/etc/shosts.equiv` on the remote machine, and the user names are the same on both sides, the user is immediately permitted to log in. Second, if `.rhosts` or `.shosts` exists in the user's home directory on the remote machine and contains a line containing the name of the client machine and the name of the user on that machine, the user is permitted to log in. This form of authentication alone is normally not allowed by the server because it is not secure.

The second (and primary) authentication method is the `rhosts` or `hosts.equiv` method combined with RSA-based host authentication. It means that if the login would be permitted by `$HOME/.rhosts`, `$HOME/.shosts`, `/etc/hosts.equiv`, or `/etc/shosts.equiv`, and if additionally the server can verify the client's host key (see `/etc/ssh_known_hosts` in the FILES section), only then is login permitted. This authentication method closes security holes due to IP spoofing, DNS spoofing, and routing spoofing.

Note to the administrator: `/etc/hosts.equiv`, `$HOME/.rhosts`, and the rlogin/rsh protocol in general, are inherently insecure and should be disabled if security is desired.

As a third authentication method, ssh supports RSA-based authentication. The scheme is based on public-key cryptography. There are cryptosystems where encryption and decryption are done using separate keys, and it is not possible to derive the decryption key from the encryption key. RSA is one such system. The idea is that each user creates a public/private key pair for authentication purposes. The server knows the public key, and only the user knows

the private key. The file `$HOME/.ssh/authorized_keys` lists the public keys that are permitted for logging in. When the user logs in, the `ssh` program tells the server which key pair it would like to use for authentication. The server checks if this key is permitted, and if so, sends the user (actually the `ssh` program running on behalf of the user) a challenge in the form of a random number, encrypted by the user's public key. The challenge can only be decrypted using the proper private key. The user's client then decrypts the challenge using the private key, proving that he or she knows the private key but without disclosing it to the server.

`ssh` implements the RSA authentication protocol automatically. The user creates his or her RSA key pair by running `ssh-keygen(1)`. This stores the private key in `$HOME/.ssh/identity` and the public key in `$HOME/.ssh/identity.pub` in the user's home directory. The user should then copy the `identity.pub` to `$HOME/.ssh/authorized_keys` in his or her home directory on the remote machine (the `authorized_keys` file corresponds to the conventional `$HOME/.rhosts` file, and has one key per line, though the lines can be very long). After this, the user can log in without giving the password. RSA authentication is much more secure than `rhosts` authentication.

The most convenient way to use RSA authentication can be with an authentication agent. See [ssh-agent\(1\)](#) for more information.

If other authentication methods fail, `ssh` prompts the user for a password. The password is sent to the remote host for checking. However, since all communications are encrypted, the password cannot be seen by someone listening on the network.

SSH Protocol Version 2 The SSH version 2 protocol supports multiple user authentication methods, some of which are similar to those available with the SSH protocol version 1. These authentication mechanisms are negotiated by the client and server, with the client trying methods in the order specified in the `PreferredAuthentications` client configuration option. The server decides when enough authentication methods have passed successfully so as to complete the authentication phase of the protocol.

When a user connects by using protocol version 2, similar authentication methods are available. Using the default values for `PreferredAuthentications`, the client tries to authenticate first by using the `hostbased` method. If this method fails, public key authentication is attempted. Finally, if this method fails, `keyboard-interactive` and `password` authentication are tried.

The public key method is similar to RSA authentication described in the previous section and allows the RSA or DSA algorithm to be used: The client uses his or her private key, `$HOME/.ssh/id_dsa` or `$HOME/.ssh/id_rsa`, to sign the session identifier and sends the result to the server. The server checks whether the matching public key is listed in `$HOME/.ssh/authorized_keys` and grants access if both the key is found and the signature is correct. The session identifier is derived from a shared Diffie-Hellman value and is only known to the client and the server.

If public key authentication fails or is not available, a password can be sent encrypted to the remote host for proving the user's identity, or an extended prompt/reply protocol can be engaged.

Additionally, `ssh` supports hostbased or challenge response authentication.

Protocol 2 provides additional mechanisms for confidentiality (the traffic is encrypted using 3DES, Blowfish, CAST128 or Arcfour) and integrity (`hmac-sha2-256`, `hmac-sha2-256-96`, `hmac-sha2-512`, `hmac-sha2-512-96`, `hmac-sha1`, and `hmac-md5`). Protocol 1 lacks a strong mechanism for ensuring the integrity of the connection.

Login Session and Remote Execution

When the user's identity has been accepted by the server, the server either executes the specified command, or logs into the machine and gives the user a normal shell on the remote machine. All communication with the remote command or shell is automatically encrypted.

If a pseudo-terminal has been allocated (normal login session), the user can use the escape characters noted below. If a pseudo-terminal has been allocated (normal login session), the user can disconnect with `~.`, and suspend `ssh` with `~^Z`. All forwarded connections can be listed with `~#`. If the session blocks waiting for forwarded X11 or TCP/IP connections to terminate, `ssh` can be backgrounded with `~&`, although this should not be used while the user shell is active, as it can cause the shell to hang. All available escapes can be listed with `~?`.

A single tilde character can be sent as `~~`, or by following the tilde with a character other than those described above. The escape character must always follow a newline to be interpreted as special. The escape character can be changed in configuration files or on the command line.

If no pseudo tty has been allocated, the session is transparent and can be used to reliably transfer binary data. On most systems, setting the escape character to "none" also makes the session transparent even if a tty is used.

The session terminates when the command or shell on the remote machine exits and all X11 and TCP/IP connections have been closed. The exit status of the remote program is returned as the exit status of `ssh`.

Escape Characters

When a pseudo-terminal has been requested, `ssh` supports a number of functions through the use of an escape character.

A single tilde character can be sent as `~~` or by following the tilde with a character other than those described below. The escape character must always follow a newline to be interpreted as special. The escape character can be changed in configuration files using the `EscapeChar` configuration directive or on the command line by the `-e` option.

The supported escapes, assuming the default `~`, are:

- `~.` Disconnect.
- `~^Z` Background `ssh`.
- `~#` List forwarded connections.

- ~& Background ssh at logout when waiting for forwarded connection / X11 sessions to terminate.
- ~? Display a list of escape characters.
- ~B Send a break to the remote system. Only useful for SSH protocol version 2 and if the peer supports it.
- ~C Open command line. Only useful for adding port forwardings using the -L and -R options).
- ~R Request rekeying of the connection. Only useful for SSH protocol version 2 and if the peer supports it.

X11 and TCP Forwarding

If the `ForwardX11` variable is set to “yes” (or, see the description of the `-X` and `-x` options described later) and the user is using X11 (the `DISPLAY` environment variable is set), the connection to the X11 display is automatically forwarded to the remote side in such a way that any X11 programs started from the shell (or command) goes through the encrypted channel, and the connection to the real X server is made from the local machine. The user should not manually set `DISPLAY`. Forwarding of X11 connections can be configured on the command line or in configuration files.

The `DISPLAY` value set by `ssh` points to the server machine, but with a display number greater than zero. This is normal behavior, because `ssh` creates a “proxy” X11 server on the server machine for forwarding the connections over the encrypted channel.

`ssh` also automatically sets up `Xauthority` data on the server machine. For this purpose, it generates a random authorization cookie, store it in `Xauthority` on the server, and verify that any forwarded connections carry this cookie and replace it by the real cookie when the connection is opened. The real authentication cookie is never sent to the server machine (and no cookies are sent in the plain).

If the `ForwardAgent` variable is set to “yes” (or, see the description of the `-A` and `-a` options described later) and the user is using an authentication agent, the connection to the agent is automatically forwarded to the remote side.

Forwarding of arbitrary TCP/IP connections over the secure channel can be specified either on the command line or in a configuration file. One possible application of TCP/IP forwarding is a secure connection to an electronic purse. Another possible application is firewall traversal.

Server Authentication

`ssh` automatically maintains and checks a database containing identifications for all hosts it has ever been used with. Host keys are stored in `$HOME/.ssh/known_hosts` in the user's home directory. Additionally, the file `/etc/ssh_known_hosts` is automatically checked for known hosts. The behavior of `ssh` with respect to unknown host keys is controlled by the `StrictHostKeyChecking` parameter. If a host's identification ever changes, `ssh` warns about this and disables password authentication to prevent a trojan horse from getting the user's

password. Another purpose of this mechanism is to prevent attacks by intermediaries which could otherwise be used to circumvent the encryption. The `StrictHostKeyChecking` option can be used to prevent logins to machines whose host key is not known or has changed.

However, when using key exchange protected by GSS-API, the server can advertise a host key. The client automatically adds this host key to its known hosts file, `$HOME/.ssh/known_hosts`, regardless of the setting of the `StrictHostKeyChecking` option, unless the advertised host key collides with an existing known hosts entry.

When the user's GSS-API credentials expire, the client continues to be able to rekey the session using the server's public host key to protect the key exchanges.

GSS-API User and Server Authentication

`ssh` uses the user's GSS-API credentials to authenticate the client to the server wherever possible, if `GssKeyEx` and/or `GssAuthentication` are set.

With `GssKeyEx`, one can have an SSHv2 server that has no host public keys, so that only `GssKeyEx` can be used. With such servers, rekeying fails if the client's credentials are expired.

GSS-API user authentication has the disadvantage that it does not obviate the need for SSH host keys, but its failure does not impact rekeying. `ssh` can try other authentication methods (such as public key, password, and so on) if GSS-API authentication fails.

Delegation of GSS-API credentials can be quite useful, but is not without danger. As with passwords, users should not delegate GSS credentials to untrusted servers, since a compromised server can use a user's delegated GSS credentials to impersonate the user.

GSS-API user authorization is covered in [gss_auth_rules\(5\)](#).

Rekeying can be used to redelegate credentials when `GssKeyEx` is “yes”. (See `~R` under `Escape Characters` above.)

Configure `ssh` with:

```
UseFIPS140 yes
```

...to run OpenSSL in FIPS-140 mode. Only SSH Protocol Version 2 is supported. SunSSH may still delegate cryptographic operations for user/host authentication to other parts of Solaris which may or may not be FIPS 140–certified. The default value of `UseOpenSSLEngine` option is `no`, and the setting of `UseOpenSSLEngine` to `yes` does not have an effect in FIPS mode. As a further requirement to run `ssh` in FIPS-140 mode, the client needs to generate the user's private key in PKCS#8 format with the `ssh-keygen -8` command.

For the case of `ssh` with FIPS-140 enabled, when logging into a non-FIPS-140 `sshd`, the supported and approved FIPS ciphers must be explicitly specified in the [sshd_config\(4\)](#), using “Ciphers” for this scenario.

选项

The following options are supported:

- 1 Forces ssh to try protocol version 1 only.
- 2 Forces ssh to try protocol version 2 only.
- 4 Forces ssh to use IPv4 addresses only.
- 6 Forces ssh to use IPv6 addresses only.
- a Disables forwarding of the authentication agent connection.
- A Enables forwarding of the authentication agent connection. This can also be specified on a per-host basis in a configuration file.

Agent forwarding should be enabled with caution. Users with the ability to bypass file permissions on the remote host (for the agent's UNIX-domain socket) can access the local agent through the forwarded connection. An attacker cannot obtain key material from the agent. However, the attacker can perform operations on the keys that enable the attacker to authenticate using the identities loaded into the agent.
- b *bind_address* Specifies the interface to transmit from on machines with multiple interfaces or aliased addresses.
- c *cipher_spec* Selects the cipher specification for encrypting the session.

For protocol version 1, *cipher_spec* is a single cipher. See the Cipher option in [ssh_config\(4\)](#) for more information.

For protocol version 2, *cipher_spec* is a comma-separated list of ciphers listed in order of preference. See the Ciphers option in [ssh_config\(4\)](#) for more information.
- C Requests compression of all data (including stdin, stdout, stderr, and data for forwarded X11 and TCP/IP connections). The compression algorithm is the same used by [gzip\(1\)](#). The [gzip](#) man page is available in the SUNW'sfman package. The “level” can be controlled by the CompressionLevel option (see [ssh_config\(4\)](#)). Compression is desirable on

-
- modem lines and other slow connections, but only slows down things on fast networks. The default value can be set on a host-by-host basis in the configuration files. See the `Compression` option in `ssh_config(4)`.
- `-D [bind_address:]port`
- Specifies a local dynamic application-level port forwarding. This works by allocating a socket to listen to port on the local side, optionally bound to the specified *bind_address*. Whenever a connection is made to this port, the connection is forwarded over the secure channel. The application protocol is then used to determine where to connect to from the remote machine. Currently, the SOCKS4 and SOCKS5 protocols are supported and `ssh` acts as a SOCKS server. Only a user with enough privileges can forward privileged ports. Dynamic port forwardings can also be specified in the configuration file.
- IPv6 addresses can be specified with an alternative syntax: `[bind_address/]port` or by enclosing the address in square brackets. By default, the local port is bound in accordance with the `GatewayPorts` setting. However, an explicit *bind_address* can be used to bind the connection to a specific address. The *bind_address* of `localhost` indicates that the listening port be bound for local use only, while an empty address or `*` indicates that the port should be available from all interfaces.
- `-e ch | ^ch | none`
- Sets the escape character for sessions with a pty (default: `~`). The escape character is only recognized at the beginning of a line. The escape character followed by a dot (`.`) closes the connection. If followed by `CTRL-z`, the escape character suspends the connection. If followed by itself, the escape character sends itself once. Setting the character to `none` disables any escapes and makes the session fully transparent.
- `-f`
- Requests `ssh` to go to background just before command execution. This is useful if `ssh` is going to ask for passwords or passphrases, but the user wants it in the background. This implies the `-n` option. The

- recommended way to start X11 programs at a remote site is with something like `ssh -f host xterm`.
- `-F configfile` Specifies an alternative per-user configuration file. If a configuration file is specified on the command line, the system-wide configuration file, `/etc/ssh_config`, is ignored. The default for the per-user configuration file is `$HOME/.ssh/config`.
- `-g` Allows remote hosts to connect to local forwarded ports.
- `-i identity_file` Selects a file from which the identity (private key) for RSA or DSA authentication is read. The default is `$HOME/.ssh/identity` for protocol version 1, and `$HOME/.ssh/id_rsa` and `$HOME/.ssh/id_dsa` for protocol version 2. Identity files can also be specified on a per-host basis in the configuration file. It is possible to have multiple `-i` options (and multiple identities specified in configuration files).
- `-I PKCS#11-URI` Works with a certificate and a private key stored in the PKCS#11 token, instead of an identify file. See the Using X.509 Certificates section in the [ssh\(1M\)](#) man page for details.
- `-l login_name` Specifies the user to log in as on the remote machine. This also can be specified on a per-host basis in the configuration file.
- `-L [bind_address:]port:host:hostport` Specifies that the specified port on the local (client) host is to be forwarded to the specified host and port on the remote side. This works by allocating a socket to listen to the port on the local side, optionally bound to the specified `bind_address`. Then, whenever a connection is made to this port, the connection is forwarded over the secure channel and a connection is made to host port `hostport` from the remote machine. Port forwardings can also be specified in the configuration file. Only a user with enough privileges can forward privileged ports. IPv6 addresses can be specified with an alternative syntax: `[bind_address/]port/host/hostport` or by enclosing the address in square brackets.
- By default, the local port is bound in accordance with the `GatewayPorts` setting. However, an explicit

	<p><i>bind_address</i> can be used to bind the connection to a specific address. The <i>bind_address</i> of <code>localhost</code> indicates that the listening port be bound for local use only, while an empty address or <code>*</code> indicates that the port should be available from all interfaces.</p>
-m <i>mac_spec</i>	<p>Additionally, for protocol version 2 a comma-separated list of MAC (message authentication code) algorithms can be specified in order of preference. See the <code>MACs</code> keyword for more information.</p>
-n	<p>Redirects <code>stdin</code> from <code>/dev/null</code> (actually, prevents reading from <code>stdin</code>). This must be used when <code>ssh</code> is run in the background. A common trick is to use this to run X11 programs on a remote machine. For example,</p> <pre>ssh -n shadows.cs.hut.fi emacs &</pre> <p>starts an <code>emacs</code> on <code>shadows.cs.hut.fi</code>, and the X11 connection is automatically forwarded over an encrypted channel. The <code>ssh</code> program is put in the background. This does not work if <code>ssh</code> needs to ask for a password or passphrase. See also the <code>-f</code> option.</p>
-N	<p>Does not execute a remote command. This is useful if you just want to forward ports (protocol version 2 only).</p>
-o <i>option</i>	<p>Can be used to give options in the format used in the configuration file. This is useful for specifying options for which there is no separate command-line flag. The option has the same format as a line in the configuration file.</p>
-p <i>port</i>	<p>Specifies the port to connect to on the remote host. This can be specified on a per-host basis in the configuration file.</p>
-P	<p>Obsoleted option. SSHv1 connections from privileged ports are not supported.</p>
-q	<p>Quiet mode. Causes all warning and diagnostic messages to be suppressed. Only fatal errors are displayed.</p>
-R [<i>bind_address</i> :] <i>port</i> : <i>host</i> : <i>hostport</i>	<p>Specifies that the specified port on the remote (server) host is to be forwarded to the specified host</p>

and port on the local side. This works by allocating a socket to listen to the port on the remote side. Then, whenever a connection is made to this port, the connection is forwarded over the secure channel and a connection is made to host port *hostport* from the local machine. Port forwardings can also be specified in the configuration file. Privileged ports can be forwarded only when logging in on the remote machine as a user with enough privileges.

IPv6 addresses can be specified by enclosing the address in square braces or using an alternative syntax: [*bind_address*/]*host*/*port*/*hostport*.

By default, the listening socket on the server is bound to the loopback interface only. This can be overridden by specifying a *bind_address*. An empty *bind_address*, or the address ***, indicates that the remote socket should listen on all interfaces. Specifying a remote *bind_address* only succeeds if the server's GatewayPorts option is enabled. See [sshd_config\(4\)](#).

- s Can be used to request invocation of a subsystem on the remote system. Subsystems are a feature of the SSH2 protocol which facilitate the use of SSH as a secure transport for other applications, for example, `sftp`. The subsystem is specified as the remote command.
- t Forces pseudo-tty allocation. This can be used to execute arbitrary screen-based programs on a remote machine, which can be very useful, for example, when implementing menu services. Multiple `-t` options force allocation, even if `ssh` has no local `tty`.
- T Disables pseudo-tty allocation (protocol version 2 only).
- v Verbose mode. Causes `ssh` to print debugging messages about its progress. This is helpful in debugging connection, authentication, and configuration problems. Multiple `-v` options increase the verbosity. Maximum is 3.
- x Disables X11 forwarding.

-X Enables X11 forwarding. This can also be specified on a per-host basis in a configuration file.

X11 forwarding should be enabled with caution. Users with the ability to bypass file permissions on the remote host (for the user's X authorization database) can access the local X11 display through the forwarded connection. An attacker can then be able to perform activities such as keystroke monitoring.

For this reason, X11 forwarding might be subjected to X11 SECURITY extension restrictions. Refer to the `ForwardX11Trusted` directive in `ssh_config(4)` for more information.

If X11 forwarding is enabled, remote X11 clients is trusted by default. This means that they have full access to the original X11 display.

环境变量

ssh normally sets the following environment variables:

DISPLAY

The DISPLAY variable must be set for X11 display forwarding to work.

SSH_ASKPASS

If ssh needs a passphrase, it reads the passphrase from the current terminal if it was run from a terminal. If ssh does not have a terminal associated with it but DISPLAY and SSH_ASKPASS are set, it executes the program specified by SSH_ASKPASS and opens an X11 window to read the passphrase. This is particularly useful when calling ssh from a .Xsession or related script. On some machines it might be necessary to redirect the input from `/dev/null` to make this work. The system is shipped with `/usr/lib/ssh/ssh-askpass` which is the default value for SSH_ASKPASS

SSH_AUTH_SOCK

Indicates the path of a unix-domain socket used to communicate with the agent.

SSH_LANGS A comma-separated list of IETF language tags (see RFC3066) indicating the languages that the user can read and write. Used for negotiation of the locale on the server.

LANG, LC_ALL, LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_MONETARY, LC_NUMERIC, LC_TIME The values of these environment variables can be set in remote sessions according to the locale settings on the client side and availability of support for those locales on the server side. Environment Variable Passing (see RFC 4254) is used for passing them over to the server side.

See the ENVIRONMENT VARIABLES section in the [ssh\(1M\)](#) man page for more information on how locale setting can be further changed depending on server side configuration.

退出状态

The status of the remote program is returned as the exit status of `ssh`. 255 is returned if an error occurred at anytime during the `ssh` connection, including the initial key exchange.

文件

`$HOME/.ssh/known_hosts` Records host keys for all hosts the user has logged into that are not in `/etc/ssh/ssh_known_hosts`. See [ssh\(1M\)](#).

`$HOME/.ssh/identity`

`$HOME/.ssh/id_dsa`

`$HOME/.ssh/id_ssa`

Contains the authentication identity of the user. These files are for protocol 1 RSA, protocol 2 DSA, and protocol 2 RSA, respectively. These files contain sensitive data and should be readable by the user but not accessible by others (read/write/execute). `ssh` ignores a private key file if it is accessible by others. It is possible to specify a passphrase when generating the key. The passphrase is used to encrypt the sensitive part of this file using 3DES.

`/etc/ssh/sshrd`

Commands in this file are executed by `ssh` when the user logs in just before the user's shell or command is started. See [sshd\(1M\)](#) for more information.

`$HOME/.ssh/rc`

Commands in this file are executed by `ssh` when the user logs in just before the user's shell or command is started. See [sshd\(1M\)](#) for more information.

`$HOME/.ssh/environment`

Contains additional definitions for environment variables. See ENVIRONMENT VARIABLES.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	network/ssh
Interface Stability	See below.

The command line syntax is Committed. The remote locale selection through passing LC_* environment variables is Uncommitted.

另请参见

[rlogin\(1\)](#), [rsh\(1\)](#), [scp\(1\)](#), [ssh-add\(1\)](#), [ssh-agent\(1\)](#), [ssh-keygen\(1\)](#), [ssh-http-proxy-connect\(1\)](#), [ssh-socks5-proxy-connect\(1\)](#), [telnet\(1\)](#), [sshd\(1M\)](#), [ssh_config\(4\)](#), [sshd_config\(4\)](#), [attributes\(5\)](#), [gss_auth_rules\(5\)](#), [kerberos\(5\)](#), [privileges\(5\)](#)

See the discussion of the `.k5login` file in [krb5_auth_rules\(5\)](#).

RFC 1928

RFC 4254

引用名 ssh-add – 将 RSA 或 DSA 标识添加到验证代理

用法概要 ssh-add [-lLdX] [-t *life*] [*file*]...

描述 ssh-add 实用程序将 RSA 或 DSA 标识添加到验证代理 `ssh-agent(1)`。如果在不带参数的情况下运行，它将尝试添加存在的所有文件（`$HOME/.ssh/identity` (RSA v1)、`$HOME/.ssh/id_rsa` (RSA v2) 和 `$HOME/.ssh/id_dsa` (DSA v2)）。如果存在多个私钥，在重新提示输入不同的口令短语之前，将会尝试使用相同的口令短语解密各个私钥。口令短语是从用户的 tty 中读取的，或通过运行在 `SSH_ASKPASS` 中定义的程序来读取的（请参见下文）。

验证代理必须正在运行。

选项 支持以下选项：

- d 此选项会从代理中删除标识，而不是添加标识。
- D 从代理中删除所有标识。
- l 列出代理当前提供的所有标识的指纹。
- L 列出代理当前提供的所有标识的公钥参数。
- t *life* 将标识添加到代理时设置最长生命周期。可以以秒为单位指定生命周期，也可以按 `sshd(1M)` 中指定的时间格式指定生命周期。
- x 使用口令锁定代理。
- X 解除锁定代理。

环境变量

DISPLAY

SSH_ASKPASS 如果 ssh-add 需要一个口令短语，当在终端中运行时，它将通过当前终端读取口令短语。如果 ssh-add 没有与之关联的终端，但设置了 DISPLAY 和 SSH_ASKPASS，它将执行 SSH_ASKPASS 指定的程序并打开一个 X11 窗口来读取口令短语。从 `.Xsession` 或相关脚本调用 ssh-add 时，这尤其有用。系统附带了 `/usr/lib/ssh/ssh-askpass`，这是 SSH_ASKPASS 的缺省值。

SSH_AUTH_SOCK 标识用于与代理进行通信的 unix 域套接字的路径。

退出状态 将返回以下退出值：

- 0 成功完成。
- 1 出现错误。

文件 除了用户外，其他任何人都无法读取这些文件。请注意，如果某个文件可由其他人访问，则 ssh-add 会忽略该文件。可以在生成密钥时指定口令短语，该口令短语用于加密此文件的私有部分。

如果这些文件存储在网络文件系统中，则假定文件自身所提供的保护或网络文件系统的传输层可为站点策略提供充分的保护。如果不是这样，则建议将密钥文件存储在可移除介质上或在本地存储在相关主机上。

DSA 和 RSA 密钥文件的建议名称：

<code>\$HOME/.ssh/identity</code>	包含协议版本 1 中用户的 RSA 验证标识。
<code>\$HOME/.ssh/identity.pub</code>	包含协议版本 1 中用户的 RSA 验证标识的公共部分。
<code>\$HOME/.ssh/id_dsa</code>	包含用户的私有 DSA 验证标识。
<code>\$HOME/.ssh/id_dsa.pub</code>	包含用户的 DSA 验证标识的公共部分。
<code>\$HOME/.ssh/id_rsa</code>	包含用户的私有 RSA 验证标识。
<code>\$HOME/.ssh/id_rsa.pub</code>	包含用户的 RSA 验证标识的公共部分。
<code>/usr/lib/ssh/ssh-askpass</code>	包含 SSH_ASKPASS 的缺省值。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	network/ssh
接口稳定性	Committed (已确定)

另请参见

[ssh\(1\)](#)、[ssh-agent\(1\)](#)、[ssh-keygen\(1\)](#)、[sshd\(1M\)](#)、[attributes\(5\)](#)

引用名 ssh-agent – 验证代理

用法概要 ssh-agent [-a *bind_address*] [-c | -s] [-d]
 [-t *life*] [*command* [*args*]...]
 ssh-agent [-c | -s] -k

描述 ssh-agent 是一个用于保存公钥验证（RSA、DSA）所使用的私钥的程序。ssh-agent 通常在登录会话开始时启动。所有其他窗口或程序作为 ssh-agent 程序的客户机启动。通过使用环境变量，可查找代理并在登录其他使用 ssh(1) 的计算机时使用代理自动进行验证。请参见《Oracle Solaris 11.1 管理：安全服务》。

如果在命令行上指定了命令，该命令将作为代理的子进程执行。命令终止时，代理也随之终止。

最初，代理没有任何私钥。密钥是使用 ssh-add(1) 添加的，此命令将标识发送给代理。一些标识可以存储在代理中；代理可以自动使用其中的任何标识。在 ssh-add(1) 中使用 -l 选项可显示代理当前拥有的标识。

代理在用户的本地主机中运行。验证数据不需要存储在任何其他计算机上，验证口令短语也决不会通过网络传输。但是，如果与代理的连接是通过 SSH 远程登录转发的，用户可以在网络中的任何位置安全地使用标识赋予的特权。

获取代理设置主要有两种方式。您可以使代理启动一个新的子命令，将某些环境变量导出到该子命令中，或者令代理输出所需的 shell 命令（可以生成 sh(1) 或 csh(1) 语法样式），之后可以在调用方 shell 中通过 eval 显示这些命令的结果。稍后，可使用 ssh(1) 查看这些变量并使用这些变量与代理建立连接。

将创建 Unix 域套接字 (/tmp/ssh-XXXXXXX/agent.*pid*) 并该套接字的名称将存储在 SSH_AUTH_SOCK 环境变量中。只有当前用户可访问该套接字。此方法很容易被 root 用户或同一用户的另一实例错误使用。

SSH_AGENT_PID 环境变量保存代理的 PID。

在命令行上指定的命令终止时，代理将自动退出。

选项 支持以下选项：

- a *bind_address* 将代理绑定到 Unix 域套接字 *bind_address*。缺省值是 /tmp/ssh-XXXXXXX/agent.*pid*。
- c 在 stdout 上生成 C-shell 命令。如果 SHELL 指出它为 csh 样式的 shell，此选项将为缺省值。
- d 调试模式。指定此选项时，ssh-agent 不执行派生。
- k 中止当前代理（由 SSH_AGENT_PID 环境变量指定）。
- s 在 stdout 上生成 Bourne shell 命令。如果 SHELL 未指出它为 csh 样式的 shell，此选项将为缺省值。

`-t life` 为添加到代理的标识设置最大生命周期 (*life*) 缺省值。*life* 可以按秒为单位指定或使用 `sshd_config(4)` 中指定的时间格式指定。使用 `ssh-add(1)` 为标识指定的 *life* 可覆盖此值。如果不指定此选项，则缺省的最大生命周期 (*life*) 为永久。

退出状态 将返回以下退出值：

0 成功完成。

1 出现错误。

文件 `/tmp/ssh-XXXXXXXX/agent.pid` 用于包含验证代理连接的 Unix 域套接字。这些套接字应仅供所有者读取。代理退出时，将删除套接字。

属性 有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	network/ssh
接口稳定性	Committed (已确定)

另请参见 `ssh(1)`、`ssh-add(1)`、`ssh-keygen(1)`、`sshd(1M)`、`sshd_config(4)`、`attributes(5)`

《Oracle Solaris 11.1 管理：安全服务》

引用名	ssh-http-proxy-connect – 用于 HTTP 的安全 Shell 代理
用法概要	<pre>/usr/lib/ssh/ssh-http-proxy-connect [-h <i>http_proxy_host</i> [-p <i>http_proxy_port</i>] connect_host connect_port</pre>
描述	针对使用 HTTP CONNECT 的 ssh(1) 的代理命令。通常用于网络外部连接只允许通过 Web 代理服务器建立的情况。
选项	支持以下选项： <ul style="list-style-type: none"> -h <i>http_proxy_host</i> 指定通过哪个 Web 代理服务器建立连接。如果设置了 HTTPPROXY 和 http_proxy 环境变量，则覆盖这两个变量。 -p <i>http_proxy_port</i> 指定 Web 代理服务器在哪个端口运行。如果未指定，则假定为端口 80。如果设置了 HTTPPROXYPORT 和 http_proxy 环境变量，则覆盖这两个变量。
操作数	支持下列操作数： <ul style="list-style-type: none"> <i>http_proxy_host</i> 代理的主机名或 IP 地址（IPv4 或 IPv6）。 <i>http_proxy_port</i> 所要连接的 <i>http_proxy_host</i> 上的数字端口号。 <i>connect_host</i> Web 代理服务器要将您的计算机连接到的远程主机的名称。 <i>connect_port</i> Web 代理服务器要将您的计算机连接到的 <i>http_proxy_host</i> 上的数字端口号。
示例	<p>代理连接命令的建议使用方法是在 ssh_config(4) 中配置 ProxyCommand（请参见示例 1 和示例 2）。示例 3 显示了在运行 ssh(1) 时如何在命令行中指定代理命令。</p> <p>示例 1 在环境中设置代理</p> <p>以下示例显示了在环境中设置代理时，如何在 ssh_config(4) 中使用 ssh-http-proxy-connect：</p> <pre>Host playtime.foo.com ProxyCommand /usr/lib/ssh/ssh-http-proxy-connect \ playtime.foo.com 22</pre> <p>示例 2 覆盖代理环境变量</p> <p>以下示例显示了如何在 ssh_config(4) 中使用 ssh-http-proxy-connect 来覆盖代理环境变量（如果未设置则进行设置）：</p> <pre>Host playtime.foo.com ProxyCommand /usr/lib/ssh/ssh-http-proxy-connect -h webcache \ -p 8080 playtime.foo.com 22</pre> <p>示例 3 使用命令行</p> <p>以下示例显示了如何在 ssh(1) 命令行中使用 ssh-http-proxy-connect：</p>

示例3 使用命令行 (续)

```
example$ ssh -o ProxyCommand="/usr/lib/ssh/ssh-http-proxy-connect \  
-h webcache -p 8080 playtime.foo.com 22" playtime.foo.com
```

环境变量

HTTPPROXY 使用 *http_proxy_host* 操作数指定缺省代理主机。如果还设置了 *http_proxy*，则覆盖 *http_proxy*。

HTTPPROXYPORT 使用 *http_proxy_port* 操作数指定缺省代理端口。如果未设置 **HTTPPROXY**，则忽略此项。

http_proxy 用于指定代理主机和端口的 URL 格式。

退出状态 将返回以下退出值：

- 0 成功完成。
- 1 出现错误。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	network/ssh
接口稳定性	Committed (已确定)

另请参见 [ssh\(1\)](#)、[ssh-socks5-proxy-connect\(1\)](#)、[ssh_config\(4\)](#)、[attributes\(5\)](#)

引用名 ssh-keygen – 生成验证密钥

用法概要

```
ssh-keygen [-q] [-b bits] -t type [-N new_passphrase]
           [-C comment] [-f output_keyfile]

ssh-keygen -p [-P old_passphrase] [-N new_passphrase]
           [-f keyfile]

ssh-keygen -i [-f input_keyfile | PKCS#11-URI]

ssh-keygen -e [-f input_keyfile]

ssh-keygen -y [-f input_keyfile]

ssh-keygen -c [-P passphrase] [-C comment] [-f keyfile]

ssh-keygen -l [-f input_keyfile | PKCS#11-URI]

ssh-keygen -B [-f input_keyfile]

ssh-keygen -F hostname [-f known_hosts_file]

ssh-keygen -H [-f known_hosts_file]

ssh-keygen -R hostname [-f known_hosts_file]
```

描述

ssh-keygen 实用程序用于为 ssh(1) 生成、管理和转换验证密钥。ssh-keygen 可以创建供 SSH 协议版本 1 使用的 RSA 密钥，以及供 SSH 协议版本 2 使用的 RSA 或 DSA 密钥。生成的密钥类型使用 -t 选项指定。ssh-keygen 还可以生成指纹或从指定为 PKCS#11 URI 的 X.509v3 证书转换公钥。

通常，希望将 SSH 与 RSA 或 DSA 验证结合使用的每个用户应运行一次此实用程序，以便在 \$HOME/.ssh/identity、\$HOME/.ssh/id_dsa 或 \$HOME/.ssh/id_rsa 中创建验证密钥。系统管理员还可以使用此实用程序生成主机密钥。

通常，此程序会生成密钥并要求提供一个存储私钥的文件。公钥存储在附加了 ".pub" 扩展名的同名文件中。该程序还要求提供口令短语。口令短语可以为空，表示没有口令短语（主机密钥必须具有空口令短语），也可以为任意长度的字符串。好的口令短语长度在 10-30 个字符，不是简单的句子或其他容易猜到的句子，由大写字母、小写字母、数字和非字母数字字符组合而成。（普通英文句子中的每个字只有 1-2 位的熵，提供的口令短语非常差。）如果设置口令短语，则长度必须至少为 4 个字符。

之后可以使用 -p 选项更改口令短语。

无法恢复丢失的口令短语。如果丢失或忘记口令短语，您必须生成一个新密钥并将相应的公钥复制到其他计算机。

对于 RSA，密钥文件中还包含一个注释字段，该字段只是为了便于用户识别密钥。comment 字段可以说明密钥的用途或提供任何有用信息。创建密钥时，注释将初始化为 "user@host"，但可以使用 -c 选项进行更改。

密钥生成后，可以使用下面的详细说明了解应将密钥放在何处，以便激活密钥。

选项

支持以下选项：

- b *bits*
指定要创建的密钥的位数。最小位数为 512 位。通常，2048 位足以满足安全需要。密钥大小超过该值并不会提高安全性，反而会降低速度。缺省值为 2048 位。
- B
显示指定的私钥或公钥文件的 bubblebabble 摘要。
- c
请求更改私钥和公钥文件中的注释。该程序会提示您提供包含私钥的文件、口令短语（如果密钥具有一个口令短语）以及新的注释。

此选项仅适用于 `rsa1` (SSHv1) 密钥。
- C *comment*
提供新注释。
- e
此选项读取 OpenSSH 私钥或公钥文件并将密钥以 "SECSH" 公钥文件格式输出到 `stdout`。此选项允许导出密钥供其他一些 SSH 实现使用。
- f
指定密钥文件的文件名。
- F
在 `known_hosts` 文件中搜索指定的 *hostname*，列出找到的任何匹配项。此选项可用于查找散列格式的主机名或地址，还可以与 `-H` 选项一起使用，以散列格式输出找到的密钥。
- H
对 `known_hosts` 文件执行散列计算。此选项使用散列形式替换指定文件内的所有主机名和地址。原始内容将移动到后缀为 `.old` 的文件中。这些散列值通常由 `ssh` 和 `sshd` 使用，即使文件内容被公开，这些散列值也并不会透露可识别的信息。此选项不会修改现有的散列主机名，因此可以放心地用于同时包含散列名称和非散列名称的文件。
- i
此选项以 SSH2 兼容格式读取未加密的私钥（或公钥）文件并将 OpenSSH 兼容的私钥（或公钥）输出到 `stdout`。`ssh-keygen` 还可读取 "SECSH" 公钥文件格式。此选项允许从其他一些 SSH 实现中导入密钥。
- l
显示指定的私钥或公钥文件的指纹。
- N *new_passphrase*
提供新口令短语。
- p
请求更改私钥文件的口令短语，而不创建新私钥。该程序会提示您提供包含私钥的文件、旧口令短语，并两次提示您输入新口令短语。

- P *phrase*
提供（旧）口令短语。
- q
退出 `ssh-keygen`。
- t *type*
指定用于生成密钥的算法，其中 *type* 是 `rsa`、`dsa` 和 `rsa1` 中的一种。`rsa1` 类型仅用于 SSHv1 协议。
- R *hostname*
从 `known_hosts` 文件中删除属于 *hostname* 的所有密钥。此选项可用于删除散列主机。请参见 `-H`。
- x
已过时。已被 `-e` 选项取代。
- X
已过时。已被 `-i` 选项取代。
- y
此选项读取 OpenSSH 私钥格式文件并将 OpenSSH 公钥输出到 `stdout`。
- 8
指定 `ssh-keygen` 生成 PKCS#8 格式的密钥。对于要生成的密钥，支持的类型为 `rsa` 或 `dsa`。

退出状态

将返回以下退出值：

- 0
成功完成。
- 1
出现错误。

文件

`$HOME/.ssh/identity`

该文件包含用于 SSHv1 协议的 RSA 私钥。除了用户外，其他任何人都应无法读取此文件。生成密钥时可以指定密码短语；该密码短语用于使用 128 位 AES 对此文件的私有部分加密。`ssh-keygen` 并不自动访问此文件，此文件是作为私钥的缺省文件提供的。尝试登录时，`sshd(1M)` 将读取该文件。

`$HOME/.ssh/identity.pub`

该文件包含用于 SSHv1 协议的 RSA 公钥。应在登录时要使用 RSA 验证的所有计算机的 `$HOME/.ssh/authorized_keys` 中添加该文件的内容。不需要将该文件的内容保密。

`$HOME/.ssh/id_dsa`

`$HOME/.ssh/id_rsa`

这两个文件分别包含用于 SSHv2 协议的 DSA 私钥或 RSA 私钥。除了用户外，其他任何人都应无法读取这些文件。生成密钥时可能会指定口令短语；该口令短语用于

对此文件的私钥部分加密（使用 3DES）。ssh-keygen 并不自动访问其中的任一文件，这两个文件是作为私钥的缺省文件提供的。尝试登录时，sshd(1M) 将读取其中的一个文件。

```
$HOME/.ssh/id_dsa.pub
```

```
$HOME/.ssh/id_rsa.pub
```

这些文件分别包含用于 SSHv2 协议的 DSA 公钥或 RSA 公钥。应在登录时要使用 DSA 或 RSA 验证的所有计算机的 \$HOME/.ssh/authorized_keys 中分别添加这两个文件的内容。不需要将这些文件的内容保密。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	network/ssh/ssh-key
接口稳定性	Committed（已确定）

另请参见

[ssh\(1\)](#)、[ssh-add\(1\)](#)、[ssh-agent\(1\)](#)、[sshd\(1M\)](#)、[attributes\(5\)](#)

引用名	ssh-keyscan – 收集大量主机的 ssh 主机公钥
用法概要	ssh-keyscan [-v46] [-p <i>port</i>] [-T <i>timeout</i>] [-t <i>type</i>] [-f <i>file</i>] [-] [<i>host...</i> <i>addrlist namelist</i>] [...]
描述	<p>ssh-keyscan 是一个用于收集大量主机的 ssh 主机公钥的实用程序。该实用程序用于帮助生成和验证 <code>ssh_known_hosts</code> 文件。ssh-keyscan 提供了适合 shell 和 perl 脚本使用的最小接口。ssh-keyscan 的输出定向到标准输出。</p> <p>ssh-keyscan 使用非阻塞套接字 I/O，以并行方式联系尽可能多的主机，因此该实用程序非常高效。包括 1,000 个主机的域中的密钥可以在几十秒内收集完毕，即使其中一些主机关闭或未运行 ssh。扫描时，不需要登录访问所扫描的计算机，扫描过程也不涉及加密。</p>
文件格式	<p>输入格式：</p> <p>1.2.3.4,1.2.4.4 <i>name.my.domain,name,n.my.domain,n,1.2.3.4,1.2.4.4</i></p> <p>rsa1 密钥的输出格式：</p> <p><i>host-or-namelist bits exponent modulus</i></p> <p>rsa 和 dsa 密钥的输出格式，其中 <i>keytype</i> 是 ssh-rsa 或 'ssh-dsa'：</p> <p><i>host-or-namelist keytype base64-encoded-key</i></p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none">-f <i>filename</i> 从该文件中读取主机或地址列表名称列表对，每行读取一个。如果指定的不是文件名，ssh-keyscan 将从标准输入中读取主机或地址列表名称列表对。-p <i>port</i> 所要连接的远程主机上的端口。-T <i>timeout</i> 设置连接尝试的超时时间。如果自发起主机连接以来或自上次从该主机中读取内容以来经过了 <i>timeout</i> 秒，则将关闭连接并且所涉及的主机将被视为不可用。<i>timeout</i> 的缺省值为 5 秒。-t <i>type</i> 指定要从被扫描的主机提取的密钥类型。<i>type</i> 的可能值包括用于协议版本 1 的 <i>rsa1</i> 和用于协议版本 2 的 <i>rsa</i> 或 <i>dsa</i>。可指定以逗号分隔的多个值。缺省值为 <i>rsa1</i>。-v 指定详细模式。输出有关进度的调试消息。-4 强制仅使用 IPv4 地址。-6 强制仅使用 IPv6 地址。

安全 如果在不验证密钥的情况下使用 `ssh-keyscan` 构建 `ssh_known_hosts` 文件，用户容易遭受中间人 (man-in-the-middle) 攻击。如果安全模型允许存在此类风险，可以在创建 `ssh_known_hosts` 文件后使用 `ssh-keyscan` 帮助检测被篡改的密钥文件或发生的中间人攻击。

示例 **示例1** 输出 `rsa1` 主机密钥
以下示例输出了计算机 `hostname` 的 `rsa1` 主机密钥：

```
$ ssh-keyscan hostname
```

示例2 查找所有主机

以下命令查找 `ssh_hosts` 文件中密钥与排序文件 `ssh_known_hosts` 中密钥不同（密钥未在该排序文件中出现）的所有主机。

```
$ ssh-keyscan -t rsa,dsa -f ssh_hosts | \
    sort -u - ssh_known_hosts | diff ssh_known_hosts -
```

文件 `/etc/ssh_known_hosts`
包含 `ssh` 主机公钥列表。

退出状态 将返回以下退出值：

0

没有用法错误。`ssh-keyscan` 可能成功，也可能失败，还有可能无法扫描一个、多个或所有给定的主机。

1

用法错误。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	network/ssh
接口稳定性	Committed（已确定）

另请参见 [ssh\(1\)](#)、[sshd\(1M\)](#)、[attributes\(5\)](#)

作者 David Mazieres 编写了初始版本，Wayne Davison 添加了对协议版本 2 的支持。

已知问题 如果服务器低于版本 2.9，`ssh-keyscan` 会在它扫描的所有计算机的控制台上生成以下消息：

```
Connection closed by remote host
```

这是因为 `ssh-keyscan` 采用以下过程：打开与 `ssh` 端口的连接，读取公钥，在获取密钥后立即放弃该连接。

引用名	ssh-socks5-proxy-connect – 用于 SOCKS5 的安全 Shell 代理
用法概要	<code>/usr/lib/ssh/ssh-socks5-proxy-connect</code> <code>[-h socks5_proxy_host]</code> <code>[-p socks5_proxy_port] connect_host connect_port</code>
描述	<p>针对使用 SOCKS5 (RFC 1928) 的 <code>ssh(1)</code> 的代理命令。通常用于网络外部连接只允许通过 socks 网关服务器建立的情况。</p> <p>该代理命令未提供任何 RFC 1928 中定义的 SOCKS5 验证机制。只可能进行匿名连接。</p>
选项	<p>支持以下选项：</p> <p><code>-h socks5_proxy_host</code> 指定通过哪个 Web 代理服务器建立连接。覆盖 SOCKS5_SERVER 环境变量。</p> <p><code>-p socks5_proxy_port</code> 指定 Web 代理服务器在哪个端口运行。如果未指定，则假定为端口 80。覆盖 SOCKS5_PORT 环境变量。</p>
操作数	<p>支持下列操作数：</p> <p><code>socks5_proxy_host</code> 代理的主机名或 IP 地址（IPv4 或 IPv6）。</p> <p><code>socks5_proxy_port</code> 所要连接的 <code>socks5_proxy_host</code> 上的数字端口号。</p> <p><code>connect_host</code> socks 网关要将您的计算机连接到的远程主机的名称。</p> <p><code>connect_port</code> socks 网关要将您的计算机连接到的 <code>connect_host</code> 上的数字端口号。</p>
示例	<p>代理连接命令的建议使用方法是在 <code>ssh_config(4)</code> 中配置 ProxyCommand（请参见示例 1 和示例 2）。示例 3 显示了在运行 <code>ssh(1)</code> 时如何在命令行中指定代理命令。</p> <p>示例 1 在环境中设置代理</p> <p>以下示例显示了在环境中设置代理时，如何在 <code>ssh_config(4)</code> 中使用 <code>ssh-socks5-proxy-connect</code>：</p> <pre>Host playtime.foo.com ProxyCommand /usr/lib/ssh/ssh-socks5-proxy-connect \ playtime.foo.com 22</pre> <p>示例 2 覆盖代理环境变量</p> <p>以下示例显示了如何在 <code>ssh_config(4)</code> 中使用 <code>ssh-socks5-proxy-connect</code> 来覆盖代理环境变量（如果未设置则进行设置）：</p> <pre>Host playtime.foo.com ProxyCommand /usr/lib/ssh/ssh-socks5-proxy-connect -h socks-gw \ -p 1080 playtime.foo.com 22</pre>

示例3 使用命令行

以下示例显示了如何在 `ssh(1)` 命令行中使用 `ssh-socks5-proxy-connect` :

```
example$ ssh -o'ProxyCommand=/usr/lib/ssh/ssh-socks5-proxy-connect \
-h socks-gw -p 1080 playtime.foo.com 22' playtime.foo.com
```

环境变量

SOCKS5_SERVER 使用 `socks5_proxy_host` 操作数指定缺省代理主机。

SOCKS5_PORT 使用 `socks5_proxy_port` 操作数指定缺省代理端口。

退出状态

将返回以下退出值：

0 成功完成。

1 出现错误。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	network/ssh
接口稳定性	Committed (已确定)

另请参见

`ssh(1)`、`ssh-http-proxy-connect(1)`、`ssh_config(4)`、`attributes(5)`

引用名 strchg, strconf - 更改或查询流配置

用法概要 strchg -h *module1* [, *module2...*]

strchg -p [-a | -u *module*]

strchg -f *filename*

strconf [-m | -t *module*]

描述 这些命令用于更改或查询用户的标准输入关联的流配置。strchg 命令推动模块到流上和/或将模块从流上弹出。strconf 命令查询流配置。只有超级用户或 STREAMS 设备的所有者才能更改该流的配置。

如果调用时不带任何参数，strconf 将输出包含流中所有模块以及最顶端的驱动程序的列表。该列表输出时，每行显示一个名称。其中，输出的第一个名称是流上的最顶端模块（如果存在），输出的最后一项是驱动程序的名称。

选项 以下选项适用于 strchg 命令。其中，-h、-f 和 -p 选项是互斥的。

-a 将最顶端的驱动程序上的所有模块从流上弹出来。该选项需和 -p 选项一起使用。

-f *filename* 指定一个包含代表所需流配置的模块列表的 *filename*。每个模块名称必须在一个单独的行上显示。其中，第一个名称代表最顶端的模块，最后一个名称代表最靠近驱动程序的模块。strchg 确定当前流配置，并弹出和推动必要的模块以获取所需的配置。

-h *module1* [, *module2...*] 推动助记符 *h*，将模块推动到流上。该助记符将一个或多个可推流模块的名称作为参数。这些模块按顺序推动，即首先推动 *module1*，其次推动 *module2* 等。

-p 弹出助记符 *p*，将模块从流中弹出。如果只有 -p 选项，strchg 会将最顶端模块从流中弹出。

-u *module* 从流中弹出除 *module* 之外的上述所有模块。该选项需和 -p 选项一起使用。

以下选项适用于 strconf。其中，-m 和 -t 选项是互斥的。

-m *module* 确定指定的 *module* 是否在流上。如果该模块在流上，strconf 命令将输出 yes 消息并返回零值。如果该模块不在流上，strconf 将输出 no 消息并返回非零值。-t 和 -m 选项是互斥的。

-t *module* 只输出最顶端的模块（如果存在）。-t 和 -m 选项是互斥的。

示例 示例 1 使用 strchg 命令

以下命令将模块 ldterm 推动到用户的标准输入关联的流上。

```
example% strchg -h ldterm
```

示例1 使用 strchg 命令 (续)

以下命令将最顶端的模块从 `/dev/term/24` 关联的流上弹出。用户必须是该设备的所有者或超级用户。

```
example% strchg -p < /dev/term/24
```

如果 `fileconf` 文件包含以下几项：

```
ttcompat
ldterm
ptem
```

则以下命令会

```
example% strchg -f fileconf
```

配置用户的标准输入流，以便将 `ptem` 模块推动到驱动程序上，`ldterm` 模块跟随在该模块的后面，而 `ttcompat` 模块被推到最靠近流头。

如果不指定任何参数，`strconf` 命令将列出该流中的模块以及最顶端的驱动程序。如果流中只有 `ldterm` 模块在 `zs` 驱动程序上推动，该命令将产生以下输出：

```
ldterm
zs
```

以下命令询问 `ldterm` 模块是否在流上：

```
example% strconf -m ldterm
```

该命令返回退出状态 0 时产生以下输出：

```
yes
```

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见

[attributes\(5\)](#)、[streamio\(7I\)](#)

诊断

如果执行成功，`strchg` 命令将返回零值。对于各种错误状态，该命令会输出错误消息并返回非零状态，包括使用错误、错误模块名称、要推动的模块太多、流上的 `ioctl` 操作故障，或未能打开 `-f` 选项指定的 `filename`。

如果执行成功，`strconf` 命令将返回零值。对于 `-m` 或 `-t` 选项，“执行成功”意味着指定的模块或最顶端的模块在流上。如果该命令在调用时指定了 `-m` 或者 `-t` 选项并且模块不在流上，则会返回非零状态。对于各种错误状态，该命令会输出错误消息并返回非零状态，包括使用错误或流上的 `ioctl` 操作故障。

附注 如果用户既不是流的所有者也不是超级用户， `strchg` 命令将会失败。如果用户不是超级用户且没有流的读取权限， `strconf` 命令将会失败。

如果模块按错误的顺序推动，用户可能会获取不按预期的方式运行的流。对于 `tty`，如果线路规程模块被推动到不正确的位置，用户的终端可能不会对任何命令作出响应。

引用名	strings – find printable strings in an object or binary file
用法概要	strings [-a -] [-t <i>format</i> -o] [-n <i>number</i> -number] [-N <i>name</i>] [<i>file</i>]...
描述	<p>The <code>strings</code> utility looks for ASCII strings in a binary file. A string is any sequence of 4 or more printing characters ending with a NEWLINE or a NULL character.</p> <p><code>strings</code> is useful for identifying random object files and many other things.</p> <p>By default, <code>strings</code> looks at program sections that are loaded in memory. Program sections are identified by the section type <code>SHT_PROGBITS</code>. Sections that are loaded in memory are identified by the section flag <code>SHF_ALLOC</code>. Use elfdump(1) to display complete section information for a file.</p> <p>All sections can be inspected with the <code>-a</code> option. Individual sections can be inspected with the <code>-N</code> option.</p>
选项	<p>The following options are supported:</p> <p><code>-a</code> <code>-</code> Look everywhere in the file for strings.</p> <p><code>-n <i>number</i></code> <code>-<i>number</i></code> Use a <i>number</i> as the minimum string length rather than the default, which is 4. An invalid number results in the default string length being used.</p> <p><code>-N <i>name</i></code> Look only in ELF section name. See elfdump(1). Multiple <code>-N</code> options can be specified to inspect multiple sections.</p> <p>If the <code>-a</code> or <code>-o</code> option is specified, all <code>-N</code> options are ignored.</p> <p><code>-o</code> Equivalent to <code>-t d</code> option.</p> <p><code>-t <i>format</i></code> Write each string preceded by its byte offset from the start of the file. The format is dependent on the single character used as the <i>format</i> option-argument:</p> <ul style="list-style-type: none"> <code>d</code> The offset is written in decimal. <code>o</code> The offset is written in octal. <code>x</code> The offset is written in hexadecimal.
操作数	<p>The following operand is supported:</p> <p><i>file</i> A path name of a regular file to be used as input. If no <i>file</i> operand is specified, the <code>strings</code> utility reads from the standard input.</p>
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of <code>strings</code> : <code>LANG</code> , <code>LC_ALL</code> , <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>NLSPATH</code> .

退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	See below.

The `strings` utility, including all options except `-N`, are specified by standards. See [standards\(5\)](#). The `-N` option is not currently specified by any standard.

另请参见

[elfdump\(1\)](#), [od\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注

The algorithm for identifying strings is extremely primitive.

For backwards compatibility, the options `-a` and `-` are interchangeable.

引用名	strip – strip symbol table, debugging and line number information from an object file
用法概要	strip [-lVx] file...
描述	<p>The <code>strip</code> command removes the symbol table <code>SHT_SYMTAB</code> and its associated string table, debugging information, and line number information from ELF object files. That is, besides the symbol table and associated string table, the following sections are removed:</p> <pre>.line .debug* .stab*</pre> <p>Once this stripping process has been done, limited symbolic debugging access is available for that file. Therefore, this command is normally run only on production modules that have been debugged and tested.</p> <p>If <code>strip</code> is executed on a common archive file (see ar.h(3HEAD)) in addition to processing the members, <code>strip</code> removes the archive symbol table. The archive symbol table must be restored by executing the ar(1) command with the <code>-s</code> option before the archive can be linked by the ld(1) command. <code>strip</code> produces appropriate warning messages when this situation arises.</p> <p><code>strip</code> is used to reduce the file storage overhead taken by the object file.</p>
选项	<p>The amount of information stripped from the ELF object file can be controlled by using any of the following options. The following options are supported:</p> <ul style="list-style-type: none"> -l Strip line number information only. Does not strip the symbol table or debugging information. -V Prints, on standard error, the version number of <code>strip</code>. -x Does not strip the symbol table. Debugging and line number information might be stripped.
操作数	<p>The following operand is supported:</p> <p><i>file</i> A path name referring to an executable file.</p>
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of <code>strip</code> : <code>LANG</code> , <code>LC_ALL</code> , <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>NLSPATH</code> .
退出状态	<p>The following exit values are returned:</p> <ul style="list-style-type: none"> 0 Successful completion. >0 An error occurred.

文件 /tmp/strip* Temporary files

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [ar\(1\)](#), [as\(1\)](#), [ld\(1\)](#), [mcs\(1\)](#), [elf\(3ELF\)](#), [tmpnam\(3C\)](#), [a.out\(4\)](#), [ar.h\(3HEAD\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注 The symbol table section is not removed if it is contained within a segment or if the file is a relocatable object.

The line number and debugging sections are not removed if they are contained within a segment or if their associated relocation section is contained within a segment.

The `strip` command is used to remove a standard predefined set of sections from an ELF object file. To remove a user specified section by name, see [mcs\(1\)](#).

引用名	stty – set the options for a terminal												
用法概要	<pre> /usr/bin/stty [-a] [-g] /usr/bin/stty [modes] /usr/xpg4/bin/stty [-a -g] /usr/xpg4/bin/stty [modes] /usr/xpg6/bin/stty [-a -g] /usr/xpg6/bin/stty [modes] </pre>												
描述	<p>The <code>stty</code> utility sets certain terminal I/O options for the device that is the current standard input. Without arguments, <code>stty</code> reports the settings of certain options.</p> <p>In this report, if a character is preceded by a caret (^), then the value of that option is the corresponding control character (for example, ^h is CTRL-h. In this case, recall that CTRL-h is the same as the BACKSPACE key). The sequence ^@ means that an option has a null value.</p> <p>See termio(7I) for detailed information about the modes listed from Control Modes through Local Modes. For detailed information about the modes listed under Hardware Flow Control Modes and Clock Modes, see termiox(7I).</p> <p>Operands described in the Combination Modes section are implemented using options in the earlier sections. Notice that many combinations of options make no sense, but no sanity checking is performed. Hardware flow control and clock modes options might not be supported by all hardware interfaces.</p>												
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -a Writes to standard output all of the option settings for the terminal. -g Reports current settings in a form that can be used as an argument to another <code>stty</code> command. Emits <code>termios</code>-type output if the underlying driver supports it. Otherwise, it emits <code>termio</code>-type output. 												
操作数	<p>The following <i>mode</i> operands are supported:</p>												
Control Modes	<table border="0"> <tr> <td style="padding-right: 20px;">parenb (-parenb)</td> <td>Enable (disable) parity generation and detection.</td> </tr> <tr> <td>parext (-parext)</td> <td>Enable (disable) extended parity generation and detection for mark and space parity.</td> </tr> <tr> <td>parodd (-parodd)</td> <td>Select odd (even) parity, or mark (space) parity if <code>parext</code> is enabled.</td> </tr> <tr> <td>cs5 cs6 cs7 cs8</td> <td>Select character size (see termio(7I)).</td> </tr> <tr> <td>0</td> <td>Hang up line immediately.</td> </tr> <tr> <td>hupcl (-hupcl)</td> <td>Hang up (do not hang up) connection on last close.</td> </tr> </table>	parenb (-parenb)	Enable (disable) parity generation and detection.	parext (-parext)	Enable (disable) extended parity generation and detection for mark and space parity.	parodd (-parodd)	Select odd (even) parity, or mark (space) parity if <code>parext</code> is enabled.	cs5 cs6 cs7 cs8	Select character size (see termio(7I)).	0	Hang up line immediately.	hupcl (-hupcl)	Hang up (do not hang up) connection on last close.
parenb (-parenb)	Enable (disable) parity generation and detection.												
parext (-parext)	Enable (disable) extended parity generation and detection for mark and space parity.												
parodd (-parodd)	Select odd (even) parity, or mark (space) parity if <code>parext</code> is enabled.												
cs5 cs6 cs7 cs8	Select character size (see termio(7I)).												
0	Hang up line immediately.												
hupcl (-hupcl)	Hang up (do not hang up) connection on last close.												

	<code>hup (-hup)</code>	Same as <code>hupcl(-hupcl)</code> .
	<code>cstopb (-cstopb)</code>	Use two (one) stop bits per character.
	<code>cread (-cread)</code>	Enable (disable) the receiver.
	<code>crtcts (-crtcts)</code>	Enable output hardware flow control. Raise the RTS (Request to Send) modem control line. Suspends output until the CTS (Clear to Send) line is raised.
	<code>crtxoff (-crtxoff)</code>	Enable input hardware flow control. Raise the RTS (Request to Send) modem control line to receive data. Suspends input when RTS is low.
	<code>cllocal (-cllocal)</code>	Assume a line without (with) modem control.
	<code>defeucw</code>	Set the widths of multibyte characters to the values defined in the current locale specified by <code>LC_CTYPE</code> . Internally, width is expressed in terms of bytes per character, and screen or display columns per character.
	110 300 600 1200 1800 2400 4800 9600 19200 38400 357600 76800 115200 153600 230400 307200 460800	Set terminal baud rate to the number given, if possible. (All speeds are not supported by all hardware interfaces.)
	<code>ispeed 0 110 300 600 1200 1800 2400 4800 9600 19200 38400 57600 76800 115200 153600 230400 307200 460800</code>	Set terminal input baud rate to the number given, if possible. (Not all hardware supports split baud rates.) If the input baud rate is set to 0, the input baud rate is specified by the value of the output baud rate.
	<code>ospeed 0 110 300 600 1200 1800 2400 4800 9600 19200 38400 57600 76800 115200 153600 230400 307200 460800</code>	Set terminal output baud rate to the number given, if possible. (Not all hardware supports split baud rates.) If the output baud rate is set to 0, the line is hung up immediately.
Input Modes	<code>ignbrk (-ignbrk)</code>	Ignore (do not ignore) break on input.
	<code>brkint (-brkint)</code>	Signal (do not signal) INTR on break.
	<code>ignpar (-ignpar)</code>	Ignore (do not ignore) parity errors.
	<code>parmrk (-parmrk)</code>	Mark (do not mark) parity errors (see termio(7I)).

	<code>inpck (-inpck)</code>	Enable (disable) input parity checking.
	<code>istrip (-istrip)</code>	Strip (do not strip) input characters to seven bits.
	<code>inlcr (-inlcr)</code>	Map (do not map) NL to CR on input.
	<code>igncr (-igncr)</code>	Ignore (do not ignore) CR on input.
	<code>icrnl (-icrnl)</code>	Map (do not map) CR to NL on input.
	<code>iuclc (-iuclc)</code>	Map (do not map) upper-case alphabets to lower case on input.
	<code>ixon (-ixon)</code>	Enable (disable) START/STOP output control. Output is stopped by sending STOP control character and started by sending the START control character.
	<code>ixany (-ixany)</code>	Allow any character (only DC1) to restart output.
	<code>ixoff (-ixoff)</code>	Request that the system send (not send) START/STOP characters when the input queue is nearly empty/full.
	<code>imaxbel (-imaxbel)</code>	Echo (do not echo) BEL when the input line is too long. If <code>imaxbel</code> is set, the ASCII BEL character (07 hex) is echoed if the input stream overflows. Further input is not stored, but any input already present is not disturbed. If <code>-imaxbel</code> is set, no BEL character is echoed, and all unread input present in the input queue is discarded if the input stream overflows.
Output Modes	<code>opost (-opost)</code>	Post-process output (do not post-process output; ignore all other output modes).
	<code>olcuc (-olcuc)</code>	Map (do not map) lower-case alphabets to upper case on output.
	<code>onlcr (-onlcr)</code>	Map (do not map) NL to CR-NL on output.
	<code>ocrnl (-ocrnl)</code>	Map (do not map) CR to NL on output.
	<code>onocr (-onocr)</code>	Do not (do) output CRs at column zero.
	<code>onlret (-onlret)</code>	On the terminal NL performs (does not perform) the CR function.
	<code>ofill (-ofill)</code>	Use fill characters (use timing) for delays.
	<code>ofdel (-ofdel)</code>	Fill characters are DELs (NULs).
	<code>cr0 cr1 cr2 cr3</code>	Select style of delay for carriage returns (see termio(7I)).
	<code>nl0 nl1</code>	Select style of delay for line-feeds (see termio(7I)).
	<code>tab0 tab1 tab2 tab3</code>	Select style of delay for horizontal tabs (see termio(7I)).
	<code>bs0 bs1</code>	Select style of delay for backspaces (see termio(7I)).
	<code>ff0 ff1</code>	Select style of delay for form-feeds (see termio(7I)).

	vt0 vt1	Select style of delay for vertical tabs (see termio(7I)).
Local Modes	isig(-isig)	Enable (disable) the checking of characters against the special control characters INTR, QUIT, SWTCH, and SUSP. For information on SWTCH, see NOTES.
	icanon(-icanon)	Enable (disable) canonical input (ERASE and KILL processing). Does not set MIN or TIME.
	xcase(-xcase)	Canonical (unprocessed) upper/lower-case presentation.
	echo(-echo)	Echo back (do not echo back) every character typed.
	echoe(-echoe)	Echo (do not echo) ERASE character as a backspace-space-backspace string. This mode erases the ERASEed character on many CRT terminals; however, it does not keep track of column position and, as a result, it might be confusing for escaped characters, tabs, and backspaces.
	echok(-echok)	Echo (do not echo) NL after KILL character.
	lfkc(-lfkc)	The same as echok(-echok); obsolete.
	echonl(-echonl)	Echo (do not echo) NL.
	noflsh(-noflsh)	Disable (enable) flush after INTR, QUIT, or SUSP.
	stwrap(-stwrap)	Disable (enable) truncation of lines longer than 79 characters on a synchronous line.
	tostop(-tostop)	Send (do not send) SIGTTOU when background processes write to the terminal.
	echoctl(-echoctl)	Echo (do not echo) control characters as <i>^char</i> , delete as <i>^?</i> .
	echoprnt(-echoprnt)	Echo (do not echo) erase character as character is “erased”.
	echoke(-echoke)	BS-SP-BS erase (do not BS-SP-BS erase) entire line on line kill.
	flusho(-flusho)	Output is (is not) being flushed.
	pendin(-pendin)	Retype (do not retype) pending input at next read or input character.
	iexten(-iexten)	Enable (disable) special control characters not currently controlled by icanon, isig, ixon, or ixoff: VEOL, VSWTCH, VREPRINT, VDISCARD, VDSUSP, VWERASE, and VLNEXT.
	stflush(-stflush)	Enable (disable) flush on a synchronous line after every write(2) .
	stappl(-stappl)	Use application mode (use line mode) on a synchronous line.

Hardware Flow Control Modes	<code>rtsxoff (- rtsxoff)</code>	Enable (disable) RTS hardware flow control on input.
	<code>ctsxon (- ctsxon)</code>	Enable (disable) CTS hardware flow control on output.
	<code>dtrxoff (- dtrxoff)</code>	Enable (disable) DTR hardware flow control on input.
	<code>cdxon (- cdxon)</code>	Enable (disable) CD hardware flow control on output.
	<code>isxoff (- isxoff)</code>	Enable (disable) isochronous hardware flow control on input.
Clock Modes	<code>xcibrg</code>	Get transmit clock from internal baud rate generator.
	<code>xctset</code>	Get the transmit clock from transmitter signal element timing (DCE source) lead, CCITT V.24 circuit 114, EIA-232-D pin 15.
	<code>xcrset</code>	Get transmit clock from receiver signal element timing (DCE source) lead, CCITT V.24 circuit 115, EIA-232-D pin 17.
	<code>rcibrg</code>	Get receive clock from internal baud rate generator.
	<code>rctset</code>	Get receive clock from transmitter signal element timing (DCE source) lead, CCITT V.24 circuit 114, EIA-232-D pin 15.
	<code>rcrset</code>	Get receive clock from receiver signal element timing (DCE source) lead, CCITT V.24 circuit 115, EIA-232-D pin 17.
	<code>tsetcoff</code>	Transmitter signal element timing clock not provided.
	<code>tsetcbrg</code>	Output receive baud rate generator on transmitter signal element timing (DTE source) lead, CCITT V.24 circuit 113, EIA-232-D pin 24.
	<code>tsetctbrg</code>	Output transmit baud rate generator on transmitter signal element timing (DTE source) lead, CCITT V.24 circuit 113, EIA-232-D pin 24.
	<code>tsetctset</code>	Output transmitter signal element timing (DCE source) on transmitter signal element timing (DTE source) lead, CCITT V.24 circuit 113, EIA-232-D pin 24.
	<code>tsetcrset</code>	Output receiver signal element timing (DCE source) on transmitter signal element timing (DTE source) lead, CCITT V.24 circuit 113, EIA-232-D pin 24.
	<code>rsetcoff</code>	Receiver signal element timing clock not provided.
	<code>rsetcbrg</code>	Output receive baud rate generator on receiver signal element timing (DTE source) lead, CCITT V.24 circuit 128, no EIA-232-D pin.
	<code>rsetctbrg</code>	Output transmit baud rate generator on receiver signal element timing (DTE source) lead, CCITT V.24 circuit 128, no EIA-232-D pin.
	<code>rsetctset</code>	Output transmitter signal element timing (DCE source) on receiver signal element timing (DTE source) lead, CCITT V.24 circuit 128, no EIA-232-D pin.
<code>rsetcrset</code>	Output receiver signal element timing (DCE source) on receiver signal element timing (DTE source) lead, CCITT V.24 circuit 128, no EIA-232-D pin.	

Control Assignments *control-character c*

Set *control-character* to *c*, where:

control-character is *ctab*, *discard*, *dsusp*, *eof*, *eol*, *eol2*, *erase*, *intr*, *kill*, *lnext*, *quit*, *reprint*, *start*, *stop*, *susp*, *swtch*, or *werase* (*ctab* is used with *-stappl*, see [termio\(7I\)](#)). For information on *swtch*, see NOTES.

c If *c* is a single character, the control character is set to that character.

In the POSIX locale, if *c* is preceded by a caret (^) indicating an escape from the shell and is one of those listed in the ^*c* column of the following table, then its value used (in the Value column) is the corresponding control character (for example, “^d” is a CTRL-d). “^?” is interpreted as DEL and “^–” is interpreted as undefined.

<i>^c</i>	Value	<i>^c</i>	Value	<i>^c</i>	Value
a, A	<SOH>	l, L	<FF>	w, W	<ETB>
b, B	<STX>	m, M	<CR>	x, X	<CAN>
c, C	<ETX>	n, N	<SO>	y, Y	
d, D	<EOT>	o, O	<SI>	z, Z	<SUB>
e, E	<ENQ>	p, P	<DLE>	[<ESC>
f, F	<ACK>	q, Q	<DC1>	\	<FS>
g, G	<BEL>	r, R	<DC2>]	<GS>
h, H	<BS>	s, S	<DC3>	^	<RS>
i, I	<HT>	t, T	<DC4>	_	<US>
j, J	<LF>	u, U	<NAK>	?	
k, K	<VT>	v, V	<SYN>		

min number

time number Set the value of *min* or *time* to *number*. MIN and TIME are used in Non-Canonical mode input processing (*-icanon*).

line i Set line discipline to *i* ($0 < i < 127$).

Combination Modes

saved settings Set the current terminal characteristics to the saved settings produced by the *-g* option.

evenp or parity Enable *parenb* and *cs7*, or disable *parodd*.

	oddp	Enable parenb, cs7, and parodd.
	spacep	Enable parenb, cs7, and parext.
	markp	Enable parenb, cs7, parodd, and parext.
	-parity, or -evenp	Disable parenb, and set cs8.
	-oddp	Disable parenb and parodd, and set cs8.
	-spacep	Disable parenb and parext, and set cs8.
	-markp	Disable parenb, parodd, and parext, and set cs8.
	raw (- raw or cooked)	Enable (disable) raw input and output. Raw mode is equivalent to setting: <pre>stty cs8 -icanon min 1 time 0 -isig -xcase \ -inpck -opost</pre>
/usr/bin/stty, /usr/xpg6/bin/stty	nl (-nl)	Unset (set) icrnl, onlcr. In addition -nl unsets inlcr, igncr, ocrnl, and onlret.
/usr/xpg4/bin/stty	nl (-nl)	Set (unset) icrnl. In addition, -nl unsets inlcr, igncr, ocrnl, and onlret; -nl sets onlcr, and nl unsets onlcr.
	lcase (-lcase)	Set (unset) xcase, iuclc, and olcuc.
	LCASE (-LCASE)	Same as lcase (-lcase).
	tabs (-tabs or tab3)	Preserve (expand to spaces) tabs when printing.
	ek	Reset ERASE and KILL characters back to normal DEL and CTRL-u, respectively.
	sane	Reset all modes to some reasonable values.
	term	Set all modes suitable for the terminal type <i>term</i> , where <i>term</i> is one of tty33, tty37, vt05, tn300, ti700, or tek.
	async	Set normal asynchronous communications where clock settings are xcibrg, rcibrg, tsetcoff and rsetcoff.
Window Size	rows <i>n</i>	Set window size to <i>n</i> rows.
	columns <i>n</i>	Set window size to <i>n</i> columns.
	cols <i>n</i>	Set window size to <i>n</i> columns. cols is a shorthand alias for columns.
	ypixels <i>n</i>	Set vertical window size to <i>n</i> pixels.
	xpixels <i>n</i>	Set horizontal window size to <i>n</i> pixels.

用法 The `-g` flag is designed to facilitate the saving and restoring of terminal state from the shell level. For example, a program can:

```
saveterm="$(stty -g)"      # save terminal state
stty (new settings)      # set new state
...                      # ...
stty $saveterm           # restore terminal state
```

Since the `-a` format is so loosely specified, scripts that save and restore terminal settings should use the `-g` option.

环境变量 See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `stty`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态 The following exit values are returned:

```
0      Successful completion.
>0     An error occurred.
```

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/stty

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

/usr/xpg4/bin/stty

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

/usr/xpg6/bin/stty

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu6
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [tabs\(1\)](#), [ioctl\(2\)](#), [write\(2\)](#), [getwidth\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#), [ldterm\(7M\)](#), [termio\(7I\)](#), [termiox\(7I\)](#)

附注 Solaris does not support any of the actions implied by `swtch`, which was used by the `sxt` driver on System V release 4. Solaris allows the `swtch` value to be set, and prints it out if set, but it does not perform the `swtch` action.

The job switch functionality on Solaris is actually handled by job control. `susp` is the correct setting for this.

引用名	stty – set the options for a terminal
用法概要	<code>/usr/ucb/stty [-a] [-g] [-h] [modes]</code>
描述	<code>stty</code> sets certain terminal I/O options for the device that is the current standard output. Without arguments, <code>stty</code> reports the settings of certain options.
选项	<p>In this report, if a character is preceded by a caret (^), then the value of that option is the corresponding CTRL character (for example, ^h is CTRL-h. In this case, recall that CTRL-h is the same as the BACKSPACE key.) The sequence ^@ means that an option has a null value.</p> <ul style="list-style-type: none"> -a Reports all of the option settings. -g Reports current settings in a form that can be used as an argument to another <code>stty</code> command. -h Reports all the option settings with the control characters in an easy to read column format. <p>Options in the last group are implemented using options in the previous groups. Many combinations of options make no sense, but no sanity checking is performed. Hardware flow control and clock modes options might not be supported by all hardware interfaces. The options are selected from the following:</p>
Special Requests	<ul style="list-style-type: none"> <code>all</code> Reports the same option settings as <code>stty</code> without arguments, but with the control characters in column format. <code>everything</code> Everything <code>stty</code> knows about is printed. Same as <code>-h</code> option. <code>speed</code> The terminal speed alone is reported on the standard output. <code>size</code> The terminal (window) sizes are printed on the standard output, first rows and then columns. This option is only appropriate if currently running a window system. <p style="margin-left: 40px;"><code>size</code> and <code>speed</code> always report on the settings of <code>/dev/tty</code>, and always report the settings to the standard output.</p>
Control Modes	<ul style="list-style-type: none"> <code>parenb (-parenb)</code> Enable (disable) parity generation and detection. <code>parext (-parext)</code> Enable (disable) extended parity generation and detection for mark and space parity. <code>parodd (-parodd)</code> Select odd (even) parity, or mark (space) parity if <code>parext</code> is enabled. <code>cs5 cs6 cs7 cs8</code> Select character size (see termio(7I)). <code>0</code> Hang up line immediately.

`110 300 600 1200 1800 2400 4800 9600 19200 exta 38400 extb`

Set terminal baud rate to the number given, if possible. (All speeds are not supported by all hardware interfaces.)

`ispeed 0 110 300 600 1200 1800 2400 4800 9600 19200 exta 38400 extb`

Set terminal input baud rate to the number given, if possible. (Not all hardware supports split baud rates.) If the input baud rate is set to zero, the input baud rate is specified by the value of the output baud rate.

`ospeed 0 110 300 600 1200 1800 2400 4800 9600 19200 exta 38400 extb`

Set terminal output baud rate to the number given, if possible. (Not all hardware supports split baud rates.) If the baud rate is set to zero, the line is hung up immediately.

`hupcl (-hupcl)`

Hang up (do not hang up) connection on last close.

`hup (-hup)`

Same as `hupcl (-hupcl)`.

`cstopb (-cstopb)`

Use two (one) stop bits per character.

`cread (-cread)`

Enable (disable) the receiver.

`clocal (-clocal)`

Assume a line without (with) modem control.

`crtsets (-crtsets)`

Enable hardware flow control. Raise the RTS (Request to Send) modem control line. Suspends output until the CTS (Clear to Send) line is raised.

`loblk (-loblk)`

Block (do not block) output from a non-current layer.

Input Modes

`ignbrk (-ignbrk)` Ignore (do not ignore) break on input.

`brkint (-brkint)` Signal (do not signal) INTR on break.

`ignpar (-ignpar)` Ignore (do not ignore) parity errors.

`parmrk (-parmrk)` Mark (do not mark) parity errors (see [termio\(7I\)](#)).

`inpck (-inpck)` Enable (disable) input parity checking.

`istrip (-istrip)` Strip (do not strip) input characters to seven bits.

`inlcr (-inlcr)` Map (do not map) NL to CR on input.

`igncr (-igncr)` Ignore (do not ignore) CR on input.

`icrnl (-icrnl)` Map (do not map) CR to NL on input.

`iuclc (-iuclc)` Map (do not map) upper-case alphabetic to lower case on input.

	<code>ixon (-ixon)</code>	Enable (disable) START/STOP output control. Output is stopped by sending an STOP and started by sending an START.
	<code>ixany (-ixany)</code>	Allow any character (only START) to restart output.
	<code>decctlq (-decctlq)</code>	Same as <code>-ixany</code> .
	<code>ixoff (-ixoff)</code>	Request that the system send (not send) START/STOP characters when the input queue is nearly empty/full.
	<code>tandem (-tandem)</code>	Same as <code>ixoff</code> .
	<code>imaxbel (-imaxbel)</code>	Echo (do not echo) BEL when the input line is too long.
	<code>iexten (-iexten)</code>	Enable (disable) extended (implementation-defined) functions for input data.
Output Modes	<code>opost (-opost)</code>	Post-process output (do not post-process output; ignore all other output modes).
	<code>olcuc (-olcuc)</code>	Map (do not map) lower-case alphabetic to upper case on output.
	<code>onlcr (-onlcr)</code>	Map (do not map) NL to CR-NL on output.
	<code>ocrn1 (-ocrn1)</code>	Map (do not map) CR to NL on output.
	<code>onocr (-onocr)</code>	Do not (do) output CRs at column zero.
	<code>onlret (-onlret)</code>	On the terminal NL performs (does not perform) the CR function.
	<code>ofill (-ofill)</code>	Use fill characters (use timing) for delays.
	<code>ofdel (-ofdel)</code>	Fill characters are DELs (NULs).
	<code>cr0 cr1 cr2 cr3</code>	Select style of delay for carriage returns (see termio(7I)).
	<code>nl0 nl1</code>	Select style of delay for line-feeds (see termio(7I)).
	<code>tab0 tab1 tab2 tab3</code>	Select style of delay for horizontal tabs (see termio(7I)).
	<code>bs0 bs1</code>	Select style of delay for backspaces (see termio(7I)).
	<code>ff0 ff1</code>	Select style of delay for form-feeds (see termio(7I)).
	<code>vt0 vt1</code>	Select style of delay for vertical tabs (see termio(7I)).
Local Modes	<code>isig (-isig)</code>	Enable (disable) the checking of characters against the special control characters INTR, QUIT, and SWTCH. For information on SWTCH, see NOTES.
	<code>icanon (-icanon)</code>	Enable (disable) canonical input (ERASE and KILL processing). Does not set MIN or TIME.
	<code>cbreak (-cbreak)</code>	Equivalent to <code>-icanon min 1 time 0</code> .

	<code>xcase (-xcase)</code>	Canonical (unprocessed) upper/lower-case presentation.
	<code>echo (-echo)</code>	Echo back (do not echo back) every character typed.
	<code>echoe (-echoe)</code>	Echo (do not echo) ERASE character as a backspace-space-backspace string. Note: This mode erases the ERASEed character on many CRT terminals; however, it does <i>not</i> keep track of column position and, as a result, can be confusing on escaped characters, tabs, and backspaces.
	<code>crterase (-crterase)</code>	Same as <code>echoe</code> .
	<code>echok (-echok)</code>	Echo (do not echo) NL after KILL character.
	<code>lfkc (-lfkc)</code>	The same as <code>echok (-echok)</code> ; obsolete.
	<code>echonl (-echonl)</code>	Echo (do not echo) NL.
	<code>noflsh (-noflsh)</code>	Disable (enable) flush after INTR, QUIT, or SWITCH. For information on SWITCH, see NOTES.
	<code>stwrap (-stwrap)</code>	Disable (enable) truncation of lines longer than 79 characters on a synchronous line. (Does not apply to the 3B2.)
	<code>tostop (-tostop)</code>	Send (do not send) SIGTTOU for background processes.
	<code>echoctl (-echoctl)</code>	Echo (do not echo) control characters as <i>^char</i> , delete as <i>^?</i>
	<code>ctlecho (-ctlecho)</code>	Same as <code>echoctl</code> .
	<code>echoprt (-echoprt)</code>	Echo (do not echo) erase character as character is “erased”.
	<code>prterase (-prterase)</code>	Same as <code>echoprt</code> .
	<code>echoke (-echoke)</code>	BS-SP-BS erase (do not BS-SP-BS erase) entire line on line kill.
	<code>crtkill (-crtkill)</code>	Same as <code>echoke</code> .
	<code>flusho (-flusho)</code>	Output is (is not) being flushed.
	<code>pendin (-pendin)</code>	Retype (do not retype) pending input at next read or input character.
	<code>stflush (-stflush)</code>	Enable (disable) flush on a synchronous line after every <code>write(2)</code> . (Does not apply to the 3B2.)
	<code>stappl (-stappl)</code>	Use application mode (use line mode) on a synchronous line. (Does not apply to the 3B2.)
Hardware Flow Control Modes	<code>rtsxoff (-rtsxoff)</code>	Enable (disable) RTS hardware flow control on input.
	<code>ctson (-ctson)</code>	Enable (disable) CTS hardware flow control on output.
	<code>dterxoff (-dterxoff)</code>	Enable (disable) DTER hardware flow control on input.

	<code>rldsdxon (-rldsdxon)</code>	Enable (disable) RLSD hardware flow control on output.
	<code>isxoff (-isxoff)</code>	Enable (disable) isochronous hardware flow control on input.
Clock Modes	<code>xcibrg</code>	Get transmit clock from internal baud rate generator.
	<code>xctset</code>	Get the transmit clock from transmitter signal element timing (DCE source) lead, CCITT V.24 circuit 114, EIA-232-D pin 15.
	<code>xcrset</code>	Get transmit clock from receiver signal element timing (DCE source) lead, CCITT V.24 circuit 115, EIA-232-D pin 17.
	<code>rcibrg</code>	Get receive clock from internal baud rate generator.
	<code>rctset</code>	Get receive clock from transmitter signal element timing (DCE source) lead, CCITT V.24 circuit 114, EIA-232-D pin 15.
	<code>rcrset</code>	Get receive clock from receiver signal element timing (DCE source) lead, CCITT V.24 circuit 115, EIA-232-D pin 17.
	<code>tsetcoff</code>	Transmitter signal element timing clock not provided.
	<code>tsetcrc</code>	Output receive clock on transmitter signal element timing (DTE source) lead, CCITT V.24 circuit 113, EIA-232-D pin 24, clock source.
	<code>tsetcxc</code>	Output transmit clock on transmitter signal element timing (DTE source) lead, CCITT V.24 circuit 113, EIA-232-D pin 24, clock source.
	<code>rsetcoff</code>	Receiver signal element timing clock not provided.
	<code>rsetcrc</code>	Output receive clock on receiver signal element timing (DTE source) lead, CCITT V.24 circuit 128, no EIA-232-D pin, clock source.
	<code>rsetcxc</code>	Output transmit clock on receiver signal element timing (DTE source) lead, CCITT V.24 circuit 128, no EIA-232-D pin, clock source.
Control Assignments	<code>control-character c</code>	Set <i>control-character</i> to <i>c</i> , where <i>control-character</i> is <code>intr</code> , <code>quit</code> , <code>erase</code> , <code>kill</code> , <code>eof</code> , <code>eol</code> , <code>eol2</code> , <code>swtch</code> , <code>start</code> , <code>stop</code> , <code>susp</code> , <code>dsusp</code> , <code>rpnt</code> , <code>flush</code> , <code>werase</code> , <code>lnext</code> , <code>min</code> , <code>ctab</code> , <code>time</code> , or <code>brk</code> (<code>ctab</code> is used with <code>-stappl</code> ; <code>min</code> and <code>time</code> are used with <code>-icanon</code> ; see termio(7I)). If <i>c</i> is preceded by an (escaped from the shell) caret (^), then the value used is the corresponding CTRL character (for example, “^d” is a Control-d). “^?” is interpreted as DEL and “^–” is interpreted as undefined. For information on <code>SWTCH</code> , see NOTES.
	<code>line i</code>	Set line discipline to <i>i</i> ($0 < i < 127$).
Combination Modes	<code>evenp or parity</code>	Enable <code>parenb</code> and <code>cs7</code> .
	<code>-evenp, or -parity</code>	Disable <code>parenb</code> , and set <code>cs8</code> .
	<code>even (-even)</code>	Same as <code>evenp (-evenp)</code> .

	<code>oddp</code>	Enable <code>parenb</code> , <code>cs7</code> , and <code>parodd</code> .
	<code>-oddp</code>	Disable <code>parenb</code> and <code>parodd</code> , and set <code>cs8</code> .
	<code>odd (-odd)</code>	Same as <code>oddp (-oddp)</code> .
	<code>spacep</code>	Enable <code>parenb</code> , <code>cs7</code> , and <code>parext</code> .
	<code>-spacep</code>	Disable <code>parenb</code> and <code>parext</code> , and set <code>cs8</code> .
	<code>markp</code>	Enable <code>parenb</code> , <code>cs7</code> , <code>parodd</code> , and <code>parext</code> .
	<code>-markp</code>	Disable <code>parenb</code> , <code>parodd</code> , and <code>parext</code> , and set <code>cs8</code> .
	<code>raw (-raw or cooked)</code>	Enable (disable) raw input and output (no ERASE, KILL, INTR, QUIT, SWTCH, EOT, or output post processing). For information on SWTCH, see NOTES.
	<code>nl (-nl)</code>	Unset (set) <code>icrnl</code> , <code>onlcr</code> . In addition <code>-nl</code> unsets <code>inlcr</code> , <code>igncr</code> , <code>ocrnl</code> , and <code>onlret</code> .
	<code>lcase (-lcase)</code>	Set (unset) <code>xcase</code> , <code>iucLC</code> , and <code>olcuc</code> .
	<code>LCASE (-LCASE)</code>	Same as <code>lcase (-lcase)</code> .
	<code>tabs (-tabs or tab3)</code>	Preserve (expand to spaces) tabs when printing.
	<code>ek</code>	Reset ERASE and KILL characters back to normal DEL and CTRL-u, respectively.
	<code>sane</code>	Reset all modes to some reasonable values.
	<code>term</code>	Set all modes suitable for the terminal type <i>term</i> , where <i>term</i> is one of <code>tty33</code> , <code>tty37</code> , <code>vt05</code> , <code>tn300</code> , <code>ti700</code> , or <code>tek</code> .
	<code>async</code>	Set normal asynchronous communications where clock settings are <code>xcibrg</code> , <code>rcibrg</code> , <code>tsetcoff</code> and <code>rsetcoff</code> .
	<code>litout (-litout)</code>	Disable (enable) <code>parenb</code> , <code>istrip</code> , and <code>opost</code> , and set <code>cs8 (cs7)</code> .
	<code>pass8 (-pass8)</code>	Disable (enable) <code>parenb</code> and <code>istrip</code> , and set <code>cs8 (cs7)</code> .
	<code>crt</code>	Set options for a CRT (<code>echoe</code> , <code>echoctl</code> , and, if ≥ 1200 baud, <code>echoke</code> .)
	<code>dec</code>	Set all modes suitable for Digital Equipment Corp. operating systems users ERASE, KILL, and INTR characters to <code>^?</code> , <code>^U</code> , and <code>^C</code> , <code>decctlq</code> , and <code>crt</code> .)
Window Size	<code>rowsn</code>	Set window size to <i>n</i> rows.
	<code>columnsn</code>	Set window size to <i>n</i> columns.
	<code>colsn</code>	An alias for <code>columnsn</code> .

`ypixels n` Set vertical window size to *n pixels*.

`xpixels n` Set horizontal window size to *n pixels*.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见

[tabs\(1\)](#), [ioctl\(2\)](#), [attributes\(5\)](#), [termio\(7I\)](#), [termiox\(7I\)](#)

附注

Solaris does not support any of the actions implied by `swt ch`, which was used by the `sxt` driver on System V release 4. Solaris allows the `swt ch` value to be set, and prints it out if set, but it does not perform the `swt ch` action.

The job switch functionality on Solaris is actually handled by job control. `susp` is the correct setting for this.

引用名	sum – print checksum and block count for a file
用法概要	<code>/usr/bin/sum [-r] [file...]</code>
描述	The <code>sum</code> lists the checksum for each of its file arguments. The standard input is read if there are no file arguments.
选项	The following option is supported: -r Use an alternate (machine-dependent) algorithm in computing the checksum.
操作数	The following operands are supported: <i>file</i> A path name of a file. If no files are named, the standard input is used.
用法	See largefile(5) for the description of the behavior of <code>sum</code> when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).
环境变量	See environ(5) for descriptions of the following environment variables that affect the execution of <code>sum</code> : <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>NLSPATH</code> .
退出状态	The following exit values are returned. 0 Successful completion. >0 An error occurred.
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTETYPE	ATTRIBUTEVALUE
Availability	system/core-os
CSI	Enabled

另请参见	cksum(1) , getconf(1) , sum(1B) , wc(1) , libmd(3LIB) , attributes(5) , environ(5) , largefile(5)
诊断	Read error is indistinguishable from end of file on most devices. Check the block count.
附注	Portable applications should use cksum(1) . The default algorithm for this command is defined in the POSIX standard and is identical across platforms. <code>sum</code> and <code>usr/ucb/sum</code> (see sum(1B)) return different checksums.

引用名 sum – calculate a checksum for a file

用法概要 /usr/ucb/sum *file...*

描述 sum calculates and displays a 16-bit checksum for the named file and displays the size of the file in kilobytes. It is typically used to look for bad spots, or to validate a file communicated over some transmission line. The checksum is calculated by an algorithm which may yield different results on machines with 16-bit ints and machines with 32-bit ints, so it cannot always be used to validate that a file has been transferred between machines with different-sized ints.

用法 See [largefile\(5\)](#) for the description of the behavior of sum when encountering files greater than or equal to 2 Gbyte (2^{31} bytes).

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [sum\(1\)](#), [wc\(1\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

诊断 Read error is indistinguishable from EOF on most devices; check the block count.

附注 sum and /usr/bin/sum (see [sum\(1\)](#)) return different checksums.

This utility is obsolete.

引用名 suspend – shell built-in function to halt the current shell

用法概要

sh suspend

csH suspend

ksh suspend

描述

sh Stops the execution of the current shell (but not if it is the login shell).

csH Stop the shell in its tracks, much as if it had been sent a stop signal with ^Z. This is most often used to stop shells started by su.

ksh Stops the execution of the current shell (but not if it is the login shell).

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [csH\(1\)](#), [kill\(1\)](#), [ksh\(1\)](#), [sh\(1\)](#), [su\(1M\)](#), [attributes\(5\)](#)

引用名	svccprop - 检索服务配置属性值
用法概要	<pre>svccprop [-afqtv] [-C -c -s <i>snapshot</i>] [-p [<i>name/</i>]<i>name</i>]... [-g <i>pgtype</i>]...-l <i>layer</i>[,...] {<i>FMRI</i> <i>pattern</i>}... svccprop -w [-fqtv] [-p [<i>name/</i>]<i>name</i>] {<i>FMRI</i> <i>pattern</i>}</pre>
描述	<p>svccprop 实用程序输出服务配置系统信息库中的属性值。属性可通过 -p 选项和操作数选定。如果未使用 -p 明确选择属性或属性组，svccprop 将不会打印 SMF 模板定义属性。在未来发行版的缺省输出中可能会隐藏更多 SMF 基础结构属性。如果使用 -a，将显示所有属性，包括 SMF 模板定义属性。</p> <p>如果不带 -C、-c 或 -s 选项，svccprop 将访问有效属性。服务的有效属性是其直接附加属性。服务实例的有效属性是以下属性的联合：该实例正在运行的快照的复合视图中的属性，以及该实例直接附加属性的复合视图的非持久属性组中的属性。有关属性编写的说明，请参见 smf(5)。如果不存在正在运行的快照，则改用该实例的直接附加属性。</p>
输出格式	<p>缺省情况下，如果选定单个属性，每个属性的值将显示在单独的行中。空的 ASCII 字符串值用一对双引号(“”)表示。ASCII 字符串值中的 Bourne shell 元字符（包括 ';'、'&'、'('、')'、' '、'^'、'<'、'>'、换行符、空格、跳格、反斜杠、'"、单引号以及“”)通过反斜杠(\)引用。</p> <p>如果选定多个属性，每个属性将显示在一行中。每行包含一个属性定义符、属性类型以及属性值（如上所述）。这几项由空格分开。缺省情况下，如果提供了一个 <i>FMRI</i> 操作数，则属性定义符由属性组名称和属性名称中间加一个斜杠(/)组成。如果提供了多个 <i>FMRI</i> 操作数，则属性定义符是一个规范的 <i>FMRI</i>。</p> <p>如果访问控制禁止读取属性值并且 -p 选项没有明确指定属性或属性组，则属性的显示就像没有任何属性值一样。如果 -p 选项指定了一个或多个属性或属性组名称，并且访问控制禁止读取任何属性值，则会生成错误。</p> <p>标准错误流将显示错误消息。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -a 显示所有属性，包括 SMF 模板定义属性组中的属性。 -C 使用直接附加属性，无需编写。 -c 对于服务实例，使用其直接附加属性的复合视图。 -f 选择多属性输出格式，将完整的 <i>FMRI</i> 作为定义符。 -g <i>pgtype</i> 仅显示类型为 <i>pgtype</i> 的属性组中的选定属性。使用多个 -g 选项将显示多个属性组类型的属性。此选项意味着多属性输出格式 (-t)。

-l layer[,...]

仅显示在选定层中定义的属性。可用层包

括：`manifest`、`system-profile`、`site-profile`和`admin`。别名`all`可用于选择所有层。使用此选项时，不会显示属于非持久性属性组的属性。

-p name

对于操作数指定的每个服务或服务实例，选择`name`属性组中的所有属性。对于操作数指定的属性组，选择`name`属性。如果与`-g`一起使用，将选择所有匹配属性组类型的`name`属性。

-p pg/prop

对于操作数指定的每个服务或服务实例，选择`pg`属性组中的`prop`属性。

-q

静默。不产生输出。

-s name

对于服务实例，使用`name`快照的复合视图。

-t

选择多属性输出格式。

-v

详细模式。对不存在的属性输出错误消息，即使同时指定了`-q`选项。

-w

等到指定的属性组或包含指定属性的属性组在输出前更改。

该选项仅在指定了单个实体时有效。如果指定了多个操作数或者一个操作数与多个实例匹配，则会输出一条错误消息并且不采取任何操作。`-C`选项是隐含的。

操作数

支持下列操作数：

FMRI

服务、服务实例、属性组或属性的 FMRI。

可以通过指定实例名称或服务名称的结尾部分使用实例或服务的缩写形式。属性和属性组必须由完整的 FMRI 指定。例如，以下给定的 FMRI：

```
svc:/network/smtp:sendmail
```

以下是有效的缩写形式：

```
sendmail
:sendmail
smtp
smtp:sendmail
network/smtp
```

以下是无效的缩写形式：

```
mailnetwork
network/smt
```

FMRI 的缩写形式还未确定，因此不应在脚本或其他永久性工具中使用。如果一个缩写与多个实例匹配，每个实例都需要运行 `svccprop`。

pattern

一种通配模式，与系统信息库中的服务和实例的 FMRI 匹配。请参见 [fmatch\(5\)](#)。如果一个模式与多个服务或实例匹配，每个服务或实例都需要运行 `svccprop`。

示例

示例1 显示单个属性的值

以下示例显示了 `system/cron` 服务的 `default` 实例的 `restarter` 属性组中 `state` 属性的值。

```
example% svccprop -p restarter/state system/cron:default
online
```

示例2 检索是否启用了服务

是否启用服务由服务的 `-general/enabled` 属性决定。该属性可立即生效，因此必须使用 `-c` 选项。

```
example% svccprop -c -p general/enabled system/cron:default
true
```

示例3 显示属性组中的所有属性

以下实例显示了 Solaris 缺省安装时，`network/ntp` 服务的每个实例的 `general` 属性组中包含的所有属性：

```
example% svccprop -p general ntp
general/package astring SUNWntpr
general/enabled boolean true
general/entity_stability astring Uncommitted
general/single_instance boolean true
```

示例4 验证属性是否存在

以下示例验证了服务 `identity` 的所有实例的 `general/enabled` 属性是否存在。

```
example% svccprop -q -p general/enabled identity:
example% echo $?
0
```

示例5 等待属性更改

以下示例等待 `sendmail` 实例更改状态。

```
example% svccprop -w -p restarter/state sendmail
```

示例6 在脚本中检索布尔属性的值

以下示例在脚本中检索布尔属性的值：

示例6 在脚本中检索布尔属性的值 (续)

```
set -- 'svccprop -c -t -p general/enabled service'
code=$?
if [ $code -ne 0 ]; then
    echo "svccprop failed with exit code $code"
    return 1
fi
if [ $2 != boolean ]; then
    echo "general/enabled has unexpected type $2"
    return 2
fi
if [ $# -ne 3 ]; then
    echo "general/enabled has wrong number of values"
    return 3
fi
value=$3
...
```

示例7 在脚本中使用 svccprop

以下示例获取服务属性的值，并将其用于脚本 (/usr/bin/Xserver):

```
fmri=$1
prop=$2
if svccprop -q -p ${prop} ${fmri} ; then
    propval="$(svccprop -p ${prop} "${fmri}")"
    if [[ "${propval}" == "\"\"" ]] ; then
        propval=""
    fi
fi
```

示例8 按属性组类型过滤输出

以下示例获取 svc:/network/ssh:default 的方法:

```
example% svccprop -p exec -g method svc:/network/ssh:default
start/exec astring /lib/svc/method/sshd\ start
stop/exec astring :kill
refresh/exec astring /lib/svc/method/sshd\ restart
unconfigure/exec astring /lib/svc/method/sshd\ -u
```

示例9 显示通过管理方式定制的属性

以下命令使用 SMF 层显示通过管理方式定制的属性。

```
example% svccprop -p config -l admin svc:/network/dns/client
config/domain      astring      admin        my.domain.com
config/nameserver  net_address admin        10.22.33.44  10.44.33.11
```

退出状态 将返回以下退出值：

- 0 成功完成。
- 1 出现错误。
- 2 指定的命令行选项无效。

属性 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os

另请参见 [svcs\(1\)](#)、[inetd\(1M\)](#)、[svcadm\(1M\)](#)、[svccfg\(1M\)](#)、[svc.startd\(1M\)](#)、[service_bundle\(4\)](#)、[attributes\(5\)](#)

引用名	svcs – 报告服务状态
用法概要	<pre>svcs [-aHpv?] [-o col[,col]]... [-R FMRI-instance]... [-sS col]... [FMRI pattern]... svcs {-d -D} [-Hpv?] [-o col[,col]]... [-sS col]... [FMRI pattern] ... svcs -n [FMRI] ... svcs -l [-v] [FMRI pattern]... svcs -x [-v] [FMRI]...</pre>
描述	<p>svcs 命令显示有关服务配置系统信息库中记录的服务实例的信息。</p> <p>第一种命令形式输出由参数指定的服务实例的单行状态列表。每个实例只列出一 次。如果不指定任何参数，则会列出所有已启用的服务实例（即使被临时禁用），这 些服务实例将包含以下各列。</p> <p>第二种命令形式输出由参数指定的服务实例的依赖项或相关项的单行状态列表。</p> <p>第三种命令形式输出有关特定服务和实例的详细信息。</p> <p>第四种命令形式说明服务实例的状态。每个参数都对应一组可读的文本，用于说明服 务的状态以及服务处于这种状态的原因。如果不指定任何参数，将描述有问题的服 务。</p> <p>标准错误流将显示错误消息。</p> <p>该命令的输出可适当用作 svcadm(1M) 命令的输入。</p>
选项	<p>支持以下选项：</p> <ul style="list-style-type: none"> -? <ul style="list-style-type: none"> 显示一条扩展的用法消息，包含列说明符。 -a <ul style="list-style-type: none"> 显示所有服务，包括禁用的以及不完整的服务。可以使用 <code>svcs -x<service></code> 进一步 说明不完整的服务。 如果选定多个服务，则该选项无效。 -d <ul style="list-style-type: none"> 显示指定的服务实例所依赖的服务或服务实例。 -D <ul style="list-style-type: none"> 显示依赖于指定的服务或服务实例的服务实例。 -H <ul style="list-style-type: none"> 省略列标题。

-l

(字母 **ell**。) 显示有关选定的服务和实例的所有可用信息。每行显示一个服务属性。不同实例的信息由空行隔开。

以下的特定属性需要进行详细说明：

dependency

有关依赖项的信息。首先显示分组与 `restart_on` 属性，并用正斜杠 (/) 隔开。然后列出每个实体及其状态。有关状态的信息，请参见 [smf\(5\)](#)。除了标准状态，每个服务依赖项还可以具有以下状态描述：

absent

系统中未定义此类服务。

invalid

故障管理资源标识符 (Fault Management Resource Identifier, FMRI) 无效。请参见 [smf\(5\)](#)。

multiple

该实体是一个有多个实例的服务。

文件依赖项只能具有以下状态描述之一：

absent

系统中不存在此类文件。

online (联机)

该文件存在。

如果该文件在 `svc.startd` 上次评估服务的依赖项时不存在，该命令可能会认为依赖项不符合条件。`svcadm refresh` 将强制重新评估依赖项。

unknown (未知)

`stat(2)` 由于 `ENOENT` 之外的原因失败。

有关依赖项、分组以及 `restart_on` 值的其他详细信息，请参见 [smf\(5\)](#)。

enabled (已启用)

是否启用服务，以及是临时启用还是禁用服务（直到系统下一次重新引导）。前者指定为 `true` 或 `false`，后者根据是否存在 (`temporary`) 的情况进行指定。

服务可能会因管理员运行 `svcadm disable -t`、使用 `svcadm milestone` 或将系统引导到特定里程碑而被临时禁用。有关详细信息，请参见 [svcadm\(1M\)](#)。

-n

显示通知参数。请参见 [smf\(5\)](#)。无论选定的 FMRI 或模式如何，该选项都会显示 FMA 事件通知参数以及系统范围内的 SMF 状态转换通知参数。

-o *col[, col]...*

显示特定的列。每个 *col* 应为一个列名。有关可用列，请参见下文的 **列**。

- p**
列出每个服务实例关联的进程。服务实例也可能没有关联的进程。将显示每个进程的进程 ID、开始时间以及命令名称（对应 **ps(1)** 命令中的 PID、STIME 以及 CMD 字段）。
- R *FMRI-instance***
选择将给定的服务实例作为其重启程序的服务实例。
- s *col***
将输出按列排序。*col* 应为一个列名。有关可用列，请参见下文的**列**。多个 **-s** 选项的行为是累积的。
- S *col***
与 **-s** 选项一样按 *col* 的相反顺序排序。
- v**
如果不指定 **-x** 或 **-l** 选项，则显示详细的列：STATE、NSTATE、STIME、CTID 和 FMRI。
如果指定了 **-x** 选项，则显示每列说明的附加信息。
如果指定了 **-l** 选项，则显示**应用程序**类型的属性组中用户可见的属性及其描述。
- x**
显示对服务状态的说明。
如果不带参数，**-x** 选项将对以下服务的状态进行说明：
- 已经启用但未运行。
 - 阻止其他已经启用的服务运行。

操作数

支持下列操作数：

FMRI

故障管理资源标识符 (Fault Management Resource Identifier, FMRI)，用于指定一个或多个实例（请参见 **smf(5)**）。可以通过指定实例名称或服务名称的结尾部分使用 FMRI 的缩写形式。例如，以下给定的 FMRI：

```
svc:/network/smtp:sendmail
```

以下是有效的缩写形式：

```
sendmail
:sendmail
smtp
smtp:sendmail
network/smtp
```

以下是无效的缩写形式：

```
mail
network
network/smt
```

如果 FMRI 指定了一个服务，则该命令适用于此服务的所有实例，但与 `-D` 选项结合使用除外。

FMRI 的缩写形式还不稳定，因此不应在脚本或其他永久性工具中使用。

pattern

一种模式，根据 [fnmatch\(5\)](#) 中描述的“通配”规则，与服务实例的 FMRI 匹配。如果模式不以 `svc:` 开头，则前置 `svc:/`。以下是一个典型的通配模式示例：

```
qexample% svcs \*key serv\*
STATE          STIME          FMRI
disabled       Aug_02         svc:/network/rpc/key serv:default
```

FMRI-instance

用于指定实例的 FMRI。

列

列名不区分大小写。缺省的输出格式等效于 `"-o state,stime,fmri"`。缺省的排序列为 STATE、STIME 以及 FMRI。

CTID

服务实例的主合同 ID。并不是所有的服务实例都有有效的主合同 ID。

DESC

根据服务的模板元素对服务进行简单介绍。服务可能没有可用的描述，在这种情况下将使用连字符 (-) 表示空值。

FMRI

服务实例的 FMRI。

INST

服务实例的实例名称。

NSTA

服务实例下一状态的缩写，如 STA 列中的描述所示。连字符表示实例没有在转换。其他方面与 STA 相同。

NSTATE

服务的下一状态。连字符用于表示实例没有在转换。其他方面与 STATE 相同。

SCOPE

服务实例的作用域名称。

SVC

服务实例的服务名称。

STA

服务实例状态的缩写（请参见 [smf\(5\)](#)）：

DGD

降级

DIS
禁用

LRC
传统的 rc*.d 脚本启动式实例

MNT
maintenance

OFF
脱机

ON
联机

UN
未初始化

不存在的状态或无法识别的状态用问号 (?) 字符表示。除非同时显示了 NSTA 或 NSTATE 列，否则正在转换的实例后面将附加一个星号 (*)。

有关服务状态的说明，请参见 [smf\(5\)](#)。

STATE
服务实例的状态。除非同时显示了 NSTA 或 NSTATE 列，否则正在转换的实例后面将附加一个星号。

有关服务状态的说明，请参见 [smf\(5\)](#)。

STIME
如果服务实例在过去 24 小时内进入了当前状态，该列将显示服务实例进入当前状态的时间。否则，该列会显示服务实例进入当前状态的日期，并在空白的位置显示下划线(_)。

示例

示例1 显示缺省输出

以下示例显示缺省输出：

```
example% svcs
STATE      STIME      FMRI
...
legacy_run 13:25:04  lrc:/etc/rc3_d/S42myscript
...
online     13:21:50  svc:/system/svc/restarter:default
...
online     13:25:03  svc:/milestone/multi-user:default
...
online     13:25:07  svc:/milestone/multi-user-server:default
...
```

示例2 列出所有的本地实例

以下示例列出 `service1` 服务的所有本地实例。

```
example% svcs -o state,nstate,fmri service1
STATE      NSTATE      FMRI
online     -           svc:/service1:instance1
disabled   -           svc:/service1:instance2
```

示例3 列出详细信息

以下示例列出详细信息。

```
example% svcs -v network/rpc/rstat:udp
STATE      NSTATE      STIME      CTID      FMRI
online     -           Aug_09     -         svc:/network/rpc/rstat:udp
```

示例4 列出详细信息

以下示例列出有关 `system/service3` 的所有实例的详细信息。可以根据管理的重启程序相应显示其他字段。

```
example% svcs -l network/rpc/rstat:udp

fmri       svc:/network/rpc/rstat:udp
enabled    true
state      online
next_state none
restarter  svc:/network/inetd:default
contract_id
dependency require_all/error svc:/network/rpc/bind (online)
```

示例5 列出进程

```
example% svcs -p sendmail
STATE      STIME      FMRI
online     13:25:13  svc:/network/smtp:sendmail
           13:25:15  100939  sendmail
13:25:15   100940  sendmail
```

示例6 使用 `svcs -x` 说明服务状态

(a) 在以下示例中，`svcs -x` 标识了被禁用的 `print/server` 服务是两个已启用但尚未联机的服务的根源。`svcs -xv` 显示这两个服务是 `print/rfc1179` 和 `print/ipp-listener`。这种情况可通过启用 `print/server` 或禁用 `rfc1179` 和 `ipp-listener` 进行纠正。

```
example% svcs -x
svc:/application/print/server:default (LP print server)
  State: disabled since Mon Feb 13 17:56:21 2006
  Reason: Disabled by an administrator.
  See: http://support.oracle.com/msg/SMF-8000-05
```


示例 6 使用 `svcs -x` 说明服务状态 (续)

See: `lpsched(1M)`

Impact: 2 dependent services are not running. (Use `-v` for list.)

(b) 在以下示例中，NFS（远程文件系统）并未运行：

```
example$ svcs nfs/client
STATE          STIME      FMRI
offline        16:03:23  svc:/network/nfs/client:default
```

(c) 以下示例表明问题在于 `nfs/status`。 `nfs/client` 正处于等待状态，因为其依赖于 `nfs/nlockmgr`，而后者依赖于 `nfs/status`：

```
example$ svcs -xv nfs/client
svc:/network/nfs/client:default (NFS client)
  State: offline since Mon Feb 27 16:03:23 2006
  Reason: Service svc:/network/nfs/status:default
          is not running because a method failed repeatedly.
  See: http://support.oracle.com/msg/SMF-8000-GE
  Path: svc:/network/nfs/client:default
        svc:/network/nfs/nlockmgr:default
        svc:/network/nfs/status:default
  See: man -M /usr/share/man -s 1M mount_nfs
  See: /var/svc/log/network-nfs-client:default.log
  Impact: This service is not running.
```

退出状态

将返回以下退出值：

0

命令调用成功。

1

致命错误。

2

指定的命令行选项无效。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	<code>system/core-os</code>
接口稳定性	请参见下文。

屏幕输出为 `Uncommitted`（未确定）。调用为 `Committed`（已确定）。

另请参见

[ps\(1\)](#)、[svcprop\(1\)](#)、[svcadm\(1M\)](#)、[svccfg\(1M\)](#)、[svc.startd\(1M\)](#)、[stat\(2\)](#)、[libscf\(3LIB\)](#)、[attribut](#)

引用名	symorder – 重新排列符号列表
用法概要	symorder [-s] <i>objectfile symbolfile</i>
描述	<p>symorder 专门在 SunOS 4.x 中使用，以便削减获取 vmunix 中的符号所产生的系统开销。通过 /dev/ksyms 动态获取内核符号项时，该选项不再适用。</p> <p>提供该脚本是为了方便需要维护在不同操作系统中可移植的脚本的软件开发人员。</p>
退出状态	symorder 退出状态为 0。
属性	有关下列属性的说明，请参见 attributes(5) ：

属性类型	属性值
可用性	developer/base-developer-utilities

另请参见 [nlist\(3ELF\)](#)、[attributes\(5\)](#)、[ksyms\(7D\)](#)。

引用名 `sys-suspend` – 挂起或关闭系统，然后关闭电源

用法概要 `/usr/bin/sys-suspend [-fnxh][-d displayname]`

描述 `sys-suspend` 提供选项来挂起或关闭整个系统。

可挂起系统，以节省电源或使系统做好传输准备。对任意硬件执行重新配置或替换时不应使用挂起。

如果挂起，则通过保持打开内存电源（挂起到内存，Suspend to RAM）或通过将状态保存到非易失性存储（挂起到磁盘，Suspend to Disk）保留当先系统状态，直至通过打开电源或唤醒事件执行恢复操作。

在 Windows 环境中执行恢复时，系统会引入锁屏，以确保只有授权人员有权限访问系统。在非 windows 环境中，会提示用户输入口令。

设备或进程执行关键或时间敏感的操作（如实时操作）时，系统有可能挂起失败。发生这种情况时，系统会保持在其当前运行状态。报告失败的消息将显示在控制台或系统日志中。一旦系统成功挂起，恢复操作始终会成功阻拦诸如硬件重新配置之类的外部影响。

如果关机，则系统会像执行了 `poweroff(1M)` 那样进行响应。

该命令强制执行 `solaris.system.power.suspend` 授权。在缺省安装中，这些与控制台用户相关联。其他用户需要包括这些授权，或包括 "Suspend"（挂起）配置文件。

选项 支持下列操作数：

`-d displayname` 连接至 `displayname` 指定的 X 服务器。

`-f` 强制挂起。如果挂起失败，会导致 `poweroff(1M)` 发生。不保存系统状态，随后会产生标准引导。

`-h` 将缺省设置从挂起更改为关闭。

`-n` 不显示消息，也不要求用户介入。

`-x` 禁用 `lockscreen`。该标志禁止在恢复时执行 `lockscreen`。

属性 有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	system/kernel/power
接口稳定性	Committed（已确定）

另请参见 `halt(1M)`、`poweroff(1M)`、`shutdown(1M)`、`attributes(5)`、`cpr(7)`

引用名

sysV-make – maintain, update, and regenerate groups of programs

用法概要

```
/usr/lib/svr4.make [-f makefile] [-eiknpqrst] [names]
```

描述

This is the *vanilla* System V version of make. If the environment variable USE_SVR4_MAKE is set, then the command make will invoke this version of make. (See also the ENVIRONMENT section.)

make allows the programmer to maintain, update, and regenerate groups of computer programs. make executes commands in *makefile* to update one or more target *names* (*names* are typically programs). If the -f option is not present, then *makefile*, *Makefile*, and the Source Code Control System (SCCS) files *s.makefile* and *s.Makefile* are tried in order. If *makefile* is '-' the standard input is taken. More than one -f *makefile* argument pair may appear.

make updates a target only if its dependents are newer than the target. All prerequisite files of a target are added recursively to the list of targets. Missing files are deemed to be outdated.

The following list of four directives can be included in *makefile* to extend the options provided by make. They are used in *makefile* as if they were targets:

- .DEFAULT: If a file must be made but there are no explicit commands or relevant built-in rules, the commands associated with the name .DEFAULT are used if it exists.
- .IGNORE: Same effect as the -i option.
- .PRECIOUS: Dependents of the .PRECIOUS entry will not be removed when quit or interrupt are hit.
- .SILENT: Same effect as the -s option.

Options

The options for make are listed below:

- e Environment variables override assignments within makefiles.
- f *makefile* Description filename (*makefile* is assumed to be the name of a description file).
- i Ignore error codes returned by invoked commands.
- k Abandon work on the current entry if it fails, but continue on other branches that do not depend on that entry.
- n No execute mode. Print commands, but do not execute them. Even command lines beginning with an '@' are printed.
- p Print out the complete set of macro definitions and target descriptions.
- q Question. make returns a zero or non-zero status code depending on whether or not the target file has been updated.
- r Do not use the built-in rules.
- s Silent mode. Do not print command lines before executing.

-t Touch the target files (causing them to be updated) rather than issue the usual commands.

Creating the makefile The makefile invoked with the `-f` option is a carefully structured file of explicit instructions for updating and regenerating programs, and contains a sequence of entries that specify dependencies. The first line of an entry is a blank-separated, non-null list of targets, then a `' :`, then a (possibly null) list of prerequisite files or dependencies. Text following a `' ;'` and all following lines that begin with a tab are shell commands to be executed to update the target. The first non-empty line that does not begin with a tab or `' #'` begins a new dependency or macro definition. Shell commands may be continued across lines with a backslash-new-line (`\-NEWLINE`) sequence. Everything printed by `make` (except the initial TAB) is passed directly to the shell as is. Thus,

```
echo a\  
b
```

will produce

```
ab
```

exactly the same as the shell would.

Number-sign (`#`) and `NEWLINE` surround comments including contained `\-NEWLINE` sequences.

The following makefile says that `pgm` depends on two files `a.o` and `b.o`, and that they in turn depend on their corresponding source files (`a.c` and `b.c`) and a common file `incl.h`:

```
pgm: a.o b.o  
      cc a.o b.o -o pgm  
a.o: incl.h a.c  
      cc -c a.c  
b.o: incl.h b.c  
      cc -c b.c
```

Command lines are executed one at a time, each by its own shell. The `SHELL` environment variable can be used to specify which shell `make` should use to execute commands. The default is `/usr/bin/sh`. The first one or two characters in a command can be the following: `@`, `-`, `@-`, or `@@`. If `@` is present, printing of the command is suppressed. If `-` is present, `make` ignores an error. A line is printed when it is executed unless the `-s` option is present, or the entry `.SILENT` is included in *makefile*, or unless the initial character sequence contains a `@`. The `-n` option specifies printing without execution; however, if the command line has the string `$(MAKE)` in it, the line is always executed (see the discussion of the `MAKEFLAGS` macro in the `make Environment` sub-section below). The `-t` (`touch`) option updates the modified date of a file without executing any commands.

Commands returning non-zero status normally terminate `make`. If the `-i` option is present, if the entry `.IGNORE:` is included in *makefile*, or if the initial character sequence of the command contains `'-'`, the error is ignored. If the `-k` option is present, work is abandoned on the current entry, but continues on other branches that do not depend on that entry.

Interrupt and quit cause the target to be deleted unless the target is a dependent of the directive `.PRECIOUS`.

make Environment

The environment is read by `make`. All variables are assumed to be macro definitions and are processed as such. The environment variables are processed before any *makefile* and after the internal rules; thus, macro assignments in a *makefile* override environment variables. The `-e` option causes the environment to override the macro assignments in a *makefile*. Suffixes and their associated rules in the *makefile* will override any identical suffixes in the built-in rules.

The `MAKEFLAGS` environment variable is processed by `make` as containing any legal input option (except `-f` and `-p`) defined for the command line. Further, upon invocation, `make` “invents” the variable if it is not in the environment, puts the current options into it, and passes it on to invocations of commands. Thus, `MAKEFLAGS` always contains the current input options. This feature proves very useful for “super-makes”. In fact, as noted above, when the `-n` option is used, the command `$(MAKE)` is executed anyway; hence, one can perform a `make -n` recursively on a whole software system to see what would have been executed. This result is possible because the `-n` is put in `MAKEFLAGS` and passed to further invocations of `$(MAKE)`. This usage is one way of debugging all of the *makefiles* for a software project without actually doing anything.

Include Files

If the string *include* appears as the first seven letters of a line in a *makefile*, and is followed by a blank or a tab, the rest of the line is assumed to be a filename and will be read by the current invocation, after substituting for any macros.

Macros

Entries of the form *string1* = *string2* are macro definitions. *string2* is defined as all characters up to a comment character or an unescaped NEWLINE. Subsequent appearances of `$(string1[:subst1=[subst2]])` are replaced by *string2*. The parentheses are optional if a single-character macro name is used and there is no substitute sequence. The optional `:subst1=subst2` is a substitute sequence. If it is specified, all non-overlapping occurrences of *subst1* in the named macro are replaced by *subst2*. Strings (for the purposes of this type of substitution) are delimited by BLANKs, TABs, NEWLINE characters, and beginnings of lines. An example of the use of the substitute sequence is shown in the `Libraries` sub-section below.

Internal Macros

There are five internally maintained macros that are useful for writing rules for building targets.

`$(*)` The macro `$(*)` stands for the filename part of the current dependent with the suffix deleted. It is evaluated only for inference rules.

- `$$` The `$$` macro stands for the full target name of the current target. It is evaluated only for explicitly named dependencies.
- `$$<` The `$$<` macro is only evaluated for inference rules or the `.DEFAULT` rule. It is the module that is outdated with respect to the target (the “manufactured” dependent file name). Thus, in the `.c.o` rule, the `$$<` macro would evaluate to the `.c` file. An example for making optimized `.o` files from `.c` files is:
- ```
.c.o:
 cc c O $*.c

or:

.c.o:
 cc c O $$<
```
- `$$?` The `$$?` macro is evaluated when explicit rules from the makefile are evaluated. It is the list of prerequisites that are outdated with respect to the target, and essentially those modules that must be rebuilt.
- `$$%` The `$$%` macro is only evaluated when the target is an archive library member of the form `lib(file.o)`. In this case, `$$` evaluates to `lib` and `$$%` evaluates to the library member, `file.o`.

Four of the five macros can have alternative forms. When an upper case D or F is appended to any of the four macros, the meaning is changed to “directory part” for D and “file part” for F. Thus, `$$@D` refers to the directory part of the string `$$`. If there is no directory part, `./` is generated. The only macro excluded from this alternative form is `$$?`.

#### Suffixes

Certain names (for instance, those ending with `.o`) have inferable prerequisites such as `.c`, `.s`, etc. If no update commands for such a file appear in *makefile*, and if an inferable prerequisite exists, that prerequisite is compiled to make the target. In this case, `make` has inference rules that allow building files from other files by examining the suffixes and determining an appropriate inference rule to use. The current default inference rules are:

---

|                    |                      |                    |                    |                    |                    |                    |                    |                    |                    |
|--------------------|----------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|--------------------|
| <code>.c</code>    | <code>.c~</code>     | <code>.f</code>    | <code>.f~</code>   | <code>.s</code>    | <code>.s~</code>   | <code>.sh</code>   | <code>.sh~</code>  | <code>.C</code>    | <code>.C~</code>   |
| <code>.c.a</code>  | <code>.c.o</code>    | <code>.c~.a</code> | <code>.c~.c</code> | <code>.c~.o</code> | <code>.f.a</code>  | <code>.f.o</code>  | <code>.f~.a</code> | <code>.f~.f</code> | <code>.f~.o</code> |
| <code>.h~.h</code> | <code>.l.c</code>    | <code>.l.o</code>  | <code>.l~.c</code> | <code>.l~.l</code> | <code>.l~.o</code> | <code>.s.a</code>  | <code>.s.o</code>  | <code>.s~.a</code> | <code>.s~.o</code> |
| <code>.s~.s</code> | <code>.sh~.sh</code> | <code>.y.c</code>  | <code>.y.o</code>  | <code>.y~.c</code> | <code>.y~.o</code> | <code>.y~.y</code> | <code>.C.a</code>  | <code>.C.o</code>  | <code>.C~.a</code> |
| <code>.C~.C</code> | <code>.C~.o</code>   | <code>.L.C</code>  | <code>.L.o</code>  | <code>.L~.C</code> | <code>.L~.L</code> | <code>.L~.o</code> | <code>.Y.C</code>  | <code>.Y.o</code>  | <code>.Y~.C</code> |
| <code>.Y~.o</code> | <code>.Y~.Y</code>   |                    |                    |                    |                    |                    |                    |                    |                    |

---



The internal rules for `make` are contained in the source file `make.rules` for the `make` program. These rules can be locally modified. To print out the rules compiled into the `make` on any machine in a form suitable for re-compilation, the following command is used:

```
make -pf -2>/dev/null < /dev/null
```

A tilde in the above rules refers to an SCCS file (see [sccsfile\(4\)](#)). Thus, the rule `.c~.o` would transform an SCCS C source file into an object file (`.o`). Because the `s.` of the SCCS files is a prefix, it is incompatible with the `make` suffix point of view. Hence, the tilde is a way of changing any file reference into an SCCS file reference.

A rule with only one suffix (for example, `.c:`) is the definition of how to build `x` from `x.c`. In effect, the other suffix is null. This feature is useful for building targets from only one source file, for example, shell procedures and simple C programs.

Additional suffixes are given as the dependency list for `.SUFFIXES`. Order is significant: the first possible name for which both a file and a rule exist is inferred as a prerequisite. The default list is:

```
.SUFFIXES: .o .c .c~ .y .y~ .l .l~ .s .s~ .sh .sh~ .h .h~ .f .f~ .C .C~ .Y .Y~ .L
.L~
```

Here again, the above command for printing the internal rules will display the list of suffixes implemented on the current machine. Multiple suffix lists accumulate; `.SUFFIXES:` with no dependencies clears the list of suffixes.

#### Inference Rules

The first example can be done more briefly.

```
pgm: a.o b.o
 cc a.o b.o o pgm
a.o b.o: incl.h
```

This abbreviation is possible because `make` has a set of internal rules for building files. The user may add rules to this list by simply putting them in the *makefile*.

Certain macros are used by the default inference rules to permit the inclusion of optional matter in any resulting commands. Again, the previous method for examining the current rules is recommended.

The inference of prerequisites can be controlled. The rule to create a file with suffix `.o` from a file with suffix `.c` is specified as an entry with `.c.o:` as the target and no dependents. Shell commands associated with the target define the rule for making a `.o` file from a `.c` file. Any target that has no slashes in it and starts with a dot is identified as a rule and not a true target.

#### Libraries

If a target or dependency name contains parentheses, it is assumed to be an archive library, the string within parentheses referring to a member within the library. Thus, `lib(file.o)` and `$(LIB)(file.o)` both refer to an archive library that contains `file.o`. (This example assumes the `LIB` macro has been previously defined.) The expression `$(LIB)(file1.o file2.o)` is not

legal. Rules pertaining to archive libraries have the form `.XX.a` where the `XX` is the suffix from which the archive member is to be made. An unfortunate by-product of the current implementation requires the `XX` to be different from the suffix of the archive member. Thus, one cannot have `lib(file.o)` depend upon `file.o` explicitly. The most common use of the archive interface follows. Here, we assume the source files are all C type source:

```
lib: lib(file1.o) lib(file2.o) lib(file3.o)
 @echo lib is now up-to-date
.c.a:
 $(CC) -c $(CFLAGS) $<
 $(AR) $(ARFLAGS) $@ $*.o
 rm -f $*.o
```

In fact, the `.c.a` rule listed above is built into `make` and is unnecessary in this example. A more interesting, but more limited example of an archive library maintenance construction follows:

```
lib: lib(file1.o) lib(file2.o) lib(file3.o)
 $(CC) -c $(CFLAGS) $(?:.o=.c)
 $(AR) $(ARFLAGS) lib $?
 rm $?
 @echo lib is now up-to-date
.c.a:;
```

Here the substitution mode of the macro expansions is used. The  `$?`  list is defined to be the set of object filenames (inside `lib`) whose C source files are outdated. The substitution mode translates the `.o` to `.c`. (Unfortunately, one cannot as yet transform to `.c~`; however, this transformation may become possible in the future.) Also note the disabling of the `.c.a` rule, which would have created each object file, one by one. This particular construct speeds up archive library maintenance considerably. This type of construct becomes very cumbersome if the archive library contains a mix of assembly programs and C programs.

## 环境变量

**USE\_SVR4\_MAKE** If this environment variable is set, then the `make` command will invoke this System V version of `make`. If this variable is not set, then the default version of `make(1S)` is invoked.

`USE_SVR4_MAKE` can be set as follows (Bourne shell):

```
$ USE_SVR4_MAKE=""; export USE_SVR4_MAKE
```

or (C shell):

```
% setenv USE_SVR4_MAKE
```

## 文件

|                                             |                        |
|---------------------------------------------|------------------------|
| <code>[Mm]akefile</code>                    |                        |
| <code>s.[Mm]akefile</code>                  | default makefiles      |
| <code>/usr/bin/sh</code>                    | default shell for make |
| <code>/usr/share/lib/make/make.rules</code> | default rules for make |

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE      |
|----------------|----------------------|
| Availability   | developer/build/make |

## 另请参见

[cd\(1\)](#), [make\(1S\)](#), [sh\(1\)](#), [printf\(3C\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#)

## 附注

Some commands return non-zero status inappropriately; use `-i` or the `'-'` command line prefix to overcome the difficulty.

Filenames containing the characters `=`, `:`, and `@` do not work. Commands that are directly executed by the shell, notably [cd\(1\)](#), are ineffectual across NEWLINES in `make`. The syntax `lib(file1.o file2.o file3.o)` is illegal. You cannot build `lib(file.o)` from `file.o`.

**引用名**            tabs – set tabs on a terminal

**用法概要**            tabs [-n | *—file*  
                      [[-code] | -a | -a2 | -c | -c2 | -c3 | -f | -p | -s | -u]]  
                      q!! [+m [n]] [-T type]  
  
                      tabs [-T type] [+ m [n]] n1 [, n2 , ...]

**描述**                The tabs utility sets the tab stops on the user's terminal according to a tab specification, after clearing any previous settings. The user's terminal must have remotely settable hardware tabs.

**选项**                The following options are supported. If a given flag occurs more than once, the last value given takes effect:

-T *type*            tabs needs to know the type of terminal in order to set tabs and margins. *type* is a name listed in [term\(5\)](#). If no -T flag is supplied, tabs uses the value of the environment variable TERM. If the value of TERM is NULL or TERM is not defined in the environment (see [environ\(5\)](#)), tabs uses ansi+tabs as the terminal type to provide a sequence that will work for many terminals.

+m[*n*]             The margin argument may be used for some terminals. It causes all tabs to be moved over *n* columns by making column *n*+1 the left margin. If +m is given without a value of *n*, the value assumed is 10. For a TerminiNet, the first value in the tab list should be 1, or the margin will move even further to the right. The normal (leftmost) margin on most terminals is obtained by +m0. The margin for most terminals is reset only when the +m flag is given explicitly.

**Tab Specification**    Four types of tab specification are accepted. They are described below: canned, repetitive (-*n*), arbitrary (*n1,n2,...*), and file (*—file*).

If no tab specification is given, the default value is -8, that is, UNIX system “standard” tabs. The lowest column number is 1. Note: For tabs, column 1 always refers to the leftmost column on a terminal, even one whose column markers begin at 0, for example, the DASI 300, DASI 300s, and DASI 450.

**Canned -code**        Use one of the codes listed below to select a canned set of tabs. If more than one code is specified, the last code option will be used. The legal codes and their meanings are as follows:

-a            1, 10, 16, 36, 72 Assembler, IBM S/370, first format

-a2          1, 10, 16, 40, 72

                      Assembler, IBM S/370, second format

-c            1, 8, 12, 16, 20, 55

                      COBOL, normal format

-c2          1, 6, 10, 14, 49

COBOL compact format (columns 1-6 omitted). Using this code, the first typed character corresponds to card column 7, one space gets you to column 8, and a tab reaches column 12. Files using this tab setup should include a format specification as follows (see [fspec\(4\)](#)):

```
<:t-c2 m6 s66 d:>
```

-c3 1, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62, 67

COBOL compact format (columns 1-6 omitted), with more tabs than -c2. This is the recommended format for COBOL. The appropriate format specification is (see [fspec\(4\)](#)):

```
<:t-c3 m6 s66 d:>
```

-f 1, 7, 11, 15, 19, 23

FORTRAN

-p 1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61

PL/I

-s 1, 10, 55

SNOBOL

-u 1, 12, 20, 44

UNIVAC 1100 Assembler

*Repetitive*

-n A *repetitive* specification requests tabs at columns  $1+n$ ,  $1+2^*n$ , etc., where  $n$  is a single-digit decimal number. Of particular importance is the value 8: this represents the UNIX system "standard" tab setting, and is the most likely tab setting to be found at a terminal. When  $-0$  is used, the tab stops are cleared and no new ones are set.

*Arbitrary*

See OPERANDS.

*File*

-file If the name of a *file* is given, tabs reads the first line of the file, searching for a format specification (see [fspec\(4\)](#)). If it finds one there, it sets the tab stops according to it, otherwise it sets them as  $-8$ . This type of specification may be used to make sure that a tabbed file is printed with correct tab settings, and would be used with the `pr` command:

```
example% tabs -file; pr file
```

Tab and margin setting is performed via the standard output.

操作数

The following operand is supported:

*n1*[,*n2*, . . .] The *arbitrary* format consists of tab-stop values separated by commas or spaces. The tab-stop values must be positive decimal integers in ascending order. Up to 40 numbers are allowed. If any number (except the first one) is preceded by a plus sign, it is taken as an increment to be added to the previous value. Thus, the formats **1,10,20,30**, and **1,10,+10,+10** are considered identical.

示例

示例 1 Using the tabs command

The following command is an example using *-code* (*canned* specification) to set tabs to the settings required by the IBM assembler: columns 1, 10, 16, 36, 72:

```
example% tabs -a
```

The next command is an example of using *-n* (*repetitive* specification), where *n* is 8, causes tabs to be set every eighth position: 1+(1\*8), 1+(2\*8), . . . which evaluate to columns 9, 17, . . . :

```
example% tabs -8
```

This command uses *n1,n2,. . .* (*arbitrary* specification) to set tabs at columns 1, 8, and 36:

```
example% tabs 1,8,36
```

The last command is an example of using *-file* (*file* specification) to indicate that tabs should be set according to the first line of \$HOME/fspec.list/att4425 (see [fspec\(4\)](#)).

```
example% tabs -${HOME}/fspec.list/att4425
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of tabs: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

**TERM** Determine the terminal type. If this variable is unset or null, and if the *-T* option is not specified, terminal type `ansi+tabs` will be used.

退出状态

The following exit values are returned:

**0** Successful completion.

**>0** An error occurred.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | system/core-os  |
| CSI                 | Enabled         |
| Interface Stability | Committed       |

---

| ATTRIBUTE TYPE | ATTRIBUTE VALUE                    |
|----------------|------------------------------------|
| Standard       | See <a href="#">standards(5)</a> . |

**另请参见**

[expand\(1\)](#), [newform\(1\)](#), [pr\(1\)](#), [stty\(1\)](#), [tput\(1\)](#), [fspec\(4\)](#), [terminfo\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [term\(5\)](#), [standards\(5\)](#)

**附注**

There is no consistency among different terminals regarding ways of clearing tabs and setting the left margin.

`tabs` clears only 20 tabs (on terminals requiring a long sequence), but is willing to set 64.

The *tabspec* used with the `tabs` command is different from the one used with the `newform` command. For example, `tabs -8` sets every eighth position; whereas `newform -i-8` indicates that tabs are set every eighth position.

**引用名** tail – deliver the last part of a file

**用法概要**

```
/usr/bin/tail [\pm s number [lbcrl]] [file]
/usr/bin/tail [-lbcrl] [file]
/usr/bin/tail [\pm number [lbcfl]] [file]
/usr/bin/tail [-lbcfl] [file]
/usr/xpg4/bin/tail [-f | -r] [-c number | -n number] [file]
/usr/xpg4/bin/tail [\pm number [l | b | c] [f]] [file]
/usr/xpg4/bin/tail [\pm number [l] [f | r]] [file]
```

**描述**

The `tail` utility copies the named file to the standard output beginning at a designated place. If no file is named, the standard input is used.

Copying begins at a point in the file indicated by the `-c number`, `-n number`, or  `$\pm$  number` options (if `+ number` is specified, begins at distance number from the beginning; if `- number` is specified, from the end of the input; if `number` is NULL, the value 10 is assumed). `number` is counted in units of lines or byte according to the `-c` or `-n` options, or lines, blocks, or bytes, according to the appended option `l`, `b`, or `c`. When no units are specified, counting is by lines.

**选项**

The following options are supported for both `/usr/bin/tail` and `/usr/xpg4/bin/tail`. The `-r` and `-f` options are mutually exclusive. If both are specified on the command line, the `-f` option is ignored.

- b Units of blocks.
- c Units of bytes.
- f Follow. If the input-file is not a pipe, the program does not terminate after the line of the input-file has been copied, but enters an endless loop, wherein it sleeps for a second and then attempts to read and copy further records from the input-file. Thus it can be used to monitor the growth of a file that is being written by some other process.
- l Units of lines.
- r Reverse. Copies lines from the specified starting point in the file in reverse order. The default for `r` is to print the entire file in reverse order.

`/usr/xpg4/bin/tail`

The following options are supported for `/usr/xpg4/bin/tail` only:

- c *number* The *number* option-argument must be a decimal integer whose sign affects the location in the file, measured in bytes, to begin the copying:
  - + Copying starts relative to the beginning of the file.
  - Copying starts relative to the end of the file.
  - none Copying starts relative to the end of the file.



The origin for counting is 1; that is, `-c +1` represents the first byte of the file, `-c -1` the last.

`-n number` Equivalent to `-c number`, except the starting location in the file is measured in lines instead of bytes. The origin for counting is 1. That is, `-n +1` represents the first line of the file, `-n -1` the last.

## 操作数

The following operand is supported:

*file* A path name of an input file. If no *file* operands are specified, the standard input is used.

## 用法

See [largefile\(5\)](#) for the description of the behavior of `tail` when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

## 示例

示例 1 Using the tail Command

The following command prints the last ten lines of the file `fred`, followed by any lines that are appended to `fred` between the time `tail` is initiated and killed.

```
example% tail -f fred
```

The next command prints the last 15 bytes of the file `fred`, followed by any lines that are appended to `fred` between the time `tail` is initiated and killed:

```
example% tail -15cf fred
```

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `tail`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

## 退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/tail

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |
| CSI            | Enabled         |

/usr/xpg4/bin/tail

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | system/xopen/xcu4 |
| CSI            | Enabled           |

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见**

[cat\(1\)](#), [head\(1\)](#), [more\(1\)](#), [pg\(1\)](#), [dd\(1M\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

**附注**

Piped tails relative to the end of the file are stored in a buffer, and thus are limited in length. Various kinds of anomalous behavior can happen with character special files.

引用名            talk – talk to another user

用法概要        talk *address* [*terminal*]

描述             The talk utility is a two-way, screen-oriented communication program.

When first invoked, talk sends a message similar to:

```
Message from TalkDaemon@ her_machine at time...
talk: connection requested by your_address
talk: respond with: talk your_address
```

to the specified *address*. At this point, the recipient of the message can reply by typing:

```
talk your_address
```

Once communication is established, the two parties can type simultaneously, with their output displayed in separate regions of the screen. Characters are processed as follows:

- Typing the alert character will alert the recipient's terminal.
- Typing Control-L will cause the sender's screen regions to be refreshed.
- Typing the erase and kill characters will affect the sender's terminal in the manner described by the [termios\(3C\)](#) interface.
- Typing the interrupt or end-of-file (EOF) characters will terminate the local talk utility. Once the talk session has been terminated on one side, the other side of the talk session will be notified that the talk session has been terminated and will be able to do nothing except exit.
- Typing characters from LC\_CTYPE classifications print or space will cause those characters to be sent to the recipient's terminal.
- When and only when the stty iexten local mode is enabled, additional special control characters and multi-byte or single-byte characters are processed as printable characters if their wide character equivalents are printable.
- Typing other non-printable characters will cause them to be written to the recipient's terminal as follows: control characters will appear as a caret ( ^ ) followed by the appropriate ASCII character, and characters with the high-order bit set will appear in “meta” notation. For example, '\003' is displayed as '^C' and '\372' as '^M-z'.

Permission to be a recipient of a talk message can be denied or granted by use of the [mesg\(1\)](#) utility. However, a user's privilege may further constrain the domain of accessibility of other users' terminals. Certain commands, such as [pr\(1\)](#), disallow messages in order to prevent interference with their output. talk will fail when the user lacks the appropriate privileges to perform the requested action.

Certain block-mode terminals do not have all the capabilities necessary to support the simultaneous exchange of messages required for talk. When this type of exchange cannot be

supported on such terminals, the implementation may support an exchange with reduced levels of simultaneous interaction or it may report an error describing the terminal-related deficiency.

## 操作数

The following operands are supported:

*address* The recipient of the talk session. One form of *address* is the *username*, as returned by the [who\(1\)](#) utility. If you wish to talk to someone on your own machine, then *username* is just the person's login name. If you wish to talk to a user on another host, then *username* is one of the following forms:

```
host!user
host.user
host:user
user@host
```

although *user@host* is perhaps preferred.

*terminal* If the recipient is logged in more than once, *terminal* can be used to indicate the appropriate terminal name. If *terminal* is not specified, the talk message will be displayed on one or more accessible terminals in use by the recipient. The format of *terminal* will be the same as that returned by [who](#).

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of talk: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

TERM Determine the name of the invoker's terminal type. If this variable is unset or null, an unspecified terminal type will be used.

## 退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred, or talk was invoked on a terminal incapable of supporting it.

## 文件

/etc/hosts host name database

/var/adm/utmpx user and accounting information for talk

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | service/network/network-servers    |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见**

[mail\(1\)](#), [mesg\(1\)](#), [pr\(1\)](#), [stty\(1\)](#), [who\(1\)](#), [write\(1\)](#), [talkd\(1M\)](#), [termios\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

**附注**

Typing Control-L redraws the screen, while the erase, kill, and word kill characters will work in `talk` as normal. To exit, type an interrupt character. `talk` then moves the cursor to the bottom of the screen and restores the terminal to its previous state.

**引用名** tar – create tape archives and add or extract files

**用法概要**

```
tar c[BDFHijlnopTVwzZ@[0-7]][bf][X...] [blocksize]
 [tarfile] [size] [exclude-file]...
 {file | -I include-file | -C directory file}...

tar r[BDFHijlnTVwzZ@[0-7]][bf] [blocksize] [tarfile]
 [size]
 {file | -I include-file | -C directory file}...

tar t[BFHijlnTvzZ[0-7]][f][X...] [tarfile] [size]
 [exclude-file]... {file | -I include-file}...

tar u[BDFHijlnTVwzZ@[0-7]][bf] [blocksize] [tarfile]
 [size] file...

tar x[BFhilmnjopTVwzZ@[0-7]][f][X...] [tarfile] [size]
 [exclude-file]... [file]...
```

**描述**

The `tar` command archives and extracts files to and from a single file called a *tarfile*. A tarfile is usually a magnetic tape, but it can be any file. `tar`'s actions are controlled by the *key* argument. The *key* is a string of characters containing exactly one function letter (`c`, `r`, `t`, `u`, or `x`) and zero or more function modifiers (letters or digits), depending on the function letter used. The *key* string contains no SPACE characters. Function modifier arguments are listed on the command line in the same order as their corresponding function modifiers appear in the *key* string.

The `-I include-file`, `-C directory file`, and *file* arguments specify which files or directories are to be archived or extracted. In all cases, appearance of a directory name refers to the files and (recursively) subdirectories of that directory. Arguments appearing within braces (`{ }`) indicate that one of the arguments must be specified.

**操作数** The following operands are supported:

`-C directory file`

Performs a `chdir` (see [cd\(1\)](#)) operation on *directory* and performs the `c` (create) or `r` (replace) operation on *file*. Use short relative path names for *file*. If *file* is `."`, archive all files in *directory*. This operand enables archiving files from multiple directories not related by a close common parent.

`-I include-file`

Opens *include-file* containing a list of files, one per line, and treats it as if each file appeared separately on the command line. Be careful of trailing white spaces. Also beware of leading white spaces, since, for each line in the included file, the entire line (apart from the newline) is used to match against the initial string of files to include. In the case where excluded files (see `X` function modifier) are also specified, they take precedence over all included files. If a file is specified in both the *exclude-file* and the *include-file* (or on the command line), it is excluded.

*file*

A path name of a regular file or directory to be archived (when the *c*, *r* or *u* functions are specified), extracted (*x*) or listed (*t*). When *file* is the path name of a directory, the action applies to all of the files and (recursively) subdirectories of that directory.

When a file is archived, and the *E* flag (see *Function Modifiers*) is not specified, the filename cannot exceed 256 characters. In addition, it must be possible to split the name between parent directory names so that the prefix is no longer than 155 characters and the name is no longer than 100 characters. If *E* is specified, a name of up to *PATH\_MAX* characters can be specified.

For example, a file whose basename is longer than 100 characters could not be archived without using the *E* flag. A file whose directory portion is 200 characters and whose basename is 50 characters could be archived (without using *E*) if a slash appears in the directory name somewhere in character positions 151-156.

## Function Letters

The function portion of the key is specified by one of the following letters:

*c*

Create. Writing begins at the beginning of the tarfile, instead of at the end.

*r*

Replace. The named *files* are written at the end of the tarfile. A file created with extended headers must be updated with extended headers (see *E* flag under *Function Modifiers*). A file created without extended headers cannot be modified with extended headers.

*t*

Table of Contents. The names of the specified files are listed each time they occur in the tarfile. If no *file* argument is specified, the names of all files and any associated extended attributes in the tarfile are listed. With the *v* function modifier, additional information for the specified files is displayed.

*u*

Update. The named *files* are written at the end of the tarfile if they are not already in the tarfile, or if they have been modified since last written to that tarfile. An update can be rather slow. A tarfile created on a 5.x system cannot be updated on a 4.x system. A file created with extended headers must be updated with extended headers (see *E* flag under *Function Modifiers*). A file created without extended headers cannot be modified with extended headers.

*x*

Extract or restore. The named *files* are extracted from the tarfile and written to the directory specified in the tarfile, relative to the current directory. Use the relative path names of files and directories to be extracted.

Absolute path names contained in the tar archive are unpacked using the absolute path names, that is, the leading forward slash (*/*) is *not* stripped off.

By default, absolute pathnames (those that begin with a / character) have the leading slash removed, therefore extracting those files and directories relative to the current directory.

If a named file matches a directory whose contents has been written to the tarfile, this directory is recursively extracted. The owner, modification time, and mode are restored, if possible. Otherwise, to restore owner, you must be the super-user. Character-special and block-special devices (created by `mknod(1M)`) can only be extracted by the super-user. If no *file* argument is specified, the entire content of the tarfile is extracted. If the tarfile contains several files with the same name, each file is written to the appropriate directory, overwriting the previous one. Filename substitution wildcards cannot be used for extracting files from the archive. Rather, use a command of the form:

```
tar xvf ... /dev/rmt/0` tar tf ... /dev/rmt/0 | \
 grep 'pattern'`
```

When extracting tapes created with the *r* or *u* functions, directory modification times can not be set correctly. These same functions cannot be used with many tape drives due to tape drive limitations such as the absence of backspace or append capabilities.

When using the *r*, *u*, or *x* functions or the *X* function modifier, the named files must match exactly the corresponding files in the *tarfile*. For example, to extract *./thisfile*, you must specify *./thisfile*, and not *thisfile*. The *t* function displays how each file was archived.

#### Function Modifiers

The characters below can be used in conjunction with the letter that selects the desired function.

##### **b** *blocksize*

Blocking Factor. Use when reading or writing to raw magnetic archives (see *f* below). The *blocksize* argument specifies the number of 512-byte tape blocks to be included in each read or write operation performed on the tarfile. The minimum is 1, the default is 20. The maximum value is a function of the amount of memory available and the blocking requirements of the specific tape device involved (see *mtio(7I)* for details.) The maximum cannot exceed `INT_MAX/512` (4194303).

When a tape archive is being read, its actual blocking factor is automatically detected, provided that it is less than or equal to the nominal blocking factor (the value of the *blocksize* argument, or the default value if the *b* modifier is not specified). If the actual blocking factor is greater than the nominal blocking factor, a read error results. See Example 5 in EXAMPLES.

##### **B**

Block. Force *tar* to perform multiple reads (if necessary) to read exactly enough bytes to fill a block. This function modifier enables *tar* to work across the Ethernet, since pipes and sockets return partial blocks even when more data is coming. When reading from standard input, “-”, this function modifier is selected by default to ensure that *tar* can recover from short reads.



## D

Data change warnings. Used with *c*, *r*, or *u* function letters. Ignored with *t* or *x* function letters. If the size of a file changes while the file is being archived, treat this condition as a warning instead of as an error. A warning message is still written, but the exit status is not affected.

## E

Write a tarfile with extended headers. (Used with *c*, *r*, or *u* function letters. Ignored with *t* or *x* function letters.) When a tarfile is written with extended headers, the modification time is maintained with a granularity of microseconds rather than seconds. In addition, filenames no longer than `PATH_MAX` characters that could not be archived without *E*, and file sizes greater than 8GB, are supported. The *E* flag is required whenever the larger files and/or files with longer names, or whose UID/GID exceed 2097151, are to be archived, or if time granularity of microseconds is desired.

## f

File. Use the *tarfile* argument as the name of the tarfile. If *f* is specified, `/etc/default/tar` is not searched. If *f* is omitted, *tar* uses the device indicated by the `TAPE` environment variable, if set. Otherwise, *tar* uses the default values defined in `/etc/default/tar`. The number matching the `archiveN` string is used as the output device with the blocking and size specifications from the file. For example,

```
tar -c 2/tmp/*
```

writes the output to the device specified as `archive2` in `/etc/default/tar`.

If the name of the tarfile is “-”, *tar* writes to the standard output or reads from the standard input, whichever is appropriate. *tar* can be used as the head or tail of a pipeline. *tar* can also be used to move hierarchies with the command:

```
example% cd fromdir; tar cf - . | (cd todir; tar xfbp -)
```

## F

With one *F* argument, *tar* excludes all directories named `SCCS` and `RCS` from the tarfile. With two arguments, *FF*, *tar* excludes all directories named `SCCS` and `RCS`, all files with `.o` as their suffix, and all files named `errs`, `core`, and `a.out`.

## h

Follow symbolic links as if they were normal files or directories. Normally, *tar* does not follow symbolic links.

## i

Ignore directory checksum errors.

## j

*c* mode only. Compress the resulting archive with `bzip2`. In extract or list modes, this option is ignored. The implementation recognizes `bzip2` compression type automatically when reading archives. Upgrade/replace first decompresses and then applies the same mechanism to compress the archive automatically.

l

Link. Output error message if unable to resolve all links to the files being archived. If `l` is not specified, no error messages are printed.

m

Modify. The modification time of the file is the time of extraction. This function modifier is valid only with the `x` function.

n

The file being read is a non-tape device. Reading of the archive is faster since `tar` can randomly seek around the archive.

o

Ownership. Assign to extracted files the user and group identifiers of the user running the program, rather than those on tarfile. This is the default behavior for users other than root. If the `o` function modifier is not set and the user is root, the extracted files takes on the group and user identifiers of the files on tarfile (see [chown\(1\)](#) for more information). The `o` function modifier is only valid with the `x` function.

p

Restore the named files to their original modes, and ACLs if applicable, ignoring the present [umask\(1\)](#). This is the default behavior if invoked as super-user with the `x` function letter specified. If super-user, SETUID, and sticky information are also extracted, and files are restored with their original owners and permissions, rather than owned by root. When this function modifier is used with the `c` function, ACLs are created in the tarfile along with other information. Errors occur when a tarfile with ACLs is extracted by previous versions of `tar`.

P

For archive creation, suppress the addition of a trailing `/` on directory entries in the archive.

For archive extraction, preserve pathnames. By default, absolute pathnames (those that begin with a `/` character) have the leading slash removed when extracting archives. Also, `tar` refuses to extract archive entries whose pathnames contain a dot-dot (`..`).

This option suppresses these behaviors.

T

This modifier is only available if the system is configured with Trusted Extensions.

When this modifier is used with the function letter `c`, `r`, or `u` for creating, replacing or updating a tarfile, the sensitivity label associated with each archived file and directory is stored in the tarfile.

Specifying `T` implies the function modifier `p`.

When used with the function letter `x` for extracting a tarfile, the `tar` program verifies that the file's sensitivity label specified in the archive equals the sensitivity label of the destination directory. If not, the file is not restored. This operation must be invoked from

the global zone. If the archived file has a relative pathname, it is restored to the corresponding directory with the same label, if available. This is done by prepending to the current destination directory the root pathname of the zone whose label equals the file. If no such zone exists, the file is not restored.

Limited support is provided for extracting labeled archives from Trusted Solaris 8. Only sensitivity labels, and multi-level directory specifications are interpreted. Privilege specifications and audit attribute flags are silently ignored. Multilevel directory specifications including symbolic links to single level directories are mapped into zone-relative pathnames if a zone with the same label is available. This support is intended to facilitate migration of home directories. Architectural differences preclude the extraction of arbitrarily labeled files from Trusted Solaris 8 into identical pathnames in Trusted Extensions. Files cannot be extracted unless their archived label matches the destination label.

v

Verbose. Output the name of each file preceded by the function letter. With the `t` function, `v` provides additional information about the tarfile entries. The listing is similar to the format produced by the `-l` option of the `ls(1)` command.

w

What. Output the action to be taken and the name of the file, then await the user's confirmation. If the response is affirmative, the action is performed; otherwise, the action is not performed. This function modifier cannot be used with the `t` function.

X

Exclude. Use the *exclude-file* argument as a file containing a list of relative path names for files (or directories) to be excluded from the tarfile when using the functions `c`, `x`, or `t`. Be careful of trailing white spaces. Also beware of leading white spaces, since, for each line in the excluded file, the entire line (apart from the newline) is used to match against the initial string of files to exclude. Lines in the exclude file are matched exactly, so an entry like `/var` does *not* exclude the `/var` directory if `tar` is backing up relative pathnames. The entry should read `./var` under these circumstances. The `tar` command does not expand shell metacharacters in the exclude file, so specifying entries like `*.o` does not have the effect of excluding all files with names suffixed with `.o`. If a complex list of files is to be excluded, the exclude file should be generated by some means such as the `find(1)` command with appropriate conditions.

Multiple `X` arguments can be used, with one *exclude-file* per argument. In the case where included files (see `-I include-file` operand) are also specified, the excluded files take precedence over all included files. If a file is specified in both the *exclude-file* and the *include-file* (or on the command line), it is excluded.

z

`c` mode only. Compress the resulting archive with `gzip`. In extract or list mode, this option is ignored. The implementation recognizes `gzip` compression type automatically when

reading archives. Upgrade/replace first decompresses and then applies the same mechanism to compress the archive automatically.

Z

c mode only. Compress the resulting archive with `compress`. See [compress\(1\)](#). In extract or list modes, this option is ignored. The implementation recognizes `compress` compression type automatically when reading archives. Upgrade/replace first decompresses and then applies the same mechanism to compress the archive automatically.

@

Include extended attributes in archive. By default, `tar` does not place extended attributes in the archive. With this flag, `tar` looks for extended attributes on the files to be placed in the archive and add them to the archive. Extended attributes go in the archive as special files with a special type label. When this modifier is used with the `x` function, extended attributes are extracted from the tape along with the normal file data. Extended attribute files can only be extracted from an archive as part of a normal file extract. Attempts to explicitly extract attribute records are ignored.

/

Include extended system attributes in archive. By default, `tar` does not place extended system attributes in the archive. With this flag, `tar` looks for extended system attributes on the files to be placed in the archive and adds them to the archive. Extended system attributes go in the archive as special files with a special type label. When this modifier is used with the `x` function, extended system attributes are extracted from the tape along with the normal file data. Extended system attribute files can only be extracted from an archive as part of a normal file extract. Attempts to explicitly extract attribute records are ignored.

[0-7]

Select an alternative drive on which the tape is mounted. The default entries are specified in `/etc/default/tar`. If no `digit` or `f` function modifier is specified, the entry in `/etc/default/tar` with digit “0” is the default.

## 用法

See [largefile\(5\)](#) for the description of the behavior of `tar` when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

The automatic determination of the actual blocking factor can be fooled when reading from a pipe or a socket (see the `B` function modifier below).

1/4” streaming tape has an inherent blocking factor of one 512-byte block. It can be read or written using any blocking factor.

This function modifier works for archives on disk files and block special devices, among others, but is intended principally for tape devices.

For information on `tar` header format, see [archives.h\(3HEAD\)](#).

## 示例

### 示例 1 Creating an Archive of Your Home Directory

The following is an example using `tar` to create an archive of your home directory on a tape mounted on drive `/dev/rmt/0`:

```
example% cd
example% tar cvf /dev/rmt/0 .
messages from tar
```

The `c` function letter means create the archive. The `v` function modifier outputs messages explaining what `tar` is doing. The `f` function modifier indicates that the tarfile is being specified (`/dev/rmt/0` in this example). The dot (`.`) at the end of the command line indicates the current directory and is the argument of the `f` function modifier.

Display the table of contents of the tarfile with the following command:

```
example% tar tvf /dev/rmt/0
```

The output is similar to the following for the POSIX locale:

```
rw-r--r-- 1677/40 2123 Nov 7 18:15 1985 ./test.c
...
example%
```

The columns have the following meanings:

- column 1 is the access permissions to `./test.c`
- column 2 is the *user-id/group-id* of `./test.c`
- column 3 is the size of `./test.c` in bytes
- column 4 is the modification date of `./test.c`. When the `LC_TIME` category is not set to the POSIX locale, a different format and date order field can be used.
- column 5 is the name of `./test.c`

To extract files from the archive:

```
example% tar xvf /dev/rmt/0
messages from tar
example%
```

If there are multiple archive files on a tape, each is separated from the following one by an EOF marker. To have `tar` read the first and second archives from a tape with multiple archives on it, the *non-rewinding* version of the tape device name must be used with the `f` function modifier, as follows:

```
example% tar xvfp /dev/rmt/0n read first archive from tape
messages from tar
example% tar xvfp /dev/rmt/0n read second archive from tape
messages from tar
example%
```

### 示例 1 Creating an Archive of Your Home Directory (续)

Notice that in some earlier releases, the above scenario did not work correctly, and intervention with `mt(1)` between `tar` invocations was necessary. To emulate the old behavior, use the non-rewind device name containing the letter `b` for BSD behavior. See the `Close Operations` section of the `mtio(7I)` manual page.

### 示例 2 Archiving Files from /usr/include and from /etc to Default Tape Drive 0

To archive files from `/usr/include` and from `/etc` to default tape drive 0:

```
example% tar c -C /usr include -C /etc .
```

The table of contents from the resulting tarfile would produce output like the following:

```
include/
include/a.out.h
and all the other files in /usr/include ...
./chown and all the other files in /etc
```

To extract all files in the `include` directory:

```
example% tar xv include
x include/, 0 bytes, 0 tape blocks \
 and all files under include ...
```

### 示例 3 Transferring Files Across the Network

The following is an example using `tar` to transfer files across the network. First, here is how to archive files from the local machine (`example`) to a tape on a remote system (`host`):

```
example% tar cvfb - 20 files | \
 ssh host dd of=/dev/rmt/0 obs=20b
messages from tar
example%
```

In the example above, we are *creating a tarfile* with the `c` key letter, asking for *verbose* output from `tar` with the `v` function modifier, specifying the name of the output *tarfile* using the `f` function modifier (the standard output is where the *tarfile* appears, as indicated by the `'-'` sign), and specifying the blocksize (`20`) with the `b` function modifier. If you want to change the blocksize, you must change the blocksize arguments both on the `tar` command *and* on the `dd` command.

### 示例 4 Retrieving Files from a Tape on the Remote System Back to the Local System

The following is an example that uses `tar` to retrieve files from a tape on the remote system back to the local system:

```
example% ssh -n host dd if=/dev/rmt/0 bs=20b | \
 tar xvBfb - 20 files
```

#### 示例 4 Retrieving Files from a Tape on the Remote System Back to the Local System (续)

```
messages from tar
example%
```

In the example above, we are *extracting* from the *tarfile* with the *x* key letter, asking for *verbose output from tar* with the *v* function modifier, telling *tar* it is reading from a pipe with the *B* function modifier, specifying the name of the input *tarfile* using the *f* function modifier (the standard input is where the *tarfile* appears, as indicated by the “-” sign), and specifying the blocksize (20) with the *b* function modifier.

#### 示例 5 Creating an Archive of the Home Directory

The following example creates an archive of the home directory on `/dev/rmt/0` with an actual blocking factor of 19:

```
example% tar cvfb /dev/rmt/0 19 $HOME
```

To recognize this archive's actual blocking factor without using the *b* function modifier:

```
example% tar tvf /dev/rmt/0
tar: blocksize = 19
...
```

To recognize this archive's actual blocking factor using a larger nominal blocking factor:

```
example% tar tvf /dev/rmt/0 30
tar: blocksize = 19
...
```

Attempt to recognize this archive's actual blocking factor using a nominal blocking factor that is too small:

```
example% tar tvf /dev/rmt/0 10
tar: tape read error
```

#### 示例 6 Creating Compressed Archives

The following example creates a compressed archive using *bzip*:

```
example% tar cjf tarfile /tmp/*
```

The compressed file name is `tarfile.bz2`

The same compressed archive would be created in this case if the following sequence of commands had been used instead:

```
example% tar cf tarfile /tmp/*
example% bzip2 tarfile
```

however, the creation and removal of the intermediate file is eliminated. The function modifiers *z* and *Z* behave similarly, but use *gzip* and *compress*, respectively.

### 示例 6 Creating Compressed Archives (续)

The following example creates a compressed archive using compress:

```
example% tar czf tarfile /tmp/*
```

The compressed file name is tarfile.Z.

The following example creates a compressed archive using gzip:

```
example% tar czf tarfile /tmp/*
```

The compressed file name is tarfile.gz.

### 示例 7 Extracting Files from a Compressed Archive

The following examples extract files from a compressed archive: For archives compressed using bzip2 compression mode:

```
example% tar xvf tarfile.bz2
example% tar xvfj tarfile.bz2
example% bzcata tarfile.bz2 | tar xvf -
```

For archives compressed using compress compression mode:

```
example% tar xvf tarfile.Z
example% tar xvfZ tarfile.Z
example% zcatal tarfile.Z | tar xvf -
```

For archives compressed using gzip compression mode:

```
example% tar xvf tarfile.gz
example% tar xvfz tarfile.gz
example% gzcatal tarfile.gz | tar xvf -
```

## 环境变量

### TMPDIR

Creates a temporary file in /tmp by default. Otherwise, tar uses the directory specified by TMPDIR.

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of tar: LC\_COLLATE, LC\_CTYPE, LC\_MESSAGES, LC\_TIME, TZ, and NLSPATH.

Affirmative responses are processed using the extended regular expression defined for the yesexpr keyword in the LC\_MESSAGES category of the user's locale. The locale specified in the LC\_COLLATE category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for yesexpr. The locale specified in LC\_CTYPE determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the yesexpr. See [locale\(5\)](#).



退出状态 The following exit values are returned:

0  
Successful completion.

>0  
An error occurred.

文件

- /dev/rmt/[0-7][b][n]
- /dev/rmt/[0-7]l[b][n]
- /dev/rmt/[0-7]m[b][n]
- /dev/rmt/[0-7]h[b][n]
- /dev/rmt/[0-7]u[b][n]
- /dev/rmt/[0-7]c[b][n]
- /etc/default/tar

Setting for /etc/default/tar might look like the following:

```
archive0=/dev/rmt/0
archive1=/dev/rmt/0n
archive2=/dev/rmt/1
archive3=/dev/rmt/1n
archive4=/dev/rmt/0
archive5=/dev/rmt/0n
archive6=/dev/rmt/1
archive7=/dev/rmt/1n
```

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | system/core-os  |
| CSI                 | Enabled         |
| Interface Stability | Committed       |

另请参见 [ar\(1\)](#), [basename\(1\)](#), [cd\(1\)](#), [chown\(1\)](#), [compress\(1\)](#), [cpio\(1\)](#), [csh\(1\)](#), [dirname\(1\)](#), [find\(1\)](#), [ls\(1\)](#), [mt\(1\)](#), [pax\(1\)](#), [setfacl\(1\)](#), [umask\(1\)](#), [mknod\(1M\)](#), [archives.h\(3HEAD\)](#), [attributes\(5\)](#), [environ\(5\)](#), [fsattr\(5\)](#), [largefile\(5\)](#), [mtio\(7I\)](#)

诊断 Diagnostic messages are output for bad key characters and tape read/write errors, and for insufficient memory to hold the link tables.

附注 There is no way to access the *n*-th occurrence of a file.

Tape errors are handled ungracefully.

The tar archive format allows UIDs and GIDs up to 2097151 to be stored in the archive header. Files with UIDs and GIDs greater than this value is archived with the UID and GID of 60001.

If an archive is created that contains files whose names were created by processes running in multiple locales, a single locale that uses a full 8-bit codeset (for example, the en\_US locale) should be used both to create the archive and to extract files from the archive.

Neither the r function letter nor the u function letter can be used with quarter-inch archive tapes, since these tape drives cannot backspace.

Since tar has no options, the standard “—” argument that is normally used in other utilities to terminate recognition of options is not needed. If used, it is recognized only as the first argument and is ignored.

Since `-C directory file` and `-I include-file` are multi-argument operands, any of the following methods can be used to archive or extract a file named `-C` or `-I`:

1. Specify them using file operands containing a / character on the command line (such as `/home/joe/-C` or `./-I`).
2. Include them in an include file with `-I include-file`.
3. Specify the directory in which the file resides:  
`-C directory -C`  
or  
`-C directory -I`
4. Specify the entire directory in which the file resides:  
`-C directory .`

**引用名**           tbl – format tables for nroff or troff

**用法概要**       tbl [-me] [-mm] [-ms] [*filename*]...

**描述**           tbl is a preprocessor for formatting tables for [nroff\(1\)](#) or [troff\(1\)](#). The input *filenames* are copied to the standard output, except that lines between .TS and .TE command lines are assumed to describe tables and are reformatted.

If no arguments are given, tbl reads the standard input, so tbl may be used as a filter. When tbl is used with [eqn\(1\)](#) or [neqn](#), the tbl command should be first, to minimize the volume of data passed through pipes.

**选项**

- me     Copy the -me macro package to the front of the output file.
- mm     Copy the -mm macro package to the front of the output file.
- ms     Copy the -ms macro package to the front of the output file.

**示例**           示例 1 Using tbl

As an example, letting '@' (at-sign) represent a TAB, which should be typed as an actual TAB character in the input file

```
.TS
c s s
c c s
c c c
l n n.
Household Population
Town@Households
@Number@Size
Bedminster@789@3.26
Bernards Twp.@3087@3.74
Bernardsville@2018@3.30
Bound Brook@3425@3.04
Branchburg@1644@3.49
.TE
```

yields

| Household Population |        |            |      |
|----------------------|--------|------------|------|
| Town                 | Number | Households | Size |
| Bedminster           | 789    | 3.26       |      |
| Bernards Twp.        | 3087   | 3.74       |      |

## 示例1 Using tbl (续)

|               |      |      |
|---------------|------|------|
| Bernardsville | 2018 | 3.30 |
| Bound Brook   | 3425 | 3.04 |
| Branchburg    | 1644 | 3.49 |

## 文件

/usr/share/lib/tmac/e -me macros  
/usr/share/lib/tmac/m -mm macros  
/usr/share/lib/tmac/s -ms macros

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | text/doctools   |

## 另请参见

[eqn\(1\)](#), [nroff\(1\)](#), [troff\(1\)](#), [attributes\(5\)](#)

**引用名**            tcopy – copy a magnetic tape

**用法概要**        tcopy *source* [*destination*]

**描述**            The tcopy utility copies the magnetic tape mounted on the tape drive specified by the *source* argument. The only assumption made about the contents of a tape is that there are two tape marks at the end.

When only a source drive is specified, tcopy scans the tape, and displays information about the sizes of records and tape files. If a destination is specified, tcopy makes a copies the source tape onto the *destination* tape, with blocking preserved. As it copies, tcopy produces the same output as it does when only scanning a tape.

The tcopy utility requires the use of Berkeley-compatible device names. For example,

```
example% tcopy /dev/rmt/1b /dev/rmt/2b
```

**属性**            See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

**另请参见**        [mt\(1\)](#), [ioctl\(2\)](#), [attributes\(5\)](#)

**附注**            tcopy will only run on systems supporting an associated set of [ioctl\(2\)](#) requests.

**引用名** tee – replicate the standard output

**用法概要** /usr/bin/tee [-ai] [*file*...]

**描述** tee copies standard input to standard output and to zero or more files. The options determine whether the specified files are overwritten or appended to. The tee utility does not buffer output. If a write to a file fails, tee continues to write to other files although it exits with a non-zero exit status.

The number of *file* operands that can be specified is limited by the underlying operating system.

**选项** The following options are supported:

-a Appends the output to the files rather than overwriting them.

-i Ignores interrupts.

**操作数** The following operands are supported:

*file* A path name of an output file.

**用法** See [largefile\(5\)](#) for the description of the behavior of tee when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

**环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of tee: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

**退出状态** The following exit values are returned:

0 Successful completion. The standard input was successfully copied to all output files.

>0 An error occurred. The number of files that could not be opened or whose status could not be obtained.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见** [cat\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | telnet – user interface to a remote system using the TELNET protocol                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 用法概要 | <pre>telnet [-8EFKLacdfrx] [-X atype] [-e escape_char]       [-k realm] [-l user] [-n file]       [ [ [!] @hop1 [@hop2... ] @] host [port]]</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 描述   | <p>The <code>telnet</code> utility communicates with another host using the TELNET protocol. If <code>telnet</code> is invoked without arguments, it enters command mode, indicated by its prompt, <code>telnet&gt;</code>. In this mode, it accepts and executes its associated commands. See <code>USAGE</code>. If it is invoked with arguments, it performs an open command with those arguments.</p> <p>If, for example, a <i>host</i> is specified as <code>@hop1@hop2@host</code>, the connection goes through hosts <i>hop1</i> and <i>hop2</i>, using loose source routing to end at <i>host</i>. If a leading <code>!</code> is used, the connection follows strict source routing. Notice that when <code>telnet</code> uses IPv6, it can only use loose source routing, and the connection ignores the <code>!</code>.</p> <p>Once a connection has been opened, <code>telnet</code> enters input mode. In this mode, text typed is sent to the remote host. The input mode entered will be either “line mode”, “character at a time”, or “old line by line”, depending upon what the remote system supports.</p> <p>In “line mode”, character processing is done on the local system, under the control of the remote system. When input editing or character echoing is to be disabled, the remote system will relay that information. The remote system will also relay changes to any special characters that happen on the remote system, so that they can take effect on the local system.</p> <p>In “character at a time” mode, most text typed is immediately sent to the remote host for processing.</p> <p>In “old line by line” mode, all text is echoed locally, and (normally) only completed lines are sent to the remote host. The “local echo character” (initially <code>^E</code>) may be used to turn off and on the local echo. (Use this mostly to enter passwords without the password being echoed.)</p> <p>If the “line mode” option is enabled, or if the <code>localchars</code> toggle is <code>TRUE</code> (the default in “old line by line” mode), the user's <code>quit</code>, <code>intr</code>, and <code>flush</code> characters are trapped locally, and sent as TELNET protocol sequences to the remote side. If “line mode” has ever been enabled, then the user's <code>susp</code> and <code>eof</code> are also sent as TELNET protocol sequences. <code>quit</code> is then sent as a TELNET <code>ABORT</code> instead of <code>BREAK</code>. The options <code>toggle autoflush</code> and <code>toggle autosynch</code> cause this action to flush subsequent output to the terminal (until the remote host acknowledges the TELNET sequence); and to flush previous terminal input, in the case of <code>quit</code> and <code>intr</code>.</p> <p>While connected to a remote host, the user can enter <code>telnet</code> command mode by typing the <code>telnet</code> escape character (initially <code>^]</code>). When in command mode, the normal terminal editing conventions are available. Pressing <code>RETURN</code> at the <code>telnet</code> command prompt causes <code>telnet</code> to exit command mode.</p> |

**选项**

The following options are supported:

- 8  
Specifies an 8-bit data path. Negotiating the TELNET BINARY option is attempted for both input and output.
- a  
Attempts automatic login. This sends the user name by means of the USER variable of the ENVIRON option, if supported by the remote system. The name used is that of the current user as returned by `getlogin(3C)` if it agrees with the current user ID. Otherwise, it is the name associated with the user ID.
- c  
Disables the reading of the user's `telnetrc` file. (See the `toggle skiprc` command on this reference page.)
- d  
Sets the initial value of the debug toggle to TRUE.
- e *escape\_char*  
Sets the initial escape character to *escape\_char*. *escape\_char* may also be a two character sequence consisting of ^ (Control key) followed by one character. If the second character is ?, the DEL character is selected. Otherwise, the second character is converted to a control character and used as the escape character. If *escape\_char* is defined as the null string (that is, -e ''), this is equivalent to -e '^@' (Control-@). To specify that no character can be the escape character, use the -E option.
- E  
Stops any character from being recognized as an escape character.
- f  
Forwards a copy of the local credentials to the remote system.
- F  
Forwards a forwardable copy of the local credentials to the remote system.
- k *realm*  
If Kerberos authentication is being used, requests that `telnet` obtain tickets for the remote host in *realm* instead of the remote host's default realm as determined in `krb5.conf(4)`.
- K  
Specifies no automatic login to the remote system.
- l *user*  
When connecting to a remote system that understands the ENVIRON option, then *user* will be sent to the remote system as the value for the ENVIRON variable USER.
- L  
Specifies an 8-bit data path on output. This causes the BINARY option to be negotiated on output.



- n *tracefile*  
Opens *tracefile* for recording trace information. See the set *tracefile* command below.
- r  
Specifies a user interface similar to `rlogin`. In this mode, the escape character is set to the tilde (~) character, unless modified by the -e option. The `rlogin` escape character is only recognized when it is preceded by a carriage return. In this mode, the `telnet` escape character, normally '^]', must still precede a `telnet` command. The `rlogin` escape character can also be followed by '\r' or '^Z', and, like `rlogin(1)`, closes or suspends the connection, respectively. This option is an uncommitted interface and may change in the future.
- x  
Turns on encryption of the data stream. When this option is turned on, `telnet` will exit with an error if authentication cannot be negotiated or if encryption cannot be turned on.
- X *atype*  
Disables the *atype* type of authentication.

## 用法

### telnet Commands

The commands described in this section are available with `telnet`. It is necessary to type only enough of each command to uniquely identify it. (This is also true for arguments to the `mode`, `set`, `toggle`, `unset`, `environ`, and `display` commands.)

#### auth *argument* ...

The `auth` command manipulates the information sent through the TELNET AUTHENTICATE option. Valid arguments for the `auth` command are as follows:

##### disable *type*

Disables the specified type of authentication. To obtain a list of available types, use the `auth disable ?` command.

##### enable *type*

Enables the specified type of authentication. To obtain a list of available types, use the `auth enable ?` command.

##### status

Lists the current status of the various types of authentication.

#### open [-l *user*] [[!] @*hop1* [@*hop2* ...]@*host* [*port* ]

Open a connection to the named host. If no port number is specified, `telnet` will attempt to contact a TELNET server at the default port. The host specification may be either a host name (see `hosts(4)`) or an Internet address specified in the “dot notation” (see `inet(7P)` or `inet6(7P)`). If the *host* is specified as @*hop1*@*hop2*@*host*, the connection goes through hosts *hop1* and *hop2*, using loose source routing to end at *host*. The @ symbol is required as a separator between the hosts specified. If a leading ! is used with IPv4, the connection follows strict source routing.

The `-l` option passes the *user* as the value of the ENVIRON variable USER to the remote system.

#### `close`

Close any open TELNET session. An EOF (in command mode) will also close a session and exit.

#### `encrypt`

The `encrypt` command manipulates the information sent through the TELNET ENCRYPT option.

Valid arguments for the `encrypt` command are as follows:

##### `disable type [input|output]`

Disables the specified type of encryption. If you omit the input and output, both input and output are disabled. To obtain a list of available types, use the `encrypt disable ?` command.

##### `enable type [input|output]`

Enables the specified type of encryption. If you omit input and output, both input and output are enabled. To obtain a list of available types, use the `encrypt enable ?` command.

##### `input`

This is the same as the `encrypt start input` command.

##### `-input`

This is the same as the `encrypt stop input` command.

##### `output`

This is the same as the `encrypt start output` command.

##### `-output`

This is the same as the `encrypt stop output` command.

##### `start [input|output]`

Attempts to start encryption. If you omit input and output, both input and output are enabled. To obtain a list of available types, use the `encrypt enable ?` command.

##### `status`

Lists the current status of encryption.

##### `stop [input|output]`

Stops encryption. If you omit input and output, encryption is on both input and output.

##### `type type`

Sets the default type of encryption to be used with later `encrypt start` or `encrypt stop` commands.

#### `quit`

Same as `close`.

z

Suspend `telnet`. This command only works when the user is using a shell that supports job control, such as `sh(1)`.

*mode type*

The remote host is asked for permission to go into the requested mode. If the remote host is capable of entering that mode, the requested mode will be entered. The argument *type* is one of the following:

*character*

Disable the TELNET LINEMODE option, or, if the remote side does not understand the LINEMODE option, then enter “character at a time” mode.

*line*

Enable the TELNET LINEMODE option, or, if the remote side does not understand the LINEMODE option, then attempt to enter “old-line-by-line” mode.

*isig (-isig)*

Attempt to enable (disable) the TRAPSIG mode of the LINEMODE option. This requires that the LINEMODE option be enabled.

*edit (-edit)*

Attempt to enable (disable) the EDIT mode of the LINEMODE option. This requires that the LINEMODE option be enabled.

*softtabs (-softtabs)*

Attempt to enable (disable) the SOFT\_TAB mode of the LINEMODE option. This requires that the LINEMODE option be enabled.

*litecho (-litecho)*

Attempt to enable (disable) the LIT\_ECHO mode of the LINEMODE option. This requires that the LINEMODE option be enabled.

*?*

Prints out help information for the mode command.

*status*

Show the current status of `telnet`. This includes the peer one is connected to, as well as the current mode.

*display*

[*argument . . .*] Display all, or some, of the set and toggle values (see `toggle argument. . .`).

*?*

[*command*] Get help. With no arguments, `telnet` prints a help summary. If a command is specified, `telnet` will print the help information for just that command.

*send argument . . .*

Send one or more special character sequences to the remote host. The following are the arguments that can be specified (more than one argument may be specified at a time):

**escape**

Send the current telnet escape character (initially ^]).

**synch**

Send the TELNET SYNCH sequence. This sequence discards all previously typed, but not yet read, input on the remote system. This sequence is sent as TCP urgent data and may not work if the remote system is a 4.2 BSD system. If it does not work, a lowercase "r" may be echoed on the terminal.

**brk or break**

Send the TELNET BRK (Break) sequence, which may have significance to the remote system.

**ip**

Send the TELNET IP (Interrupt Process) sequence, which aborts the currently running process on the remote system.

**abort**

Send the TELNET ABORT (Abort Process) sequence.

**ao**

Send the TELNET AO (Abort Output) sequence, which flushes all output from the remote system to the user's terminal.

**ayt**

Send the TELNET AYT (Are You There) sequence, to which the remote system may or may not respond.

**ec**

Send the TELNET EC (Erase Character) sequence, which erases the last character entered.

**el**

Send the TELNET EL (Erase Line) sequence, which should cause the remote system to erase the line currently being entered.

**eof**

Send the TELNET EOF (End Of File) sequence.

**eor**

Send the TELNET EOR (End Of Record) sequence.

**ga**

Send the TELNET GA (Go Ahead) sequence, which probably has no significance for the remote system.

**getstatus**

If the remote side supports the TELNET STATUS command, getstatus will send the subnegotiation to request that the server send its current option status.

**nop**

Send the TELNET NOP (No Operation) sequence.

`susp`

Send the TELNET SUSP (Suspend Process) sequence.

`do option`

`dont option`

`will option`

`wont option`

Send the TELNET protocol option negotiation indicated. Option may be the text name of the protocol option, or the number corresponding to the option. The command will be silently ignored if the option negotiation indicated is not valid in the current state. If the *option* is given as `help` or `?`, the list of option names known is listed. This command is mostly useful for unusual debugging situations.

`?`

Print out help information for the send command.

`set argument [value]`

`unset argument`

Set any one of a number of `telnet` variables to a specific value. The special value `off` turns off the function associated with the variable. The values of variables may be interrogated with the `display` command. If *value* is omitted, the value is taken to be true, or “on”. If the `unset` form is used, the value is taken to be false, or `off`. The variables that may be specified are:

`echo`

This is the value (initially `^E`) that, when in “line by line” mode, toggles between local echoing of entered characters for normal processing, and suppressing echoing of entered characters, for example, entering a password.

`escape`

This is the `telnet` escape character (initially `^]`) that enters `telnet` command mode when connected to a remote system.

`interrupt`

If `telnet` is in `localchars` mode (see `toggle`, `localchars`) and the `interrupt` character is typed, a TELNET IP sequence (see `send` and `ip`) is sent to the remote host. The initial value for the `interrupt` character is taken to be the terminal's `intr` character.

`quit`

If `telnet` is in `localchars` mode and the `quit` character is typed, a TELNET BRK sequence (see `send`, `brk`) is sent to the remote host. The initial value for the `quit` character is taken to be the terminal's `quit` character.

`flushoutput`

If `telnet` is in `localchars` mode and the `flushoutput` character is typed, a TELNET AO sequence (see `send`, `ao`) is sent to the remote host. The initial value for the `flush` character is taken to be the terminal's `flush` character.

**erase**

If `telnet` is in `localchars` mode *and* operating in “character at a time” mode, then when the `erase` character is typed, a TELNET EC sequence (see `send, ec`) is sent to the remote system. The initial value for the `erase` character is taken to be the terminal's `erase` character.

**kill**

If `telnet` is in `localchars` mode *and* operating in “character at a time” mode, then when the `kill` character is typed, a TELNET EL sequence (see `send, el`) is sent to the remote system. The initial value for the `kill` character is taken to be the terminal's `kill` character.

**eof**

If `telnet` is operating in “line by line”/ mode, entering the `eof` character as the first character on a line sends this character to the remote system. The initial value of `eof` is taken to be the terminal's `eof` character.

**ayt**

If `telnet` is in `localchars` mode, or `LINEMODE` is enabled, and the status character is typed, a TELNET AYT (“Are You There”) sequence is sent to the remote host. (See `send, ayt` above.) The initial value for `ayt` is the terminal's status character.

**forw1****forw2**

If `telnet` is operating in `LINEMODE`, and the `forw1` or `forw2` characters are typed, this causes the forwarding of partial lines to the remote system. The initial values for the forwarding characters come from the terminal's `eo1` and `eo2` characters.

**lnext**

If `telnet` is operating in `LINEMODE` or “old line by line” mode, then the `lnext` character is assumed to be the terminal's `lnext` character. The initial value for the `lnext` character is taken to be the terminal's `lnext` character.

**reprint**

If `telnet` is operating in `LINEMODE` or “old line by line” mode, then the `reprint` character is assumed to be the terminal's `reprint` character. The initial value for `reprint` is taken to be the terminal's `reprint` character.

**rlogin**

This is the `rlogin` escape character. If set, the normal `telnet` escape character is ignored, unless it is preceded by this character at the beginning of a line. The `rlogin` character, at the beginning of a line followed by a “.” closes the connection. When followed by a ^Z, the `rlogin` command suspends the `telnet` command. The initial state is to disable the `rlogin` escape character.

**start**

If the `TELNET TOGGLE-FLOW-CONTROL` option has been enabled, then the `start` character is taken to be the terminal's `start` character. The initial value for the `kill` character is taken to be the terminal's `start` character.

**stop**

If the TELNET TOGGLE-*FLOW-CONTROL* option has been enabled, then the *stop* character is taken to be the terminal's *stop* character. The initial value for the *kill* character is taken to be the terminal's *stop* character.

**susp**

If *telnet* is in *localchars* mode, or *LINEMODE* is enabled, and the suspend character is typed, a TELNET SUSP sequence (see *send*, *susp* above) is sent to the remote host. The initial value for the suspend character is taken to be the terminal's suspend character.

**tracefile**

This is the file to which the output, generated when the *netdata* or the *debug* option is *TRUE*, will be written. If *tracefile* is set to "-", then tracing information will be written to standard output (the default).

**worderase**

If *telnet* is operating in *LINEMODE* or "old line by line" mode, then this character is taken to be the terminal's *worderase* character. The initial value for the *worderase* character is taken to be the terminal's *worderase* character.

**?**

Displays the legal set and unset commands.

***s\c state***

The *s\c* (Set Local Characters) command is used to set or change the state of special characters when the TELNET *LINEMODE* option has been enabled. Special characters are characters that get mapped to TELNET commands sequences (like *ip* or *quit*) or line editing characters (like *erase* and *kill*). By default, the local special characters are exported. The following values for *state* are valid:

**check**

Verifies the settings for the current special characters. The remote side is requested to send all the current special character settings. If there are any discrepancies with the local side, the local settings will switch to the remote values.

**export**

Switches to the local defaults for the special characters. The local default characters are those of the local terminal at the time when *telnet* was started.

**import**

Switches to the remote defaults for the special characters. The remote default characters are those of the remote system at the time when the TELNET connection was established.

**?**

Prints out help information for the *s\c* command.

***toggle argument...***

Toggle between *TRUE* and *FALSE* the various flags that control how *telnet* responds to events. More than one argument may be specified. The state of these flags may be interrogated with the *display* command. Valid arguments are:

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| authdebug   | Turns on debugging information for the authentication code.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| autodecrypt | When the TELNET ENCRYPT option is negotiated, by default the actual encryption (decryption) of the data stream does not start automatically. The autoencrypt (autodecrypt) command states that encryption of the output (input) stream should be enabled as soon as possible.                                                                                                                                                                                                                                   |
| autologin   | If the remote side supports the TELNET AUTHENTICATION option, telnet attempts to use it to perform automatic authentication. If the AUTHENTICATION option is not supported, the user's login name is propagated through the TELNET ENVIRON option. This command is the same as specifying the -a option on the open command.                                                                                                                                                                                    |
| autoflush   | If autoflush and localchars are both TRUE, then when the ao, intr, or quit characters are recognized (and transformed into TELNET sequences; see set for details), telnet refuses to display any data on the user's terminal until the remote system acknowledges (using a TELNET Timing Mark option) that it has processed those TELNET sequences. The initial value for this toggle is TRUE if the terminal user has not done an "stty noflsh". Otherwise, the value is FALSE (see <a href="#">stty(1)</a> ). |
| autosynch   | If autosynch and localchars are both TRUE, then when either the interrupt or quit characters are typed (see set for descriptions of interrupt and quit), the resulting TELNET sequence sent is followed by the TELNET SYNCH sequence. This procedure <i>should</i> cause the remote system to begin throwing away all previously typed input until both of the TELNET sequences have been read and acted upon. The initial value of this toggle is FALSE.                                                       |
| binary      | Enable or disable the TELNET BINARY option on both input and output.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| inbinary    | Enable or disable the TELNET BINARY option on input.                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| outbinary   | Enable or disable the TELNET BINARY option on output.                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| crlf        | Determines how carriage returns are sent. If the value is TRUE, then carriage returns will be sent as <CR><LF>. If the value is FALSE, then carriage returns will be sent as <CR><NUL>. The initial value for this toggle is FALSE.                                                                                                                                                                                                                                                                             |
| crmod       | Toggle RETURN mode. When this mode is enabled, most RETURN characters received from the remote host will be mapped into a RETURN followed by a line feed. This mode does not affect those characters typed by the user, only those received from the remote                                                                                                                                                                                                                                                     |



---

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | host. This mode is useful only for remote hosts that send RETURN but never send LINEFEED. The initial value for this toggle is FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| debug           | Toggle socket level debugging (only available to the super-user). The initial value for this toggle is FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| encdebug        | Turns on debugging information for the encryption code.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| localchars      | If this toggle is TRUE, then the flush, interrupt, quit, erase, and kill characters (see set) are recognized locally, and transformed into appropriate TELNET control sequences, respectively ao, ip, brk, ec, and eI (see send). The initial value for this toggle is TRUE in “line by line” mode, and FALSE in “character at a time” mode. When the LINEMODE option is enabled, the value of localchars is ignored, and assumed always to be TRUE. If LINEMODE has ever been enabled, then quit is sent as abort, and eof and suspend are sent as eof and susp (see send above). |
| netdata         | Toggle the display of all network data (in hexadecimal format). The initial value for this toggle is FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| options         | Toggle the display of some internal TELNET protocol processing (having to do with telnet options). The initial value for this toggle is FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| prettydump      | When the netdata toggle is enabled, if prettydump is enabled, the output from the netdata command will be formatted in a more user readable format. Spaces are put between each character in the output. The beginning of any TELNET escape sequence is preceded by an asterisk (*) to aid in locating them.                                                                                                                                                                                                                                                                       |
| skiprc          | When the skiprc toggle is TRUE, TELNET skips the reading of the .telnetrc file in the user's home directory when connections are opened. The initial value for this toggle is FALSE.                                                                                                                                                                                                                                                                                                                                                                                               |
| termdata        | Toggles the display of all terminal data (in hexadecimal format). The initial value for this toggle is FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| verbose_encrypt | When the verbose_encrypt flag is TRUE, TELNET prints out a message each time encryption is enabled or disabled. The initial value for this toggle is FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| ?               | Display the legal toggle commands.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

#### *environ argument . . .*

The `environ` command is used to manipulate variables that may be sent through the TELNET ENVIRON option. The initial set of variables is taken from the users environment. Only the DISPLAY and PRINTER variables are exported by default. Valid arguments for the `environ` command are:

**define *variable value***

Define *variable* to have a value of *value*. Any variables defined by this command are automatically exported. The *value* may be enclosed in single or double quotes, so that tabs and spaces may be included.

**undefine *variable***

Remove *variable* from the list of environment variables.

**export *variable***

Mark the *variable* to be exported to the remote side.

**unexport *variable***

Mark the *variable* to not be exported unless explicitly requested by the remote side.

**list**

List the current set of environment variables. Those marked with an asterisk (\*) will be sent automatically. Other variables will be sent only if explicitly requested.

**?**

Prints out help information for the `envi ron` command.

**logout**

Sends the telnet logout option to the remote side. This command is similar to a `close` command. However, if the remote side does not support the `logout` option, nothing happens. If, however, the remote side does support the `logout` option, this command should cause the remote side to close the TELNET connection. If the remote side also supports the concept of suspending a user's session for later reattachment, the `logout` argument indicates that the remote side should terminate the session immediately.

**文件**

`$HOME/.telnetrc` file that contains commands to be executed before initiating a telnet session

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | network/telnet  |

**另请参见**

[rlogin\(1\)](#), [sh\(1\)](#), [stty\(1\)](#), [getlogin\(3C\)](#), [hosts\(4\)](#), [krb5.conf\(4\)](#), [nologin\(4\)](#), [telnetrc\(4\)](#), [attributes\(5\)](#), [inet\(7P\)](#), [inet6\(7P\)](#)

**诊断**

NO LOGINS: System going down in *N* minutes

The machine is in the process of being shut down and logins have been disabled.

**附注**

On some remote systems, `echo` has to be turned off manually when in “line by line” mode.

In “old line by line” mode, or LINEMODE, the terminal's EOF character is only recognized (and sent to the remote system) when it is the first character on a line.

The telnet protocol only uses single DES for session protection—clients request service tickets with single DES session keys. The KDC must know that host service principals that offer the telnet service support single DES, which, in practice, means that such principals must have single DES keys in the KDC database.

引用名            test – evaluate condition(s)

用法概要        /usr/bin/test [*condition*]

[ [*condition*] ]

sh                test [*condition*]

[ [*condition*] ]

cs                test [*condition*]

[ [*condition*] ]

ksh88            test [*condition*]

[ [*condition*] ]

ksh               test [*condition*]

[ [*condition*] ]

描述             The test utility evaluates the *condition* and indicates the result of the evaluation by its exit status. An exit status of zero indicates that the condition evaluated as true and an exit status of 1 indicates that the condition evaluated as false.

In the first form of the utility shown using the SYNOPSIS:

```
test [condition]
```

the square brackets denote that *condition* is an optional operand and are not to be entered on the command line.

In the second form of the utility shown using the SYNOPSIS:

```
[[condition]]
```

the first open square bracket, [, is the required utility name. *condition* is optional, as denoted by the inner pair of square brackets. The final close square bracket, ], is a required operand.

See [largefile\(5\)](#) for the description of the behavior of test when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

The test and [ utilities evaluate the condition *condition* and, if its value is true, set exit status to 0. Otherwise, a non-zero (false) exit status is set. test and [ also set a non-zero exit status if there are no arguments. When permissions are tested, the effective user ID of the process is used.

All operators, flags, and brackets (brackets used as shown in the last SYNOPSIS line) must be separate arguments to these commands. Normally these arguments are separated by spaces.

## 操作数

The primaries listed below with two elements of the form:

*-primary\_operator primary\_operand*

are known as *unary primaries*. The primaries with three elements in either of the two forms:

*primary\_operand -primary\_operator primary\_operand*

*primary\_operand primary\_operator primary\_operand*

are known as *binary primaries*.

If any file operands except for *-h* and *-L* primaries refer to symbolic links, the symbolic link is expanded and the test is performed on the resulting file.

If you test a file you own (the *-r* *-w* or *-x* tests), but the permission tested does not have the *owner* bit set, a non-zero (false) exit status is returned even though the file can have the *group* or *other* bit set for that permission.

The *=* and *!=* primaries have a higher precedence than the unary primaries. The *=* and *!=* primaries always expect arguments; therefore, *=* and *!=* cannot be used as an argument to the unary primaries.

The following primaries can be used to construct *condition*:

- a file* True if *file* exists. (Not available in sh.)
- b file* True if *file* exists and is a block special file.
- c file* True if *file* exists and is a character special file.
- d file* True if *file* exists and is a directory.
- e file* True if *file* exists. (Not available in sh.)
- f file* True if *file* exists and is a regular file. Alternatively, if */usr/bin/sh* users specify */usr/ucb* before */usr/bin* in their *PATH* environment variable, then test returns true if *file* exists and is (not-a-directory). The *csh* test and *[* built-ins always use this alternative behavior.
- g file* True if *file* exists and its set group ID flag is set.
- G file* True if *file* exists and its group matches the effective group ID of this process. (Not available in sh.)
- h file* True if *file* exists and is a symbolic link.
- k file* True if *file* exists and has its sticky bit set.
- L file* True if *file* exists and is a symbolic link.
- n string* True if the length of *string* is non-zero.

|                                   |                                                                                                                                                                                                                                           |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-o option</code>            | True if option named <i>option</i> is on. This option is not available in <code>csh</code> or <code>sh</code> .                                                                                                                           |
| <code>-O file</code>              | True if <i>file</i> exists and is owned by the effective user ID of this process. This option is not available in <code>sh</code> .                                                                                                       |
| <code>-p file</code>              | True if <i>file</i> is a named pipe (FIFO).                                                                                                                                                                                               |
| <code>-r file</code>              | True if <i>file</i> exists and is readable.                                                                                                                                                                                               |
| <code>-s file</code>              | True if <i>file</i> exists and has a size greater than zero.                                                                                                                                                                              |
| <code>-S file</code>              | True if <i>file</i> exists and is a socket. This option is not available in <code>sh</code> .                                                                                                                                             |
| <code>-t [file_descriptor]</code> | True if the file whose file descriptor number is <i>file_descriptor</i> is open and is associated with a terminal. If <i>file_descriptor</i> is not specified, 1 is used as a default value.                                              |
| <code>-u file</code>              | True if <i>file</i> exists and its set-user-ID flag is set.                                                                                                                                                                               |
| <code>-w file</code>              | True if <i>file</i> exists and is writable. True indicates only that the write flag is on. The <i>file</i> is not writable on a read-only file system even if this test indicates true.                                                   |
| <code>-x file</code>              | True if <i>file</i> exists and is executable. True indicates only that the execute flag is on. If <i>file</i> is a directory, true indicates that <i>file</i> can be searched.                                                            |
| <code>-z string</code>            | True if the length of string <i>string</i> is zero.                                                                                                                                                                                       |
| <code>file1 -nt file2</code>      | True if <i>file1</i> exists and is newer than <i>file2</i> . This option is not available in <code>sh</code> .                                                                                                                            |
| <code>file1 -ot file2</code>      | True if <i>file1</i> exists and is older than <i>file2</i> . This option is not available in <code>sh</code> .                                                                                                                            |
| <code>file1 -ef file2</code>      | True if <i>file1</i> and <i>file2</i> exist and refer to the same file. This option is not available in <code>sh</code> .                                                                                                                 |
| <code>string</code>               | True if the string <i>string</i> is not the null string.                                                                                                                                                                                  |
| <code>string1 = string2</code>    | True if the strings <i>string1</i> and <i>string2</i> are identical.                                                                                                                                                                      |
| <code>string1 != string2</code>   | True if the strings <i>string1</i> and <i>string2</i> are not identical.                                                                                                                                                                  |
| <code>n1 -eq n2</code>            | True if the numbers <i>n1</i> and <i>n2</i> are algebraically equal. A number may be integer, floating point or floating-point constant (such as <code>[+/-]Inf</code> , <code>[+/-]NaN</code> ) in any format specified by C99/XPG6/SUS. |

---

|                                 |                                                                                                                                                                                                                                        |
|---------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>n1 -ne n2</i>                | True if the numbers <i>n1</i> and <i>n2</i> are not algebraically equal. A number may be integer, floating point or floating-point constant (such as [+/-]Inf, [+/-]NaN) in any format specified by C99/XPG6/SUS.                      |
| <i>n1 -gt n2</i>                | True if the number <i>n1</i> is algebraically greater than the number <i>n2</i> . A number may be integer, floating point or floating-point constant (such as [+/-]Inf, [+/-]NaN) in any format specified by C99/XPG6/SUS.             |
| <i>n1 -ge n2</i>                | True if the number <i>n1</i> is algebraically greater than or equal to the number <i>n2</i> . A number may be integer, floating point or floating-point constant (such as [+/-]Inf, [+/-]NaN) in any format specified by C99/XPG6/SUS. |
| <i>n1 -lt n2</i>                | True if the number <i>n1</i> is algebraically less than the number <i>n2</i> . A number may be integer, floating point or floating-point constant (such as [+/-]Inf, [+/-]NaN) in any format specified by C99/XPG6/SUS.                |
| <i>n1 -le n2</i>                | True if the number <i>n1</i> is algebraically less than or equal to the number <i>n2</i> . A number may be integer, floating point or floating-point constant (such as [+/-]Inf, [+/-]NaN) in any format specified by C99/XPG6/SUS.    |
| <i>condition1 -a condition2</i> | True if both <i>condition1</i> and <i>condition2</i> are true. The -a binary primary is left associative and has higher precedence than the -o binary primary.                                                                         |
| <i>condition1 -o condition2</i> | True if either <i>condition1</i> or <i>condition2</i> is true. The -o binary primary is left associative.                                                                                                                              |

These primaries can be combined with the following operators:

|                      |                                                                                                                                                                                                 |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>! condition</i>   | True if <i>condition</i> is false.                                                                                                                                                              |
| <i>( condition )</i> | True if <i>condition</i> is true. The parentheses ( ) can be used to alter the normal precedence and associativity. The parentheses are meaningful to the shell and, therefore, must be quoted. |

The algorithm for determining the precedence of the operators and the return value that is generated is based on the number of arguments presented to `test`. (However, when using the `[...]` form, the right-bracket final argument is not counted in this algorithm.)

In the following list, \$1, \$2, \$3 and \$4 represent the arguments presented to `test` as a *condition*, *condition1*, or *condition2*.

*0 arguments:* Exit false (1).

*1 argument:* Exit true (0) if \$1 is not null. Otherwise, exit false.

*2 arguments:*

- If \$1 is !, exit true if \$2 is null, false if \$2 is not null.
- If \$1 is a unary primary, exit true if the unary test is true, false if the unary test is false.
- Otherwise, produce unspecified results.

*3 arguments:*

- If \$2 is a binary primary, perform the binary test of \$1 and \$3.
- If \$1 is !, negate the two-argument test of \$2 and \$3.
- Otherwise, produce unspecified results.

*4 arguments:*

- If \$1 is !, negate the three-argument test of \$2, \$3, and \$4.
- Otherwise, the results are unspecified.

## 用法

Scripts should be careful when dealing with user-supplied input that could be confused with primaries and operators. Unless the application writer knows all the cases that produce input to the script, invocations like `test "$1" -a "$2"` should be written as `test "$1" && test "$2"` to avoid problems if a user supplied values such as \$1 set to ! and \$2 set to the null string. That is, in cases where maximal portability is of concern, replace `test expr1 -a expr2` with `test expr1 && test expr2`, and replace `test expr1 -o expr2` with `test expr1 || test expr2`. But notice that, in `test`, `-a` has *higher* precedence than `-o`, while `&&` and `||` have *equal* precedence in the shell.

Parentheses or braces can be used in the shell command language to effect grouping.

Parentheses must be escaped when using `sh`. For example:

```
test \(expr1 -a expr2 \) -o expr3
```

This command is not always portable outside XSI-conformant systems. The following form can be used instead:

```
(test expr1 && test expr2) || test expr3
```

The two commands:

```
test "$1"
test ! "$1"
```

could not be used reliably on some historical systems. Unexpected results would occur if such a *string* condition were used and \$1 expanded to !, (, or a known unary primary. Better constructs are, respectively,



```
test -n "$1"
test -z "$1"
```

Historical systems have also been unreliable given the common construct:

```
test "$response" = "expected string"
```

One of the following is a more reliable form:

```
test "X$response" = "Xexpected string"
test "expected string" = "$response"
```

The second form assumes that `expected string` could not be confused with any unary primary. If `expected string` starts with `-`, `(`, `!` or even `=`, the first form should be used instead. Using the preceding rules without the marked extensions, any of the three comparison forms is reliable, given any input. (However, observe that the strings are quoted in all cases.)

Because the string comparison binary primaries, `=` and `!=`, have a higher precedence than any unary primary in the `>4` argument case, unexpected results can occur if arguments are not properly prepared. For example, in

```
test -d $1 -o -d $2
```

If `$1` evaluates to a possible directory name of `=`, the first three arguments are considered a string comparison, which causes a syntax error when the second `-d` is encountered. is encountered. One of the following forms prevents this; the second is preferred:

```
test \(-d "$1" \) -o \(-d "$2" \)
test -d "$1" || test -d "$2"
```

Also in the `>4` argument case:

```
test "$1" = "bat" -a "$2" = "ball"
```

Syntax errors occur if `$1` evaluates to `(` or `!`. One of the following forms prevents this; the third is preferred:

```
test "X$1" = "Xbat" -a "X$2" = "Xball"
test "$1" = "bat" && test "$2" = "ball"
test "X$1" = "Xbat" && test "X$2" = "Xball"
```

## 示例

In the `if` command examples, three conditions are tested, and if all three evaluate as true or successful, then their validities are written to the screen. The three tests are:

- if a variable set to 1 is greater than 0,
- if a variable set to 2 is equal to 2, and
- if the word `root` is included in the text file `/etc/passwd`.

/usr/bin/test

示例1 Using /usr/bin/test

Perform a mkdir if a directory does not exist:

```
test ! -d tempdir && mkdir tempdir
```

Wait for a file to become non-readable:

```
while test -r thefile
do
 sleep 30
done
echo "thefile" is no longer readable'
```

Perform a command if the argument is one of three strings (two variations), using the open bracket version [ of the test command:

```
if ["$1" = "pear"] || ["$1" = "grape"] || ["$1" = "apple"]
then
 command
fi
case "$1" in
 pear|grape|apple) command;;
esac
```

示例2 Using /usr/bin/test for the -e option

If one really wants to use the -e option in sh, use /usr/bin/test, as in the following:

```
if [! -h $PKG_INSTALL_ROOT$rLink] && /usr/bin/test -e
$PKG_INSTALL_ROOT/usr/bin/$rFile ; then
 ln -s $rFile $PKG_INSTALL_ROOT$rLink
fi
```

The test built-in

The two forms of the test built-in follow the Bourne shell's if example.

示例3 Using the sh built-in

```
ZERO=0 ONE=1 TWO=2 ROOT=root
```

```
if [$ONE -gt $ZERO]
```

```
[$TWO -eq 2]
```

```
grep $ROOT /etc/passwd >&1 > /dev/null # discard output
```

```
then
```

```
 echo "$ONE is greater than 0, $TWO equals 2, and $ROOT is" \
 "a user-name in the password file"
```

示例3 Using the sh built-in (续)

```
else
 echo "At least one of the three test conditions is false"
fi
```

示例4 Using the test built-in

Examples of the test built-in:

```
test `grep $ROOT /etc/passwd >&1 /dev/nul`l # discard output
```

```
echo $? # test for success
[`grep nosuchname /etc/passwd >&1 /dev/nul`l]
```

```
echo $? # test for failure
```

csch

示例5 Using the csh built-in

```
@ ZERO = 0; @ ONE = 1; @ TWO = 2; set ROOT = root
grep $ROOT /etc/passwd >&1 /dev/null # discard output
$status must be tested for immediately following grep
if ("$status" == "0" && $ONE > $ZERO && $TWO == 2) then
 echo "$ONE is greater than 0, $TWO equals 2, and $ROOT is" \
 "a user-name in the password file"
endif
```

ksh88

示例6 Using the ksh88/ksh built-in

```
ZERO=0 ONE=1 TWO=$((ONE+ONE)) ROOT=root
if ((ONE > ZERO)) # arithmetical comparison
[[$TWO = 2]] # string comparison
[`grep $ROOT /etc/passwd >&1 /dev/nul`l] # discard output
then
 echo "$ONE is greater than 0, $TWO equals 2, and $ROOT is" \
 "a user-name in the password file"

else
 echo "At least one of the three test conditions is false"
fi
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of test: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

退出状态

The following exit values are returned:

- 0 *condition* evaluated to true.
- 1 *condition* evaluated to false or *condition* was missing.

>1 An error occurred.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

/usr/bin/test, csh,  
ksh88, sh

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

ksh

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | system/core-os  |
| Interface Stability | Uncommitted     |

## 另请参见

[csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [test\(1B\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

## 附注

The `not-a-directory` alternative to the `-f` option is a transition aid for BSD applications and may not be supported in future releases.

XPG4 sh, ksh88, ksh

Use arithmetic expressions such as

```
$((x > 3.1)) #
```

instead of

```
$ /usr/bin/test "$x" -gt 3.1 #)
```

when comparing two floating-point variables or a constant and a floating-point variable to prevent rounding errors (caused by the base16 to base10 transformation) to affect the result. Additionally the built-in arithmetic support in XPG4 sh, ksh88 and ksh is significantly faster because it does not require the explicit transformation to strings for each comparison.

**引用名** test – condition evaluation command

**用法概要** /usr/ucb/test *expression*  
*expression*

**描述** test evaluates the expression *expression* and, if its value is true, sets 0 (true) exit status; otherwise, a non-zero (false) exit status is set. test also sets a non-zero exit status if there are no arguments. When permissions are tested, the effective user ID of the process is used.

All operators, flags, and brackets (brackets used as shown in the second SYNOPSIS line) must be separate arguments to the test command; normally these items are separated by spaces.

## 用法

### Primitives

The following primitives are used to construct *expression*:

- r *filename* True if *filename* exists and is readable.
- w *filename* True if *filename* exists and is writable.
- x *filename* True if *filename* exists and is executable.
- f *filename* True if *filename* exists and is a regular file. Alternatively, if /usr/bin/sh users specify /usr/ucb before /usr/bin in their PATH environment variable, then test will return true if *filename* exists and is (not-a-directory). This is also the default for /usr/bin/csh users.
- d *filename* True if *filename* exists and is a directory.
- c *filename* True if *filename* exists and is a character special file.
- b *filename* True if *filename* exists and is a block special file.
- p *filename* True if *filename* exists and is a named pipe (fifo).
- u *filename* True if *filename* exists and its set-user- ID bit is set.
- g *filename* True if *filename* exists and its set-group- ID bit is set.
- k *filename* True if *filename* exists and its sticky bit is set.
- s *filename* True if *filename* exists and has a size greater than zero.
- t[ *filides* ] True if the open file whose file descriptor number is *filides* (1 by default) is associated with a terminal device.
- z *s1* True if the length of string *s1* is zero.
- n *s1* True if the length of the string *s1* is non-zero.
- s1* = *s2* True if strings *s1* and *s2* are identical.
- s1* != *s2* True if strings *s1* and *s2* are *not* identical.

*s1* True if *s1* is *not* the null string.

*n1* `-eq` *n2* True if the integers *n1* and *n2* are algebraically equal. Any of the comparisons `-ne`, `-gt`, `-ge`, `-lt`, and `-le` may be used in place of `-eq`.

Operators These primaries may be combined with the following operators:

`!` Unary negation operator.

`-a` Binary *and* operator.

`-o` Binary *or* operator (`-a` has higher precedence than `-o`).

*(expression)* Parentheses for grouping. Notice also that parentheses are meaningful to the shell and, therefore, must be quoted.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

另请参见 [find\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#)

附注 The `not-a-directory` alternative to the `-f` option is a transition aid for BSD applications and may not be supported in future releases.

If you test a file you own (the `-r`, `-w`, or `-x` tests), but the permission tested does not have the *owner* bit set, a non-zero (false) exit status will be returned even though the file may have the *group* or *other* bit set for that permission. The correct exit status will be set if you are super-user.

The `=` and `!=` operators have a higher precedence than the `-r` through `-n` operators, and `=` and `!=` always expect arguments; therefore, `=` and `!=` cannot be used with the `-r` through `-n` operators.

If more than one argument follows the `-r` through `-n` operators, only the first argument is examined; the others are ignored, unless a `-a` or a `-o` is the second argument.

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名      | tftp – trivial file transfer program                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 用法概要     | tftp [ <i>host</i> [ <i>port</i> ]]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 描述       | tftp is the user interface to the Internet TFTP (Trivial File Transfer Protocol), which allows users to transfer files to and from a remote machine. The remote <i>host</i> and optional <i>port</i> may be specified on the command line, in which case tftp uses <i>host</i> as the default host, and if specified, <i>port</i> as the default port, for future transfers. See the connect command below.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 用法       | Once tftp is running, it issues the prompt tftp> and recognizes the following commands:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Commands | <p>connect <i>host-name</i> [<i>port</i> ]</p> <p>Set the <i>host</i>, and optionally <i>port</i>, for transfers. The TFTP protocol, unlike the FTP protocol, does not maintain connections between transfers; thus, the connect command does not actually create a connection, but merely remembers what host is to be used for transfers. You do not have to use the connect command; the remote host can be specified as part of the get or put commands.</p> <p>mode <i>transfer-mode</i></p> <p>Set the mode for transfers; <i>transfer-mode</i> may be one of <code>ascii</code> or <code>binary</code>. The default is <code>ascii</code>.</p> <p>put <i>filename</i></p> <p>put <i>localfile remotefile</i></p> <p>put <i>filename1 filename2 . . . filenameN remote-directory</i></p> <p>Transfer a file, or a set of files, to the specified remote file or directory. The destination can be in one of two forms: a filename on the remote host if the host has already been specified, or a string of the form:</p> <p><i>host:filename</i></p> <p>to specify both a <i>host</i> and <i>filename</i> at the same time. If the latter form is used, the specified host becomes the default for future transfers. If the remote-directory form is used, the remote host is assumed to be running the UNIX system.</p> <p>The <i>host</i> can be a host name (see <a href="#">hosts(4)</a>) or an IPv4 or IPv6 address string (see <a href="#">inet(7P)</a> or <a href="#">inet6(7P)</a>). Since IPv6 addresses already contain “:”s, the <i>host</i> should be enclosed in square brackets when an IPv6 address is used. Otherwise, the first occurrence of a colon will be interpreted as the separator between the <i>host</i> and the <i>filename</i>. For example,</p> <p>[1080::8:800:200c:417A]:myfile</p> <p>Files may be written only if they already exist and are publicly writable. See <a href="#">in.tftpd(1M)</a>.</p> <p>get <i>filename</i></p> <p>get <i>remotename localname</i></p> <p>get <i>filename1 filename2 filename3 . . . filenameN</i></p> <p>Get a file or set of files (three or more) from the specified remote <i>sources</i>. <i>source</i> can be in one of two forms: a filename on the remote host if the host has already been specified, or a string of the form:</p> |

*host:filename*

to specify both a host and filename at the same time. If the latter form is used, the last host specified becomes the default for future transfers. See the `put` command regarding specifying a *host*.

`quit`

Exit `tftp`. An EOF also exits.

`verbose`

Toggle verbose mode.

`trace`

Toggle packet tracing.

`status`

Show current status.

`rext retransmission-timeout`

Set the per-packet retransmission timeout, in seconds.

`timeout total-transmission-timeout`

Set the total transmission timeout, in seconds.

`ascii`

Shorthand for mode `ascii`.

`binary`

Shorthand for mode `binary`.

`blksize transfer-blocksize`

The value of the transfer blocksize option to negotiate with the server. A value of 0 disables the negotiation of this option.

`srext server-retransmission-timeout`

The value of the retransmission timeout option to request that the server uses. A value of 0 disables the negotiation of this option.

`tsize`

A toggle that sends the transfer size option to the server. By default, the option is not sent. The transfer size option is not sent with a `write` request when the *transfer-mode* is `ascii`.

? [ *command-name . . .* ]

Print help information.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE      |
|----------------|----------------------|
| Availability   | service/network/tftp |



## 另请参见

`in.tftpd(1M)`, `hosts(4)`, `attributes(5)`, `inet(7P)`, `inet6(7P)`

Malkin, G. and Harkin, A. RFC 2347, TFTP Option Extension. The Internet Society. May 1998

Malkin, G. and Harkin, A. RFC 2348, TFTP Blocksize Option. The Internet Society. May 1998

Malkin, G. and Harkin, A. RFC 2349, TFTP Timeout Interval and Transfer Size Options. The Internet Society. May 1998

Sollins, K.R. RFC 1350, The TFTP Protocol (Revision 2). Network Working Group. July 1992.

## 附注

The default *transfer-mode* is `ascii`. This differs from pre-SunOS 4.0 and pre-4.3BSD systems, so explicit action must be taken when transferring non-ASCII binary files such as executable commands.

Because there is no user-login or validation within the TFTP protocol, many remote sites restrict file access in various ways. Approved methods for file access are specific to each site, and therefore cannot be documented here.

When using the `get` command to transfer multiple files from a remote host, three or more files must be specified. If two files are specified, the second file is used as a local file.

With the default block size of 512 octets and a 16-bit block counter, some TFTP implementations might have problems with files over 33,553,919 octets (513 octets short of 32MB) in size. The Solaris implementation can transfer files up to 4GB in size.

By default, the Solaris TFTP client does not enable the `blocksize` or `transfer size` options. Setting the `blocksize` option to a higher value is sometimes useful as a workaround when dealing with peers that have a 32MB limit.

**引用名**           time – time a simple command

**用法概要**       time [-p] *utility* [*argument*]...

**描述**           The `time` utility invokes *utility* operand with *argument*, and writes a message to standard error that lists timing statistics for *utility*. The message includes the following information:

- The elapsed (real) time between invocation of *utility* and its termination.
- The User CPU time, equivalent to the sum of the `tms_utime` and `tms_cutime` fields returned by the `times(2)` function for the process in which *utility* is executed.
- The System CPU time, equivalent to the sum of the `tms_stime` and `tms_cstime` fields returned by the `times()` function for the process in which *utility* is executed.

When `time` is used as part of a pipeline, the times reported are unspecified, except when it is the sole command within a grouping command in that pipeline. For example, the commands on the left are unspecified; those on the right report on utilities `a` and `c`, respectively:

```
time a | b | c { time a } | b | c
a | b | time c a | b | (time c)
```

**选项**           The following option is supported:

`-p`     Writes the timing output to standard error in the following format:  
          real %f\nuser %f\nsys %f\n < real seconds>, <user seconds>,  
          <system seconds>

**操作数**       The following operands are supported:

*utility*       The name of the utility that is to be invoked.  
*argument*     Any string to be supplied as an argument when invoking *utility*.

**用法**           The `time` utility returns exit status 127 if an error occurs so that applications can distinguish “failure to find a utility” from “invoked utility exited with an error indication.” The value 127 was chosen because it is not commonly used for other meanings. Most utilities use small values for “normal error conditions” and the values above 128 can be confused with termination due to receipt of a signal. The value 126 was chosen in a similar manner to indicate that the utility could be found, but not invoked.

**示例**           示例 1 Using the `time` command

It is frequently desirable to apply `time` to pipelines or lists of commands. This can be done by placing pipelines and command lists in a single file. This single file can then be invoked as a utility, and the `time` applies to everything in the file.

Alternatively, the following command can be used to apply `time` to a complex command:

```
example% time sh -c 'complex-command-line'
```

**示例 2** Using time in the csh shell

The following two examples show the differences between the csh version of `time` and the version in `/usr/bin/time`. These examples assume that csh is the shell in use.

```
example% time find / -name csh.1 -print
/usr/share/man/man1/csh.1
95.0u 692.0s 1:17:52 16% 0+0k 0+0io 0pf+0w
```

See [csh\(1\)](#) for an explanation of the format of time output.

```
example% /usr/bin/time find / -name csh.1 -print
/usr/share/man/man1/csh.1
real 1:23:31.5
user 1:33.2
sys 11:28.2
```

**环境变量**

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `time`: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, LC\_NUMERIC, NLS\_PATH, and PATH.

**退出状态**

If *utility* is invoked, the exit status of `time` will be the exit status of *utility*. Otherwise, the `time` utility will exit with one of the following values:

- 1–125 An error occurred in the `time` utility.
- 126 *utility* was found but could not be invoked.
- 127 *utility* could not be found.

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见**

[csh\(1\)](#), [shell\\_builtins\(1\)](#), [timex\(1\)](#), [times\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

**附注**

When the `time` command is run on a multiprocessor machine, the total of the values printed for `user` and `sys` can exceed `real`. This is because on a multiprocessor machine it is possible to divide the task between the various processors.

When the command being timed is interrupted, the timing values displayed may not always be accurate.

## 已知问题

Elapsed time is accurate to the second, while the CPU times are measured to the 100th second. Thus the sum of the CPU times can be up to a second larger than the elapsed time.

**引用名** times – shell built-in function to report time usages of the current shell

## 用法概要

sh times

ksh times

## 描述

sh Print the accumulated user and system times for processes run from the shell.

ksh Print the accumulated user and system times for the shell and for processes run from the shell.

On this man page, [ksh\(1\)](#) commands that are preceded by one or two \* (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by \*\* that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

**另请参见** [ksh\(1\)](#), [sh\(1\)](#), [time\(1\)](#), [attributes\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | timex – time a command; report process data and system activity                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 用法概要 | timex [-o] [-p [-fhkmrt]] [-s] <i>command</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 描述   | <p>The given <i>command</i> is executed; the elapsed time, user time and system time spent in execution are reported in seconds. Optionally, process accounting data for the <i>command</i> and all its children can be listed or summarized, and total system activity during the execution interval can be reported.</p> <p>The output of <code>timex</code> is written on standard error.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"><li>-o Report the total number of blocks read or written and total characters transferred by <i>command</i> and all its children. This option works only if the process accounting software is installed.</li><li>-p List process accounting records for <i>command</i> and all its children. This option works only if the process accounting software is installed. Suboptions <code>f</code>, <code>h</code>, <code>k</code>, <code>m</code>, <code>r</code>, and <code>t</code> modify the data items reported. The options are as follows:<ul style="list-style-type: none"><li>-f Print the <code>fork(2)/exec(2)</code> flag and system exit status columns in the output.</li><li>-h Instead of mean memory size, show the fraction of total available CPU time consumed by the process during its execution. This “hog factor” is computed as (total CPU time)/(elapsed time).</li><li>-k Instead of memory size, show total kcore-minutes.</li><li>-m Show mean core size (the default).</li><li>-r Show CPU factor (user time/(system-time + user-time)).</li><li>-t Show separate system and user CPU times. The number of blocks read or written and the number of characters transferred are always reported.</li></ul></li><li>-s Report total system activity (not just that due to <i>command</i>) that occurred during the execution interval of <i>command</i>. All the data items listed in <code>sar(1)</code> are reported.</li></ul> |

示例 1 Examples of `timex`.

A simple example:

```
example% timex -ops sleep 60
```

A terminal session of arbitrary complexity can be measured by timing a sub-shell:

```
example% timex -opskmt sh
 session commands
```

```
EOT
```

---

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE                     |
|----------------|-------------------------------------|
| Availability   | system/accounting/legacy-accounting |

**另请参见**

[sar\(1\)](#), [time\(1\)](#), [exec\(2\)](#), [fork\(2\)](#), [times\(2\)](#), [attributes\(5\)](#)

**附注**

Process records associated with command are selected from the accounting file `/var/adm/pacct` by inference, since process genealogy is not available. Background processes having the same user ID, terminal ID, and execution time window will be spuriously included.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | tip - 连接至远程系统                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 用法概要 | tip [-v] [-speed-entry] {hostname   phone-number   device}                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 描述   | <p>tip 实用程序可建立到远程主机的全双工终端连接。建立连接后，使用 tip 的远程会话行为将类似本地终端上的交互式会话。</p> <p>remote 文件包含描述 tip 使用的远程系统和线速的条目。</p> <p>每个主机都具有缺省连接波特率，或者可以使用 <i>-speed-entry</i> 命令行参数指定一个速度。</p> <p>指定了 <i>phone-number</i> 后，tip 将在 remote 文件中查找以下格式的条目：</p> <pre>tip -speed-entry</pre> <p>tip 找到此类条目时，其将设置相应的连接速度。如果未找到此类条目，tip 会将 <i>-speed-entry</i> 当做系统名进行解释，从而导致错误消息。</p> <p>如果省略 <i>-speed-entry</i>，tip 将使用 <i>tip0</i> 条目设置连接速度。</p> <p>指定了 <i>device</i> 后，tip 将尝试使用用户的访问特权打开该设备，而不是通过 tip 的常用访问特权 (<i>setuid uucp</i>)。用户必须对此设备具有读取/写入权限。tip 实用程序会将任何以反斜杠字符 (/) 开头的字符串解释为设备名。</p> <p>建立连接时，tip 将向远程系统发送连接消息。该消息的缺省值位于 remote 文件中。</p> <p>tip 尝试连接至远程系统时，其将使用以独占方式打开的 <i>ioctl(2)</i> 调用打开关联设备。因此，一次仅能一位用户访问设备。这可以防止多个进程对终端线抽样。此外，tip 遵循 <i>uucp(1C)</i> 使用的锁定协议。</p> <p>tip 启动后，其将从主目录中的文件 <i>.tiprc</i> 读取命令。</p> |
| 选项   | -v 随着执行 <i>.tiprc</i> 文件的命令来显示这些命令。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 用法   | <p>键入的字符通常会直接传输至远程计算机，此计算机也会执行回显操作。</p> <p>无论何时 tip 提示需要参数（例如，文件传输设置期间），都可以使用标准清除和删除字符对键入的行进行编辑。对提示做出响应的空行或中断都将中止对话并将您返回至远程计算机。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 命令   | <p>作为某行的第一个字符显示的波浪号 (~) 是转义信号，其将指示 tip 执行一些特殊操作。tip 可识别以下转义序列：</p> <pre>~^D</pre> <p>~. 丢弃连接并退出（您可能仍登录在远程计算机上）。<b>请注意：</b>如果执行 <i>rlogin</i>，然后在远程主机上运行 tip，必须键入 <i>~.</i>（波浪线波浪线点）以结束 tip 会话。如果键入 <i>~.</i>（波浪线点），其将终止 <i>rlogin</i>。</p> <pre>~c [name]</pre> <p>将目录更改为 <i>name</i>。无参数表示更改至主目录。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |



- ~! 退回到本地计算机上的交互式 shell。退出 shell 会将您返回至 tip。
- ~> 将文件从本地复制到远程。
- ~< 将文件从远程复制到本地。
- ~p *from* [ *to* ] 将文件发送到运行 UNIX 系统的远程主机。使用 put 命令时，远程系统将在
- ```
cat > to
```
- tip 向其发送 *from* 文件时运行该命令字符串。如果未指定 *to* 文件，将使用 *from* 文件名。该命令实际是一个 UNIX 系统特定版本的 "->" 命令。
- ~t *from* [*to*] 从运行 UNIX 系统的远程主机获取文件。如在 put 命令中一样，如果未指定 *to* 文件，其缺省为 *from* 文件名。远程主机将执行以下命令字符串
- ```
cat from ; echo ^A
```
- 以将文件发送至 tip。
- ~| 将输出从远程命令管道传输至本地进程。发送至本地系统的命令字符串由 shell 进行处理。
- ~C 将程序连接至远程计算机。发送至该程序的命令字符串由 shell 进行处理。程序将文件描述符 0 继承为远程行输入，将 1 继承为远程行输出，将 2 继承为 tty 标准错误输出。
- ~\$ 将输出从本地进程管道传输至远程主机。发送至本地系统的命令字符串由 shell 进行处理。
- ~# 将 BREAK 发送至远程系统。
- ~s 设置变量（请参见以下讨论）。
- ~^Z 停止 tip。只有在支持作业控制的 shell（如 C shell）下运行时才可用。
- ~^Y 仅停止 tip 的“本地端”。只有在支持作业控制的 shell（如 C shell）下运行时才可用。保留运行 tip 的“远程端”，即显示来自远程主机的输出的一端。
- ~? 获取波浪线转义符的摘要。

复制文件需要远程主机提供一些协作。当使用 ~> 或 ~< 转义符发送文件时，tip 将提示输入（要传输或接收的）文件名，如果正从远程系统传输文件，则将向远程系统发送一条命令。tip 传输文件时，已传输的行数将持续显示在屏幕上。文件传输可以因为中断而中止。

## 自动呼叫单元

通过一些自动呼叫单元 (auto-call unit, ACU)，可使用 `tip` 对远程系统进行拨号。远程系统说明包含 `du` 功能时，`tip` 使用呼叫单元 (`cu`)、ACU 类型 (`at`) 和提供的电话号码 (`pn`)。通常情况下，`tip` 将在其拨号时显示详细的消息。

根据用于建立连接的自动拨号程序的类型，远程主机可能会在连接时向其发送一些乱码。请始终不要认为向外部主机键入的第一个字符就是显示给其的第一个字符。推荐的做法是在建立连接时立即键入一个 `kill` 字符（大多数 UNIX 系统既支持 `@` 也支持 `Control-U` 作为初始 `kill` 字符）。

`tip` 当前支持 Ventel MD-212+ 调制解调器和 DC Hayes 兼容的调制解调器。

`tip` 初始化 Hayes 兼容的调制解调器以进行拨号时，其会将该调制解调器设置为自动应答。通常情况下，会话完成后，`tip` 将丢弃 DTR，这将导致该调制解调器“挂起”。

大多数调制解调器可进行配置，以便在 DTR 丢弃时，这些调制解调器可以将自身重新初始化为预编程状态。这可用于重置调制解调器并禁用自动应答（如果需要）。

此外，可以使用 `Hayes S` 命令启动电话号码，以便可以在拨号前配置调制解调器。例如，要禁用自动应答，可使用诸如 `pn=S0=0DT5551212` 的设置在 `/etc/remote` 中设置所有电话号码。`S0=0` 将禁用自动应答。

## 远程主机描述

远程主机的描述通常位于系统范围文件 `/etc/remote` 中。然而，用户可以通过定义和导出 `REMOTE` shell 变量维护个人描述文件（和电话号码）。`remote` 文件必须可由 `tip` 读取，但描述电话号码的辅助文件则可以保持仅可由用户读取。该辅助电话号码文件是 `/etc/phones`，除非定义并导出了 shell 变量 `PHONES`。该电话号码文件包含以下格式的行：

```
system-name phone-number
```

针对系统找到的每个电话号码都会进行试拨，直至建立连接，或到达文件结尾。电话号码构造自 `'0123456789-=*'`，其中 `'='` 和 `'*'` 用于指示应等待的第二个拨号音（取决于 ACU）。

## tip 内部变量

`tip` 维护一组用于正常运行的变量。其中一些变量对于普通用户是只读的（`root` 用户可以根据需要更改任何变量）。可以通过 `~s` 转义符显示和设置变量。变量的语法仿效 `vi(1)` 和 `mail(1)`。将 `all` 作为参数提供给 `~s` 转义符将显示用户可以读取的所有变量。或者，用户可以通过将 `?` 附加到末尾请求显示特定变量。例如，`'~s escape?'` 将显示当前转义符。

变量可以是数字 (`num`)、字符串 (`str`)、字符 (`char`) 或布尔 (`bool`) 值。布尔变量只能通过指定其名称进行设置。它们可以通过在名称前加上 `!` 进行重置。其他变量类型通过附加 `=` 和值进行设置。整个赋值中不得包含任何空白。可使用单个 `set` 命令询问以及设置一些变量。

通过将 `set` 命令（没有 `~s` 前缀）放入某人的主目录中的 `.tiprc` 文件中，可以在运行时初始化变量。通过 `-v` 选项，`tip` 可以随着进行设置而显示这些设置。前面有 `#` 符号的注释可显示在 `.tiprc` 文件中。

最后，必须指定完整的变量名或者可以提供缩写。下表详细列出了对于 `tip` 已知的变量。

|                          |                                                                                                                                                                                                                                                    |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>beautify</code>    | (bool) 编写会话脚本时废弃不可打印的字符；缩写为 <code>be</code> 。如果提供了 <code>nb</code> 功能，则 <code>beautify</code> 初始设置为 <code>off</code> 。否则， <code>beautify</code> 初始设置为 <code>on</code> 。                                                                            |
| <code>baudrate</code>    | (num) 建立连接所用的波特率；缩写为 <code>ba</code> 。如果在命令行上指定了波特率，则 <code>baudrate</code> 初始设置为该指定值。或者，如果提供了 <code>br</code> 功能，则 <code>baudrate</code> 初始设置为该功能的值。否则，将 <code>baudrate</code> 设置为 300 波特。启动 <code>tip</code> 后， <code>baudrate</code> 只可由超级用户更改。 |
| <code>dialtimeout</code> | (num) 拨电话号码时，等待连接建立的时间（以秒为单位）；缩写为 <code>dial</code> 。 <code>dialtimeout</code> 初始设置为 60 秒，并只可由超级用户更改。                                                                                                                                              |
| <code>disconnect</code>  | (str) 发送至远程主机以从其断开连接的字符串；缩写为 <code>di</code> 。如果提供了 <code>di</code> 功能，则 <code>disconnect</code> 初始设置为该功能的值。否则，将 <code>disconnect</code> 设置为空字符串 ("")。                                                                                             |
| <code>echocheck</code>   | (bool) 通过等待传输的最后一个字符的回显，在文件传输期间与远程主机同步；缩写为 <code>ec</code> 。如果提供了 <code>ec</code> 功能，则 <code>echocheck</code> 初始设置为 <code>on</code> 。否则， <code>echocheck</code> 初始设置为 <code>off</code> 。                                                           |
| <code>eofread</code>     | (str) 表示 <code>~&lt;</code> 文件传输命令期间传输结束的字符集合；缩写为 <code>eofr</code> 。如果提供了 <code>ie</code> 功能，则 <code>eofread</code> 初始设置为该功能的值。否则，将 <code>eofread</code> 设置为空字符串 ("")。                                                                            |
| <code>eofwrite</code>    | (str) 发送以指示 <code>~&gt;</code> 文件传输命令期间传输结束的字符串；缩写为 <code>eofw</code> 。如果提供了 <code>oe</code> 功能，则 <code>eofread</code> 初始设置为该功能的值。否则，将 <code>eofread</code> 设置为空字符串 ("")。                                                                          |
| <code>eol</code>         | (str) 指示行结束的字符集合。 <code>tip</code> 仅识别行结束之后的转义符。如果提供了 <code>el</code> 功能，则 <code>eol</code> 初始设置为该功能的值。否则，将 <code>eol</code> 设置为空字符串 ("")。                                                                                                         |
| <code>escape</code>      | (char) 命令前缀（转义）字符；缩写为 <code>es</code> 。如果提供了 <code>es</code> 功能，则 <code>escape</code> 初始设置为该功能的值。否则，将 <code>escape</code> 设置为 <code>'~'</code> 。                                                                                                   |
| <code>etimeout</code>    | (num) 设置了 <code>echocheck</code> 后， <code>tip</code> 应等待回显检查响应的时量（以秒为单位）；缩写为 <code>et</code> 。如果提供了 <code>et</code> 功能，则 <code>etimeout</code> 初始设置为该功能的值。否则，将 <code>etimeout</code> 设置为 10 秒。                                                     |
| <code>exceptions</code>  | (str) 由于美化开关而不应丢弃的字符集合；缩写为 <code>ex</code> 。如果提供了 <code>ex</code> 功能，则 <code>exceptions</code> 初始设置为该功能的值。否则，将 <code>exceptions</code> 设置为 <code>'\t\n\f\b'</code> 。                                                                               |
| <code>force</code>       | (char) 用于强制执行文字数据传输的字符；缩写为 <code>fo</code> 。如果提供了 <code>fo</code> 功能，则 <code>force</code> 初始设置为该功能的值。否则，将 <code>force</code> 设置为 <code>\377</code> （这将禁用该项）。                                                                                       |

|              |                                                                                                                                                                                                                              |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| framesize    | (num)接收文件时，文件系统写入操作之间要缓冲的数据量（以字节为单位）；缩写为 fr。如果提供了 fs 功能，则 framesize 初始设置为该功能的值。否则，将 framesize 设置为 1024。                                                                                                                      |
| halfduplex   | (bool)因为主机是半双工而执行本地回显；缩写为 hdx。如果提供了 hd 功能，则 halfduplex 初始设置为 on。否则，halfduplex 初始设置为 off。                                                                                                                                     |
| hardwareflow | (bool)执行硬件流控制；缩写为 hf。如果提供了 hf 功能，则 hardwareflow 初始设置为 on。否则，hardwareflowcontrol 初始设置为 off。                                                                                                                                   |
| host         | (str)连接到的主机名；缩写为 ho。host 永久设置为命令行上或 HOST 环境变量中给定的名称。                                                                                                                                                                         |
| localecho    | (bool)halfduplex 的同义词；缩写为 le。                                                                                                                                                                                                |
| log          | (str)向其中记录有关外拨电话呼叫信息的文件的名称。log 初始设置为 /var/adm/aculog，且只能由超级用户检查或更改。                                                                                                                                                          |
| parity       | (str)与远程主机对话时要生成和校验的奇偶；缩写为 par。可能值有：<br>none><br>zero 不对输入校验奇偶，输出上的奇偶位设置为零。<br>one 不对输入校验奇偶，输出上的奇偶位设置为一。<br>even 对输入校验偶数奇偶并在输出上生成偶数奇偶。<br>odd 对输入校验奇数奇偶并在输出上生成奇数奇偶。<br>如果提供了 pa 功能，则 parity 初始设置为该功能的值；否则，将 parity 设置为 none。 |
| phones       | 要在其中查找隐藏的电话号码的文件。如果设置了环境变量 PHONES，则将 phones 设置为 PHONES 的值。否则，将 phones 设置为 /etc/phones。phones 的值不能从 tip 内进行更改。                                                                                                                |
| prompt       | (char)指示远程主机上行结束的字符；缩写为 pr。该值用于数据传输期间的同步。文件传输命令期间传输的行计数基于该字符的接收。如果提供了 pr 功能，则 prompt 初始设置为该功能的值。否则，将 prompt 设置为 \n。                                                                                                          |
| raise        | (bool)大写映射模式；缩写为 ra。启用该模式后，所有小写字母都将由 tip 映射为大写，以便传输至远程计算机。如果提供了 ra 功能，则 raise 初始设置为 on。否则，raise 初始设置为 off。                                                                                                                   |
| raisechar    | (char)用于切换大写映射模式的输入字符；缩写为 rc。如果提供了 rc 功能，则 raisechar 初始设置为该功能的值。否则，将 raisechar 设置为 \377（这将禁用该项）。                                                                                                                             |

|                        |                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>rawftp</code>    | (bool) 文件传输期间发送所有字符；不过滤非可打印字符，不执行诸如从 <code>\n</code> 到 <code>\r</code> 的转换。缩写为 <code>raw</code> 。如果提供了 <code>rw</code> 功能，则 <code>rawftp</code> 初始设置为 <code>on</code> 。否则， <code>rawftp</code> 初始设置为 <code>off</code> 。                                                                                                                                                                            |
| <code>record</code>    | (str) 其中记录了会话脚本的文件的名称；缩写为 <code>rec</code> 。如果提供了 <code>re</code> 功能，则 <code>record</code> 初始设置为该功能的值。否则，将 <code>record</code> 设置为 <code>tip.record</code> 。                                                                                                                                                                                                                                       |
| <code>remote</code>    | 要在其中查找远程系统描述的文件。如果设置了环境变量 <code>REMOTE</code> ，则将 <code>remote</code> 设置为 <code>REMOTE</code> 的值。否则，将 <code>remote</code> 设置为 <code>/etc/remote</code> 。 <code>remote</code> 的值不能从 <code>tip</code> 内进行更改。                                                                                                                                                                                         |
| <code>script</code>    | (bool) 会话脚本编写模式；缩写为 <code>sc</code> 。 <code>script</code> 为 <code>on</code> 时， <code>tip</code> 将在 <code>record</code> 中指定的脚本记录文件中记录远程计算机传输的所有内容。如果 <code>beautify</code> 开关为开启，脚本文件中将仅包括可打印 ASCII 字符（这些字符介于 040 与 0177 之间）。变量 <code>exceptions</code> 用于指示对于一般美化规则而言是例外的字符。如果提供了 <code>sc</code> 功能，则 <code>script</code> 初始设置为 <code>on</code> 。否则， <code>script</code> 初始设置为 <code>off</code> 。 |
| <code>tabexpand</code> | (bool) 在文件传输期间，将 TAB 字符扩展至 SPACE 字符；缩写为 <code>tab</code> 。 <code>tabexpand</code> 为 <code>on</code> 时，每个 <code>tab</code> 都将扩展至八个 SPACE 字符。如果提供了 <code>tb</code> 功能， <code>tabexpand</code> 初始设置为 <code>on</code> 。否则， <code>tabexpand</code> 初始设置为 <code>off</code> 。                                                                                                                             |
| <code>tandem</code>    | (bool) 使用 XON/XOFF 流控制来限制远程主机发送数据的速率；缩写为 <code>ta</code> 。如果提供了 <code>nt</code> 功能，则 <code>tandem</code> 初始设置为 <code>off</code> 。否则， <code>tandem</code> 初始设置为 <code>on</code> 。                                                                                                                                                                                                                   |
| <code>verbose</code>   | (bool) 详细模式；缩写为 <code>verb</code> ；启用详细模式后， <code>tip</code> 将在拨号时打印消息，显示文件传输运行期间传输的当前行数等等。如果提供了 <code>nv</code> 功能，则 <code>verbose</code> 初始设置为 <code>off</code> 。否则， <code>verbose</code> 初始设置为 <code>on</code> 。                                                                                                                                                                                |
| <code>SHELL</code>     | (str) 用于 <code>~!</code> 命令的 shell 名称；缺省值是 <code>/bin/sh</code> ，或从环境中获取。                                                                                                                                                                                                                                                                                                                          |
| <code>HOME</code>      | (str) 用于 <code>~c</code> 命令的主目录。缺省值从环境中获取。                                                                                                                                                                                                                                                                                                                                                         |

## 示例

示例 1 使用 `tip` 命令

以下是一个用于传输文件的对话框示例。

```
arpa% tip monet
[connected]
...(assume we are talking to a UNIX system)...
ucbmonet login: sam
Password:
monet% cat sylvester.c
~> Filename: sylvester.c
```

### 示例1 使用 tip 命令 (续)

```

32 lines transferred in 1 minute 3 seconds
monet%
monet% ~< Filename: reply.c
List command for remote host: cat reply.c
65 lines transferred in 2 minutes
monet%
...(or, equivalently)...
monet% ~p sylvester.c
...(actually echoes as ~[put] sylvester.c)...
32 lines transferred in 1 minute 3 seconds
monet%
monet% ~t reply.c
...(actually echoes as ~[take] reply.c)...
65 lines transferred in 2 minutes
monet%
...(to print a file locally)...
monet% ~|Local command: pr h sylvester.c | lpr
List command for remote host: cat sylvester.c
monet% ~^D
[EOT]
...(back on the local system)...

```

### 环境变量

以下环境变量由 tip 读取。

REMOTE remote 文件的位置。  
 PHONES 包含私人电话号码的文件的位置。  
 HOST 要连接的缺省主机。  
 HOME 某人的登录目录（针对 chdirs）。  
 SHELL 在 '~!' 转义符上分叉的 shell。

### 文件

/etc/phones  
 /etc/remote  
 /var/spool/locks/LCK.\* 锁定文件以避免与 UUCP 冲突  
 /var/adm/acuolog 在其中记录外拨呼叫的文件  
 ~/.tiprc 初始化文件

### 属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

---

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

另请参见

[cu\(1C\)](#)、[mail\(1\)](#)、[uucp\(1C\)](#)、[vi\(1\)](#)、[ioctl\(2\)](#)、[attributes\(5\)](#)

已知问题

有两个当前未实现的其他变量：`chardelay` 和 `linedelay`。

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名   | touch, settime – change file access and modification times                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 用法概要  | <pre>touch [-acm] [-r <i>ref_file</i>   -t <i>time</i>   -d <i>date_time</i>] <i>file</i>...</pre> <pre>touch [-acm] [<i>time_spec</i>] <i>file</i>...</pre> <pre>settime [-f <i>ref_file</i>] [<i>time_spec</i>] <i>file</i>...</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 描述    | <p>The touch utility sets the access and modification times of each file. The <i>file</i> operand is created if it does not already exist.</p> <p>The time used can be specified by <code>-t <i>time</i></code>, by <code>-d <i>date_time</i></code>, by the corresponding time fields of the file referenced by <code>-r <i>ref_file</i></code>, or by the <i>time_spec</i> operand. If none of these are specified, touch uses the current time.</p> <p>If neither the <code>-a</code> nor <code>-m</code> options are specified, touch updates both the modification and access times.</p> <p>A user with write access to a file, but who is not the owner of the file or a super-user, can change the modification and access times of that file only to the current time. Attempts to set a specific time with touch results in an error.</p> <p>The <code>settime</code> utility is equivalent to <code>touch -c [<i>time_spec</i>] <i>file</i></code>.</p> |
| 选项    | The following options are supported in the touch and settime utilities:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| touch | The following options are supported for the touch utility:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|       | <ul style="list-style-type: none"><li>-a<br/>Changes the access time of <i>file</i>. Does not change the modification time unless <code>-m</code> is also specified.</li><li>-c<br/>Does not create a specified <i>file</i> if it does not exist. Does not write any diagnostic messages concerning this condition.</li><li>-d <i>date_time</i><br/>Uses the specified <i>date_time</i> instead of the current time. The option-argument must be a string of the form:<br/><pre>YYYY-MM-DDThh:mm:SS[.<i>frac</i>][<i>tz</i>]</pre>or<br/><pre>YYYY-MM-DDThh:mm:SS[,<i>frac</i>][<i>tz</i>]</pre>where<ul style="list-style-type: none"><li>▪ YYYY is at least four decimal digits giving the year</li><li>▪ MM, DD, hh, mm, and SS are as with <code>-t <i>time</i></code></li><li>▪ T is either the letter T or a single SPACE character</li></ul></li></ul>                                                                                                     |



- [*frac*] and [*frac*] are either empty, or a period (.) or a comma (,) respectively, followed by one or more decimal digits, specifying a fractional second
- [*tz*] is either empty, signifying local time, or the letter Z, signifying UTC. If [*tz*] is empty, the resulting time is affected by the value of the TZ environment variable

-m

Changes the modification time of *file*. Does not change the access time unless -a is also specified.

-r *ref\_file*

Uses the corresponding times of the file named by *ref\_file* instead of the current time.

-t *time*

Uses the specified *time* instead of the current time. *time* is a decimal number of the form:

[[CC]YY]MMDDhhmm[.SS]

where each two digits represent the following:

*MM*

The month of the year [01-12].

*DD*

The day of the month [01-31].

*hh*

The hour of the day [00-23].

*mm*

The minute of the hour [00-59].

*CC*

The first two digits of the year.

*YY*

The second two digits of the year.

*SS*

The second of the minute [00-61].

Both *CC* and *YY* are optional. If neither is given, the current year is assumed. If *YY* is specified, but *CC* is not, *CC* is derived as follows:

| If YY is: | CC becomes: |
|-----------|-------------|
| 69-99     | 19          |
| 00-38     | 20          |
| 39-68     | ERROR       |

The resulting time is affected by the value of the TZ environment variable. The range of valid times is the Epoch to January 18, 2038.

The range for SS is [00-61] rather than [00-59] because of leap seconds. If SS is 60 or 61, and the resulting time, as affected by the TZ environment variable, does not refer to a leap second, the resulting time is one or two seconds after a time where SS is 59. If SS is not given, it is assumed to be 0.

settime The following option is supported for the settime utility:

*-f ref\_file*

Uses the corresponding times of the file named by *ref\_file* instead of the current time.

操作数 The following operands are supported for the touch and settime utilities:

*file*

A path name of a file whose times are to be modified.

*time\_spec*

Uses the specified *time\_spec* instead of the current time. This operand is a decimal number of the form:

*MMDDhhmm[YY]*

where each two digits represent the following:

*MM*

The month of the year [01-12].

*DD*

The day of the month [01-31].

*hh*

The hour of the day [00-23].

*mm*

The minute of the hour [00-59].

*YY*

The second two digits of the year.

*YY* is optional. If it is omitted, the current year is assumed. If *YY* is specified, the year is derived as follows:

| YY    | Corresponding Year |
|-------|--------------------|
| 69-99 | 1969-1999          |
| 00-38 | 2000-2038          |
| 39-68 | ERROR              |

If no `-d`, `-r`, or `-t` option is specified, at least two operands are specified, and the first operand is an eight- or ten-digit decimal integer, the first operand is assumed to be a *time\_spec* operand. Otherwise, the first operand is assumed to be a *file* operand.

## 用法

See [largefile\(5\)](#) for the description of the behavior of touch when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of touch: LANG, LC\_ALL, LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

### TZ

Determine the timezone to be used for interpreting the *time* or *date\_time* option-argument or the *time\_spec* operand.

## 退出状态

The following exit values are returned:

0

The touch utility executed successfully and all requested changes were made.

>0

An error occurred. The touch utility returned the number of files for which the times could not be successfully modified.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

## 另请参见

[futimens\(2\)](#), [stat\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

## 附注

Users familiar with the BSD environment find that for the touch utility, the `-f` option is accepted but ignored. The `-f` option is unnecessary because touch succeeds for all files owned by the user regardless of the permissions on the files.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | touch – change file access and modification times                                                                                                                                                                                                                                                                                                                                                              |
| 用法概要 | <code>/usr/ucb/touch [-acfm] file...</code>                                                                                                                                                                                                                                                                                                                                                                    |
| 描述   | touch sets the access and modification times of each file to the current time. file is created if it does not already exist.                                                                                                                                                                                                                                                                                   |
| 选项   | <ul style="list-style-type: none"> <li>-a Change the access time of file. Do not change the modification time unless -m is also specified.</li> <li>-c Do not create file if it does not exist.</li> <li>-f Attempt to force the touch in spite of read and write permissions on file.</li> <li>-m Change the modification time of file. Do not change the access time unless -a is also specified.</li> </ul> |
| 用法   | See <a href="#">largefile(5)</a> for the description of the behavior of touch when encountering files greater than or equal to 2 Gbyte ( $2^{31}$ bytes).                                                                                                                                                                                                                                                      |
| 退出状态 | <p>The following exit values are returned:</p> <ul style="list-style-type: none"> <li>0 touch executed successfully and all requested changes were made.</li> <li>&gt;0 An error occurred. touch returns the number of files for which the times could not be successfully modified.</li> </ul>                                                                                                                |
| 属性   | See <a href="#">attributes(5)</a> for descriptions of the following attributes:                                                                                                                                                                                                                                                                                                                                |

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

另请参见 [touch\(1\)](#), [attributes\(5\)](#), [largefile\(5\)](#)

**引用名** tplot, t300, t300s, t4014, t450, tek, ver – 不同绘图仪的图形过滤器

**用法概要** /usr/bin/tplot [-T *terminal*]

**描述** tplot 读取标准输入中的绘图说明，然后在标准输出中生成适用于特定 *terminal* 的绘图说明。

如果未指定 *terminal*，则使用环境变量 TERM。缺省 *terminal* 是 tek。

**文件**

- /usr/lib/t300
- /usr/lib/t300s
- /usr/lib/t4014
- /usr/lib/t450
- /usr/lib/tek
- /usr/lib/vplot

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

**另请参见** [vi\(1\)](#)、[attributes\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | tput – initialize a terminal or query terminfo database                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 用法概要 | tput [-T <i>type</i> ] <i>capname</i> [ <i>parm</i> ]...<br>tput -S <<                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 描述   | The tput utility uses the terminfo database to make the values of terminal-dependent capabilities and information available to the shell (see <a href="#">sh(1)</a> ); to clear, initialize or reset the terminal; or to return the long name of the requested terminal type. tput outputs a string if the capability attribute ( <i>capname</i> ) is of type string, or an integer if the attribute is of type integer. If the attribute is of type boolean, tput simply sets the exit status (0 for TRUE if the terminal has the capability, 1 for FALSE if it does not), and produces no output. Before using a value returned on standard output, the user should test the exit status (\$?, see <a href="#">sh(1)</a> ) to be sure it is 0. See the EXIT STATUS section.                                                                                                                                                                                           |
| 选项   | The following options are supported:<br><br>-T <i>type</i> Indicates the <i>type</i> of terminal. Normally this option is unnecessary, because the default is taken from the environment variable TERM. If -T is specified, then the shell variables LINES and COLUMNS and the layer size will not be referenced.<br><br>-S            Allows more than one capability per invocation of tput. The capabilities must be passed to tput from the standard input instead of from the command line (see the example in the EXAMPLES section). Only one <i>capname</i> is allowed per line. The -S option changes the meaning of the 0 and 1 boolean and string exit statuses (see the EXAMPLES section).                                                                                                                                                                                                                                                                   |
| 操作数  | The following operands are supported:<br><br><i>capname</i> Indicates the capability attribute from the terminfo database. See <a href="#">terminfo(4)</a> for a complete list of capabilities and the <i>capname</i> associated with each.<br><br>The following strings will be supported as operands by the implementation in the "C" locale:<br><br>clear            Display the clear-screen sequence.<br><br>init            If the terminfo database is present and an entry for the user's terminal exists (see -T <i>type</i> , above), the following will occur:<br><ol style="list-style-type: none"><li>1. if present, the terminal's initialization strings will be output (<i>is1</i>, <i>is2</i>, <i>is3</i>, <i>if</i>, <i>iprog</i>),</li><li>2. any delays (for instance, newline) specified in the entry will be set in the tty driver,</li><li>3. tabs expansion will be turned on or off according to the specification in the entry, and</li></ol> |

4. if tabs are not expanded, standard tabs will be set (every 8 spaces). If an entry does not contain the information needed for any of the four above activities, that activity will silently be skipped.
- reset** Instead of putting out initialization strings, the terminal's reset strings will be output if present (*rs1*, *rs2*, *rs3*, *rf*). If the reset strings are not present, but initialization strings are, the initialization strings will be output. Otherwise, **reset** acts identically to **init**.
- longname** If the **terminfo** database is present and an entry for the user's terminal exists (see **-Ttype** above), then the long name of the terminal will be put out. The long name is the last name in the first line of the terminal's description in the **terminfo** database (see **term(5)**).
- parm** If the attribute is a string that takes parameters, the argument *parm* will be instantiated into the string. An all numeric argument will be passed to the attribute as a number.

## 示例

### 示例 1 Initializing the terminal according to TERM

This example initializes the terminal according to the type of terminal in the environment variable **TERM**. This command should be included in everyone's **.profile** after the environment variable **TERM** has been exported, as illustrated on the **profile(4)** manual page.

```
example% tput init
```

### 示例 2 Resetting a terminal

This example resets an AT&T 5620 terminal, overriding the type of terminal in the environment variable **TERM**:

```
example% tput -T5620 reset
```

### 示例 3 Moving the cursor

The following example sends the sequence to move the cursor to row 0, column 0 (the upper left corner of the screen, usually known as the "home" cursor position).

```
example% tput cup 0 0
```

This next example sends the sequence to move the cursor to row 23, column 4.

```
example% tput cup 23 4
```

**示例 4** Echoing the clear-screen sequence

This example echos the clear-screen sequence for the current terminal.

```
example% tput clear
```

**示例 5** Printing the number of columns

This command prints the number of columns for the current terminal.

```
example% tput cols
```

The following command prints the number of columns for the 450 terminal.

```
example% tput -T450 cols
```

**示例 6** Setting shell variables

This example sets the shell variables `bold`, to begin stand-out mode sequence, and `offbold`, to end standout mode sequence, for the current terminal. This might be followed by a prompt:

```
echo "${bold}Please type in your name: ${offbold}\c"
example% bold='tput smso'
example% offbold='tput rmso'
```

**示例 7** Setting the exit status

This example sets the exit status to indicate if the current terminal is a hardcopy terminal.

```
example% tput hc
```

**示例 8** Printing the long name from terminfo

This command prints the long name from the `terminfo` database for the type of terminal specified in the environment variable `TERM`.

```
example% tput longname
```

**示例 9** Processing several capabilities with one invocation

This example shows `tput` processing several capabilities in one invocation. This example clears the screen, moves the cursor to position `10, 10` and turns on `bold` (extra bright) mode. The list is terminated by an exclamation mark (!) on a line by itself.

```
example% tput -S <<!
> clear
> cup 10 10
> bold
> !
```

**环境变量**

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `tput`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.



**TERM** Determine the terminal type. If this variable is unset or null, and if the `-T` option is not specified, an unspecified default terminal type will be used.

## 退出状态

The following exit values are returned:

0

- If *capname* is of type boolean and `-S` is not specified, indicates TRUE.
- If *capname* is of type string and `-S` is not specified, indicates *capname* is defined for this terminal type.
- If *capname* is of type boolean or string and `-S` is specified, indicates that all lines were successful.
- *capname* is of type integer.
- The requested string was written successfully.

1

- If *capname* is of type boolean and `-S` is not specified, indicates FALSE.
- If *capname* is of type string and `-S` is not specified, indicates that *capname* is not defined for this terminal type.

2 Usage error.

3 No information is available about the specified terminal type.

4 The specified operand is invalid.

>4 An error occurred.

-1 *capname* is a numeric variable that is not specified in the terminfo database. For instance, `tput -T450 lines` and `tput -T2621 xmc`.

## 文件

`/usr/include/curses.h` [curses\(3CURSES\)](#) header

`/usr/include/term.h` terminfo header

`/usr/lib/tabset/*` Tab settings for some terminals, in a format appropriate to be output to the terminal (escape sequences that set margins and tabs). For more information, see the "Tabs and Initialization" section of [terminfo\(4\)](#)

`/usr/share/lib/terminfo/??/*` compiled terminal description database

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

另请参见

[clear\(1\)](#), [sh\(1\)](#), [stty\(1\)](#), [tabs\(1\)](#), [curses\(3CURSES\)](#), [profile\(4\)](#), [terminfo\(4\)](#),  
[attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#), [term\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | tr – translate characters                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 用法概要 | <pre> /usr/bin/tr [-cds] [string1 [string2]] /usr/xpg4/bin/tr [-cs] string1 string2 /usr/xpg4/bin/tr -s   -d [-c] string1 /usr/xpg4/bin/tr -ds [-c] string1 string2 /usr/xpg6/bin/tr [-c   -C] [-s] string1 string2 /usr/xpg6/bin/tr -s [-c   -C] string1 /usr/xpg6/bin/tr -d [-c   -C] string1 /usr/xpg6/bin/tr -ds [-c   -C] string1 string2 </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 描述   | The <code>tr</code> utility copies the standard input to the standard output with substitution or deletion of selected characters. The options specified and the <code>string1</code> and <code>string2</code> operands control translations that occur while copying characters and single-character collating elements.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-c   Complements the set of values specified by <code>string1</code>.</li> <li>-C   Complements the set of characters specified by <code>string1</code>.</li> <li>-d   Deletes all occurrences of input characters that are specified by <code>string1</code>.</li> <li>-s   Replaces instances of repeated characters with a single character.</li> </ul> <p>When the <code>-d</code> option is not specified:</p> <ul style="list-style-type: none"> <li>▪ Each input character found in the array specified by <code>string1</code> is replaced by the character in the same relative position in the array specified by <code>string2</code>. When the array specified by <code>string2</code> is shorter than the one specified by <code>string1</code>, the results are unspecified.</li> <li>▪ If the <code>-c</code> option is specified, the complements of the values specified by <code>string1</code> are placed in the array in ascending order by binary value.</li> <li>▪ If the <code>-C</code> option is specified, the complements of the characters specified by <code>string1</code> (the set of all characters in the current character set, as defined by the current setting of <code>LC_CTYPE</code>, except for those actually specified in the <code>string1</code> operand) are placed in the array in ascending collation sequence, as defined by the current setting of <code>LC_COLLATE</code>.</li> <li>▪ Because the order in which characters specified by character class expressions or equivalence class expressions is undefined, such expressions should only be used if the intent is to map several characters into one. An exception is case conversion, as described previously.</li> </ul> <p>When the <code>-d</code> option is specified:</p> <ul style="list-style-type: none"> <li>▪ Input characters found in the array specified by <code>string1</code> are deleted.</li> </ul> |

- When the `-C` option is specified with `-d`, all values except those specified by *string1* are deleted. The contents of *string2* are ignored, unless the `-s` option is also specified.
- If the `-c` option is specified, the complements of the values specified by *string1* are placed in the array in ascending order by binary value.
- The same string cannot be used for both the `-d` and the `-s` option. When both options are specified, both *string1* (used for deletion) and *string2* (used for squeezing) are required.

When the `-s` option is specified, after any deletions or translations have taken place, repeated sequences of the same character is replaced by one occurrence of the same character, if the character is found in the array specified by the last operand. If the last operand contains a character class, such as the following example:

```
tr -s '[:space:]'
```

the last operand's array contains all of the characters in that character class. However, in a case conversion, as described previously, such as

```
tr -s '[:upper:]' '[:lower:]'
```

the last operand's array contains only those characters defined as the second characters in each of the `toupper` or `tolower` character pairs, as appropriate. (See [toupper\(3C\)](#) and [tolower\(3C\)](#)).

An empty string used for *string1* or *string2* produces undefined results.

## 操作数

The following operands are supported:

*string1*

*string2* Translation control strings. Each string represents a set of characters to be converted into an array of characters used for the translation.

The operands *string1* and *string2* (if specified) define two arrays of characters. The constructs in the following list can be used to specify characters or single-character collating elements. If any of the constructs result in multi-character collating elements, `tr` excludes, without a diagnostic, those multi-character elements from the resulting array.

*character* Any character not described by one of the conventions below represents itself.

*\octal*

Octal sequences can be used to represent characters with specific coded values. An octal sequence consists of a backslash followed by the longest sequence of one-, two-, or three-octal-digit characters (01234567). The sequence causes the character whose encoding is represented by the one-, two- or three-digit octal integer to be placed into the array. Multi-byte characters require multiple, concatenated escape sequences of this type, including the leading `\` for each byte.

`\character` The backslash-escape sequences `\a`, `\b`, `\f`, `\n`, `\r`, `\t`, and `\v` are supported. The results of using any other character, other than an octal digit, following the backslash are unspecified.

`/usr/xpg4/bin/tr`

`c-c`

`/usr/bin/tr`

`[c-c]`

In the POSIX locale, this construct represents the range of collating elements between the range endpoints (as long as neither endpoint is an octal sequence of the form `\octal`), inclusively, as defined by the collation sequence. The characters or collating elements in the range are placed in the array in ascending collation sequence. If the second endpoint precedes the starting endpoint in the collation sequence, it is unspecified whether the range of collating elements is empty, or this construct is treated as invalid. In locales other than the POSIX locale, this construct has unspecified behavior.

If either or both of the range endpoints are octal sequences of the form `\octal`, represents the range of specific coded binary values between two range endpoints, inclusively.

`[ :class: ]`

Represents all characters belonging to the defined character class, as defined by the current setting of the LC\_CTYPE locale category. The following character class names are accepted when specified in *string1*:

```
alnum blank digit lower punct upper
alpha cntrl graph print space xdigit
```

In addition, character class expressions of the form `[ :name: ]` are recognized in those locales where the *name* keyword has been given a `charclass` definition in the LC\_CTYPE category.

When both the `-d` and `-s` options are specified, any of the character class names are accepted in *string2*. Otherwise, only character class names `lower` or `upper` are valid in *string2* and then only if the corresponding character class `upper` and `lower`, respectively, is specified in the same relative position in *string1*. Such a specification is interpreted as a request for case conversion. When `[ :lower: ]` appears in *string1* and `[ :upper: ]` appears in *string2*, the arrays contain the characters from the `toupper` mapping in the LC\_CTYPE category of the current locale. When `[ :upper: ]` appears in *string1* and `[ :lower: ]` appears in *string2*, the arrays contain the characters from the `tolower` mapping in the LC\_CTYPE category of the current locale. The first character from each mapping pair is in the array for *string1* and the second character from each mapping pair is in the array for *string2* in the same relative position.

Except for case conversion, the characters specified by a character class expression are placed in the array in an unspecified order.

If the name specified for *class* does not define a valid character class in the current locale, the behavior is undefined.

`[=equiv=]` Represents all characters or collating elements belonging to the same equivalence class as *equiv*, as defined by the current setting of the LC\_COLLATE locale category. An equivalence class expression is allowed only in *string1*, or in *string2* when it is being used by the combined `-d` and `-s` options. The characters belonging to the equivalence class are placed in the array in an unspecified order.

`[x*n]` Represents *n* repeated occurrences of the character *x*. Because this expression is used to map multiple characters to one, it is only valid when it occurs in *string2*. If *n* has a leading 0, it is interpreted as an octal value. Otherwise, it is interpreted as a decimal value.

If *n* is omitted or is 0, `/usr/bin/tr` interprets this as huge; `/usr/xpg4/bin/tr` and `/usr/xpg6/bin/tr` interprets this as large enough to extend the *string2*-based sequence to the length of the *string1*-based sequence.

**用法** See [largefile\(5\)](#) for the description of the behavior of `tr` when encountering files greater than or equal to 2 Gbyte (  $2^{31}$  bytes).

**示例** 示例 1 Creating a list of words

The following example creates a list of all words in *file1*, one per line in *file2*, where a word is taken to be a maximal string of letters.

```
tr -cs "[:alpha:]" "[\n*]" <file1 >file2
```

示例 2 Translating characters

This example translates all lower-case characters in *file1* to upper-case and writes the results to standard output.

```
tr "[:lower:]" "[:upper:]" <file1
```

Notice that the caveat expressed in the corresponding example in XPG3 is no longer in effect. This case conversion is now a special case that employs the `tolower` and `toupper` classifications, ensuring that proper mapping is accomplished (when the locale is correctly defined).

示例 3 Identifying equivalent characters

This example uses an equivalence class to identify accented variants of the base character *e* in *file1*, which are stripped of diacritical marks and written to *file2*.

```
tr "[=e=]" e <file1 >file2
```

**环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `tr`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

**退出状态** The following exit values are returned:

- `0` All input was processed successfully.
- `>0` An error occurred.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/tr`

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |
| CSI            | Enabled         |

`/usr/xpg4/bin/tr`

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/xopen/xcu4                  |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

`/usr/xpg6/bin/tr`

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/xopen/xcu6                  |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见** [ed\(1\)](#), [sed\(1\)](#), [sh\(1\)](#), [tolower\(3C\)](#), [toupper\(3C\)](#), [ascii\(5\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [regex\(5\)](#), [standards\(5\)](#)

**附注** Unlike some previous versions, `/usr/xpg4/bin/tr` correctly processes NUL characters in its input stream. NUL characters can be stripped by using `tr -d '\000'`.

引用名 tr – translate characters

用法概要 /usr/ucb/tr [-cds] [*string1* [*string2*]]

描述 The `tr` utility copies the standard input to the standard output with substitution or deletion of selected characters. The arguments *string1* and *string2* are considered sets of characters. Any input character found in *string1* is mapped into the character in the corresponding position within *string2*. When *string2* is short, it is padded to the length of *string1* by duplicating its last character.

In either string the notation:

*a–b*

denotes a range of characters from *a* to *b* in increasing ASCII order. The character `\`, followed by 1, 2 or 3 octal digits stands for the character whose ASCII code is given by those digits. As with the shell, the escape character `\`, followed by any other character, escapes any special meaning for that character.

选项 Any combination of the options `-c`, `-d`, or `-s` may be used:

- c Complement the set of characters in *string1* with respect to the universe of characters whose ASCII codes are 01 through 0377 octal.
- d Delete all input characters in *string1*.
- s Squeeze all strings of repeated output characters that are in *string2* to single characters.

示例 示例 1 Creating a list of all the words in a filename

The following example creates a list of all the words in *filename1*, one per line, in *filename2*, where a word is taken to be a maximal string of alphabets. The second string is quoted to protect `\` from the shell. 012 is the ASCII code for NEWLINE.

```
example% tr -cs A-Za-z '\012' < filename1 > filename2
```

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

另请参见 [ed\(1\)](#), [ascii\(5\)](#), [attributes\(5\)](#)

附注 Will not handle ASCII NUL in *string1* or *string2*. `tr` always deletes NUL from input.



引用名                    trap, onintr – shell built-in functions to respond to (hardware) signals

## 用法概要

```
sh trap [argument n [n2]...]
csh onintr [-| label]
ksh88 *trap [arg sig [sig2...]]
ksh +trap [-p] [action condition...]
```

## 描述

sh                    The `trap` command *argument* is to be read and executed when the shell receives numeric or symbolic signal(s) (*n*). (Note: *argument* is scanned once when the trap is set and once when the trap is taken.) Trap commands are executed in order of signal number or corresponding symbolic names. Any attempt to set a trap on a signal that was ignored on entry to the current shell is ineffective. An attempt to trap on signal 11 (memory fault) produces an error. If *argument* is absent all trap(s) *n* are reset to their original values. If *argument* is the null string this signal is ignored by the shell and by the commands it invokes. If *n* is 0 the command *argument* is executed on exit from the shell. The `trap` command with no arguments prints a list of commands associated with each signal number.

csh                   `onintr` controls the action of the shell on interrupts. With no arguments, `onintr` restores the default action of the shell on interrupts. (The shell terminates shell scripts and returns to the terminal command input level). With the `-` argument, the shell ignores all interrupts. With a *label* argument, the shell executes a `goto label` when an interrupt is received or a child process terminates because it was interrupted.

ksh88                `trap` uses *arg* as a command to be read and executed when the shell receives signal(s) *sig*. *arg* is scanned once when the trap is set and once when the trap is taken. Each *sig* can be specified as a number or as the name of the signal. `trap` commands are executed in order of signal number. Any attempt to set a trap on a signal that was ignored on entry to the current shell is ineffective. If *arg* is omitted or is `-`, then the trap(s) for each *sig* are reset to their original values. If *arg* is the null (the empty string, for example, `""`) string then this signal is ignored by the shell and by the commands it invokes. If *sig* is ERR then *arg* are executed whenever a command has a non-zero exit status. If *sig* is DEBUG then *arg* are executed after each command. If *sig* is 0 or EXIT for a `trap` set outside any function then the command *arg* is executed on exit from the shell. The `trap` command with no arguments prints a list of commands associated with each signal number.

On this manual page, [ksh88\(1\)](#) commands that are preceded by one or two \* (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.

3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by **\*\*** that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.

**ksh** `trap` is a special built-in that defines actions to be taken when conditions such as receiving a signal occur. `trap` can also be used to display the current trap settings on standard output.

If *action* is `-`, `trap` resets each condition to the default value. If *action* is an empty string, the shell ignores each of the conditions if they arise. Otherwise, the argument *action* is read and executed by the shell as if it were processed by `eval` when one of the corresponding conditions arise. The action of the trap overrides any previous action associated with each specified condition. The value of `$_` is not altered by the trap execution.

*condition* can be the name or number of a signal, or one of the following:

- |                    |                                                                                                                                                                                                                                                                          |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>EXIT</code>  | Execute this trap when the shell exits. If defined within a function with the <code>function</code> reserved word, executes the trap in the caller's environment when the function returns. The trap action is restored to the value it had when it called the function. |
| <code>0</code>     | Same as <code>EXIT</code> .                                                                                                                                                                                                                                              |
| <code>DEBUG</code> | Execute before each simple command is executed but after the arguments are expanded.                                                                                                                                                                                     |
| <code>ERR</code>   | Execute whenever <code>set -e</code> would cause the shell to exit.                                                                                                                                                                                                      |
| <code>KEYBD</code> | Execute when a key is entered from a terminal device.                                                                                                                                                                                                                    |

Signal names are case insensitive and the `sig` prefix is optional. Signals that were ignored on entry to a non-interactive shell cannot be trapped or reset although doing so does not report an error. The use of signal numbers other than 1, 2, 3, 6, 9, 14, and 15 are not portable.

Although `trap` is a special built-in, specifying a condition that the shell does not know about causes `trap` to exit with a non-zero exit status, but does not terminate the invoking shell.

If no action or conditions are specified then all the current trap settings are written to standard output.

The following options are supported by the `trap` built-in command in `ksh`:

- `-p` Causes the current traps to be output in a format that can be processed as input to the shell to recreate the current traps.

The trap built-in in `ksh` exits with one of the following values:

- `0` Successful completion.

>0 An error occurred.

On this manual page, [ksh\(1\)](#) commands that are preceded by one or two + (plus signs) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. They are not valid function names.
5. Words, following a command preceded by ++ that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

## 另请参见

[csh\(1\)](#), [eval\(1\)](#), [exit\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | troff – typeset or format documents                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 用法概要 | troff [-a] [-f] [-Fdir] [-i] [-mname] [-nN] [-olist] [-raN]<br>[-sN] [-uN] [-z] [filename]...                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 描述   | troff formats text in the <i>filenames</i> for typesetting or laser printing. Input to troff is expected to consist of text interspersed with formatting requests and macros. If no <i>filename</i> argument is present, troff reads standard input. A minus sign (–) as a <i>filename</i> indicates that standard input should be read at that point in the list of input files.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 选项   | <p>The following options are supported. They may appear in any order, but all must appear before the first <i>filename</i>.</p> <ul style="list-style-type: none"> <li>-a            Send an ASCII approximation of formatted output to standard output. (Note: a rough ASCII version can also be printed out on ordinary terminals with an old and rarely used command, <code>/usr/bin/ta</code>.)</li> <li>-f            Do not print a trailer after the final page of output or cause the postprocessor to relinquish control of the device.</li> <li>-Fdir        Search directory <i>dir</i> for font width or terminal tables instead of the system default directory.</li> <li>-i            Read standard input after all input files are exhausted.</li> <li>-mname      Prepend the macro file <code>/usr/share/lib/tmac/name</code> to the input <i>filenames</i>. Note: most references to macro packages include the leading <i>m</i> as part of the name; for example, the <code>man(5)</code> macros reside in <code>/usr/share/lib/tmac/an</code>. The macro directory can be changed by setting the TROFFMACS environment variable to a specific path. Be certain to include the trailing <code>'/'</code> (slash) at the end of the path.</li> <li>-nN         Number the first generated page <i>N</i>.</li> <li>-olist      Print only pages whose page numbers appear in the comma-separated <i>list</i> of numbers and ranges. A range <i>N–M</i> means pages <i>N</i> through <i>M</i>; an initial <i>–N</i> means from the beginning to page <i>N</i>; and a final <i>N–</i> means from <i>N</i> to the end.</li> <li>-q           Quiet mode in <code>nroff</code>; ignored in troff.</li> <li>-raN        Set register <i>a</i> (one-character names only) to <i>N</i>.</li> <li>-sN         Stop the phototypesetter every <i>N</i> pages. On some devices, troff produces a trailer so you can change cassettes; resume by pressing the typesetter's start button.</li> <li>-uN         Set the emboldening factor for the font mounted in position 3 to <i>N</i>. If <i>N</i> is missing, then set the emboldening factor to 0.</li> <li>-z           Suppress formatted output. Only diagnostic messages and messages output using the <code>.tm</code> request are output.</li> </ul> |

- 操作数**           The following operand is supported:
- filename*    The file containing text to be processed by `troff`.
- 文件**
- `/tmp/trtmp`           temporary file
  - `/usr/share/lib/tmac/*`   standard macro files
  - `/usr/lib/font/*`       font width tables for alternate mounted `troff` fonts
  - `/usr/share/lib/nterm/*`   terminal driving tables for `nroff`
- 属性**            See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | text/doctools   |

**另请参见**       [checknr\(1\)](#), [col\(1\)](#), [eqn\(1\)](#), [man\(1\)](#), [nroff\(1\)](#), [tbl\(1\)](#), [attributes\(5\)](#), [man\(5\)](#), [me\(5\)](#), [ms\(5\)](#)

**附注**            `troff` is not 8-bit clean because it is by design based on 7-bit ASCII.

Previous documentation incorrectly described the numeric register `yr` as being the Last two digits of current year. `yr` is in actuality the number of years since 1900. To correctly obtain the last two digits of the current year through the year 2099, the definition given below of string register `yy` may be included in a document and subsequently used to display a two-digit year. Note that any other available one- or two-character register name may be substituted for `yy`.

```

.\" definition of new string register yy--last two digits of year
.\" use yr (# of years since 1900) if it is < 100
.ie \n(yr<100 .ds yy \n(yr
.el \{
 .\" else, subtract 100 from yr, store in ny
 .nr ny \n(yr-100
 .ie \n(ny>9 \{
 .\" use ny if it is two digits
 .ds yy \n(ny
 .\" remove temporary number register ny
 .rr ny \}
 .el \{.ds yy 0
 .\" if ny is one digit, append it to 0
 .as yy \n(ny
 .rr ny \} \}

```

引用名 true, false – provide truth values

用法概要 true  
false

描述 The `true` utility does nothing, successfully. The `false` utility does nothing, unsuccessfully. They are typically used in a shell script `sh` as:

```
while true
do
 command
done
```

which executes *command* forever.

退出状态 `true` has exit status 0.

`false` always will exit with a non-zero value.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

另请参见 [sh\(1\)](#), [attributes\(5\)](#), [standards\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | truss - 跟踪系统调用和信号                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 用法概要 | <pre>truss [-fcaeilDDE] [- [tTvX] [!] syscall ,...]       [- [sS] [!] signal ,...] [- [mM] [!] fault ,...]       [- [rw] [!] fd ,...]       [- [uU] [!] lib ,... : [:] [!] func ,...]       [-o outfile] command   -p pid[/lwps]...</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 描述   | <p><b>truss</b> 实用程序执行指定的命令并跟踪记录其执行的系统调用、其收到的信号以及其引发的计算机故障。跟踪输出的每一行都会报告故障或信号名称，或者系统调用名及其参数和返回值。如果可以，将使用相关系统头中的定义通过符号方式显示系统调用参数。对于任何路径名指针参数，将显示指向字符串。使用 <a href="#">Intro(3)</a> 中介绍的错误代码名称报告返回的错误。在出错情况下，如果内核报告缺少特权，则在错误代码名称后在方括号 [ ] 中报告 <a href="#">privileges(5)</a> 中介绍的特权名称。有关错误报告的更多信息，请参见“附注”部分。</p> <p>某些情况下（请参见 <code>-u</code> 选项），<b>truss</b> 还将记录由受跟踪进程执行的用户级函数调用的进入/退出，以缩进形式表示嵌套。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 选项   | <p>对于具有列表参数的选项，名称 <code>all</code> 可用作指定列表所有可能成员的快速方法。如果列表以 <code>!</code> 开头，则该选项的含义是否定的（如，排除而不是跟踪）。可以指定同一选项多次出现。对于列表中的相同名称，后续选项（右侧选项）将覆盖之前的选项（左侧选项）。</p> <p>支持以下选项：</p> <ul style="list-style-type: none"> <li><code>-a</code> 显示在各个 <code>exec()</code> 系统调用中传递的参数字符串。</li> <li><code>-c</code> 对跟踪的系统调用、故障和信号计数，而不是逐行显示跟踪。跟踪的命令终止后或 <code>truss</code> 被中断时，将生成汇总报告。如果还指定了 <code>-f</code>，则计数将包括子进程的所有受跟踪系统调用、故障和信号。</li> <li><code>-d</code> 在跟踪输出的每一行上包含时间戳。时间戳作为一个包含 <code>seconds . fraction</code> 的字段显示在行开头。这表示相对于跟踪开始的时间，以秒为单位。跟踪输出的第一行显示单个时间戳开始测量的基准时间，可以是某个时间点以来的秒数（请参见 <a href="#">time(2)</a>），也可以是日期字符串（请参见 <a href="#">ctime(3C)</a> 和 <a href="#">date(1)</a>）。报告的时间是正在谈及的事件发生的时间。对于所有系统共用，事件是系统调用的完成，而不是系统调用的开始。</li> <li><code>-D</code> 在跟踪输出的每一行上包含时间增量。该值显示为一个包含 <code>seconds . fraction</code> 的字段，表示自上次报告的由 LWP 引发的事件以来，引发该事件的 LWP 所用的时间。具体来说，对于系统调用，这不属于系统调用所花费的时间。</li> </ul> |

- e** 显示在各个 `exec()` 系统调用中传递的环境字符串。
- E** 在跟踪输出的每一行上包含时间增量。该值显示为一个包含 *seconds.fraction* 的字段，表示系统调用开始与结束之间所用的时间。
- 与 **-D** 选项相反，这是在系统调用内花费的时间量。
- f** 跟随 `fork()` 或 `vfork()` 创建的所有子项并在跟踪输出中包含其信号、故障和系统调用。通常，仅跟踪第一级命令或进程。如果指定了 **-f**，进程 ID 将包含在跟踪输出的各个行中，以指示哪个进程执行了系统调用或接收了信号。
- i** 不会显示可中断的休眠系统调用。某些系统调用（如终端设备或管道上的 `open()` 和 `read()`）可以无限期休眠，并可中断。`truss` 通常会在此类休眠系统调用保持休眠状态超过一秒时报告这些系统调用。系统调用在完成后会再次进行报告。**-i** 选项将使此类系统调用仅在完成时报告一次。
- l** 在跟踪输出的各个行中包含负责轻量级进程 (lightweight process, LWP) 的 ID。如果还指定了 **-f**，进程 ID 和 LWP ID 均会包含在内。
- m [!]fault,...** 要跟踪或排除的计算机故障。将跟踪在以逗号分隔的列表中指定的那些故障跟踪。故障可以按名称或编号进行指定（请参见 `<sys/fault.h>`）。如果列表以 **!** 开头，将从跟踪输出中排除指定的故障。缺省值为 `-mall -m !fltpage`。
- M [!]fault,...** 停止进程的计算机故障。将指定的故障添加至 **-m** 指定的集合中。如果引发指定的故障之一，则 `truss` 将保留该进程处于停止和弃用状态（请参见 **-T** 选项）。缺省值是 `-M!all`。
- o outfile** 要用于跟踪输出的文件。缺省情况下，输出为标准错误。
- p** 将 `truss` 的 *command* 参数解释为现有进程的进程 ID 列表（请参见 `ps(1)`），而不是要执行的命令。`truss` 控制各个进程，并在进程的用户 ID 和组 ID 与该用户的用户 ID 和组 ID 匹配或该用户是特权用户时，开始对该进程进行跟踪。用户可以通过将 `/thread-id` 附加到进程 ID 来仅跟踪选定的线程。使用 **-** 和 **,** 分隔符可以选定多个线程。例如，`/1,2,7-9` 将跟踪线程 1、2、7、8 以及 9。也可在目录 `/proc` 中使用进程名指定进程，例如 `/proc/12345`。



- `-r [!]fd,...` 显示任一指定文件描述符上每个 `read()` 的 I/O 缓冲区的完整内容。输出格式化为每行 32 字节，其中各个字节显示为 ASCII 字符（前面留有一个空白）或双字符 C 语言转义序列，以便控制诸如水平制表符 (`\t`) 和换行 (`\n`) 等的字符。如果不能进行 ASCII 解释，则字节将采用双字符十六进制表示。（每个受跟踪 `print >read()` 的 I/O 缓冲区的前 12 个字节即使在没有 `-r` 的情况下也会显示。）缺省值是 `-r!all`。
- `-s [!]signal,...` 要跟踪或排除的信号。将跟踪在逗号分隔的列表中指定的那些信号。跟踪输出将报告各个指定信号的接收情况，即使该信号受到忽略（未阻塞）。（不会收到阻塞的信号，直至将其解除阻塞。）信号可以按名称或编号进行指定（请参见 `<sys/signal.h>`）。如果列表以 `!` 开头，将从跟踪输出中排除指定的信号。缺省值是 `-sall`。
- `-S [!]signal,...` 停止进程的信号。将指定的信号添加至 `-s` 指定的集合中。如果收到一个指定信号，则 `truss` 将保留该进程处于停止和弃用状态（请参见 `-T` 选项）。缺省值是 `-S!all`。
- `-t [!]syscall ,...` 要跟踪或排除的系统调用。将跟踪在逗号分隔的列表中指定的这些系统调用。如果列表以 `!` 开头，则指定的系统调用将从跟踪输出中排除。缺省值是 `-tall`。
- `-T [!]syscall ,...` 指定停止进程的系统调用。将指定的系统调用添加至 `-t` 指定的集合中。如果遇到指定的系统调用之一，`truss` 将保留该进程处于停止和弃用状态。即，`truss` 将释放该进程并退出，但在涉及的系统调用完成时保留该进程处于停止状态。然后，可以向该停止的进程应用调试程序或其他进程检查工具（请参见 `proc(1)`）。可将具有相同或不同选项的 `truss` 重新应用于该停止的进程以继续跟踪。缺省值是 `-T!all`。
- 以此种方式保持停止状态的进程无法通过应用 `kill -CONT` 重新启动，因为它通过 `/proc` 基于特定事件停止的，而不是由停止信号的缺省操作停止的（请参见 `signal.h(3HEAD)`）。`prun(1)` 命令（`proc(1)` 中有述）可用于将已停止进程设置为再次运行。
- `-u [!]lib, ... :[:][!]func, ...` 用户级函数调用跟踪。`lib, ...` 是动态库名的逗号分隔列表，不包括 `".so.n"` 前缀。`func, ...` 是函数名的逗号分隔列表。在两种情况下，名称都可以包括名称匹配元字符 `*,?,[ ]`，它们与 `sh(1)` 的那些元字符具有相同意

义，但是它们应用于库/函数名称空间，而不是文件。空库或函数列表的缺省设置是\*，跟踪库中所有库或函数。两个列表中任一个上出现前导!将指定不受跟踪的排除列表、库或函数的名称。排除一个库将排除该库中的所有函数；库排除列表后的任何函数列表都将被忽略。

单：可以分离库列表与函数列表，意味着跟踪从库外到库内的调用，但忽略库中其他函数对同一库中函数的调用。双：:意味着跟踪所有调用，而不管来源。

库模式与可执行文件或动态链接程序均不匹配，除非存在确切匹配（l\*与ld.so.1不匹配）。要跟踪这些目标文件中的函数，必须准确指定名称，如：

```
truss -u a.out -u ld ...
```

a.out 是用于实现此目的的文字名称；其不代表可执行文件的名称。跟踪a.out函数调用意指所有调用（缺省值是::）。

可以指定多个-u选项，其遵循自左向右的顺序。执行函数调用的轻量级进程的ID和线程的ID包括在该调用的跟踪输出中，分别以斜杠(/)和at符号(@)字符分隔。

**-U [!]lib, ... :[:][!]func, ...**

停止进程的用户级函数调用。将指定的函数添加至-u指定的集合中。如果调用指定的函数之一，则truss将保留该进程处于停止和弃用状态（请参见-T选项）。

**-v [!]syscall, ...**

详细模式。显示按地址传递至指定系统调用（如果由-t跟踪）的任何结构的内容。将显示输入值以及由操作系统返回的值。对于用作输入和输出的任何字段，仅显示输出值。缺省值是-v!all。

**-w [!]fd, ...**

显示任一指定文件描述符上每个write()的I/O缓冲区的内容（请参见-r选项）。缺省值是-w!all。

**-x [!]syscall, ...**

以原始格式（通常为十六进制）而不是符号方式显示指定系统调用（如果由-t跟踪）的参数。有些技术高手希望看到原始位。缺省值是-x!all。

有关-t、-T、-v和-x接受的系统调用名称，请参见《[man pages section 2: System Calls](#)》。系统调用编号也会被接受。

如果使用 `truss` 初始化并跟踪指定的命令且使用了 `-o` 选项，或者如果将标准错误输出重定向至非终端文件，则 `truss` 在运行中将忽略挂起、中断和退出信号。这将简化对从终端捕捉中断和退出信号的交互式程序的跟踪。

如果跟踪输出保持定向至终端，或者如果现有进程受到跟踪（`-p` 选项），则 `truss` 将通过释放所有受跟踪的进程并退出来响应挂起、终端和退出信号。这使得用户能够终止过多的跟踪输出并释放之前就存在的进程。释放的进程将正常继续运行，好像从未被碰过一样。

跟踪现有进程时，`truss` 将在退出时释放这些进程并将其设置为运行状态。其中包括由于信号导致的退出，如 `SIGINT`、`SIGHUP` 或 `SIGQUIT`。这使得用户能够终止过多的跟踪输出并释放之前就存在的进程。释放的进程将正常继续运行，好像从未被碰过一样。

## 示例

### 示例1 跟踪命令

以下示例将跟踪记录终端上的 `find(1)` 命令：

```
example$ truss find . -print >find.out
```

### 示例2 跟踪常见系统调用

以下示例仅显示了打开、关闭、读取和写入系统调用的跟踪记录：

```
example$ truss -t open,close,read,write find . -print >find.out
```

### 示例3 跟踪 Shell 脚本

以下示例将跟踪记录文件 `truss.out` 上的 `spell(1)` 命令：

```
example$ truss -f -o truss.out spell document
```

`spell` 是一个 shell 脚本，因此需要 `-f` 标志以跟踪 shell 以及由此 shell 创建的进程。（该 `spell` 脚本将运行一个八进程的流水线。）

### 示例4 缩短输出

以下示例将缩短输出：

```
example$ truss nroff -mm document >nroff.out
```

因为 97% 的输出报告 `lseek()`、`read()` 和 `write()` 系统调用。要缩短输出：

```
example$ truss -t !lseek,read,write nroff -mm document >nroff.out
```

### 示例5 跟踪来自 C 库外部的库调用

以下示例将跟踪从 C 库外部对 C 库内任意函数进行的所有用户级调用：

```
example$ truss -u libc ...
```

**示例 6** 跟踪来自 C 库内部的库调用

以下示例包括从 C 库自身内对 C 库中函数进行的调用：

```
example$ truss -u libc:: ...
```

**示例 7** 跟踪非 C 库的库调用

以下示例将跟踪对除 C 库以外的任意库进行的所有用户级调用：

```
example$ truss -u '*' -u !libc ...
```

**示例 8** 跟踪 printf 和 scanf 函数调用

以下示例将跟踪对 C 库内所含 printf 和 scanf 系列中的函数进行的所有用户级调用：

```
example$ truss -u 'libc:*printf,*scanf' ...
```

**示例 9** 跟踪每个用户级函数调用

以下示例将跟踪从任意位置到任意位置的每个用户级函数调用：

```
example$ truss -u a.out -u ld:: -u :: ...
```

**示例 10** 详细跟踪系统调用

以下示例将详细跟踪进程 #1 `init(1M)` 的系统调用活动（如果您是特权用户）：

```
example# truss -p -v all 1
```

中断 truss 会将 init 返回至正常运行状态。

文件

/proc/\* 进程文件

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值            |
|------|----------------|
| 可用性  | system/core-os |

另请参见

[date\(1\)](#)、[find\(1\)](#)、[proc\(1\)](#)、[ps\(1\)](#)、[sh\(1\)](#)、[spell\(1\)](#)、[init\(1M\)](#)、[Intro\(3\)](#)、[exec\(2\)](#)、[fork\(2\)](#)、[lseek\(2\)](#)

《[man pages section 2: System Calls](#)》

附注

《[man pages section 2: System Calls](#)》中描述的一些系统调用与实际操作系统界面不同。请勿对与该文档中描述的跟踪输出之间的微小差异表示惊讶。

每个计算机故障（缺页除外）都会向引发该故障的 LWP 发送信号。已接收信号的报告紧跟在每个计算机故障（缺页除外）报告后，除非该信号被阻塞。

操作系统将对进程跟踪强制执行某些安全限制。尤其是，如果命令的目标文件（`a.out`）不能由用户读取，则该用户无法跟踪此命令；`set-uid` 和 `set-gid` 命令只能由特

权用户进行跟踪。除非由特权用户运行，否则 `truss` 将无法控制执行 `set-id` 或不可读目标文件的 `exec()` 的任何进程；此类进程将继续从 `exec()` 的点继续正常运行，但与 `truss` 无关。

为避免与其他控制进程冲突，如果通过 `/proc` 接口检测到某个进程正在由另一个进程控制，则 `truss` 不会跟踪该进程。这使得 `truss` 可以应用于基于 [proc\(4\)](#) 的调试器以及它自身的其他实例。

假设标准制表停止位置已设置的情况下（每八个位置），跟踪输出将包含制表符。

多个进程或多线程进程（包含多个 LWP 的进程）的跟踪输出不会按严格的时间顺序生成。例如，某个管道上的 `read()` 可能在对应的 `write()` 之前报告。对于任何一个 LWP（传统进程只包含一个），输出将严格按照时间顺序生成。

跟踪多个进程时，`truss` 将作为每个受跟踪进程的一个控制进程运行。对于上述 `spell` 命令的示例，`spell` 自身将使用 9 个进程槽，一个用于该 shell，8 个用于包含 8 个成员的流水线，同时 `truss` 会添加另外 9 个进程，所以总共是 18 个。

并非所有可能系统调用中传递的所有可能结构都会显示在 `-v` 选项之下。

当 `truss` 报告由于缺少特权导致系统调用返回错误时，`truss` 在错误代码后显示简单的特权名称，或复杂的特权描述。请参见 [privileges\(5\)](#)。该复杂的描述可以由以下内容组成：

- [ALL]           对于请求的操作，此进程需要所有特权。
- [MULTIPLE]     此进程缺少多个特权。
- [ZONE]          此进程缺少区域中的可用特权之一（ALL 的区域本地变体）
- [GLOBAL]        请求的操作要求进程在全局区域中运行。
- [MWAC]          请求的操作违反适用于进程的 [mwac\(5\)](#) 策略。

**引用名** tset, reset – establish or restore terminal characteristics

## 用法概要

```
tset [-InQrs] [-ec] [-kc]
 [-m [port-ID [baudrate] : type]...] [type]

reset [-] [-ec] [-I] [-kc]
 [-n] [-Q] [-r] [-s]
 [-m [indent] [test baudrate] : type]... [type]
```

## 描述

The `tset` utility sets up your terminal, typically when you first log in. It does terminal dependent processing such as setting erase and kill characters, setting or resetting delays, sending any sequences needed to properly initialize the terminal, and the like. `tset` first determines the type of terminal involved, and then does necessary initializations and mode settings. If a port is not wired permanently to a specific terminal (not hardwired) it is given an appropriate generic identifier such as `dia1up`.

`reset` clears the terminal settings by turning off CBREAK and RAW modes, output delays and parity checking, turns on NEWLINE translation, echo and TAB expansion, and restores undefined special characters to their default state. It then sets the modes as usual, based on the terminal type (which will probably override some of the above). See [stty\(1\)](#) for more information. All arguments to `tset` may be used with `reset`. `reset` also uses `rs=` and `rf=` to reset the initialization string and file. This is useful after a program dies and leaves the terminal in a funny state. Often in this situation, characters will not echo as you type them. You may have to type `LINEFEED reset LINEFEED` since RETURN may not work.

When no arguments are specified, `tset` reads the terminal type from the `TERM` environment variable and re-initializes the terminal, and performs initialization of mode, environment and other options at login time to determine the terminal type and set up terminal modes.

When used in a startup script (`.profile` for [sh\(1\)](#) users or `.login` for [csh\(1\)](#) users) it is desirable to give information about the type of terminal you will usually use on ports that are not hardwired. Any of the alternate generic names given in the file `/etc/termcap` are possible identifiers. Refer to the `-m` option below for more information. If no mapping applies and a final type option, not preceded by a `-m`, is given on the command line then that type is used.

It is usually desirable to return the terminal type, as finally determined by `tset`, and information about the terminal's capabilities, to a shell's environment. This can be done using the `-`, `-s`, or `-S` options.

For the Bourne shell, put this command in your `.profile` file:

```
eval 'tset -s options...'
```

or using the C shell, put these commands in your `.login` file:

```
set noglob
eval 'tset -s options...'unset noglob
```

With the C shell, it is also convenient to make an alias in your `.cshrc` file:

```
alias ts 'eval `tset -s \!*`'
```

This also allows the command:

```
ts 2621
```

to be invoked at any time to set the terminal and environment. It is not possible to get this aliasing effect with a Bourne shell script, because shell scripts cannot set the environment of their parent. If a process could set its parent's environment, none of this nonsense would be necessary in the first place.

Once the terminal type is known, `tset` sets the terminal driver mode. This normally involves sending an initialization sequence to the terminal, setting the single character erase (and optionally the line-kill (full line erase)) characters, and setting special character delays. TAB and NEWLINE expansion are turned off during transmission of the terminal initialization sequence.

On terminals that can backspace but not overstrike (such as a CRT), and when the erase character is `#`, the erase character is changed as if `-e` had been used.

## 选项

- The name of the terminal finally decided upon is output on the standard output. This is intended to be captured by the shell and placed in the `TERM` environment variable.
- ec Set the erase character to be the named character *c* on all terminals. Default is the BACKSPACE key on the keyboard, usually `^H` (CTRL-H). The character *c* can either be typed directly, or entered using the circumflex-character notation used here.
- ic Set the interrupt character to be the named character *c* on all terminals. Default is `^C` (CTRL-C). The character *c* can either be typed directly, or entered using the circumflex-character notation used here.
- I Suppress transmitting terminal-initialization strings.
- kc Set the line kill character to be the named character *c* on all terminals. Default is `^U` (CTRL-U). The kill character is left alone if `-k` is not specified. Control characters can be specified by prefixing the alphabetical character with a circumflex (as in CTRL-U) instead of entering the actual control key itself. This allows you to specify control keys that are currently assigned.
- n Specify that the new tty driver modes should be initialized for this terminal. Probably useless since `stty new` is the default.
- Q Suppress printing the 'Erase set to' and 'Kill set to' messages.
- r In addition to other actions, reports the terminal type.
- s Output commands to set and export `TERM`. This can be used with

```
set noglob
eval 'tset -s ...'
unset noglob
```

to bring the terminal information into the environment. Doing so makes programs such as `vi(1)` start up faster. If the SHELL environment variable ends with `cs`, C shell commands are output, otherwise Bourne shell commands are output.

`-m [port-ID [baudrate] : type] ...` Specify (map) a terminal type when connected to a generic port (such as *dialup* or *plugboard*) identified by *port-ID*. The *baudrate* argument can be used to check the baudrate of the port and set the terminal type accordingly. The target rate is prefixed by any combination of the following operators to specify the conditions under which the mapping is made:

- > Greater than
- @ Equals or “at”
- < Less than
- ! It is not the case that (negates the above operators)
- ? Prompt for the terminal type. If no response is given, then *type* is selected by default.

In the following example, the terminal type is set to `adm3a` if the port is a dialup with a speed of greater than 300 or to `dw2` if the port is a dialup at 300 baud or less. In the third case, the question mark preceding the terminal type indicates that the user is to verify the type desired. A NULL response indicates that the named type is correct. Otherwise, the user's response is taken to be the type desired.

```
tset -m 'dialup>300:adm3a' -m 'dialup<=300:dw2' -m 'plugboard:?adm3a'
```

To prevent interpretation as metacharacters, the entire argument to `-m` should be enclosed in single quotes. When using the C shell, exclamation points should be preceded by a backslash (`\`).

## 示例

These examples all use the `-s` option. A typical use of `tset` in a `.profile` or `.login` will also use the `-e` and `-k` options, and often the `-n` or `-Q` options as well. These options have been omitted here to keep the examples short.



**示例 1** Selecting a terminal

To select a 2621, you might put the following sequence of commands in your `.login` file (or `.profile` for Bourne shell users).

```
set noglob
eval 'tset -s 2621'
unset noglob
```

If you want to make the selection based only on the baud rate, you might use the following:

```
set noglob
eval 'tset -s -m '>1200:wy' 2621'
unset noglob
```

**示例 2** Selecting terminals according to speed or baud rate

If you have a switch which connects to various ports (making it impractical to identify which port you may be connected to), and use various terminals from time to time, you can select from among those terminals according to the *speed* or baud rate. In the example below, `tset` will prompt you for a terminal type if the baud rate is greater than 1200 (say, 9600 for a terminal connected by an RS-232 line), and use a Wyse® 50 by default. If the baud rate is less than or equal to 1200, it will select a 2621. Note the placement of the question mark, and the quotes to protect the `>` and `?` from interpretation by the shell.

```
set noglob
eval 'tset -s -m 'switch>1200:?wy' -m 'switch<=1200:2621''
unset noglob
```

**示例 3** Selecting the terminal used most often

The following entry is appropriate if you always dial up, always at the same baud rate, on many different kinds of terminals, and the terminal you use most often is an `adm3a`.

```
set noglob
eval 'tset -s ?adm3a'
unset noglob
```

**示例 4** Selecting a terminal with specific settings

The following example quietly sets the erase character to `BACKSPACE`, and kill to `CTRL-U`. If the port is switched, it selects a Concept™ 100 for speeds less than or equal to 1200, and asks for the terminal type otherwise (the default in this case is a Wyse 50). If the port is a direct dialup, it selects Concept 100 as the terminal type. If logging in over the ARPANET, the terminal type selected is a Datamedia® 2500 terminal or emulator. Note the backslash escaping the `NEWLINE` at the end of the first line in the example.

```
set noglob
eval 'tset -e -k^U -Q -s -m 'switch<=1200:concept100' -m\
'switch:?wy' -m dialup:concept100 -m arpanet:dm2500'
```

示例 4 Selecting a terminal with specific settings (续)

```
unset noglob
```

文件

```
.login
```

```
.profile
```

```
/etc/termcap
```

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

另请参见

[csh\(1\)](#), [sh\(1\)](#), [stty\(1\)](#), [vi\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注

The `tset` command is one of the first commands a user must master when getting started on a UNIX system. Unfortunately, it is one of the most complex, largely because of the extra effort the user must go through to get the environment of the `login` shell set. Something needs to be done to make all this simpler, either the `login` program should do this stuff, or a default shell alias should be made, or a way to set the environment of the parent should exist.

This program cannot intuit personal choices for erase, interrupt and line kill characters, so it leaves these set to the local system standards.

It could well be argued that the shell should be responsible for ensuring that the terminal remains in a sane state; this would eliminate the need for the `reset` program.

|      |                                                                                                                                                                                                                                                                                                                                                                                      |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | tsort – topological sort                                                                                                                                                                                                                                                                                                                                                             |
| 用法概要 | tsort [ <i>file</i> ]                                                                                                                                                                                                                                                                                                                                                                |
| 描述   | <p>The <code>tsort</code> command produces on the standard output a totally ordered list of items consistent with a partial ordering of items mentioned in the input <i>file</i>.</p> <p>The input consists of pairs of items (non-empty strings) separated by blanks. Pairs of different items indicate ordering. Pairs of identical items indicate presence, but not ordering.</p> |
| 操作数  | <p>The following operand is supported:</p> <p><i>file</i>    A path name of a text file to order. If no <i>file</i> operand is given, the standard input is used.</p>                                                                                                                                                                                                                |
| 示例   | <p>示例 1 An example of the <code>tsort</code> command</p> <p>The command:</p> <pre>example% <b>tsort</b> &lt;&lt;EOF <b>a b c c d e</b> <b>g g</b> <b>f g e f</b> <b>EOF</b></pre> <p>produces the output:</p> <pre>a b c d e f g</pre>                                                                                                                                               |
| 环境变量 | See <a href="#">environ(5)</a> for descriptions of the following environment variables that affect the execution of <code>tsort</code> : <code>LANG</code> , <code>LC_ALL</code> , <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>NLSPATH</code> .                                                                                                                     |
| 退出状态 | <p>The following exit values are returned:</p> <p>0        Successful completion.</p> <p>&gt;0      An error occurred.</p>                                                                                                                                                                                                                                                           |
| 属性   | See <a href="#">attributes(5)</a> for descriptions of the following attributes:                                                                                                                                                                                                                                                                                                      |

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

另请参见

[lorder\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断

Odd data: there are an odd number of fields in the input file.

|      |                                                                                                                                                                                                                                                                                                                     |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | tty – return user's terminal name                                                                                                                                                                                                                                                                                   |
| 用法概要 | /usr/bin/tty [-l] [-s]                                                                                                                                                                                                                                                                                              |
| 描述   | The <code>tty</code> utility writes to the standard output the name of the terminal that is open as standard input. The name that is used is equivalent to the string that would be returned by the <a href="#">ttyname(3C)</a> function.                                                                           |
| 选项   | The following options are supported: <ul style="list-style-type: none"> <li>-l Prints the synchronous line number to which the user's terminal is connected, if it is on an active synchronous line.</li> <li>-s Inhibits printing of the terminal path name, allowing one to test just the exit status.</li> </ul> |
| 环境变量 | See <a href="#">environ(5)</a> for descriptions of the following environment variables that affect the execution of <code>tty</code> : LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.                                                                                                                            |
| 退出状态 | The following exit values are returned: <ul style="list-style-type: none"> <li>0 Standard input is a terminal.</li> <li>1 Standard input is not a terminal.</li> <li>&gt;1 An error occurred.</li> </ul>                                                                                                            |
| 属性   | See <a href="#">attributes(5)</a> for descriptions of the following attributes:                                                                                                                                                                                                                                     |

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                |
|---------------------|--------------------------------|
| Availability        | system/core-os                 |
| CSI                 | Enabled                        |
| Interface Stability | Committed                      |
| Standard            | <a href="#">standards(5)</a> . |

|      |                                                                                                                                                                                                     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 另请参见 | <a href="#">isatty(3C)</a> , <a href="#">ttyname(3C)</a> , <a href="#">attributes(5)</a> , <a href="#">environ(5)</a> , <a href="#">standards(5)</a>                                                |
| 诊断   | <p>not on an active synchronous line    The standard input is not a synchronous terminal and -l is specified.</p> <p>not a tty    The standard input is not a terminal and -s is not specified.</p> |
| 附注   | The -s option is useful only if the exit status is wanted. It does not rely on the ability to form a valid path name. Portable applications should use <code>test -t</code> .                       |

**引用名** type – write a description of command type

**用法概要** type *name*...

**描述** The type utility indicates how each *name* operand would be interpreted if used as a command. type displays information about each operand identifying the operand as a shell built-in, function, alias, hashed command, or keyword, and where applicable, may display the operand's path name.

There is also a shell built-in version of type that is similar to the type utility.

**操作数** The following operand is supported:

*name* A name to be interpreted.

**环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of type: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

PATH Determine the location of *name*.

**退出状态** The following exit values are returned:

0 Successful completion.

>0 An error occurred.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见** [typeset\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | typeset, whence – shell built-in functions to set/get attributes and values for shell variables and functions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 用法概要 | typeset [ -CDHLRZfilrtux [ <i>n</i> ] [ <i>name</i> [= <i>value</i> ]]...<br>whence [-pv] <i>name</i> ...                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 描述   | <p>typeset sets attributes and values for shell variables and functions. When typeset is invoked inside a function, a new instance of the variables <i>name</i> is created. The variables <i>value</i> and type are restored when the function completes. The following list of attributes is supported:</p> <ul style="list-style-type: none"> <li>-C Compound variable. Each name is a compound variable. If <i>value</i> names a compound variable it is copied to <i>name</i>. Otherwise if the variable already exists, it is first to be unset</li> <li>-D Reserved for future use.</li> <li>-H Provide UNIX to hostname file mapping on non-UNIX machines.</li> <li>-L Left justify and remove leading blanks from value. If <i>n</i> is non-zero it defines the width of the field. Otherwise, it is determined by the width of the value of first assignment. When the variable is assigned to, it is filled on the right with blanks or truncated, if necessary, to fit into the field. Leading zeros are removed if the -Z flag is also set. The -R flag is turned off.</li> <li>-R Right justify and fill with leading blanks. If <i>n</i> is non-zero it defines the width of the field, otherwise it is determined by the width of the value of first assignment. The field is left filled with blanks or truncated from the end if the variable is reassigned. The -L flag is turned off.</li> <li>-Z Right justify and fill with leading zeros if the first non-blank character is a digit and the -L flag has not been set. If <i>n</i> is non-zero it defines the width of the field. Otherwise, it is determined by the width of the value of first assignment.</li> <li>-f All uppercase characters are converted to lowercase. The uppercase flag, -u is turned off.</li> </ul> <p>The FPATH variable is searched to find the function definition when the function is referenced. The flag -x allows the function definition to remain in effect across shell procedures invoked by name.</p> <ul style="list-style-type: none"> <li>-i Parameter is an integer. This makes arithmetic faster. If <i>n</i> is non-zero it defines the output arithmetic base. Otherwise, the first assignment determines the output base.</li> <li>-l All uppercase characters are converted to lowercase. The uppercase flag, -u is turned off.</li> <li>-m Move. The value is the name of a variable whose value is moved to name. The original variable is unset. Cannot be used with any other options.</li> </ul> |

- r The specified names are marked read-only and these names cannot be changed by subsequent assignment.
- t Tags the variables. Tags are user definable and have no special meaning to the shell.
- u All lowercase characters are converted to uppercase characters. The lowercase flag, -l is turned off.
- x The specified names are marked for automatic export to the environment of subsequently-executed commands.

The *i* attribute can not be specified along with -R, -L, -Z, or -f.

Using + rather than - causes these flags to be turned off. If no name arguments are specified but flags are specified, a list of names (and optionally the values) of the variables which have these flags set is printed. Using + rather than - keeps the values from being printed. If no names and flags are specified, the names and attributes of all variables are printed.

For each name, whence indicates how it would be interpreted if used as a command name.

The -v flag produces a more verbose report.

The -p flag does a path search for *name* even if *name* is an alias, a function, or a reserved word.

On this manual page, [ksh\(1\)](#) commands that are preceded by one or two \* (asterisks) are treated specially in the following ways:

1. Variable assignment lists preceding the command remain in effect when the command completes.
2. I/O redirections are processed after variable assignments.
3. Errors cause a script that contains them to abort.
4. Words, following a command preceded by \*\* that are in the format of a variable assignment, are expanded with the same rules as a variable assignment. This means that tilde substitution is performed after the = sign and word splitting and file name generation are not performed.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

## 另请参见

[ksh\(1\)](#), [ksh88\(1\)](#), [set\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#)



|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | ul – do underlining                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 用法概要 | ul [-i] [-t <i>terminal</i> ] [ <i>filename</i> ]...                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 描述   | ul reads the named <i>filenames</i> (or the standard input if none are given) and translates occurrences of underscores to the sequence which indicates underlining for the terminal in use, as specified by the environment variable TERM. ul uses the /usr/share/lib/terminfo entry to determine the appropriate sequences for underlining. If the terminal is incapable of underlining, but is capable of a standout mode then that is used instead. If the terminal can overstrike, or handles underlining automatically, ul degenerates to <a href="#">cat(1)</a> . If the terminal cannot underline, underlining is ignored. |
| 选项   | <p>-t <i>terminal</i>    Override the terminal kind specified in the environment. If the terminal cannot underline, underlining is ignored. If the terminal name is not found, no underlining is attempted.</p> <p>-i                Indicate underlining by a separate line containing appropriate dashes ‘-’; this is useful when you want to look at the underlining which is present in an <a href="#">nroff(1)</a> output stream on a CRT-terminal.</p>                                                                                                                                                                       |
| 返回值  | ul returns exit code 1 if the file specified is not found.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 文件   | /usr/share/lib/terminfo/*                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 属性   | See <a href="#">attributes(5)</a> for descriptions of the following attributes:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | text/doctools   |

另请参见 [cat\(1\)](#), [man\(1\)](#), [nroff\(1\)](#), [attributes\(5\)](#)

已知问题  
nroff usually generates a series of backspaces and underlines intermixed with the text to indicate underlining. ul makes attempt to optimize the backward motion.

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名   | umask – get or set the file mode creation mask                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 用法概要  | <code>/usr/bin/umask [-S] [<i>mask</i>]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| sh    | <code>umask [<i>ooo</i>]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| csch  | <code>umask [<i>ooo</i>]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| ksh88 | <code>umask [-S] [<i>mask</i>]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| ksh   | <code>umask [-S] [<i>mask</i>]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 描述    | <p>The <code>umask</code> utility sets the file mode creation mask of the current shell execution environment to the value specified by the <i>mask</i> operand. This mask affects the initial value of the file permission bits of subsequently created files. If <code>umask</code> is called in a subshell or separate utility execution environment, such as one of the following:</p> <pre>(umask 002) nohup umask ... find . -exec umask ...</pre> <p>it does not affect the file mode creation mask of the caller's environment. For this reason, the <code>/usr/bin/umask</code> utility cannot be used to change the <code>umask</code> in an ongoing session. Its usefulness is limited to checking the caller's <code>umask</code>. To change the <code>umask</code> of an ongoing session you must use one of the shell builtins.</p> <p>If the <i>mask</i> operand is not specified, the <code>umask</code> utility writes the value of the invoking process's file mode creation mask to standard output.</p>                                               |
| sh    | <p>The user file-creation mode mask is set to <i>ooo</i>. The three octal digits refer to read/write/execute permissions for owner, group, and other, respectively (see <a href="#">chmod(1)</a>, <a href="#">chmod(2)</a>, and <a href="#">umask(2)</a>). The value of each specified digit is subtracted from the corresponding "digit" specified by the system for the creation of a file (see <a href="#">creat(2)</a>). For example, <code>umask 022</code> removes write permission for group and other. Files (and directories) normally created with mode 777 become mode 755. Files (and directories) created with mode 666 become mode 644.</p> <ul style="list-style-type: none"><li>▪ If <i>ooo</i> is omitted, the current value of the mask is printed.</li><li>▪ <code>umask</code> is recognized and executed by the shell.</li><li>▪ <code>umask</code> can be included in the user's <code>.profile</code> (see <a href="#">profile(4)</a>) and invoked at login to automatically set the user's permissions on files or directories created.</li></ul> |
| csch  | See the description above for the Bourne shell (sh) <code>umask</code> built-in.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| ksh88 | The user file-creation mask is set to <i>mask</i> . <i>mask</i> can either be an octal number or a symbolic value as described in <a href="#">chmod(1)</a> . If a symbolic value is given, the new <code>umask</code> value is the complement of the result of applying <i>mask</i> to the complement of the previous <code>umask</code> value. If <i>mask</i> is omitted, the current value of the mask is printed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

ksh           umask sets the file creation mask of the current shell execution environment to the value specified by the *mask* operand. This mask affects the file permission bits of subsequently created files. *mask* can either be an octal number or a symbolic value as described in [chmod\(1\)](#). If a symbolic value is specified, the new file creation mask is the complement of the result of applying *mask* to the complement of the current file creation mask. If *mask* is not specified, umask writes the value of the file creation mask for the current process to standard output.

## 选项

ksh88           The following option is supported for `/usr/bin/umask` and `umask` in ksh88:

-S     Produces symbolic output.

The default output style is unspecified, but will be recognized on a subsequent invocation of `umask` on the same system as a *mask* operand to restore the previous file mode creation mask.

ksh           The following option is supported in ksh:

-S     Causes the file creation mask to be written or treated as a symbolic value rather than an octal number.

## 操作数

The following operand is supported:

*mask*     A string specifying the new file mode creation mask. The string is treated in the same way as the *mode* operand described in the [chmod\(1\)](#) manual page.

For a *symbolic\_mode* value, the new value of the file mode creation mask is the logical complement of the file permission bits portion of the file mode specified by the *symbolic\_mode* string.

In a *symbolic\_mode* value, the permissions *op* characters + and – are interpreted relative to the current file mode creation mask. + causes the bits for the indicated permissions to be cleared in the mask. – causes the bits of the indicated permissions to be set in the mask.

The interpretation of *mode* values that specify file mode bits other than the file permission bits is unspecified.

The file mode creation mask is set to the resulting numeric value.

The default output of a prior invocation of `umask` on the same system with no operand will also be recognized as a *mask* operand. The use of an operand obtained in this way is not obsolescent, even if it is an octal number.

## Output

When the *mask* operand is not specified, the `umask` utility will write a message to standard output that can later be used as a `umask mask` operand.

If `-S` is specified, the message will be in the following format:

```
"u=%s,g=%s,o=%s\n", owner permissions, group permissions, \
 other permissions
```

where the three values will be combinations of letters from the set `{r, w, x}`. The presence of a letter will indicate that the corresponding bit is clear in the file mode creation mask.

If a *mask* operand is specified, there will be no output written to standard output.

## 示例

示例 1 Using the `umask` Command

The examples in this section refer to the `/usr/bin/umask` utility and the `ksh88 umask` builtin.

Either of the commands:

```
umask a=rx,ug+w
umask 002
```

sets the mode mask so that subsequently created files have their `S_IWOTH` bit cleared.

After setting the mode mask with either of the above commands, the `umask` command can be used to write the current value of the mode mask:

```
example$ umask
0002
```

The output format is unspecified, but historical implementations use the obsolescent octal integer mode format.

```
example$ umask -S
u=rwx,g=rwx,o=rx
```

Either of these outputs can be used as the *mask* operand to a subsequent invocation of the `umask` utility.

Assuming the mode mask is set as above, the command:

```
umask g-w
```

sets the mode mask so that subsequently created files have their `S_IWGRP` and `S_IWOTH` bits cleared.

The command:

```
umask --w
```

sets the mode mask so that subsequently created files have all their write bits cleared. Notice that *mask* operands `r`, `w`, `x`, or anything beginning with a hyphen (`-`), must be preceded by `-` to keep it from being interpreted as an option.

**环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `umask`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

**退出状态** The following exit values are returned:

- `0` The file mode creation mask was successfully changed, or no *mask* operand was supplied.
- `>0` An error occurred.

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/umask`, `csh`,  
`ksh88`, `sh`

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

`ksh`

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE |
|---------------------|-----------------|
| Availability        | system/core-os  |
| Interface Stability | Volatile        |

**另请参见** [chmod\(1\)](#), [csh\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [sh\(1\)](#), [chmod\(2\)](#), [creat\(2\)](#), [umask\(2\)](#), [profile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

**引用名**                    `uname` – print name of current system

**用法概要**                `uname [-aimnprsvX]`

`uname [-S system_name]`

**描述**                    The `uname` utility prints information about the current system on the standard output. When options are specified, symbols representing one or more system characteristics will be written to the standard output. If no options are specified, `uname` prints the current operating system's name. The options print selected information returned by [uname\(2\)](#), [sysinfo\(2\)](#), or both.

**选项**                    The following options are supported:

- a  
Prints basic information currently available from the system.
- i  
Prints the name of the platform. For machines of the sun4v architecture, the `-i` option returns: sun4v. Use [prtconf\(1M\)](#) with the `-b` option to obtain the platform name for a sun4v machine.
- m  
Prints the machine hardware name (class). Use of this option is discouraged. Use `uname -p` instead. See NOTES section below.
- n  
Prints the nodename (the nodename is the name by which the system is known to a communications network).
- p  
Prints the current host's ISA or processor type.
- r  
Prints the operating system release level.
- s  
Prints the name of the operating system. This is the default.
- S *system\_name*  
The nodename may be changed by specifying a system name argument. The system name argument is restricted to `SYS_NMLN` characters. `SYS_NMLN` is an implementation specific value defined in `<sys/utsname.h>`. Only the super-user is allowed this capability. This change does not persist across reboots of the system.
- v  
Prints the operating system version.
- X  
Prints expanded system information, one information element per line, as expected by SCO UNIX. The displayed information includes:
  - system name, node, release, version, machine, and number of CPUs.

- BusType, Serial, and Users (set to unknown in Solaris)
- OEM# and Origin# (set to 0 and 1, respectively)

## 示例

示例 1 Printing the OS Name and Release Level

The following command prints the operating system name and release level, separated by one SPACE character:

```
example% uname -sr
```

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `uname`: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

## 退出状态

The following exit values are returned:

0

Successful completion.

>0

An error occurred.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

## 另请参见

[arch\(1\)](#), [isalist\(1\)](#), [prtconf\(1M\)](#), [sysinfo\(2\)](#), [uname\(2\)](#), [nodename\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

## 附注

Independent software vendors (ISVs) and others who need to determine detailed characteristics of the platform on which their software is either being installed or executed should use the `uname` command.

To determine the operating system name and release level, use `uname -sr`. To determine only the operating system release level, use `uname -r`. Notice that operating system release levels are not guaranteed to be in *x.y* format (such as 5.3, 5.4, 5.5, and so forth); future releases could be in the *x.y.z* format (such as 5.3.1, 5.3.2, 5.4.1, and so forth).

In SunOS 4.x releases, the [arch\(1\)](#) command was often used to obtain information similar to that obtained by using the `uname` command. The [arch\(1\)](#) command output `sun4` was often incorrectly interpreted to signify a SunOS SPARC system. If hardware platform information is desired, use `uname -sp`.

The `arch -k` and `uname -m` commands return equivalent values; however, the use of either of these commands by third party programs is discouraged, as is the use of the `arch` command in general. To determine the machine's Instruction Set Architecture (ISA or processor type), use `uname` with the `-p` option.



---

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | unifdef – resolve and remove ifdefed lines from C program source                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 用法概要 | unifdef [-c] [-Dname] [-Uname] [-iDname] [-iUname] ...<br>[filename]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 描述   | <p>unifdef removes ifdefed lines from a file while otherwise leaving the file alone. It is smart enough to deal with the nested ifdefs, comments, single and double quotes of C syntax, but it does not do any including or interpretation of macros. Neither does it strip out comments, though it recognizes and ignores them. You specify which symbols you want defined with -D options, and which you want undefined with -U options. Lines within those ifdefs will be copied to the output, or removed, as appropriate. Any ifdef, ifndef, else, and endif lines associated with <i>filename</i> will also be removed.</p> <p>ifdefs involving symbols you do not specify are untouched and copied out along with their associated ifdef, else, and endif lines.</p> <p>If an ifdefX occurs nested inside another ifdefX, then the inside ifdef is treated as if it were an unrecognized symbol. If the same symbol appears in more than one argument, only the first occurrence is significant.</p> <p>unifdef copies its output to the standard output and will take its input from the standard input if no <i>filename</i> argument is given.</p> |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-c            Complement the normal operation. Lines that would have been removed or blanked are retained, and vice versa.</li> <li>-l            Replace “lines removed” lines with blank lines.</li> <li>-t            Plain text option. unifdef refrains from attempting to recognize comments and single and double quotes.</li> <li>-Dname       Lines associated with the defined symbol <i>name</i>.</li> <li>-Uname       Lines associated with the undefined symbol <i>name</i>.</li> <li>-iDname      Ignore, but print out, lines associated with the defined symbol <i>name</i>. If you use ifdefs to delimit non-C lines, such as comments or code which is under construction, then you must tell unifdef which symbols are used for that purpose so that it will not try to parse for quotes and comments within them.</li> <li>-iUname      Ignore, but print out, lines associated with the undefined symbol <i>name</i>.</li> </ul>                                                                                                                    |
| 退出状态 | <p>The following exit values are returned:</p> <ul style="list-style-type: none"> <li>0            Successful operation.</li> <li>1            Operation failed.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE                    |
|----------------|------------------------------------|
| Availability   | developer/base-developer-utilities |

**另请参见**

[diff\(1\)](#), [attributes\(5\)](#)

**诊断**

Premature EOF    Inappropriate else or endif.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | uniq – report or filter out repeated lines in a file                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 用法概要 | <pre> /usr/bin/uniq [-c   -d   -u ] [ -f <i>fields</i>] [-s <i>char</i>] <input_file> [<i>output_file</i>]  /usr/bin/uniq [-c   -d   -u ] [-n ] [+<i>m</i> ] <input_file> [<i>output_file</i>] </input_file></input_file></pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 描述   | <p>uniq reads an input, comparing adjacent lines, and writing one copy of each input line on the output. The second and succeeding copies of the repeated adjacent lines are not written.</p> <p>If the output file, <i>output_file</i>, is not specified, uniq writes to standard output. If no <i>input_file</i> is given, or if the <i>input_file</i> is -, uniq reads from standard input with the start of the file is defined as the current offset.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-c Precedes each output line with a count of the number of times the line occurred in the input.</li> <li>-d Suppresses the writing of lines that are not repeated in the input.</li> <li>-f Ignores the first <i>fields</i> fields on each input line when doing comparisons, where <i>fields</i> is a positive decimal integer. A <i>field</i> is the maximal string matched by the basic regular expression: <pre>[[[:blank:]]*^[^[:blank:]]*</pre> <p>If <i>fields</i> specifies more <i>fields</i> than appear on an input line, a null string are used for comparison.</p> </li> <li>-s Ignores the first <i>chars</i> characters when doing comparisons, where <i>chars</i> is a positive decimal integer. If specified in conjunction with the -f option, the first <i>chars</i> characters after the first <i>fields</i> fields are ignored. If <i>chars</i> specifies more characters than remain on an input line, a null string are used for comparison.</li> <li>-u Suppresses the writing of lines that are repeated in the input.</li> <li>-n Equivalent to -f <i>fields</i> with <i>fields</i> set to <i>n</i>.</li> <li>+<i>m</i> Equivalent to -s <i>chars</i> with <i>chars</i> set to <i>m</i>.</li> </ul> |
| 操作数  | <p>The following operands are supported:</p> <p><i>input_file</i> A path name of the input file. If <i>input_file</i> is not specified, or if the <i>input_file</i> is -, the standard input is used.</p> <p><i>output_file</i> A path name of the output file. If <i>output_file</i> is not specified, the standard output is used. The results are unspecified if the file named by <i>output_file</i> is the file named by <i>input_file</i>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

**示例**

示例 1 Using the `uniq` Command

The following example lists the contents of the `uniq.test` file and outputs a copy of the repeated lines.

```
example% cat uniq.test
This is a test.
This is a test.
TEST.
Computer.
TEST.
TEST.
Software.
```

```
example% uniq -d uniq.test
This is a test.
TEST.
example%
```

The next example outputs just those lines that are not repeated in the `uniq.test` file.

```
example% uniq -u uniq.test
TEST.
Computer.
Software.
example%
```

The last example outputs a report with each line preceded by a count of the number of times each line occurred in the file:

```
example% uniq -c uniq.test
 2 This is a test.
 1 TEST.
 1 Computer.
 2 TEST.
 1 Software.
example%
```

**环境变量**

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `uniq`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

**退出状态**

The following exit values are returned:

```
0 Successful completion.
>0 An error occurred.
```

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

---

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

另请参见

[comm\(1\)](#), [pack\(1\)](#), [pcat\(1\)](#), [sort\(1\)](#), [uncompress\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

**引用名** units – converts quantities expressed in standard scales to other scales

**用法概要** units

**描述** units converts quantities expressed in various standard scales to their equivalents in other scales. It works interactively in this fashion:

```
You have: ~inch
You want: ~cm
 * 2.540000e+00
/ 3.937008e-01
```

A quantity is specified as a multiplicative combination of units optionally preceded by a numeric multiplier. Powers are indicated by suffixed positive integers, division by the usual sign:

```
You have: ~15 lbs force/in2
You want: ~atm
 * 1.020689e+00
 / 9.797299e-01
```

units only does multiplicative scale changes; thus it can convert Kelvin to Rankine, but not Celsius to Fahrenheit. Most familiar units, abbreviations, and metric prefixes are recognized, together with a generous leavening of exotica and a few constants of nature including:

pi        ratio of circumference to diameter,  
c        speed of light,  
e        charge on an electron,  
g        acceleration of gravity,  
force    same as g,  
mole     Avogadro's number,  
water    pressure head per unit height of water,  
au       astronomical unit.

Pound is not recognized as a unit of mass; lb is. Compound names are run together, (for example, lightyear). British units that differ from their U.S. counterparts are prefixed thus: brgallon. For a complete list of units, type:

```
cat /usr/share/lib/unittab
```

**文件** /usr/share/lib/unittab

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

---

| ATTRIBUTETYPE | ATTRIBUTE VALUE |
|---------------|-----------------|
| Availability  | system/core-os  |

另请参见

[attributes\(5\)](#)

**引用名**                    unix2dos – convert text file from ISO format to DOS format

**用法概要**                unix2dos [-ascii] [-iso] [-7]  
                              [-437 | -850 | -860 | -863 | -865] *originalfile convertedfile*

**描述**                    The unix2dos utility converts ISO standard characters to the corresponding characters in the DOS extended character set.

This command may be invoked from either DOS or SunOS. However, the filenames must conform to the conventions of the environment in which the command is invoked.

If the original file and the converted file are the same, unix2dos will rewrite the original file after converting it.

**选项**                    The following options are supported:

- ascii     Adds carriage returns and converts end of file characters in SunOS format text files to conform to DOS requirements.
- iso        This is the default. Converts ISO standard characters to the corresponding character in the DOS extended character set.
- 7          Converts 8 bit SunOS characters to 7 bit DOS characters.

On non-i386 systems, unix2dos will attempt to obtain the keyboard type to determine which code page to use. Otherwise, the default is US. The user may override the code page with one of the following options:

- 437     Use US code page
- 850     Use multilingual code page
- 860     Use Portuguese code page
- 863     Use French Canadian code page
- 865     Use Danish code page

**操作数**                The following operands are required:

- originalfile*     The original file in ISO format that is being converted to DOS format.
- convertedfile*    The new file in DOS format that has been converted from the original ISO file format.

**属性**                    See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |



## 另请参见

[dos2unix\(1\)](#), [ls\(1\)](#), [attributes\(5\)](#)

## 诊断

File *filename* not found, or no read permission

The input file you specified does not exist, or you do not have read permission. Check with the SunOS command, `ls -l` (see [ls\(1\)](#)).

Bad output filename *filename*, or no write permission

The output file you specified is either invalid, or you do not have write permission for that file or the directory that contains it. Check also that the drive is not write-protected.

Error while writing to temporary file

An error occurred while converting your file, possibly because there is not enough space on the current drive. Check the amount of space on the current drive using the DIR command. Also be certain that the default drive is write-enabled (not write-protected). When this error occurs, the original file remains intact.

Translated tmpfile name = *filename*.

Could not rename tmpfile to *filename*.

The program could not perform the final step in converting your file. Your converted file is stored under the name indicated on the second line of this message.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | updatehome – 更新起始目录副本，然后链接当前标签的文件                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 用法概要 | <code>/usr/bin/updatehome [-cirs]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 描述   | <p>updatehome 读取用户的最小标签副本和链接控制文件（<code>.copy_files</code> 和 <code>.link_files</code>）。这些文件中包含要复制的和要从用户最小标签起始目录以符号方式链接到当前标签处的用户起始目录的文件列表。</p> <p>缺省情况下，在 <code>label_encodings(4)</code> 中指定最小用户标签，并且可在 <code>user_attr(4)</code> 中明确指定最小用户标签。使用 <code>txzonemgr(1M)</code> 创建公共区域时，会为公共区域分配缺省最小标签，并且会将公共区域配置为多级别 NFS 服务器。公共区域中的授权管理员可使用 <code>share(1M)</code> 命令导出起始目录，从而可由较高级别的区域在只读模式下挂载这些目录。通过 <code>txzonemgr(1M)</code> 方式创建的其他区域将配置有 <code>automount(1M)</code> 项，该项在 <code>/zone/public/home</code> 处挂载公共区域的起始目录。</p> <p>如果用户的最小起始目录已按照这种方式进行了共享，则用户可以在较高级别的区域中运行 <code>updatehome</code> 命令（手动或通过启动文件中执行）。例如，用户可能需要指向诸如 <code>.profile</code>、<code>.login</code>、<code>.cshrc</code>、<code>.exrc</code>、<code>.mailrc</code> 和 <code>~/bin</code> 等的文件的符号链接。<code>updatehome</code> 命令为完成该符号链接提供了一种方便的机制。用户可将文件添加到那些要复制的（<code>.copy_files</code>）和要以符号方式链接（<code>.link_files</code>）的文件。</p> |
| 选项   | <ul style="list-style-type: none"> <li>-c 替换当前标签处的现有起始目录副本。缺省设置是跳过现有副本。</li> <li>-i 忽略遇到的错误。缺省设置出错时中止。</li> <li>-r 替换当前标签处的现有起始目录副本或符号链接。该选项表示选项 -c 和 -s。缺省设置是跳过现有副本或符号链接。</li> <li>-s 替换当前标签处的现有起始目录符号链接。缺省设置是跳过现有符号链接。</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 退出状态 | 一旦成功， <code>updatehome</code> 返回 0。一旦失败， <code>updatehome</code> 返回 1，并且会将诊断消息写入标准错误。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 示例   | <p>示例1 样例 <code>.copy_files</code> 文件</p> <p>可在每个用户标签处对 <code>.copy_files</code> 中列出的文件进行修改。</p> <pre>.cshrc .mailrc .mozilla/bookmarks.html</pre> <p>示例2 样例 <code>.link_files</code> 文件</p> <p>可在最低级别标签处对 <code>.link_files</code> 中所列的文件进行修改。更改将传播到用户可用的其他标签。</p> <pre>~/bin .mozilla/preferences .xrc .rhosts</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

示例3 更新已链接文件和已复制文件

用户在最小标签处对 `.copy_files` 和 `.link_files` 进行了更新。在较高级别标签处，用户刷新副本和链接。运行该命令不需要特权。

```
% updatehome -r
```

文件

```
$HOME/.copy_files 要复制的文件列表
```

```
$HOME/.link_files 要以符号方式链接的文件列表
```

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值             |
|-------|-----------------|
| 可用性   | system/trusted  |
| 接口稳定性 | Committed (已确定) |

另请参见

[automount\(1M\)](#)、[share\(1M\)](#)、[txzonemgr\(1M\)](#)、[label\\_encodings\(4\)](#)、[user\\_attr\(4\)](#)、[attributes\(5\)](#)

《[Trusted Extensions 配置和管理](#)》中的“[.copy\\_files](#) 和 [.link\\_files](#) 文件”

附注

仅当系统配置有 Trusted Extensions 时，本手册页中介绍的功能才可用。

**引用名** uptime – show how long the system has been up

**用法概要** uptime

**描述** The uptime command prints the current time, the length of time the system has been up, and the average number of jobs in the run queue over the last 1, 5 and 15 minutes. It is, essentially, the first line of a [w\(1\)](#) command.

**示例** Below is an example of the output uptime provides:

```
example% uptime
10:47am up 27 day(s), 50 mins, 1 user, load average: 0.18, 0.26, 0.20
```

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

**另请参见** [w\(1\)](#), [who\(1\)](#), [whodo\(1M\)](#), [attributes\(5\)](#)

**附注** who -b gives the time the system was last booted.

**引用名** userattr – 输出授予用户或角色的属性值

**用法概要** userattr [-v] *attribute\_name* [*user*]

**描述** userattr 命令在标准输出中输出为属性 *attribute\_name* 找到的第一个值。如果未指定用户，则将从进程的实际用户 ID 中选择用户。属性名称即 [user\\_attr\(4\)](#) 和 [prof\\_attr\(4\)](#) 中定义的名称。对分配给用户的配置文件使用 [profiles\(1\)](#) 命令。对分配给用户的授权使用 [auths\(1\)](#) 命令。搜索顺序是先搜索用户的 *user\_attr* 项，然后搜索用户的配置文件。

如果未将属性 *attribute\_name* 分配给用户，则对于任何错误，userattr 都会返回非零退出代码。否则，userattr 会返回零退出代码。

-v 选项可额外输出找到属性的位置。

**示例** 示例 1 使用 userattr

```
example% userattr lock_after_retries root
no
```

**文件** /etc/user\_attr

/etc/security/policy.conf

/etc/security/prof\_attr

**退出状态** 将返回以下退出值：

0 成功完成。

1 出现错误。

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型  | 属性值    |
|-------|--------|
| 接口稳定性 | 请参见下文。 |

退出代码是 "Committed"（已确定）。-v 选项的输出不是接口。

**另请参见** [auths\(1\)](#)、[profiles\(1\)](#)、[policy.conf\(4\)](#)、[prof\\_attr\(4\)](#)、[user\\_attr\(4\)](#)、[attributes\(5\)](#)

**引用名** users – display a compact list of users logged in

**用法概要** /usr/ucb/users [*filename*]

**描述** The users utility lists the login names of the users currently on the system in a compact, one-line format.

Specifying *filename* tells users where to find its information; by default it checks /var/adm/utmpx.

Typing users is equivalent to typing who -q.

**示例** 示例 1 Listing current users

```
example% users
paul george ringoexample%
```

**文件** /var/adm/utmpx

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

**另请参见** [who\(1\)](#), [attributes\(5\)](#)

|        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名    | uucp, uulog, uuname – UNIX-to-UNIX system copy                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 用法概要   | <pre>uucp [-c   -C] [-d   -f] [-ggrade] [-jmr] [-nuser] [-sfile]       [-xdebug_level] source-file destination-file  uulog [-ssys] [-fsystem] [-x] [-number] system  uuname [-c   -l]</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 描述     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| uucp   | The uucp utility copies files named by the <i>source-file</i> arguments to the <i>destination-file</i> argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| uulog  | The uulog utility queries a log file of uucp or uuxqt transactions in file <i>/var/uucp/.Log/uucico/system</i> or <i>/var/uucp/.Log/uuxqt/system</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| uuname | The uuname utility lists the names of systems known to uucp.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 选项     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| uucp   | <p>The following options are supported by uucp:</p> <ul style="list-style-type: none"> <li>-c Does not copy local file to the spool directory for transfer to the remote machine (default).</li> <li>-C Forces the copy of local files to the spool directory for transfer.</li> <li>-d Makes all necessary directories for the file copy (default).</li> <li>-f Does not make intermediate directories for the file copy.</li> <li>-g <i>grade</i> <i>grade</i> can be either a single letter, number, or a string of alphanumeric characters defining a service grade. The <code>uuglist</code> command can determine whether it is appropriate to use the single letter, number, or a string of alphanumeric characters as a service grade. The output from the <code>uuglist</code> command is a list of service grades that are available, or a message that says to use a single letter or number as a grade of service.</li> <li>-j Prints the uucp job identification string on standard output. This job identification can be used by <code>uustat</code> to obtain the status of a uucp job or to terminate a uucp job. The uucp job is valid as long as the job remains queued on the local system.</li> <li>-m Sends mail to the requester when the copy is complete.</li> <li>-n <i>user</i> Notifies <i>user</i> on the remote system that a file was sent.<br/><br/>When multiple -n options are passed in, uucp only retains the value specified for the last -n option. This is the only user notified.</li> <li>-r Does not start the file transfer, just queue the job.</li> </ul> |

- s *file* Reports status of the transfer to *file*. This option is accepted for compatibility, but it is ignored because it is insecure.
- x *debug\_level* Produce debugging output on standard output. *debug\_level* is a number between 0 and 9. As *debug\_level* increases to 9, more detailed debugging information is given. This option may not be available on all systems.

uulog The following options cause uulog to print logging information:

- s *sys* Prints information about file transfer work involving system *sys*.
- f *system* Executes a `tail -f` command of the file transfer log for *system*. You must press BREAK to exit this function.

Other options used in conjunction with the above options are:

- x Looks in the uuxqt log file for the given system.
- number* Executes a `tail` command of *number* lines.

uuname The following options are supported by uuname:

- c Displays the names of systems known to cu. The two lists are the same, unless your machine is using different `Systems` files for cu and uucp. See the `Systems` file.
- l Displays the local system name.

## 操作数

The source file name may be a path name on your machine, or may have the form:

*system-name!pathname*

where *system-name* is taken from a list of system names that uucp knows about. *source\_file* is restricted to no more than one *system-name*. The destination *system-name* may also include a list of system names such as

*system-name!system-name!...!system-name!pathname*

In this case, an attempt is made to send the file, using the specified route, to the destination. Care should be taken to ensure that intermediate nodes in the route are willing to forward information. See NOTES for restrictions.

For C-Shell users, the exclamation point (!) character must be surrounded by single quotes ('), or preceded by a backslash (\).

The shell metacharacters ?, \* and [...] appearing in *pathname* are expanded on the appropriate system.

Pathnames may be one of the following:

1. An absolute pathname.



2. A pathname preceded by *~user* where *user* is a login name on the specified system and is replaced by that user's login directory.
3. A pathname preceded by *~/destination* where *destination* is appended to `/var/spool/uucppublic`. This destination is treated as a filename unless more than one file is being transferred by this request or the destination is already a directory. To ensure that the destination is a directory, follow it with a forward slash (/). For example, `~/dan/` as the destination creates the directory `/var/spool/uucppublic/dan` if it does not exist and put the requested file(s) in that directory.

Anything else is prefixed by the current directory.

If the result is an erroneous path name for the remote system, the copy fails. If the *destination-file* is a directory, the last part of the *source-file* name is used.

Invoking `uucp` with shell wildcard characters as the remote *source-file* invokes the `uux(1C)` command to execute the `uucp` command on the remote machine. The remote `uucp` command spools the files on the remote machine. After the first session terminates, if the remote machine is configured to transfer the spooled files to the local machine, the remote machine initiates a call and send the files; otherwise, the user must "call" the remote machine to transfer the files from the spool directory to the local machine. This call can be done manually using `Uutry(1M)`, or as a side effect of another `uux(1C)` or `uucp` call.

Notice that the local machine must have permission to execute the `uucp` command on the remote machine in order for the remote machine to send the spooled files.

`uucp` removes execute permissions across the transmission and gives `0666` read and write permissions (see `chmod(2)`).

## 环境变量

See `environ(5)` for descriptions of the following environment variables that affect the execution of `uucp`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, `NLSPATH`, and `TZ`.

## 退出状态

The following exit values are returned:

- `0` Successful completion.
- `>0` An error occurred.

## 文件

|                                      |                                            |
|--------------------------------------|--------------------------------------------|
| <code>/etc/uucp/*</code>             | other data files                           |
| <code>/var/spool/uucp</code>         | spool directories                          |
| <code>/usr/lib/uucp/*</code>         | other program files                        |
| <code>/var/spool/uucppublic/*</code> | public directory for receiving and sending |

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | service/network/uucp               |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

## 另请参见

[mail\(1\)](#), [uuglist\(1C\)](#), [uustat\(1C\)](#), [uux\(1C\)](#), [Uutry\(1M\)](#), [uuxqt\(1M\)](#), [chmod\(2\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

## 附注

For security reasons, the domain of remotely accessible files may be severely restricted. You probably are not able to access files by path name. Ask a responsible person on the remote system to send them to you. For the same reasons you are probably not able to send files to arbitrary path names. As distributed, the remotely accessible files are those whose names begin `/var/spool/uucppublic` (equivalent to `~/`).

All files received by uucp are owned by uucp.

The `-m` option only works when sending files or receiving a single file. Receiving multiple files specified by special shell characters `?`, `&`, and `[ . . . ]` does not activate the `-m` option.

The forwarding of files through other systems may not be compatible with the previous version of uucp. If forwarding is used, all systems in the route must have compatible versions of uucp.

Protected files and files that are in protected directories that are owned by the requester can be sent by uucp. However, if the requester is root, and the directory is not searchable by "other" or the file is not readable by "other", the request fails.

Strings that are passed to remote systems may not be evaluated in the same locale as the one in use by the process that invoked uucp on the local system.

Configuration files must be treated as C (or POSIX) locale text files.

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | uuencode, uudecode – encode a binary file, or decode its encoded representation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 用法概要 | <pre>uuencode [source-file] decode_pathname uuencode [-m] [source-file] decode_pathname uudecode [-p] [encoded-file] uudecode [-o outfile] [encoded-file]</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 描述   | <p>These commands encode and decode files as follows:</p> <p><b>uuencode</b> The uuencode utility converts a binary file into an encoded representation that can be sent using <code>mail(1)</code>. It encodes the contents of <i>source-file</i>, or the standard input if no <i>source-file</i> argument is given. The <i>decode_pathname</i> argument is required. The <i>decode_pathname</i> is included in the encoded file's header as the name of the file into which uudecode is to place the binary (decoded) data. uuencode also includes the permission modes of <i>source-file</i> (except <code>setuid</code>, <code>setgid</code>, and sticky-bits), so that <i>decode_pathname</i> is recreated with those same permission modes.</p> <p><b>uudecode</b> The uudecode utility reads an <i>encoded-file</i>, strips off any leading and trailing lines added by mailer programs, and recreates the original binary data with the filename and the mode specified in the header.</p> <p>The encoded file is an ordinary portable character set text file; it can be edited by any text editor. It is best only to change the mode or <i>decode_pathname</i> in the header to avoid corrupting the decoded binary.</p> |
| 选项   | <p>The following options are supported:</p> <p><b>uuencode</b> <code>-m</code> Encodes <i>source-file</i> using Base64 encoding and sends it to standard output.</p> <p><b>uudecode</b> <code>-o outfile</code> Specifies a file pathname that should be used instead of any pathname contained in the input data. Specifying an <i>outfile</i> option-argument of <code>/dev/stdout</code> indicates standard output. This allows uudecode to be used in a pipeline.</p> <p><code>-p</code> Decodes <i>encoded-file</i> and sends it to standard output. This allows uudecode to be used in a pipeline.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 操作数  | <p>The following operands are supported by uuencode and uudecode:</p> <p><b>uuencode</b> <i>decode_pathname</i> The pathname of the file into which the uudecode utility will place the decoded file. If there are characters in <i>decode_pathname</i> that are not in the portable filename character set, the results are unspecified.</p> <p><i>source-file</i> A pathname of the file to be encoded.</p> <p><b>uudecode</b> <i>encoded-file</i> The pathname of a file containing the output of uuencode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

- 用法** See [largefile\(5\)](#) for the description of the behavior of uuencode and uudecode when encountering files greater than or equal to 2 Gbyte ( $2^{31}$  bytes).
- 环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of uuencode and uudecode: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLS\_PATH.
- Output** stdout
- uuencode Base64 Algorithm The standard output is a text file, encoded in the character set of the current locale, that begins with the line:

```
begin-base64 %s %s\n, mode, decode_pathname
```

and ends with the line:

```
====\n
```

In both cases, the lines have no preceding or trailing blank characters.

The encoding process represents 24-bit groups of input bits as output strings of four encoded characters. Proceeding from left to right, a 24-bit input group is formed by concatenating three 8-bit input groups. Each 24-bit input group is then treated as four concatenated 6-bit groups, each of which is translated into a single digit in the Base64 alphabet. When encoding a bit stream by means of the Base64 encoding, the bit stream is presumed to be ordered with the most-significant bit first. That is, the first bit in the stream is the high-order bit in the first byte, and the eighth bit is the low-order bit in the first byte, and so on. Each 6-bit group is used as an index into an array of 64 printable characters, as shown in the following table.

| Value | Encoding | Value | Encoding | Value | Encoding | Value | Encoding |
|-------|----------|-------|----------|-------|----------|-------|----------|
| 0     | A        | 17    | R        | 34    | i        | 51    | z        |
| 1     | B        | 18    | S        | 35    | j        | 52    | 0        |
| 2     | C        | 19    | T        | 36    | k        | 53    | 1        |
| 3     | D        | 20    | U        | 37    | l        | 54    | 2        |
| 4     | E        | 21    | V        | 38    | m        | 55    | 3        |
| 5     | F        | 22    | W        | 39    | n        | 56    | 4        |
| 6     | G        | 23    | X        | 40    | o        | 57    | 5        |
| 7     | H        | 24    | Y        | 41    | p        | 58    | 6        |
| 8     | I        | 25    | Z        | 42    | q        | 59    | 7        |
| 9     | J        | 26    | a        | 43    | r        | 60    | 8        |
| 10    | K        | 27    | b        | 44    | s        | 61    | 9        |
| 11    | L        | 28    | c        | 45    | t        | 62    | +        |
| 12    | M        | 29    | d        | 46    | u        | 63    | /        |
| 13    | N        | 30    | e        | 47    | v        |       |          |
| 14    | O        | 31    | f        | 48    | w        | (pad) | =        |
| 15    | P        | 32    | g        | 49    | x        |       |          |
| 16    | Q        | 33    | h        | 50    | y        |       |          |

The character referenced by the index is placed in the output string.

The output stream (encoded bytes) is represented in lines of no more than 76 characters each. All line breaks or other characters not found in the table are ignored by decoding software (see `uudecode`).

Special processing is performed if fewer than 24 bits are available at the end of a message or encapsulated part of a message. A full encoding quantum is always completed at the end of a message. When fewer than 24 input bits are available in an input group, zero bits are added on the right to form an integral number of 6-bit groups. Output character positions that are not required to represent actual input data are set to the equals (=) character. Since all Base64 input is an integral number of octets, only the following cases can arise:

1. The final quantum of encoding input is an integral multiple of 24 bits. Here, the final unit of encoded output is an integral multiple of four characters with no '=' padding.
2. The final quantum of encoding input is exactly 16 bits. Here, the final unit of encoded output is three characters followed by one '=' padding character.
3. The final quantum of encoding input is exactly 8 bits. Here, the final unit of encoded output is two characters followed by two '=' padding characters.

A terminating "====" evaluates to nothing and denotes the end of the encoded data.

#### uuencode Historical Algorithm

The standard output is a text file (encoded in the character set of the current locale) that begins with the line:

```
begin %s %s\n, mode, decode_pathname
```

and ends with the line:

```
end\n
```

In both cases, the lines have no preceding or trailing blank characters.

The algorithm that is used for lines between `begin` and `end` takes three octets as input and writes four characters of output by splitting the input at six-bit intervals into four octets, containing data in the lower six bits only. These octets are converted to characters by adding a value of `0x20` to each octet, so that each octet is in the range `0x20–0x5f`, and each octet is assumed to represent a printable character. Each octet is then translated into the corresponding character codes for the codeset in use in the current locale. For example, the octet `0x41`, representing 'A', would be translated to 'A' in the current codeset, such as `0xc1` if the codeset were EBCDIC.

Where the bits of two octets are combined, the least significant bits of the first octet are shifted left and combined with the most significant bits of the second octet shifted right. Thus, the three octets A, B, C are converted into the four octets:

$$0x20 + ((A \gg 2) \ll 4) \& 0x3F)$$

$$0x20 + (((A \ll 4) \& 0xF) \ll 2) \& 0x3F)$$

$$0x20 + (((B \ll 2) \& 0x3) \ll 6) \& 0x3F)$$

$0x20 + ((C \quad \quad \quad) \& 0x3F)$

These octets are then translated into the local character set.

Each encoded line contains a length character, equal to the number of characters to be decoded plus  $0x20$  translated to the local character set as described above, followed by the encoded characters. The maximum number of octets to be encoded on each line is 45.

## 退出状态

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

## 另请参见

[mail\(1\)](#), [mailx\(1\)](#), [uucp\(1C\)](#), [uux\(1C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

## 附注

The size of the encoded file is expanded by 35% (3 bytes become 4, plus control information), causing it to take longer to transmit than the equivalent binary.

The user on the remote system who is invoking `uudecode` (typically `uucp`) must have write permission on the file specified in the *decode\_pathname*.

If you invoke `uuencode` and then execute `uudecode` on a file in the same directory, you will overwrite the original file.

- 引用名** uuglist – print the list of service grades that are available on this UNIX system
- 用法概要** uuglist [-u]
- 描述** uuglist prints the list of service grades that are available on the system to use with the -g option of [uucp\(1C\)](#) and [uux\(1C\)](#).
- 选项** -u List the names of the service grades that the user is allowed to use with the -g option of the uucp and uux commands.
- 文件** /etc/uucp/Grades contains the list of service grades
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE      |
|----------------|----------------------|
| Availability   | service/network/uucp |

- 另请参见** [uucp\(1C\)](#), [uux\(1C\)](#), [attributes\(5\)](#)

**引用名** uustat – uucp status inquiry and job control

**用法概要**

```
uustat
 [[-m] | [-p] | [-q] | [-k jobid [-n]] | [-r jobid [-n]]]
uustat [-a] [-s system [-j]] [-u user] [-S qric]
uustat -t system [-c] [-d number]
```

**描述** The uustat utility functions in the following three areas:

1. Displays the general status of, or cancels, previously specified uucp commands.
2. Provides remote system performance information, in terms of average transfer rates or average queue times.
3. Provides general remote system-specific and user-specific status of uucp connections to other systems.

**选项** The following options are supported:

General Status These options obtain general status of, or cancel, previously specified uucp commands:

- a Lists all jobs in queue.
- j Lists the total number of jobs displayed. The -j option can be used in conjunction with the -a or the -s option.
- k*jobid* Kills the uucp request whose job identification is *jobid*. The killed uucp request must belong to the user issuing the uustat command unless the user is the super-user or uucp administrator. If the job is killed by the super-user or uucp administrator, electronic mail is sent to the user.
- m Reports the status of accessibility of all machines.
- n Suppresses all standard output, but not standard error. The -n option is used in conjunction with the -k and -r options.
- p Executes the command `ps -flp` for all the process-ids that are in the lock files.
- q Lists the jobs queued for each machine. If a status file exists for the machine, its date, time and status information are reported. In addition, if a number appears in parentheses next to the number of C or X files, it is the age in days of the oldest C./X. file for that system. The Retry field represents the number of hours until the next possible call. The Count is the number of failure attempts. *Note:* For systems with a moderate number of outstanding jobs, this could take 30 seconds or more of real-time to execute. An example of the output produced by the -q option is:

```
eagle 3C 04/07-11:07 NO DEVICES AVAILABLE
mh3bs3 2C 07/07-10:42 SUCCESSFUL
```

This indicates the number of command files that are waiting for each system. Each command file may have zero or more files to be sent (zero means to call the system



and see if work is to be done). The date and time refer to the previous interaction with the system followed by the status of the interaction.

**-rjobid** Rejuvenates *jobid*. The files associated with *jobid* are touched so that their modification time is set to the current time. This prevents the cleanup daemon from deleting the job until the jobs' modification time reaches the limit imposed by the daemon.

**Remote System Status** These options provide remote system performance information, in terms of average transfer rates or average queue times. The **-c** and **-d** options can only be used in conjunction with the **-t** option:

**-tsystem** Reports the average transfer rate or average queue time for the past 60 minutes for the remote *system*. The following parameters can only be used with this option:

**-c** Average queue time is calculated when the **-c** parameter is specified and average transfer rate when **-c** is not specified. For example, the command:

```
example% uustat -teagle -d50 -c
```

produces output in the following format:

```
average queue time to eagle for last 50 minutes:
 5 seconds
```

The same command without the **-c** parameter produces output in the following format:

```
average transfer rate with eagle for last 50 minutes:
 2000.88 bytes/sec
```

**-dnumber** *number* is specified in minutes. Used to override the 60 minute default used for calculations. These calculations are based on information contained in the optional performance log and therefore may not be available. Calculations can only be made from the time that the performance log was last cleaned up.

**User- or  
System-Specific Status**

These options provide general remote system-specific and user-specific status of uucp connections to other systems. Either or both of the following options can be specified with `uustat`. The **-j** option can be used in conjunction with the **-s** option to list the total number of jobs displayed:

**-ssystem** Reports the status of all uucp requests for remote system *system*.

**-uuser** Reports the status of all uucp requests issued by *user*.

Output for both the **-s** and **-u** options has the following format:

```
eagleN1bd7 4/07-11:07 S eagle dan 522 /home/dan/A
eagleC1bd8 4/07-11:07 S eagle dan 59 D.3b2a12ce4924
4/07-11:07 S eagle dan rmail mike
```

With the above two options, the first field is the *jobid* of the job. This is followed by the date/time. The next field is an S if the job is sending a file or an R if the job is requesting a file. The next field is the machine where the file is to be transferred. This is followed by the user-id of the user who queued the job. The next field contains the size of the file, or in the case of a remote execution (*rmail* is the command used for remote mail), the name of the command. When the size appears in this field, the file name is also given. This can either be the name given by the user or an internal name (for example, *D.3b2a1ce4924*) that is created for data files associated with remote executions (*rmail* in this example).

*-Sqric* Reports the job state:

- q for queued jobs
- r for running jobs
- i for interrupted jobs
- c for completed jobs

A job is queued if the transfer has not started. A job is running when the transfer has begun. A job is interrupted if the transfer began but was terminated before the file was completely transferred. A completed job is a job that successfully transferred. The completed state information is maintained in the accounting log, which is optional and therefore may be unavailable. The parameters can be used in any combination, but at least one parameter must be specified. The *-S* option can also be used with *-s* and *-u* options. The output for this option is exactly like the output for *-s* and *-u* except that the job states are appended as the last output word. Output for a completed job has the following format:

```
eagleC1bd3 completed
```

When no options are given, *uustat* writes to standard output the status of all *uucp* requests issued by the current user.

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of *uustat*: *LANG*, *LC\_ALL*, *LC\_COLLATE*, *LC\_CTYPE*, *LC\_MESSAGES*, *LC\_TIME*, *NLSPATH*, and *TZ*.

## 退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

## 文件

|                                  |                   |
|----------------------------------|-------------------|
| <i>/var/spool/uucp/*</i>         | spool directories |
| <i>/var/uucp/.Admin/account</i>  | accounting log    |
| <i>/var/uucp/.Admin/perfllog</i> | performance log   |

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | service/network/uucp               |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见**

[uucp\(1C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

**诊断**

The -t option produces no message when the data needed for the calculations is not being recorded.

**附注**

After the user has issued the uucp request, if the file to be transferred is moved, deleted or was not copied to the spool directory (-C option) when the uucp request was made, uustat reports a file size of -99999. This job will eventually fail because the file(s) to be transferred can not be found.

**引用名** uuto, uupick – public UNIX-to-UNIX system file copy

**用法概要** uuto [-mp] *source-file*... *destination*

uupick [-s *system*]

## 描述

uuto sends *source-file* to *destination*. uuto uses the [uucp\(1C\)](#) facility to send files, while it allows the local system to control the file access. A source-file name is a path name on your machine. Destination has the form:

*system*[!*system*]...!*user*

where *system* is taken from a list of system names that uucp knows about. *User* is the login name of someone on the specified system.

The files (or sub-trees if directories are specified) are sent to PUBDIR on *system*, where PUBDIR is a public directory defined in the uucp source. By default, this directory is `/var/spool/uucppublic`. Specifically the files are sent to

`PUBDIR/receive/user/mysystem/files`.

The recipient is notified by [mail\(1\)](#) of the arrival of files.

uupick accepts or rejects the files transmitted to the user. Specifically, uupick searches PUBDIR for files destined for the user. For each entry (file or directory) found, the following message is printed on standard output:

from system *sysname*: [file *file-name*] [dir *dirname*] ?

uupick then reads a line from standard input to determine the disposition of the file:

<new-line> Go to next entry.

d Delete the entry.

m [*dir*] Move the entry to named directory *dir*. If *dir* is not specified as a complete path name (in which \$HOME is legitimate), a destination relative to the current directory is assumed. If no destination is given, the default is the current directory.

a [*dir*] Same as m above, except it moves all the files sent from *system*.

p Print the content of the file.

q Stop.

EOT (control-d) Same as q.

!command Escape to the shell to do command.

\* Print a command summary.

## 选项

**uuto** The following options are supported by **uuto**:

- m Send mail to the sender when the copy is complete.
- p Copy the source file into the spool directory before transmission.

**uupick** The following option is supported by **uupick**:

- s *system* Search only the PUBDIR for files sent from *system*.

## 操作数

The following operands are supported for **uuto**:

*destination* A string of the form:

*system-name ! user*

where *system-name* is taken from a list of system names that **uucp** knows about; see **uname**. The argument *user* is the login name of someone on the specified system. The destination *system-name* can also be a list of names such as

*system-name ! system-name ! . . . ! system-name ! user*

in which case, an attempt is made to send the file via the specified route to the destination. Care should be taken to ensure that intermediate nodes in the route are willing to forward information.

*source-file* A pathname of a file on the local system to be copied to *destination*.

## 环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of **uuto** and **uupick**: **LC\_TYPE**, **LC\_MESSAGES**, and **NLSPATH**.

## 退出状态

The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.

## 文件

**PUBDIR** /var/spool/uucppublic public directory

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTETYPE | ATTRIBUTEVALUE       |
|---------------|----------------------|
| Availability  | service/network/uucp |

另请参见

[mail\(1\)](#), [uucp\(1C\)](#), [uustat\(1C\)](#), [uux\(1C\)](#), [uucleanup\(1M\)](#), [attributes\(5\)](#)

附注

In order to send files that begin with a dot (for instance, `.profile`), the files must be qualified with a dot. For example, the following files are correct:

```
.profile .prof* .profil?
```

The following files are incorrect:

```
prof ?profile
```

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | uux – UNIX-to-UNIX system command execution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 用法概要 | uux [-] [-bcCjnprz] [-a <i>name</i> ] [-g <i>grade</i> ]<br>[-s <i>filename</i> ] [-x <i>debug_level</i> ] <i>command-string</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 描述   | <p>The uux utility will gather zero or more files from various systems, execute a command on a specified system and then send standard output to a file on a specified system.</p> <p><i>Note:</i> For security reasons, most installations limit the list of commands executable on behalf of an incoming request from uux, permitting only the receipt of mail (see <a href="#">mail(1)</a>). (Remote execution permissions are defined in <code>/etc/uucp/Permissions</code>.)</p> <p>The <i>command-string</i> is made up of one or more arguments that look like a shell command line, except that the command and file names may be prefixed by <i>system-name!</i>. A null <i>system-name</i> is interpreted as the local system.</p> <p>File names may be one of the following:</p> <ul style="list-style-type: none"> <li>▪ An absolute path name.</li> <li>▪ A path name preceded by <code>~xxx</code>, where <i>xxx</i> is a login name on the specified system and is replaced by that user's login directory.</li> </ul> <p>Anything else is prefixed by the current directory.</p> <p>As an example, the command:</p> <pre>example% uux "!"diff sys1!/home/dan/filename1 \                 sys2!/a4/dan/filename2 &gt; !~/dan/filename.diff"</pre> <p>will get the <code>filename1</code> and <code>filename2</code> files from the <code>sys1</code> and <code>sys2</code> machines, execute a <a href="#">diff(1)</a> command and put the results in <code>filename.diff</code> in the local <code>PUBDIR/dan/</code> directory. <code>PUBDIR</code> is a public directory defined in the uucp source. By default, this directory is <code>/var/spool/uucppublic</code>.</p> <p>Any special shell characters (such as <code>&lt;</code> <code>&gt;</code> <code>;</code> <code> </code>) should be quoted either by quoting the entire <i>command-string</i>, or quoting the special characters as individual arguments. The redirection operators <code>&gt;&gt;</code>, <code>&lt;&lt;</code>, <code>&gt; </code>, and <code>&gt;&amp;</code> cannot be used.</p> <p>uux will attempt to get all appropriate files to the specified system where they will be processed. For files that are output files, the file name must be escaped using parentheses. For example, the command:</p> <pre>example% uux "a!cut -f1 b!/usr/filename &gt; c!/usr/filename"</pre> <p>gets <code>/usr/filename</code> from system <code>b</code> and sends it to system <code>a</code>, performs a <code>cut</code> command on that file and sends the result of the <code>cut</code> command to system <code>c</code>.</p> <p>uux will notify you if the requested command on the remote system was disallowed. This notification can be turned off by the <code>-n</code> option. The response comes by remote mail from the remote machine.</p> |

- 选项
- The following options are supported:
- The standard input to uux is made the standard input to the *command-string*.
  - a *name* Uses *name* as the user job identification replacing the initiator user-id. (Notification will be returned to user-id *name*.)
  - b Returns whatever standard input was provided to the uux command if the exit status is non-zero.
  - c Does not copy local file to the spool directory for transfer to the remote machine (default).
  - C Forces the copy of local files to the spool directory for transfer.
  - g *grade* *grade* can be either a single letter, number, or a string of alphanumeric characters defining a service grade. The `uuglist(1C)` command determines whether it is appropriate to use the single letter, number, or a string of alphanumeric characters as a service grade. The output from the `uuglist` command will be a list of service grades that are available or a message that says to use a single letter or number as a grade of service.
  - j Outputs the jobid string on the standard output which is the job identification. This job identification can be used by `uustat(1C)` to obtain the status or terminate a job.
  - n Does not notify the user if the command fails.
  - p Same as –. The standard input to uux is made the standard input to the *command-string*.
  - r Does not start the file transfer, but just queues the job.
  - s *filename* Reports status of the transfer in *filename*. This option is accepted for compatibility, but it is ignored because it is insecure.
  - x *debug\_level* Produces debugging output on the standard output. *debug\_level* is a number between 0 and 9. As *debug\_level* increases to 9, more detailed debugging information is given.
  - z Sends success notification to the user.

环境变量 See `environ(5)` for descriptions of the following environment variables that affect the execution of uux: LANG, LC\_ALL, LC\_CTYPE, LC\_MESSAGES, and NLSPATH.

退出状态 The following exit values are returned:

- 0 Successful completion.
- >0 An error occurred.



|    |                       |                              |
|----|-----------------------|------------------------------|
| 文件 | /etc/uucp/*           | other data and programs      |
|    | /etc/uucp/Permissions | remote execution permissions |
|    | /usr/lib/uucp/*       | other programs               |
|    | /var/spool/uucp       | spool directories            |

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | service/network/uucp               |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

另请参见 [cut\(1\)](#), [mail\(1\)](#), [uucp\(1C\)](#), [uuglist\(1C\)](#), [uustat\(1C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注 The execution of commands on remote systems takes place in an execution directory known to the uucp system.

All files required for the execution will be put into this directory unless they already reside on that machine. Therefore, the simple file name (without path or machine reference) must be unique within the uux request. The following command will NOT work:

```
example% uux "a!diff b!/home/dan/xyz c!/home/dan/xyz > !xyz.diff"
```

But the command:

```
example% uux "a!diff a!/home/dan/xyz c!/home/dan/xyz > !xyz.diff"
```

will work (if `diff` is a permitted command.)

Protected files and files that are in protected directories that are owned by the requester can be sent in commands using uux. However, if the requester is root, and the directory is not searchable by "other", the request will fail.

The following restrictions apply to the shell pipeline processed by uux:

- In gathering files from different systems, pathname expansion is not performed by uux. Thus, a request such as
 

```
uux "c89 remsys!~/*.c"
```

 would attempt to copy the file named literally `*.c` to the local system.
- Only the first command of a shell pipeline may have a `system-name!`. All other commands are executed on the system of the first command.
- The use of the shell metacharacter `*` will probably not do what you want it to do.

- The shell tokens << and >> are not implemented.
- The redirection operators >>, <<, >|, and >& cannot be used.
- The reserved word ! cannot be used at the head of the pipeline to modify the exit status.
- Alias substitution is not performed.

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名                         | vacation – reply to mail automatically                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 用法概要                        | <pre>vacation [-I] vacation [-a <i>alias</i>] [-e <i>filter_file</i>] [-f <i>database_file</i>]           [-j] [-m <i>message_file</i>] [-s <i>sender</i>] [-tN] <i>username</i> vacation [-f <i>database_file</i>] -l</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 描述                          | The <code>vacation</code> utility automatically replies to incoming mail.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Installation                | <p>The installation consists of an interactive program which sets up <code>vacation</code>'s basic configuration.</p> <p>To install <code>vacation</code>, type it with no arguments on the command line. The program creates a <code>.vacation.msg</code> file, which contains the message that is automatically sent to all senders when <code>vacation</code> is enabled, and starts an editor for you to modify the message. (See USAGE section.) Which editor is invoked is determined by the <code>VISUAL</code> or <code>EDITOR</code> environment variable, or <code>vi(1)</code> if neither of those environment variables are set.</p> <p>A <code>.forward</code> file is also created if one does not exist in your home directory. Once created, the <code>.forward</code> file will contain a line of the form:</p> <p>One copy of an incoming message is sent to the <code>username</code> and another copy is piped into <code>vacation</code>:</p> <pre><code>username, " /usr/bin/vacation username"</code></pre> <p>If a <code>.forward</code> file is present in your home directory, it will ask whether you want to remove it, which disables <code>vacation</code> and ends the installation.</p> <p>The program automatically creates <code>.vacation.pag</code> and <code>.vacation.dir</code>, which contain a list of senders when <code>vacation</code> is enabled.</p> |
| Activation and Deactivation | The presence of the <code>.forward</code> file determines whether or not <code>vacation</code> is disabled or enabled. To disable <code>vacation</code> , remove the <code>.forward</code> file, or move it to a new name.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Initialization              | The <code>-I</code> option clears the <code>vacation</code> log files, <code>.vacation.pag</code> and <code>.vacation.dir</code> , erasing the list of senders from a previous <code>vacation</code> session. (See OPTIONS section.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Additional Configuration    | <code>vacation</code> provides configuration options that are not part of the installation, these being <code>-a</code> , <code>-e</code> , <code>-f</code> , <code>-j</code> , <code>-m</code> , <code>-s</code> , and <code>-t</code> . (See OPTIONS section.)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Reporting                   | <code>vacation</code> provides a reporting option, <code>-l</code> . See OPTIONS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 选项                          | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li><code>-I</code>    Initializes the <code>.vacation.pag</code> and <code>.vacation.dir</code> files and enables <code>vacation</code>. If the <code>-I</code> flag is not specified, and a <code>user</code> argument is given, <code>vacation</code> reads the first line from the standard input (for a <code>From:</code> line, no colon). If absent, it produces an error message.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

Options `-a`, `-e`, `-f`, `-j`, `-m`, `-s`, and `-t` are configuration options to be used in conjunction with `vacation` in the `.forward` file, not on the command line. For example,

```
\username, "|/usr/bin/vacation -t1m username"
```

repeats replies to the sender every minute.

- `-a alias` Indicates that *alias* is one of the valid aliases for the user running `vacation`, so that mail addressed to that alias generates a reply.
- `-e filter_file` Uses *filter\_file* instead of `.vacation.filter` as the source of the domain and email address filters.
- `-f database_file` Uses *database\_file* instead of `.vacation` as the base name for the database file.
- `-j` Does not check whether the recipient appears in the `To:` or the `Cc:` line. Warning: use of this option can result in `vacation` replies being sent to mailing lists and other inappropriate places; its use is therefore strongly discouraged.
- `-m message_file` Uses `~/message_file` as the message to send for the reply instead of `~/vacation.msg`. *message\_file* is a relative path to the desired vacation message file. To prevent directory/file “not found” errors, *message\_file* should be on the same disk partition as `~/forward`.
- `-s sender` Replies to *sender* instead of the value read from the UNIX `From` line of the incoming message.
- `-tN` Changes the interval between repeat replies to the same sender. The default is 1 week. A trailing `s`, `m`, `h`, `d`, or `w` scales *N* to seconds, minutes, hours, days, or weeks, respectively.

The `-l` option is neither for initialization nor configuration., but for reporting. The `-f` option can also be used in conjunction with the `-l`.

- `-l` Lists the addresses to which a reply has been sent since the last invocation of `vacation` `-I`, along with a date and time stamp.

## 用法

`.vacation.msg` should include a header with at least a `Subject:` line (it should not include a `To:` line). For example:

```
Subject: I am on vacation
I am on vacation until July 22. If you have something urgent,
please contact Joe Jones (jones@fB0).
--John
```

If the string `$SUBJECT` appears in the `.vacation.msg` file, it is replaced with the subject of the original message when the reply is sent. Thus, a `.vacation.msg` file such as

---

```

Subject: I am on vacation
I am on vacation until July 22.
Your mail regarding "$SUBJECT" will be read when I return.
If you have something urgent, please contact
Joe Jones (jones@fB0).
--John

```

will include the subject of the message in the reply.

No message is sent if the To: or the Cc: line does not list the user to whom the original message was sent or one of a number of aliases for them, if the initial From line includes the string -REQUEST@, or if a Precedence: bulk or Precedence: junk line is included in the header.

vacation will also not respond to mail from either postmaster or Mailer-Daemon.

In addition to the above criteria, if a .vacation.filter file exists, it is used to constrain further the set of addresses to which a reply is sent. Each line in that file should be either a domain name, an email address, a negated domain name or a negated email address. A negated line starts with the single character !.

Each line is compared in the order listed to the sender address. A line containing an email address matches if the sender address is exactly the same except for case, which is ignored. A line containing a domain name matches if the sender address is *something@domain-name* or *something@something.domain-name*. A reply is sent if the first match is an entry that is not negated. If the first match is a negated entry, or if no lines match, then no reply is sent.

A sample filter file might look like the following:

```

!host.subdomain.sun.com
sun.com
!wife@mydomain.com
mydomain.com
onefriend@hisisp.com
anotherfriend@herisp.com

```

Blank lines and lines starting with “#” are ignored.

文件

```

~/ .forward
~/ .vacation.filter
~/ .vacation.msg

```

A list of senders is kept in the dbm format files .vacation.pag and .vacation.dir in your home directory. These files are dbm files and cannot be viewed directly with text editors.

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE               |
|----------------|-------------------------------|
| Availability   | service/network/smtp/sendmail |

**另请参见**

[vi\(1\)](#), [sendmail\(1M\)](#), [getusershell\(3C\)](#), [aliases\(4\)](#), [shells\(4\)](#), [attributes\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | vc – version control                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 用法概要 | vc [-a] [-t] [-c <i>char</i> ] [-s]<br>[ <i>keyword=value</i> ... <i>keyword=value</i> ]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 描述   | <p>This command is obsolete and will be removed in the next release.</p> <p>The vc command copies lines from the standard input to the standard output under control of its arguments and of “control statements” encountered in the standard input. In the process of performing the copy operation, user-declared <i>keywords</i> may be replaced by their string <i>value</i> when they appear in plain text and/or control statements.</p> <p>The copying of lines from the standard input to the standard output is conditional, based on tests (in control statements) of keyword values specified in control statements or as vc command arguments.</p> <p>A control statement is a single line beginning with a control character, except as modified by the -t keyletter (see below). The default control character is colon (:), except as modified by the -c keyletter (see below). Input lines beginning with a backslash (\) followed by a control character are not control lines and are copied to the standard output with the backslash removed. Lines beginning with a backslash followed by a non-control character are copied in their entirety.</p> <p>A keyword is composed of 9 or less alphanumeric; the first must be alphabetic. A value is any ASCII string that can be created with ed; a numeric value is an unsigned string of digits. Keyword values may not contain blanks or tabs.</p> <p>Replacement of keywords by values is done whenever a keyword surrounded by control characters is encountered on a version control statement. The -a keyletter (see below) forces replacement of keywords in all lines of text. An uninterpreted control character may be included in a value by preceding it with \. If a literal \ is desired, then it too must be preceded by \.</p> |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-a        Forces replacement of keywords surrounded by control characters with their assigned value in all text lines and not just in vc statements.</li> <li>-t        All characters from the beginning of a line up to and including the first tab character are ignored for the purpose of detecting a control statement. If a control statement is found, all characters up to and including the tab are discarded.</li> <li>-c<i>char</i>   Specifies a control character to be used in place of the “:” default.</li> <li>-s        Silences warning messages (not error) that are normally printed on the diagnostic output.</li> </ul> <p>vc recognizes the following version control statements:</p> <p>:dc1 <i>keyword</i>[, ..., <i>keyword</i>]    Declare keywords. All keywords must be declared.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

:asg *keyword=value*

Assign values to keywords. An asg statement overrides the assignment for the corresponding keyword on the vc command line and all previous asg statements for that keyword. Keywords that are declared but are not assigned values have null values.

:if *condition*

...

:end

Skip lines of the standard input. If the condition is true, all lines between the if statement and the matching end statement are copied to the standard output. If the condition is false, all intervening lines are discarded, including control statements. Note: Intervening if statements and matching end statements are recognized solely for the purpose of maintaining the proper if-end matching.

The syntax of a condition is:

```

<cond> ::= ["not"] <or>
<or> ::= <and> | <and> " | " <or>
<and> ::= <exp> | <exp> "&" <and>
<exp> ::= "(" <or> ")" | <value> <op> <value>
<op> ::= "=" | "!=" | "<" | ">"
<value> ::= <arbitrary ASCII string> | <numeric string>

```

The available operators and their meanings are:

|     |                                                                                                      |
|-----|------------------------------------------------------------------------------------------------------|
| =   | equal                                                                                                |
| !=  | not equal                                                                                            |
| &   | and                                                                                                  |
|     | or                                                                                                   |
| >   | greater than                                                                                         |
| <   | less than                                                                                            |
| ()  | used for logical groupings                                                                           |
| not | may only occur immediately after the if, and when present, inverts the value of the entire condition |



The `>` and `<` operate only on unsigned integer values (for example, `: 012 > 12` is false). All other operators take strings as arguments (for example, `: 012 != 12` is true).

The precedence of the operators (from highest to lowest) is:

`= != > <` all of equal precedence

`&`

`|`

Parentheses may be used to alter the order of precedence.

Values must be separated from operators or parentheses by at least one blank or tab.

`::text`

Replace keywords on lines that are copied to the standard output. The two leading control characters are removed, and keywords surrounded by control characters in text are replaced by their value before the line is copied to the output file. This action is independent of the `-a` keyletter.

`:on`

`:off`

Turn on or off keyword replacement on all lines.

`:ctl char`

Change the control character to *char*.

`:msg message`

Print *message* on the diagnostic output.

`:err message`

Print *message* followed by:

ERROR: *err statement* on line ... (915)

on the diagnostic output. `vc` halts execution, and returns an exit code of 1.

## 属性

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE      |
|----------------|----------------------|
| Availability   | developer/build/make |

## 另请参见

[ed\(1\)](#), [attributes\(5\)](#)

**引用名** vgrind – 格式化易阅读的程序列表

**用法概要** vgrind [-2fntwx] [-d *defs-file*] [-h *header*] [-l *language*]  
[-s *n*] [-o *pagelist*] [-P *printer*] [-T *output-device*] *filename...*

**描述** vgrind 实用程序使用 **troff(1)** 命令以易于阅读的风格格式化由 *filename* 参数指定的程序源。注释以斜体显示，关键字以粗体显示，遇到的每个函数的名称在页边空白中列出。

vgrind 以两种基本模式运行：过滤模式或常规模式。在过滤模式中，vgrind 以一种类似于 **tbl(1)** 命令的方式充当过滤器。标准输入直接被传递到标准输出，但用 troff 样式的宏括起来的行除外：

```
.vS 开始处理
.vE 结束处理
```

这些行按照以上所述进行格式化。可以将此过滤器的输出传递到 troff 进行输出。eqn(1) 或 tbl(1) 无需特定的顺序。

在常规模式中，vgrind 接受输入的 *filename*，进行处理并将它们传递到 troff 以便输出。使用连字符 (-) 指定标准输入。否则，vgrind 将会退出，而不会尝试读取标准输入。文件名必须在所有其他选项参数后指定。

在常规模式中，如果指定了 -t 或 -P 选项，则输出如下：

- 在指定了 -t 选项的情况下，以 troff 格式发出到标准输出。
- 在指定了 -P 选项的情况下，以 PostScript 格式输出到指定的打印机。

否则，输出如下：

- 在定义了系统缺省打印机并且该命令的标准输出为 tty 的情况下，以 PostScript 格式输出到该打印机。
- 在标准输出非 tty（即标准输出是到文件的管道或重定向）的情况下，以 PostScript 格式发出到标准输出。

在这两种模式中，vgrind 将传递以小数点开头的任意行，而不进行转换。

**选项** 支持以下选项：

- 2 产生两列输出。指定该选项会将缺省的点尺寸更改为 8（就像提供了 -s8 选项一样）。此外，还会将输出调整为以横向模式显示。
- f 强制使用过滤模式。
- n 不将关键字设置为粗体。
- w 考虑使制表符相隔四列，而不是通常的八列。

- x 以一种易于阅读的格式输出索引文件。只要在运行 `vgrind` 时当前目录中存在一个称为 `index` 的文件，就会自动生成索引文件本身。然后，可以通过在 `vgrind` 中指定 `-x` 选项以及将 `index` 文件作为参数，从而关闭函数定义索引。
- d *defs-file* 指定备用的语言定义文件（缺省为 `/usr/lib/vgrindefs`）。
- h *header* 指定要显示在每个输出页的中间位置的页眉。使用引号指定嵌入了空格的页眉。
- l *language* 指定要使用的语言。当前已知的语言有： Bourne shell (`-lsh`)、C (`-lc`，缺省语言)、C++ (`-lc++`)、C shell (`-lcsh`)、emacs MLisp (`-lmL`)、FORTRAN (`-lf`)、Icon (`-lI`)、ISP (`-i`)、LDL (`-lLDL`)、Model (`-lm`)、Pascal (`-lp`) 以及 RATFOR (`-lr`)。
- P *printer* 将输出发送到指定的 *printer*。
- s *n* 指定要在输出上使用的点尺寸（与 `troff .ps` 点尺寸请求的参数完全相同）。

`vgrind` 将下列选项传递给由 `TROFF` 环境变量指定的格式化程序。请参见“环境变量”部分。

- t 与 `troff` 中的同一选项类似，即格式化文本输出到标准输出。
- o *pagelist* 只输出以逗号分隔的页号及页号范围 *pagelist* 中出现的页号对应的页面。例如，页号范围 `N-M` 表示页面 `N` 到页面 `M`；初始的 `-N` 表示起始页到页面 `N`；最后的 `N-` 表示从页面 `N` 到最后一页。
- T *output-device* 格式化指定的 *output-device* 的输出。

**操作数** 支持下列操作数：

*filename* 要由 `vgrind` 处理的程序源的名称。使用 `'-'` 指定标准输入。

**环境变量** 在常规模式中，`vgrind` 会将其中间输出提供给 `TROFF` 环境变量的值指定的文本格式化程序，或提供给 `/usr/bin/troff`（如果环境中未定义该变量）。该机制允许 `troff` 名称出现局部变化。

|           |                                              |              |
|-----------|----------------------------------------------|--------------|
| <b>文件</b> | <code>index</code>                           | 在其中创建了索引源的文件 |
|           | <code>/usr/lib/vgrindefs</code>              | 语言描述         |
|           | <code>/usr/lib/vfontedpr</code>              | 预处理程序        |
|           | <code>/usr/share/lib/tmac/tmac.vgrind</code> | 宏软件包         |

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值           |
|------|---------------|
| 可用性  | text/doctools |

另请参见

[csh\(1\)](#)、[ctags\(1\)](#)、[eqn\(1\)](#)、[tbl\(1\)](#)、[troff\(1\)](#)、[attributes\(5\)](#)、[vgrindefs\(5\)](#)

已知问题

vgrind 假定遵循某种编程风格：

**C** 每行的函数名称前面只能添加空格、制表符或星号(\*)。括号中的参数还必须在同一行。

**FORTTRAN** 函数名称需和关键字 `function` 或 `subroutine` 显示在同一行。

**MLisp** 函数名称不应和前面的 `defun` 显示在同一行。

**Model** 函数名称需和关键字 `is beginproc` 显示在同一行。

**Pascal** 函数名称需和关键字 `function` 或 `procedure` 显示在同一行。

如果不遵守以上约定，索引建立以及边缘函数名称注释机制将会失败。

更普遍的情况是，任意选择程序格式化样式通常会导致未见过的结果。准备程序以进行 vgrind 输出时，请使用制表符（而非空格字符）将源代码正确对齐，因为 vgrind 使用可变宽度的字体。

此处应使用 [ctags\(1\)](#) 的函数识别机制。

`-w` 选项令人厌烦，但除此之外别无它法能达到所需效果。

`tmac.vgrind` 中定义的宏无法与其他宏软件包中的宏正常共存，这使得过滤模式很难有效利用。

vgrind 无法正确处理 [csh\(1\)](#) 脚本中的某些特殊字符。

`tmac.vgrind` 格式化宏不支持两列模式中使用的页面高度与宽度，这使得两列输出只对美国标准页面大小（8.5 英寸 x 11 英寸）有效，而对其他页面大小毫无用处。对于其他页面大小，需要编辑 `tmac.vgrind` 中给定的大小值。更好的解决方案就是，创建专门针对横向输出的 `troff` 输出设备规范并在其中记录大小信息。

**引用名** vi, view, vedit – screen-oriented (visual) display editor based on ex

**用法概要**

```

/usr/bin/vi [-] -s [-l] [-L] [-R] [-r [filename]] [-S]
 [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

/usr/bin/view [-] -s [-l] [-L] [-R] [-r [filename]] [-S]
 [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

/usr/bin/vedit [-] -s [-l] [-L] [-R] [-r [filename]] [-S]
 [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

/usr/xpg4/bin/vi [-] -s [-l] [-L] [-R] [-r [filename]]
 [-S] [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

/usr/xpg4/bin/view [-] -s [-l] [-L] [-R] [-r [filename]]
 [-S] [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

/usr/xpg4/bin/vedit [-] -s [-l] [-L] [-R] [-r [filename]]
 [-S] [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

/usr/xpg6/bin/vi [-] -s [-l] [-L] [-R] [-r [filename]]
 [-S] [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

/usr/xpg6/bin/view [-] -s [-l] [-L] [-R] [-r [filename]]
 [-S] [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

/usr/xpg6/bin/vedit [-] -s [-l] [-L] [-R] [-r [filename]]
 [-S] [-t tag] [-v] [-V] [-wn]
 [+command | -c command] filename...

```

**描述**

The `vi` (visual) utility is a display-oriented text editor based on an underlying line editor `ex`. It is possible to use the command mode of `ex` from within `vi` and to use the command mode of `vi` from within `ex`. The visual commands are described on this manual page; how to set options (like automatically numbering lines and automatically starting a new output line when you type carriage return) and all `ex` line editor commands are described on the [ex\(1\)](#) manual page.

When using `vi`, changes you make to the file are reflected in what you see on your terminal screen. The position of the cursor on the screen indicates the position within the file.

The `view` invocation is the same as `vi` except that the `readonly` flag is set.

The `vedit` invocation is intended for beginners. It is the same as `vi` except that the `report` flag is set to 1, the `showmode` and `novice` flags are set, and `magic` is turned off. These defaults make it easier to learn how to use `vi`.

## 选项

The following options are supported:

### Invocation Options

The following invocation options are interpreted by `vi` (previously documented options are discussed under NOTES):

- `- | -s`                      Suppresses all interactive user feedback. This is useful when processing editor scripts.
- `-l`                              Sets up for editing LISP programs.
- `-L`                              Lists the name of all files saved as the result of an editor or system crash.
- `-r filename`                Edits *filename* after an editor or system crash. (Recovers the version of *filename* that was in the buffer when the crash occurred.)
- `-R`                              Readonly mode. The `readonly` flag is set, preventing accidental overwriting of the file.
- `-S`                              This option is used in conjunction with the `-t tag` option to tell `vi` that the tags file can not be sorted and that, if the binary search (which relies on a sorted tags file) for *tag* fails to find it, the much slower linear search should also be done. Since the linear search is slow, users of large tags files should ensure that the tags files are sorted rather than use this flag. Creation of tags files normally produces sorted tags files. See [ctags\(1\)](#) for more information on tags files.
- `-t tag`                        Edits the file containing *tag* and position the editor at its definition. It is an error to specify more than one `-t` option.
- `-v`                              Starts up in display editing state, using `vi`. You can achieve the same effect by typing the `vi` command itself.
- `-V`                              Verbose. When `ex` commands are read by means of standard input, the input is echoed to standard error. This can be useful when processing `ex` commands within shell scripts.
- `-wn`                            Sets the default window size to *n*. This is useful when using the editor over a slow speed line.
- `-command | -c command`    Begins editing by executing the specified editor *command* (usually a search or positioning command).

/usr/xpg4/bin/vi and  
/usr/xpg6/bin/vi

If both the `-t tag` and the `-c command` options are given, the `-t tag` option is processed first. That is, the file containing `tag` is selected by `-t` and then the command is executed.

## 操作数

The following operands are supported:

*filename* A file to be edited.

## Command Summary

vi Modes

The vi command modes are summarized in this section.

**Command** Normal and initial mode. Other modes return to command mode upon completion. ESC (escape) is used to cancel a partial command.

**Input** Entered by setting any of the following options:

a A i I o O c C s S R

Arbitrary text can then be entered. Input mode is normally terminated with the ESC character, or, abnormally, with an interrupt.

**Last line** Reading input for `:` `/` `?` or `!`. Terminate by typing a carriage return. An interrupt cancels termination.

## Sample Commands

In the descriptions, CR stands for carriage return and ESC stands for the escape key.

←, →

down-arrow

up-arrow arrow keys move the cursor

h j k l same as arrow keys

i text ESC insert *text*

cw new ESC change word to *new*

ea ESC pluralize word (end of word; append s; escape from input state)

x delete a character

dw delete a word

dd delete a line

3dd delete 3 lines

u undo previous change

ZZ exit vi, saving changes

:q! CR quit, discarding changes

/text CR search for *text*

^U ^D scroll up or down

:cmd CR any ex or ed command

Counts Before vi Commands      Numbers can be typed as a prefix to some commands. They are interpreted in one of these ways:

line/column number      z G |  
scroll amount            ^D ^U  
repeat effect            most of the rest

Interrupting, Canceling    ESC    end insert or incomplete command  
                                 DEL    (delete or rubout) interrupts

File Manipulation          ZZ            if file modified, write and exit; otherwise, exit  
                                 :wCR        write back changes  
                                 :w!CR       forced write, if permission originally not valid  
                                 :qCR        quit  
                                 :q!CR       quit, discard changes  
                                 :e *name*CR   edit file *name*  
                                 :e!CR       reedit, discard changes  
                                 :e + *name*CR   edit, starting at end  
                                 :e +*n*CR    edit, starting at line *n*  
                                 :e #CR      edit alternate file  
                                 :e! #CR     edit alternate file, discard changes  
                                 :w *name*CR   write file *name*  
                                 :w! *name*CR   overwrite file *name*  
                                 :shCR      run shell, then return  
                                 :!*cmd*CR    run *cmd*, then return  
                                 :nCR        edit next file in arglist  
                                 :n *args*CR   specify new arglist  
                                 ^G          show current file and line  
                                 :ta *tag*CR   position cursor to *tag*

In general, any ex or ed command (such as *substitute* or *global*) can be typed, preceded by a colon and followed by a carriage return.



---

|                           |                      |                                                                                          |                                        |
|---------------------------|----------------------|------------------------------------------------------------------------------------------|----------------------------------------|
| Positioning Within a File | F                    | forward screen                                                                           |                                        |
|                           | ^B                   | backward screen                                                                          |                                        |
|                           | ^D                   | scroll down half screen                                                                  |                                        |
|                           | ^U                   | scroll up half screen                                                                    |                                        |
|                           | <i>n</i> G           | go to the beginning of the specified line (end default), where <i>n</i> is a line number |                                        |
|                           | <i>/pat</i>          | next line matching <i>pat</i>                                                            |                                        |
|                           | ? <i>pat</i>         | previous line matching <i>pat</i>                                                        |                                        |
|                           | n                    | repeat last / or ? command                                                               |                                        |
|                           | N                    | reverse last / or ? command                                                              |                                        |
|                           | <i>/pat/+n</i>       | <i>n</i> th line after <i>pat</i>                                                        |                                        |
|                           | ? <i>pat?-n</i>      | <i>n</i> th line before <i>pat</i>                                                       |                                        |
|                           | ]]                   | next section/function                                                                    |                                        |
|                           | [[                   | previous section/function                                                                |                                        |
|                           | (                    | beginning of sentence                                                                    |                                        |
|                           | )                    | end of sentence                                                                          |                                        |
|                           | {                    | beginning of paragraph                                                                   |                                        |
|                           | }                    | end of paragraph                                                                         |                                        |
|                           | %                    | find matching ( ) or { }                                                                 |                                        |
|                           | Adjusting the Screen | ^L                                                                                       | clear and redraw window                |
|                           |                      | ^R                                                                                       | clear and redraw window if ^L is → key |
| zCR                       |                      | redraw screen with current line at top of window                                         |                                        |
| z-CR                      |                      | redraw screen with current line at bottom of window                                      |                                        |
| z.CR                      |                      | redraw screen with current line at center of window                                      |                                        |
| <i>/pat/z-CR</i>          |                      | move <i>pat</i> line to bottom of window                                                 |                                        |
| <i>zn.CR</i>              |                      | use <i>n</i> -line window                                                                |                                        |
| ^E                        |                      | scroll window down one line                                                              |                                        |
| ^Y                        |                      | scroll window up one line                                                                |                                        |
| Marking and Returning     |                      | ``                                                                                       | move cursor to previous context        |
|                           | ^^                   | move cursor to first non-white space in line                                             |                                        |

|                                 |                                              |                                                                                             |
|---------------------------------|----------------------------------------------|---------------------------------------------------------------------------------------------|
|                                 | <code>mx</code>                              | mark current position with the ASCII lower-case letter <i>x</i>                             |
|                                 | <code>`x</code>                              | move cursor to mark <i>x</i>                                                                |
|                                 | <code>´x</code>                              | move cursor to first non-white space in line marked by <i>x</i>                             |
| Line Positioning                | <code>H</code>                               | top line on screen                                                                          |
|                                 | <code>L</code>                               | last line on screen                                                                         |
|                                 | <code>M</code>                               | middle line on screen                                                                       |
|                                 | <code>+</code>                               | next line, at first non-white space character                                               |
|                                 | <code>-</code>                               | previous line, at first non-white space character                                           |
|                                 | <code>CR</code>                              | return, same as <code>+</code>                                                              |
|                                 | <code>down-arrow</code><br>or <code>j</code> | next line, same column                                                                      |
|                                 | <code>up-arrow</code><br>or <code>k</code>   | previous line, same column                                                                  |
| Character Positioning           | <code>^</code>                               | first non-white space character                                                             |
|                                 | <code>0</code>                               | beginning of line                                                                           |
|                                 | <code>\$</code>                              | end of line                                                                                 |
|                                 | <code>l</code> or <code>→</code>             | forward                                                                                     |
|                                 | <code>h</code> or <code>←</code>             | backward                                                                                    |
|                                 | <code>^H</code>                              | same as <code>←</code> (backspace)                                                          |
|                                 | <code>space</code>                           | same as <code>→</code> (space bar)                                                          |
|                                 | <code>fx</code>                              | find next <i>x</i>                                                                          |
|                                 | <code>Fx</code>                              | find previous <i>x</i>                                                                      |
|                                 | <code>tx</code>                              | move to character following the next <i>x</i>                                               |
|                                 | <code>Tx</code>                              | move to character following the previous <i>x</i>                                           |
|                                 | <code>;</code>                               | repeat last <code>f</code> , <code>F</code> , <code>t</code> , or <code>T</code>            |
|                                 | <code>,</code>                               | repeat inverse of last <code>f</code> , <code>F</code> , <code>t</code> , or <code>T</code> |
|                                 | <code>n </code>                              | move to column <i>n</i>                                                                     |
|                                 | <code>%</code>                               | find matching <code>( )</code> or <code>{ }</code>                                          |
| Words, Sentences,<br>Paragraphs | <code>w</code>                               | forward a word                                                                              |
|                                 | <code>b</code>                               | back a word                                                                                 |

---

|                           |                   |                                                                                                                   |
|---------------------------|-------------------|-------------------------------------------------------------------------------------------------------------------|
|                           | e                 | end of word                                                                                                       |
|                           | )                 | to next sentence                                                                                                  |
|                           | }                 | to next paragraph                                                                                                 |
|                           | (                 | back a sentence                                                                                                   |
|                           | {                 | back a paragraph                                                                                                  |
|                           | W                 | forward a blank-delimited word                                                                                    |
|                           | B                 | back a blank-delimited word                                                                                       |
|                           | E                 | end of a blank-delimited word                                                                                     |
| Corrections During Insert | ^H                | erase last character (backspace)                                                                                  |
|                           | ^W                | erase last word                                                                                                   |
|                           | erase             | your erase character, same as ^H (backspace)                                                                      |
|                           | kill              | your kill character, erase this line of input                                                                     |
|                           | \                 | quotes your erase and kill characters                                                                             |
|                           | ESC               | ends insertion, back to command mode                                                                              |
|                           | Control-C         | interrupt, suspends insert mode                                                                                   |
|                           | ^D                | backtab one character; reset left margin of <i>autoindent</i>                                                     |
|                           | ^^D               | caret (^) followed by control-d (^D); backtab to beginning of line; do not reset left margin of <i>autoindent</i> |
|                           | 0^D               | backtab to beginning of line; reset left margin of <i>autoindent</i>                                              |
|                           | ^V                | quote non-printable character                                                                                     |
| Insert and Replace        | a                 | append after cursor                                                                                               |
|                           | A                 | append at end of line                                                                                             |
|                           | i                 | insert before cursor                                                                                              |
|                           | I                 | insert before first non-blank                                                                                     |
|                           | o                 | open line below                                                                                                   |
|                           | O                 | open line above                                                                                                   |
|                           | rx                | replace single character with <i>x</i>                                                                            |
|                           | R <i>text</i> ESC | replace characters                                                                                                |

|                          |                                                                                                                                                                                                                                                                                          |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Operators                | Operators are followed by a cursor motion and affect all text that would have been moved over. For example, since <code>w</code> moves over a word, <code>dw</code> deletes the word that would be moved over. Double the operator, for example <code>dd</code> , to affect whole lines. |
|                          | <code>d</code> delete                                                                                                                                                                                                                                                                    |
|                          | <code>c</code> change                                                                                                                                                                                                                                                                    |
|                          | <code>y</code> yank lines to buffer                                                                                                                                                                                                                                                      |
|                          | <code>&lt;</code> left shift                                                                                                                                                                                                                                                             |
|                          | <code>&gt;</code> right shift                                                                                                                                                                                                                                                            |
|                          | <code>!</code> filter through command                                                                                                                                                                                                                                                    |
| Miscellaneous Operations | <code>C</code> change rest of line ( <code>c\$</code> )                                                                                                                                                                                                                                  |
|                          | <code>D</code> delete rest of line ( <code>d\$</code> )                                                                                                                                                                                                                                  |
|                          | <code>s</code> substitute characters ( <code>c\l</code> )                                                                                                                                                                                                                                |
|                          | <code>S</code> substitute lines ( <code>cc</code> )                                                                                                                                                                                                                                      |
|                          | <code>J</code> join lines                                                                                                                                                                                                                                                                |
|                          | <code>x</code> delete characters ( <code>d\l</code> )                                                                                                                                                                                                                                    |
|                          | <code>X</code> delete characters before cursor ( <code>dh</code> )                                                                                                                                                                                                                       |
|                          | <code>Y</code> yank lines ( <code>yy</code> )                                                                                                                                                                                                                                            |
| Yank and Put             | Put inserts the text most recently deleted or yanked; however, if a buffer is named (using the ASCII lower-case letters <code>a - z</code> ), the text in that buffer is put instead.                                                                                                    |
|                          | <code>3yy</code> yank 3 lines                                                                                                                                                                                                                                                            |
|                          | <code>3yl</code> yank 3 characters                                                                                                                                                                                                                                                       |
|                          | <code>p</code> put back text after cursor                                                                                                                                                                                                                                                |
|                          | <code>P</code> put back text before cursor                                                                                                                                                                                                                                               |
|                          | <code>"xp</code> put from buffer <code>x</code>                                                                                                                                                                                                                                          |
|                          | <code>"xy</code> yank to buffer <code>x</code>                                                                                                                                                                                                                                           |
|                          | <code>"xd</code> delete into buffer <code>x</code>                                                                                                                                                                                                                                       |
| Undo, Redo, Retrieve     | <code>u</code> undo last change                                                                                                                                                                                                                                                          |
|                          | <code>U</code> restore current line                                                                                                                                                                                                                                                      |
|                          | <code>.</code> repeat last change                                                                                                                                                                                                                                                        |
|                          | <code>"dp</code> retrieve <code>d</code> 'th last delete                                                                                                                                                                                                                                 |

- 用法** See [largefile\(5\)](#) for the description of the behavior of vi and view when encountering files greater than or equal to 2 Gbyte (  $2^{31}$  bytes).
- 环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of vi: LANG, LC\_ALL, LC\_COLLATE, LC\_CTYPE, LC\_TIME, LC\_MESSAGES, NLSPATH, PATH, SHELL, and TERM.
- COLUMNS** Override the system-selected horizontal screen size.
- EXINIT** Determine a list of ex commands that are executed on editor start-up, before reading the first file. The list can contain multiple commands by separating them using a vertical-line (|) character.
- LINES** Override the system-selected vertical screen size, used as the number of lines in a screenful and the vertical screen size in visual mode.
- 文件**
- `/var/tmp` default directory where temporary work files are placed; it can be changed using the `directory` option (see the [ex\(1\)](#) command)
  - `/usr/share/lib/terminfo/??/*` compiled terminal description database
  - `/usr/lib/.COREterm/??/*` subset of compiled terminal description database
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/vi,`  
`/usr/bin/view,`  
`/usr/bin/vedit`

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |
| CSI            | Not enabled     |

`/usr/xpg4/bin/vi,`  
`/usr/xpg4/bin/view,`  
`/usr/xpg4/bin/vedit`

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/xopen/xcu4                  |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

`/usr/xpg6/bin/vi,`  
`/usr/xpg6/bin/view,`  
`/usr/xpg6/bin/vedit`

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE   |
|---------------------|-------------------|
| Availability        | system/xopen/xcu6 |
| CSI                 | Enabled           |
| Interface Stability | Standard          |

**另请参见**

[Intro\(1\)](#), [ctags\(1\)](#), [ed\(1\)](#), [edit\(1\)](#), [ex\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

《Solaris Advanced User's Guide》

**Author**

vi and ex were developed by The University of California, Berkeley California, Computer Science Division, Department of Electrical Engineering and Computer Science.

**附注**

Two options, although they continue to be supported, have been replaced in the documentation by options that follow the Command Syntax Standard (see [Intro\(1\)](#)). An `-r` option that is not followed with an option-argument has been replaced by `-L` and `+command` has been replaced by `-c` command.

The message file too large to recover with `-r` option, which is seen when a file is loaded, indicates that the file can be edited and saved successfully, but if the editing session is lost, recovery of the file with the `-r` option is not possible.

The editing environment defaults to certain configuration options. When an editing session is initiated, vi attempts to read the EXINIT environment variable. If it exists, the editor uses the values defined in EXINIT; otherwise the values set in `$HOME/.exrc` are used. If `$HOME/.exrc` does not exist, the default values are used.

To use a copy of `.exrc` located in the current directory other than `$HOME`, set the `exrc` option in EXINIT or `$HOME/.exrc`. Options set in EXINIT can be turned off in a local `.exrc` only if `exrc` is set in EXINIT or `$HOME/.exrc`. In order to be used, `.exrc` in `$HOME` or the current directory must fulfill these conditions:

- It must exist.
- It must be owned by the same userid as the real userid of the process, or the process has appropriate privileges.
- It is not writable by anyone other than the owner.

Tampering with entries in `/usr/share/lib/terminfo/??/*` or `/usr/share/lib/terminfo/??/*` (for example, changing or removing an entry) can affect programs such as vi that expect the entry to be present and correct. In particular, removing the “dumb” terminal can cause unexpected problems.

Software tabs using `^T` work only immediately after the *autoindent*.

Left and right shifts on intelligent terminals do not make use of insert and delete character operations in the terminal.

Loading an alternate `malloc()` library using the environment variable `LD_PRELOAD` can cause problems for `/usr/bin/vi`.

The vi utility currently has the following limitations:

1. Lines, including the trailing NEWLINE character, can contain no more than 4096 bytes.

If a longer line is found, Line too long is displayed in the status line.

2. The editor's temporary work file can be no larger than 128Mb.

If a larger temporary file is needed, Tmp file too large is displayed in the status line.

**引用名** vipw – edit the password file

**用法概要** /usr/ucb/vipw

**描述** vipw edits the password file while setting the appropriate locks, and does any necessary processing after the password file is unlocked. If the password file is already being edited, then you will be told to try again later. The [vi\(1\)](#) editor will be used unless the environment variable VISUAL or EDITOR indicates an alternate editor.

vipw performs a number of consistency checks on the password entry for root, and will not allow a password file with a “mangled” root entry to be installed. It also checks the /etc/shells file to verify the login shell for root.

**文件** /etc/ptmp

/etc/shells

**属性** See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE | ATTRIBUTE VALUE   |
|----------------|-------------------|
| Availability   | compatibility/ucb |

**另请参见** [passwd\(1\)](#), [vi\(1\)](#), [passwd\(4\)](#), [attributes\(5\)](#)



|      |                                                                                                                                                                                                                                                  |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | volcheck – 检查驱动器中的介质                                                                                                                                                                                                                             |
| 用法概要 | volcheck [-v] [-i secs] [-t secs] <i>pathname</i>                                                                                                                                                                                                |
| 描述   | <p>volcheck 实用程序会指示卷管理依次查看每个 <i>dev/pathname</i>，然后确定新介质是否已插入到驱动器中。</p> <p>缺省操作是对由卷管理来管理的所有可检查介质执行 volcheck。</p>                                                                                                                                 |
| 选项   | <p>支持以下选项：</p> <ul style="list-style-type: none"> <li>-i <i>secs</i> 将设备检查的频率设置为 <i>secs</i> 秒。缺省值是 2 秒。最小频率是 1 秒。</li> <li>-t <i>secs</i> 在下一个 <i>secs</i> 秒内检查指定设备。允许的最大秒数是 28800 秒，即 8 小时。检查的频率由 -i 指定。没有缺省的总时间。</li> <li>-v 详细模式。</li> </ul> |
| 操作数  | <p>支持下列操作数：</p> <p><i>pathname</i> 介质设备的路径名称。</p>                                                                                                                                                                                                |
| 文件   | <i>/dev/volctl</i> 卷管理控制端口                                                                                                                                                                                                                       |
| 属性   | 有关下列属性的说明，请参见 <a href="#">attributes(5)</a> ：                                                                                                                                                                                                    |

| 属性类型 | 属性值                                 |
|------|-------------------------------------|
| 可用性  | system/storage/media-volume-manager |

另请参见 [eject\(1\)](#)、[rmmount\(1M\)](#)、[attributes\(5\)](#)

**引用名** volrmount – 调用 rmmount 来挂载或卸载介质

**用法概要** volrmount [-i | -e] [*name* | *nickname*]

volrmount [-d]

**描述** volrmount 实用程序调用 [rmmount\(1M\)](#) 实际上是为了模拟插入 (-i) 或弹出 (-e)。模拟插入通常意味着 rmmount 将挂载介质。与此相反，模拟弹出通常意味着 rmmount 将卸载介质。但是，这些操作可能因 rmmount 配置和介质类型而异。

例如，使用缺省设置时，如果您插入一张音乐 CD，可能无法挂载该 CD。但是，您可以配置 rmmount，使其在任何时候插入音乐 CD 都会调用 workman。

volrmount 实用程序允许您覆盖卷管理的常用介质处理。

**选项** 支持以下选项：

- i 通过调用 rmmount 模拟插入指定介质。
- e 通过调用 rmmount 模拟弹出指定介质。
- d 为要处理的 volrmount 显示缺省设备的名称。如果没有提供 *name* 或 *nickname*，则使用该设备。

**操作数** 支持下列操作数：

*name* 卷管理识别为设备名称的名称。

*nickname* 设备名称的缩短形式。下面列出了识别的别名：

| 昵称      | 路径                       |
|---------|--------------------------|
| cdrom0  | /dev/rdisk/cXtYdZ//label |
| zip0    | /dev/rdisk/cXtYdZ//label |
| jaz0    | /dev/rdisk/cXtYdZ//label |
| rmdisk0 | /dev/rdisk/cXtYdZ//label |

**文件** /dev/volctl 卷管理控制端口

**属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

| 属性类型 | 属性值                                 |
|------|-------------------------------------|
| 可用性  | system/storage/media-volume-manager |

**另请参见** [cpio\(1\)](#)、[eject\(1\)](#)、[tar\(1\)](#)、[rmmount\(1M\)](#)、[attributes\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | w – display information about currently logged-in users                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 用法概要 | w [-hlsuw] [ <i>user</i> ]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 描述   | <p>The <code>w</code> command displays a summary of the current activity on the system, including what each user is doing. The heading line shows the current time, the length of time the system has been up, the number of users logged into the system, and the average number of jobs in the run queue over the last 1, 5 and 15 minutes.</p> <p>The fields displayed are: the user's login name, the name of the tty the user is on, the time of day the user logged on (in <i>hours:minutes</i>), the idle time—that is, the number of minutes since the user last typed anything (in <i>hours:minutes</i>), the CPU time used by all processes and their children on that terminal (in <i>minutes:seconds</i>), the CPU time used by the currently active processes (in <i>minutes:seconds</i>), and the name and arguments of the current process.</p> |
| 选项   | <p>The following options are supported:</p> <ul style="list-style-type: none"> <li>-h Suppresses the heading.</li> <li>-l Produces a long form of output, which is the default.</li> <li>-s Produces a short form of output. In the short form, the tty is abbreviated, the login time and CPU times are left off, as are the arguments to commands.</li> <li>-u Produces the heading line which shows the current time, the length of time the system has been up, the number of users logged into the system, and the average number of jobs in the run queue over the last 1, 5 and 15 minutes.</li> <li>-w Produces a long form of output, which is also the same as the default.</li> </ul>                                                                                                                                                               |
| 操作数  | <i>user</i> Name of a particular user for whom login information is displayed. If specified, output is restricted to that user.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 示例   | <p>示例 1 Sample Output From the <code>w</code> Command</p> <pre>example% w  10:54am up 27 day(s), 57 mins, 1 user, load average: 0.28, 0.26, 0.22 User      tty          login@      idle        JCPU        PCPU        what ralph    console    7:10am     1           10:05      4:31        w</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 环境变量 | See <a href="#">environ(5)</a> for descriptions of the following environment variables that affect the execution of <code>w</code> : <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>LC_TIME</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 文件   | <code>/var/adm/utmpx</code> user and accounting information                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 属性   | See <a href="#">attributes(5)</a> for descriptions of the following attributes:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

---

| ATTRIBUTE TYPE | ATTRIBUTE VALUE |
|----------------|-----------------|
| Availability   | system/core-os  |

另请参见

[ps\(1\)](#), [who\(1\)](#), [whodo\(1M\)](#), [utmpx\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#)

附注

The notion of the “current process” is unclear. The current algorithm is “the highest numbered process on the terminal that is not ignoring interrupts, or, if there is none, the highest numbered process on the terminal”. This fails, for example, in critical sections of programs like the shell and editor, or when faulty programs running in the background fork and fail to ignore interrupts. In cases where no process can be found, `w` prints `-`.

The CPU time is only an estimate, in particular, if someone leaves a background process running after logging out, the person currently on that terminal is “charged” with the time.

Background processes are not shown, even though they account for much of the load on the system.

Sometimes processes, typically those in the background, are printed with null or garbaged arguments. In these cases, the name of the command is printed in parentheses.

`w` does not know about the conventions for detecting background jobs. It will sometimes find a background job instead of the right one.

引用名            wait – await process completion

## 用法概要

/bin/sh            wait [*pid*]...

/bin/jsh /bin/ksh88    wait [*pid*]...

/usr/xpg4/bin/sh    wait [% *jobid*...]

/bin/csh            wait

ksh                wait [*job*...]

## 描述

The shell itself executes `wait`, without creating a new process. If you get the error message `cannot fork, too many processes`, try using the `wait` command to clean up your background processes. If this doesn't help, the system process table is probably full or you have too many active foreground processes. There is a limit to the number of process IDs associated with your login, and to the number the system can keep track of.

Not all the processes of a pipeline with three or more stages are children of the shell, and thus cannot be waited for.

/bin/sh, /bin/jsh    Wait for your background process whose process ID is *pid* and report its termination status. If *pid* is omitted, all your shell's currently active background processes are waited for and the return code is 0. The `wait` utility accepts a job identifier, when Job Control is enabled (`jsh`), and the argument, *jobid*, is preceded by a percent sign (%).

If *pid* is not an active process ID, the `wait` utility returns immediately and the return code is 0.

csh                Wait for your background processes.

ksh88             When an asynchronous list is started by the shell, the process ID of the last command in each element of the asynchronous list becomes known in the current shell execution environment.

If the `wait` utility is invoked with no operands, it waits until all process IDs known to the invoking shell have terminated and exit with an exit status of 0.

If one or more *pid* or *jobid* operands are specified that represent known process IDs (or jobids), the `wait` utility waits until all of them have terminated. If one or more *pid* or *jobid* operands are specified that represent unknown process IDs (or jobids), `wait` treats them as if they were known process IDs (or jobids) that exited with exit status 127. The exit status returned by the `wait` utility is the exit status of the process requested by the last *pid* or *jobid* operand.

The known process IDs are applicable only for invocations of `wait` in the current shell execution environment.

ksh wait with no operands, waits until all jobs known to the invoking shell have terminated. If one or more job operands are specified, wait waits until all of them have completed. Each job can be specified as one of the following:

*number*    *number* refers to a process ID.  
-*number*    *number* refers to a process group ID.  
%*number*    *number* refers to a job number  
%*string*    Refers to a job whose name begins with *string*  
%?*string*    Refers to a job whose name contains *string*  
%+  
%%           Refers to the current job  
%-           Refers to the previous job

If one or more job operands is a process id or process group id not known by the current shell environment, wait treats each of them as if it were a process that exited with status 127.

## 操作数

The following operands are supported:

*pid*        The unsigned decimal integer process ID of a command, for which the utility is to wait for the termination.  
*jobid*      A job control job ID that identifies a background process group to be waited for. The job control job ID notation is applicable only for invocations of wait in the current shell execution environment, and only on systems supporting the job control option.

## 用法

On most implementations, wait is a shell built-in. If it is called in a subshell or separate utility execution environment, such as one of the following,

```
(wait)
nohup wait ...
find . -exec wait ... \;
```

it returns immediately because there is no known process IDs to wait for in those environments.

## 示例

示例 1 Using A Script To Identify The Termination Signal

Although the exact value used when a process is terminated by a signal is unspecified, if it is known that a signal terminated a process, a script can still reliably figure out which signal is using kill, as shown by the following (/bin/ksh88 and /usr/pkg4/bin/sh):

```
sleep 1000&
pid=$!
kill -kill $pid
wait $pid
```

**示例 1 Using A Script To Identify The Termination Signal (续)**

```
echo $pid was terminated by a SIG$(kill -l ${ $?-128}) signal.
```

**示例 2 Returning The Exit Status Of A Process**

If the following sequence of commands is run in less than 31 seconds (/bin/ksh88 and /usr/xpg4/bin/sh):

```
sleep 257 | sleep 31 &
```

```
jobs -l %%
```

then either of the following commands returns the exit status of the second `sleep` in the pipeline:

```
wait <pid of sleep 31>
```

```
wait %%
```

**环境变量**

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `wait`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

**退出状态****ksh**

The following exit values are returned by the `wait` built-in in `ksh`:

- 0**        `wait` was invoked with no operands. All processes known by the invoking process have terminated.
- 127**     `job` is a process id or process group id that is unknown to the current shell environment.

**属性**

See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

**另请参见**

[csh\(1\)](#), [jobs\(1\)](#), [ksh\(1\)](#), [ksh88\(1\)](#), [pwait\(1\)](#), [sh\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 引用名  | <code>wc</code> – display a count of lines, words and characters in a file                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 用法概要 | <code>/usr/bin/wc [-c   -m   -C ] [ -lw ] [file...]</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 描述   | <p><code>wc</code> reads one or more input files and, by default, for each file writes a line containing the number of NEWLINES, words, and bytes contained in each file followed by the file name to standard output in that order. A word is defined to be a non-zero length string delimited by <code>isspace(3C)</code> characters.</p> <p>If more than one file is specified, <code>wc</code> writes a total count for all of the named files with total written instead of the file name.</p> <p>By default, <code>wc</code> writes all three counts. Options can be specified so that only certain counts are written. The <code>-c</code> and <code>-m</code> options are mutually exclusive.</p> <p>If no file is specified, or if the file is <code>-</code>, <code>wc</code> reads from standard input and no filename is written to standard output. The start of the file is defined as the current offset.</p> |
| 选项   | <p>The following options are supported for both <code>/usr/bin/wc</code> and <code>ksh</code>. The long form of the options are only available with <code>ksh</code>:</p> <ul style="list-style-type: none"><li><code>-c</code> Counts bytes.</li><li><code>-C</code> Counts characters. Same as <code>-m</code>.</li><li><code>-l</code> Counts lines.</li><li><code>-m</code> Counts characters. Same as <code>-C</code>.</li><li><code>-w</code> Counts words delimited by white space characters or new line characters. Delimiting characters are Extended Unix Code (EUC) characters from any code set defined by <code>isspace(3C)</code>.</li></ul> <p>If no option is specified, the default is <code>-lwc</code> (counts lines, words, and bytes.)</p>                                                                                                                                                             |
| 操作数  | <p>The following operand is supported:</p> <p><i>file</i> A path name of an input file. If no <i>file</i> operands are specified, the standard input is used.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 用法   | See <a href="#">largefile(5)</a> for the description of the behavior of <code>wc</code> when encountering files greater than or equal to 2 Gbyte ( $2^{31}$ bytes).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 环境变量 | See <a href="#">environ(5)</a> for descriptions of the following environment variables that affect the execution of <code>wc</code> : <code>LANG</code> , <code>LC_ALL</code> , <code>LC_CTYPE</code> , <code>LC_MESSAGES</code> , and <code>NLSPATH</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |



退出状态      0      Successful completion.  
                  >0      An error occurred.

属性            See [attributes\(5\)](#) for descriptions of the following attributes:

| ATTRIBUTE TYPE      | ATTRIBUTE VALUE                    |
|---------------------|------------------------------------|
| Availability        | system/core-os                     |
| CSI                 | Enabled                            |
| Interface Stability | Committed                          |
| Standard            | See <a href="#">standards(5)</a> . |

另请参见      [cksum\(1\)](#), [isspace\(3C\)](#), [iswalph\(3C\)](#), [iswspace\(3C\)](#), [setlocale\(3C\)](#), [attributes\(5\)](#), [environ\(5\)](#), [largefile\(5\)](#), [standards\(5\)](#)

- 引用名**            what – extract SCCS version information from a file
- 用法概要**        what [-s] *filename*...
- 描述**            The `what` utility searches each *filename* for occurrences of the pattern `@(#)` that the SCCS `get` command (see [sccs-get\(1\)](#)) substitutes for the `@(#)` ID keyword, and prints what follows up to a `"`, `>`, `NEWLINE`, `\`, or `NULL` character.
- 选项**            The following option is supported:
- s     Stops after the first occurrence of the pattern.
- 示例**            示例 1 Extracting SCCS version information
- If a C program in file `program.c` contains
- ```
char sccsid[ ] = "@(#)identification information ";
```
- and `program.c` is compiled to yield `program.o` and `a.out`, the command:
- ```
example% what program.c program.o a.out
```
- produces:
- ```
program.c:  identification information
program.o:  identification information
a.out:     identification information
```
- 退出状态** The following exit values are returned:
- 0 Any matches were found.
 - 1 No matches found.
- 环境变量** See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `what`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.
- 属性** See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/build/make
Interface Stability	Committed
Standard	See standards(5) .

另请参见 [sccs\(1\)](#), [sccs-admin\(1\)](#), [sccs-cdc\(1\)](#), [sccs-comb\(1\)](#), [sccs-delta\(1\)](#), [sccs-get\(1\)](#), [sccs-help\(1\)](#), [sccs-prs\(1\)](#), [sccs-prt\(1\)](#), [sccs-rmdel\(1\)](#), [sccs-sact\(1\)](#), [sccs-sccsdiff\(1\)](#), [sccs-unget\(1\)](#), [sccs-val\(1\)](#), [sccsfile\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断

Use the `sccs -help` command for explanations of SCCS commands. See [sccs-help\(1\)](#).

已知问题

There is a remote possibility that a spurious occurrence of the `@(#)` pattern could be found by `what`.

引用名

whatis – display a one-line summary about a keyword

用法概要

whatis *command*...

描述

whatis looks up a given *command* and displays the header line from the manual section. You can then run the [man\(1\)](#) command to get more information. If the line starts *name(section)*... you can do `man -ssection name` to get the documentation for it. Try `whatis ed` and then you should do `man -s 1 ed` to get the manual page for [ed\(1\)](#).

whatis is actually just the `-f` option to the [man\(1\)](#) command.

whatis uses the `/usr/share/man/man_index/*` index files. The index files are either automatically generated by an SMF service as described in [man\(1\)](#) and [man\(5\)](#) or manually generated by using [catman\(1M\)](#) with the `-w` option. If the index files do not exist, `whatis` is run slower since it looks directly into manual page files.

文件

`/usr/share/man/man_index/*` Table of Contents and keyword database.

Generated files include:

- `/usr/share/man/man_index/man.idx`
- `/usr/share/man/man_index/man.dic`
- `/usr/share/man/man_index/man.frq`
- `/usr/share/man/man_index/man.pos`

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/doctools
CSI	Enabled
Interface Stability	Committed

另请参见

[apropos\(1\)](#), [man\(1\)](#), [catman\(1M\)](#), [attributes\(5\)](#), [man\(5\)](#)

引用名	whereis – locate the binary, source, and manual page files for a command
用法概要	<code>/usr/ucb/whereis [-bmsu] [-BMS directory... -f] filename...</code>
描述	<p>The whereis utility locates source/binary and manuals sections for specified files. The supplied names are first stripped of leading pathname components and any (single) trailing extension of the form <i>.ext</i>, for example, <i>.c</i>. Prefixes of <i>s.</i> resulting from use of source code control are also dealt with. whereis then attempts to locate the desired program in a list of standard places:</p> <pre> etc /sbin /usr/bin /usr/lang /usr/lbin /usr/lib /usr/sbin /usr/ucb /usr/ucblib /usr/ucbinclude /usr/games /usr/local /usr/local/bin /usr/new /usr/old /usr/hosts /usr/include </pre>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -b Searches only for binaries. -B Changes or otherwise limits the places where whereis searches for binaries. -f Terminates the last directory list and signals the start of file names, and <i>must</i> be used when any of the -B, -M, or -S options are used. -m Searches only for manual sections. -M Changes or otherwise limits the places where whereis searches for manual sections. -s Searches only for sources. -S Changes or otherwise limit the places where whereis searches for sources. -u Searches for unusual entries. A file is said to be unusual if it does not have one entry of each requested type. Thus <code>'whereis -m -u *'</code> asks for those files in the current directory which have no documentation.

示例

示例1 Finding files

Find all files in `/usr/bin` which are not documented in `/usr/share/man/man1` with source in `/usr/src/cmd`:

```
example% cd /usr/ucb
```

```
example% whereis -u -M /usr/share/man/man1 -S /usr/src/cmd -f *
```

文件

`/usr/src/*`

`/usr/{doc,man}/*`

`/etc, /usr/{lib,bin,ucb,old,new,local}`

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见

[chdir\(2\)](#), [attributes\(5\)](#)

已知问题

Since `whereis` uses [chdir\(2\)](#) to run faster, pathnames given with the `-M`, `-S`, or `-B` must be full; that is, they must begin with a `/`.

引用名	which – locate a command and display its pathname or alias
用法概要	which [<i>name</i>]. . .
描述	<p>which takes a list of names and determines which alias or utility would be executed had these names been given as commands.</p> <p>For each <i>name</i> operand, if it names an alias the alias is expanded. Otherwise the user's path is searched for a utility name matching <i>name</i>. Aliases are taken from the user's <i>.cshrc</i> file. <i>path</i> is taken from the current shell execution environment.</p>
操作数	<p>The following operand is supported:</p> <p><i>name</i> The name of a command to be located.</p>
退出状态	<p>The following exit values are returned:</p> <p>0 Successful completion.</p> <p>>0 One or more <i>name</i> operands were not located or an error occurred.</p>
文件	~/ <i>.cshrc</i> source of aliases and path values
属性	See attributes(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见	csh(1) , attributes(5)
诊断	A diagnostic is given for names which are aliased to more than a single word, or if an executable file with the argument name was not found in the path.
附注	The <i>which</i> utility is not a shell built-in command.
已知问题	To compensate for ~/ <i>.cshrc</i> files in which aliases depend upon the prompt variable being set, <i>which</i> sets this variable to NULL. If the ~/ <i>.cshrc</i> produces output or prompts for input when prompt is set, <i>which</i> can produce some strange results.

引用名 **who** – who is on the system

用法概要

```
/usr/bin/who [-abdHlmpqrstTu] [file]  
/usr/bin/who -q [-n x] [file]  
/usr/bin/who am i  
/usr/bin/who am I  
/usr/xpg4/bin/who [-abdHlmpqrstTu] [file]  
/usr/xpg4/bin/who -q [-n x] [file]  
/usr/xpg4/bin/who -s [-bdHlmpqrtu] [file]  
/usr/xpg4/bin/who am i  
/usr/xpg4/bin/who am I
```

描述

The `who` utility can list the user's name, terminal line, login time, elapsed time since activity occurred on the line, and the process-ID of the command interpreter (shell) for each current UNIX system user. It examines the `/var/adm/utmpx` file to obtain its information. If *file* is given, that file (which must be in `utmpx(4)` format) is examined. Usually, *file* will be `/var/adm/wtmpx`, which contains a history of all the logins since the file was last created.

The general format for output is:

```
name [state] line time [idle] [pid] [comment] [exit]
```

where:

<i>name</i>	User's login name
<i>state</i>	Capability of writing to the terminal
<i>line</i>	Name of the line found in <code>/dev</code>
<i>time</i>	Time since user's login
<i>idle</i>	Time elapsed since the user's last activity
<i>pid</i>	User's process id
<i>comment</i>	Comment line in <code>inittab(4)</code>
<i>exit</i>	Exit status for dead processes

选项 The following options are supported:

- a Processes `/var/adm/utmpx` or the named *file* with -b, -d, -l, -p, -r, -t, -T, and -u options turned on.
- b Indicates the time and date of the last reboot.

-
- d Displays all processes that have expired and not been respawned by `init`. The `exit` field appears for dead processes and contains the termination and exit values (as returned by `wait(3C)`), of the dead process. This can be useful in determining why a process terminated.
 - H Outputs column headings above the regular output.
 - l Lists only those lines on which the system is waiting for someone to login. The `name` field is `LOGIN` in such cases. Other fields are the same as for user entries except that the `state` field does not exist.
 - m Outputs only information about the current terminal.
 - n *x* Takes a numeric argument, *x*, which specifies the number of users to display per line. *x* must be at least 1. The `-n` option can only be used with `-q`.
 - p Lists any other process that is currently active and has been previously spawned by `init`. The `name` field is the name of the program executed by `init` as found in `/usr/sbin/inittab`. The `state`, `line`, and `idle` fields have no meaning. The `comment` field shows the `id` field of the line from `/usr/sbin/inittab` that spawned this process. See `inittab(4)`.
 - q (Quick who) Displays only the names and the number of users currently logged on. When this option is used, all other options are ignored.
 - r Indicates the current *run-level* of the `init` process.
 - s (Default) Lists only the `name`, `line`, and `time` fields.
 - /usr/bin/who -T Same as the `-s` option, except that the `state`, `idle`, `pid`, and `comment`, fields are also written. `state` is one of the following characters:
 - + The terminal allows write access to other users.
 - The terminal denies write access to other users.
 - ? The terminal write-access state cannot be determined.
 - /usr/xpg4/bin/who -T Same as the `-s` option, except that the `state` field is also written. `state` is one of the characters listed under the `/usr/bin/who` version of this option. If the `-u` option is used with `-T`, the idle time is added to the end of the previous format.
 - t Indicates the last change to the system clock (using the `date` utility) by root. See `su(1M)` and `date(1)`.
 - u Lists only those users who are currently logged in. The `name` is the user's login name. The `line` is the name of the line as found in the directory `/dev`. The `time` is the time that the user logged in. The `idle` column contains the number of hours and minutes since activity last occurred on that particular line. A dot (.) indicates that the terminal has seen activity in the last minute and is therefore current. If more than twenty-four hours have elapsed or the line has not been used since boot time, the entry is marked `old`.

This field is useful when trying to determine whether a person is working at the terminal or not. The *pid* is the process-ID of the user's shell. The *comment* is the comment field associated with this line as found in `/usr/sbin/inittab` (see [inittab\(4\)](#)). This can contain information about where the terminal is located, the telephone number of the dataset, type of terminal if hard-wired, and so forth.

操作数

The following operands are supported:

`am i`

`am I` In the C locale, limits the output to describing the invoking user, equivalent to the `-m` option. The `am` and `i` or `I` must be separate arguments.

file Specifies a path name of a file to substitute for the database of logged-on users that who uses by default.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `who`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, `LC_TIME`, and `NLSPATH`.

退出状态

The following exit values are returned:

`0` Successful completion.

`>0` An error occurred.

文件

`/usr/sbin/inittab` Script for `init`

`/var/adm/utmpx` Current user and accounting information

`/var/adm/wtmpx` Historic user and accounting information

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

`/usr/bin/who`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

`/usr/xpg4/bin/who`

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/xopen/xcu4
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[date\(1\)](#), [login\(1\)](#), [mesg\(1\)](#), [init\(1M\)](#), [su\(1M\)](#), [wait\(3C\)](#), [inittab\(4\)](#), [utmpx\(4\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

附注

Superuser: After a shutdown to the single-user state, who returns a prompt. Since `/var/adm/utmpx` is updated at login time and there is no login in single-user state, who cannot report accurately on this state. The command, `who am i`, however, returns the correct information.

引用名 whoami – display the effective current username

用法概要 /usr/ucb/whoami

描述 whoami displays the login name corresponding to the current effective user ID. If you have used su to temporarily adopt another user, whoami will report the login name associated with that user ID. whoami gets its information from the geteuid and getpwuid library routines (see getuid and [getpwnam\(3C\)](#), respectively).

文件 /etc/passwd username data base

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	compatibility/ucb

另请参见 [su\(1M\)](#), [who\(1\)](#), [getuid\(2\)](#), [getpwnam\(3C\)](#), [attributes\(5\)](#)

- 引用名** whocalls – 报告对特定过程的调用
- 用法概要** whocalls [-l *wholib*] [-s] *funcname executable* [*arguments*]...
- 描述** whocalls 是一个基于 [ld.so.1\(1\)](#)（允许跟踪给定的函数调用）的 *Link-Auditing* 功能的实用程序的简单示例。有关 *Link-Auditing* 机制的详细描述，请参见《[链接程序和库指南](#)》。*executable* 通常可以和任何关联的参数一起运行。每次调用 *funcname* 过程时，都会在标准输出中显示该过程的参数以及栈跟踪。
- 选项** 支持以下选项：
- l *wholib* 指定要使用的备用 *who.so Link-Auditing* 库。
 - s 如果可用，检查并使用 *.symtab* 符号表中的局部符号。尽管这比使用 *.dynsym* 符号表成本高一点，但可以生成更详细的栈跟踪信息。

- 示例** 示例1 跟踪函数调用
- 以下示例跟踪一个简单的 *helloworld* 程序对 *printf()* 的调用：

```
example% whocalls printf helloworld
printf(0x106e4, 0xef625310, 0xef621ba8)
    helloworld:main+0x10
    helloworld:_start+0x5c
Hello World
```

- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	developer/base-developer-utilities

- 另请参见** [ld.so.1\(1\)](#)、[sotruss\(1\)](#)、[attributes\(5\)](#)
[《链接程序和库指南》](#)

引用名 whois – Internet user name directory service

用法概要 whois [-h *host*] *identifier*

描述 whois searches for an Internet directory entry for an *identifier* which is either a name (such as ‘Smith’) or a handle (such as ‘SRI-NIC’). To force a name-only search, precede the name with a period; to force a handle-only search, precede the handle with an exclamation point.

To search for a group or organization entry, precede the argument with * (an asterisk). The entire membership list of the group will be displayed with the record.

You may of course use an exclamation point and asterisk, or a period and asterisk together.

示例 示例 1 Using The whois Command

The command:

```
example% whois Smith
```

looks for the name or handle SMITH.

The command:

```
example% whois !SRI-NIC
```

looks for the handle SRI-NIC only.

The command:

```
example% whois .Smith, John
```

looks for the name JOHN SMITH only.

Adding . . . to the name or handle argument will match anything from that point; that is, ZU . . . will match ZUL, ZUM, and so on.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	service/network/network-clients

另请参见 [attributes\(5\)](#)

引用名 write – write to another user

用法概要 write *user* [*terminal*]

描述 The write utility reads lines from the user's standard input and writes them to the terminal of another user. When first invoked, it writes the message:

```
Message from sender-login-id (sending-terminal) [date]...
```

to *user*. When it has successfully completed the connection, the sender's terminal will be alerted twice to indicate that what the sender is typing is being written to the recipient's terminal.

If the recipient wants to reply, this can be accomplished by typing

```
write sender-login-id [sending-terminal]
```

upon receipt of the initial message. Whenever a line of input as delimited by a NL, EOF, or EOL special character is accumulated while in canonical input mode, the accumulated data will be written on the other user's terminal. Characters are processed as follows:

- Typing the alert character will write the alert character to the recipient's terminal.
- Typing the erase and kill characters will affect the sender's terminal in the manner described by the [termios\(3C\)](#) interface.
- Typing the interrupt or end-of-file characters will cause write to write an appropriate message (EOT\n in the C locale) to the recipient's terminal and exit.
- Typing characters from LC_CTYPE classifications print or space will cause those characters to be sent to the recipient's terminal.
- When and only when the stty iexten local mode is enabled, additional special control characters and multi-byte or single-byte characters are processed as printable characters if their wide character equivalents are printable.
- Typing other non-printable characters will cause them to be written to the recipient's terminal as follows: control characters will appear as '^' followed by the appropriate ASCII character, and characters with the high-order bit set will appear in "meta" notation. For example, '\003' is displayed as '^C' and '\372' as 'M-z'.

To write to a user who is logged in more than once, the *terminal* argument can be used to indicate which terminal to write to. Otherwise, the recipient's terminal is the first writable instance of the user found in /usr/adm/utmpx, and the following informational message will be written to the sender's standard output, indicating which terminal was chosen:

```
user is logged on more than one place.
You are connected to terminal.
Other locations are:terminal
```

Permission to be a recipient of a `wri te` message can be denied or granted by use of the `mes g` utility. However, a user's privilege may further constrain the domain of accessibility of other users' terminals. The `wri te` utility will fail when the user lacks the appropriate privileges to perform the requested action.

If the character `!` is found at the beginning of a line, `wri te` calls the shell to execute the rest of the line as a command.

`wri te` runs `setgid()` (see [setuid\(2\)](#)) to the group ID `tty`, in order to have write permissions on other users' terminals.

The following protocol is suggested for using `wri te`: when you first write to another user, wait for them to write back before starting to send. Each person should end a message with a distinctive signal (that is, `(o)` for *over*) so that the other person knows when to reply. The signal `(oo)` (for *over and out*) is suggested when conversation is to be terminated.

操作数

The following operands are supported:

user User (login) name of the person to whom the message will be written. This operand must be of the form returned by the [who\(1\)](#) utility.

terminal Terminal identification in the same format provided by the `who` utility.

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `wri te`: `LANG`, `LC_ALL`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

退出状态

The following exit values are returned:

`0` Successful completion.

`>0` The addressed user is not logged on or the addressed user denies permission.

文件

`/var/adm/utmpx` User and accounting information for `wri te`

`/usr/bin/sh` Bourne shell executable file

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[mail\(1\)](#), [mesg\(1\)](#), [pr\(1\)](#), [sh\(1\)](#), [talk\(1\)](#), [who\(1\)](#), [setuid\(2\)](#), [termios\(3C\)](#), [attributes\(5\)](#),
[environ\(5\)](#), [standards\(5\)](#)

诊断

user is not logged on

The person you are trying to write to is not logged on.

Permission denied

The person you are trying to write to denies that permission (with mesg).

Warning: cannot respond, set mesg-y

Your terminal is set to mesg n and the recipient cannot respond to you.

Can no longer write to user

The recipient has denied permission (mesg n) after you had started writing.

引用名 xargs – construct argument lists and invoke utility

用法概要 xargs [-t] [-p] [-e[*eofstr*]] [-E *eofstr*]
[-I *replstr*] [-i[*replstr*]] [-L *number*] [-l[*number*]]
[-n *number* [-x]] [-s *size*] [*utility* [*argument...*]]

描述 The xargs utility constructs a command line consisting of the *utility* and *argument* operands specified followed by as many arguments read in sequence from standard input as fit in length and number constraints specified by the options. The xargs utility then invokes the constructed command line and waits for its completion. This sequence is repeated until an end-of-file condition is detected on standard input or an invocation of a constructed command line returns an exit status of 255.

Arguments in the standard input must be separated by unquoted blank characters, or unescaped blank characters or newline characters. A string of zero or more non-double-quote (") and non-newline characters can be quoted by enclosing them in double-quotes. A string of zero or more non-apostrophe (') and non-newline characters can be quoted by enclosing them in apostrophes. Any unquoted character can be escaped by preceding it with a backslash (\). The *utility* are executed one or more times until the end-of-file is reached. The results are unspecified if the utility named by *utility* attempts to read from its standard input.

The generated command line length is the sum of the size in bytes of the utility name and each argument treated as strings, including a null byte terminator for each of these strings. The xargs utility limits the command line length such that when the command line is invoked, the combined argument and environment lists can not exceed {ARG_MAX}–2048 bytes. Within this constraint, if neither the -n nor the -s option is specified, the default command line length is at least {LINE_MAX}.

选项 The following options are supported:

- 0 Input items are terminated by a null character instead of by white space or a NEWLINE, and the quotes and backslash are not special, that is, every character is taken literally. The end of file string is also disabled and is treated like any other argument. This is useful when input items might contain white space, quote marks, or backslashes. The find -print0 option produces input suitable for this mode.
- e[*eofstr*] Uses *eofstr* as the logical end-of-file string. Underscore (_) is assumed for the logical EOF string if neither -e nor -E is used. When the *eofstr* option-argument is omitted, the logical EOF string capability is disabled and underscores are taken literally. The xargs utility reads standard input until either end-of-file or the logical EOF string is encountered.
- E *eofstr* Specifies a logical end-of-file string to replace the default underscore. xargs reads standard input until either end-of-file or the logical EOF string is encountered. When *eofstr* is a null string, the logical end-of-file string capability is disabled and underscore characters are taken literally.

- I replstr** Insert mode. *utility* is executed for each line from standard input, taking the entire line as a single argument, inserting it in *arguments* for each occurrence of *replstr*. A maximum of five arguments in *arguments* can each contain one or more instances of *replstr*. Any blank characters at the beginning of each line are ignored. Constructed arguments cannot grow larger than 255 bytes. Option **-x** is forced on. The **-I** and **-i** options are mutually exclusive; the last one specified takes effect.
- i [replstr]** This option is equivalent to **-I replstr**. The string { } is assumed for *replstr* if the option-argument is omitted.
- L number** The *utility* is executed for each non-empty *number* lines of arguments from standard input. The last invocation of *utility* is with fewer lines of arguments if fewer than *number* remain. A line is considered to end with the first newline character unless the last character of the line is a blank character; a trailing blank character signals continuation to the next non-empty line, inclusive. The **-L**, **-l**, and **-n** options are mutually exclusive; the last one specified takes effect.
- l [number]** (The letter ell.) This option is equivalent to **-L number**. If *number* is omitted, 1 is assumed. Option **-x** is forced on.
- n number** Invokes *utility* using as many standard input arguments as possible, up to *number* (a positive decimal integer) arguments maximum. Fewer arguments are used if:
- The command line length accumulated exceeds the size specified by the **-s** option (or {LINE_MAX} if there is no **-s** option), or
 - The last iteration has fewer than *number*, but not zero, operands remaining.
- p** Prompt mode. The user is asked whether to execute *utility* at each invocation. Trace mode (**-t**) is turned on to write the command instance to be executed, followed by a prompt to standard error. An affirmative response (specific to the user's locale) read from /dev/tty executes the command; otherwise, that particular invocation of *utility* is skipped.
- s size** Invokes *utility* using as many standard input arguments as possible yielding a command line length less than *size* (a positive decimal integer) bytes. Fewer arguments are used if:
- The total number of arguments exceeds that specified by the **-n** option, or
 - The total number of lines exceeds that specified by the **-L** option, or
 - End of file is encountered on standard input before *size* bytes are accumulated.

Values of *size* up to at least {LINE_MAX} bytes are supported, provided that the constraints specified in DESCRIPTION are met. It is not considered an error if

a value larger than that supported by the implementation or exceeding the constraints specified in DESCRIPTION is specified. xargs uses the largest value it supports within the constraints.

- t Enables trace mode. Each generated command line is written to standard error just prior to invocation.
- x Terminates if a command line containing *number* arguments (see the -n option above) or *number* lines (see the -L option above) does not fit in the implied or specified size (see the -s option above).

操作数

The following operands are supported:

- utility* The name of the utility to be invoked, found by search path using the PATH environment variable. (see [environ\(5\)](#).) If *utility* is omitted, the default is the [echo\(1\)](#) utility. If the *utility* operand names any of the special built-in utilities in [shell_builtins\(1\)](#), the results are undefined.
- argument* An initial option or operand for the invocation of *utility*.

用法

The 255 exit status allows a utility being used by xargs to tell xargs to terminate if it knows no further invocations using the current data stream succeeds. Thus, *utility* should explicitly exit with an appropriate value to avoid accidentally returning with 255.

Notice that input is parsed as lines. Blank characters separate arguments. If xargs is used to bundle output of commands like `find dir -print` or `ls` into commands to be executed, unexpected results are likely if any filenames contain any blank characters or newline characters. This can be fixed by using `find` to call a script that converts each file found into a quoted string that is then piped to xargs. Notice that the quoting rules used by xargs are not the same as in the shell. They were not made consistent here because existing applications depend on the current rules and the shell syntax is not fully compatible with it. An easy rule that can be used to transform any string into a quoted form that xargs interprets correctly is to precede each character in the string with a backslash (\).

On implementations with a large value for {ARG_MAX}, xargs can produce command lines longer than {LINE_MAX}. For invocation of utilities, this is not a problem. If xargs is being used to create a text file, users should explicitly set the maximum command line length with the -s option.

The xargs utility returns exit status 127 if an error occurs so that applications can distinguish “failure to find a utility” from “invoked utility exited with an error indication.” The value 127 was chosen because it is not commonly used for other meanings; most utilities use small values for “normal error conditions” and the values above 128 can be confused with termination due to receipt of a signal. The value 126 was chosen in a similar manner to indicate that the utility could be found, but not invoked.

示例

示例 1 Using the xargs command

The following example moves all files from directory \$1 to directory \$2, and echo each move command just before doing it:

```
example% ls $1 | xargs -I {} -t mv $1/{} $2/{} 
```

The following command combines the output of the parenthesised commands onto one line, which is then written to the end of file log:

```
example% (logname; date; printf "%s\n" "$0 $*") | xargs >>log
```

The following command invokes `diff` with successive pairs of arguments originally typed as command line arguments (assuming there are no embedded blank characters in the elements of the original argument list):

```
example% printf "%s\n" "$*" | xargs -n 2 -x diff
```

The user is asked which files in the current directory are to be archived. The files are archived into `arch`; a, one at a time, or b, many at a time:

```
example% ls | xargs -p -L 1 ar -r arch
ls | xargs -p -L 1 | xargs ar -r arch
```

The following executes with successive pairs of arguments originally typed as command line arguments:

```
example% echo $* | xargs -n 2 diff
```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of `xargs`: `LANG`, `LC_ALL`, `LC_COLLATE`, `LC_CTYPE`, `LC_MESSAGES`, and `NLSPATH`.

`PATH` Determine the location of *utility*.

Affirmative responses are processed using the extended regular expression defined for the `yesexpr` keyword in the `LC_MESSAGES` category of the user's locale. The locale specified in the `LC_COLLATE` category defines the behavior of ranges, equivalence classes, and multi-character collating elements used in the expression defined for `yesexpr`. The locale specified in `LC_CTYPE` determines the locale for interpretation of sequences of bytes of text data a characters, the behavior of character classes used in the expression defined for the `yesexpr`. See [locale\(5\)](#).

退出状态

The following exit values are returned:

- 0 All invocations of *utility* returned exit status 0.
- 1–125 A command line meeting the specified requirements could not be assembled, one or more of the invocations of *utility* returned a non-zero exit status, or some other error occurred.
- 126 The utility specified by *utility* was found but could not be invoked.

127 The utility specified by *utility* could not be found.

If a command line meeting the specified requirements cannot be assembled, the utility cannot be invoked, an invocation of the utility is terminated by a signal, or an invocation of the utility exits with exit status 255, the `xargs` utility writes a diagnostic message and exit without processing any remaining input.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os
CSI	Enabled
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[echo\(1\)](#), [shell_builtins\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

引用名	xgettext – extract gettext call strings from C programs
用法概要	<pre>xgettext [-ns] [-a [-x <i>exclude-file</i>]] [-c <i>comment-tag</i>] [-d <i>default-domain</i>] [-j] [-m <i>prefix</i>] [-M <i>suffix</i>] [-p <i>pathname</i>] - <i>filename...</i></pre> <p>xgettext -h</p>
描述	<p>The xgettext utility is used to automate the creation of portable message files (.po). A .po file contains copies of “C” strings that are found in ANSI C source code in <i>filename</i> or the standard input if ‘-’ is specified on the command line. The .po file can be used as input to the msgfmt(1) utility, which produces a binary form of the message file that can be used by application during run-time.</p> <p>xgettext writes <i>msgid</i> strings from gettext(3C) calls in <i>filename</i> to the default output file <i>messages.po</i>. The default output file name can be changed by -d option. <i>msgid</i> strings in <code>dgettext()</code> calls are written to the output file <i>domainname.po</i> where <i>domainname</i> is the first parameter to the <code>dgettext()</code> call.</p> <p>By default, xgettext creates a .po file in the current working directory, and each entry is in the same order that the strings are extracted from <i>filenames</i>. When the -p option is specified, the .po file is created in the <i>pathname</i> directory. An existing .po file is overwritten.</p> <p>Duplicate <i>msgids</i> are written to the .po file as comment lines. When the -s option is specified, the .po is sorted by the <i>msgid</i> string, and all duplicated <i>msgids</i> are removed. All <i>msgstr</i> directives in the .po file are empty unless the -m option is used.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -n Add comment lines to the output file indicating file name and line number in the source file where each extracted string is encountered. These lines appear before each <i>msgid</i> in the following format: # # File: <i>filename</i>, line: <i>line-number</i> -s Generate output sorted by <i>msgids</i> with all duplicate <i>msgids</i> removed. -a Extract all strings, not just those found in gettext(3C), and <code>dgettext()</code> calls. Only one .po file is created. -c <i>comment-tag</i> The comment block beginning with <i>comment-tag</i> as the first token of the comment block is added to the output .po file as # delimited comments. For multiple domains, xgettext directs comments and messages to the prevailing text domain. -d <i>default-domain</i> Rename default output file from <i>messages.po</i> to <i>default-domain.po</i>. -j Join messages with existing message files. If a .po file does not exist, it is created. If a .po file does exist, new messages are appended. Any duplicate <i>msgids</i> are commented out in the resulting .po file. Domain

- directives in the existing `.po` file are ignored. Results not guaranteed if the existing message file has been edited.
- `-m prefix` Fill in the `msgstr` with `prefix`. This is useful for debugging purposes. To make `msgstr` identical to `msgid`, use an empty string ("") for `prefix`.
 - `-M suffix` Fill in the `msgstr` with `suffix`. This is useful for debugging purposes.
 - `-p pathname` Specify the directory where the output files will be placed. This option overrides the current working directory.
 - `-x exclude-file` Specify a `.po` file that contains a list of `msgid`s that are not to be extracted from the input files. The format of `exclude-file` is identical to the `.po` file. However, only the `msgid` directive line in `exclude-file` is used. All other lines are simply ignored. The `-x` option can only be used with the `-a` option.
 - `-h` Print a help message on the standard output.

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	text/locale

另请参见

[msgfmt\(1\)](#), [gettext\(3C\)](#), [attributes\(5\)](#)

附注

`xgettext` is not able to extract cast strings, for example ANSI C casts of literal strings to `(const char *)`. This is unnecessary anyway, since the prototypes in `<libintl.h>` already specify this type.

In messages and translation notes, lines greater than 2048 characters are truncated to 2048 characters and a warning message is printed to `stderr`.

引用名	xstr – extract strings from C programs to implement shared strings
用法概要	<pre>xstr -c filename [-v] [-l array] xstr [-l array] xstr filename [-v] [-l array]</pre>
描述	<p>xstr maintains a file called <code>strings</code> into which strings in component parts of a large program are hashed. These strings are replaced with references to this common area. This serves to implement shared constant strings, which are most useful if they are also read-only.</p> <p>The command:</p> <pre>example% xstr -c filename</pre> <p>extracts the strings from the C source in <code>name</code>, replacing string references by expressions of the form <code>&xstr[number]</code> for some number. An appropriate declaration of <code>xstr</code> is prepended to the file. The resulting C text is placed in the file <code>x.c</code>, to then be compiled. The strings from this file are placed in the <code>strings</code> data base if they are not there already. Repeated strings and strings which are suffixes of existing strings do not cause changes to the data base.</p> <p>After all components of a large program have been compiled, a file declaring the common <code>xstr</code> space called <code>xs.c</code> can be created by a command of the form:</p> <pre>example% xstr</pre> <p>This <code>xs.c</code> file should then be compiled and loaded with the rest of the program. If possible, the array can be made read-only (shared) saving space and swap overhead.</p> <p><code>xstr</code> can also be used on a single file. A command:</p> <pre>example% xstr filename</pre> <p>creates files <code>x.c</code> and <code>xs.c</code> as before, without using or affecting any <code>strings</code> file in the same directory.</p> <p>It may be useful to run <code>xstr</code> after the C preprocessor if any macro definitions yield strings or if there is conditional code which contains strings which may not, in fact, be needed. <code>xstr</code> reads from the standard input when the argument <code>'-'</code> is given. An appropriate command sequence for running <code>xstr</code> after the C preprocessor is:</p> <pre>example% cc -E name.c xstr -c - example% cc -c x.c example% mv x.o name.o</pre> <p><code>xstr</code> does not touch the file <code>strings</code> unless new items are added; thus make(1S) can avoid remaking <code>xs.o</code> unless truly necessary.</p>

选项	<code>-c filename</code>	Take C source text from <i>filename</i> .
	<code>-v</code>	Verbose: display a progress report indicating where new or duplicate strings were found.
	<code>-l array</code>	Specify the named <i>array</i> in program references to abstracted strings. The default array name is <code>xstr</code> .
文件	<code>strings</code>	data base of strings
	<code>x.c</code>	massaged C source
	<code>xs.c</code>	C source for definition of array " <code>xstr*(rq</code>
	<code>/tmp/xs*</code>	temp file when <code>xstr filename</code> doesn't touch <code>strings</code>
属性	See attributes(5) for descriptions of the following attributes:	

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [make\(1S\)](#), [attributes\(5\)](#)

已知问题
If a string is a suffix of another string in the data base, but the shorter string is seen first by `xstr` both strings will be placed in the data base, when just placing the longer one there would do.

附注
Be aware that `xstr` indiscriminately replaces all strings with expressions of the form `&xstr[number]` regardless of the way the original C code might have used the string. For example, you will encounter a problem with code that uses `sizeof()` to determine the length of a literal string because `xstr` will replace the literal string with a pointer that most likely will have a different size than the string's.

To circumvent this problem:

- use `strlen()` instead of `sizeof()`; note that `sizeof()` returns the size of the array (including the null byte at the end), whereas `strlen()` doesn't count the null byte. The equivalent of `sizeof("xxx")` really is `(strlen("xxx")+1)`.
- use `#define` for operands of `sizeof()` and use the define'd version. `xstr` ignores `#define` statements. Make sure you run `xstr` on *filename* before you run it on the preprocessor.

You will also encounter a problem when declaring an initialized character array of the form

```
char x[] = "xxx";
```

`xstr` will replace `xxx` with an expression of the form `&xstr[number]` which will not compile. To circumvent this problem, use `static char *x = "xxx"` instead of `static char x[] = "xxx"`.

引用名	yacc – yet another compiler-compiler
用法概要	yacc [-dltVv] [-b <i>file_prefix</i>] [-Q [y n]] [-P <i>parser</i>] [-p <i>sym_prefix</i>] <i>file</i>
描述	<p>The yacc command converts a context-free grammar into a set of tables for a simple automaton that executes an LALR(1) parsing algorithm. The grammar can be ambiguous. Specified precedence rules are used to break ambiguities.</p> <p>The output file, <i>y.tab.c</i>, must be compiled by the C compiler to produce a function <code>yyparse()</code>. This program must be loaded with the lexical analyzer program, <code>yylex()</code>, as well as <code>main()</code> and <code>yyerror()</code>, an error handling routine. These routines must be supplied by the user. The <code>lex(1)</code> command is useful for creating lexical analyzers usable by yacc.</p>
选项	<p>The following options are supported:</p> <ul style="list-style-type: none"> -b <i>file_prefix</i> Uses <i>file_prefix</i> instead of <i>y</i> as the prefix for all output files. The code file <i>y.tab.c</i>, the header file <i>y.tab.h</i> (created when <code>-d</code> is specified), and the description file <i>y.output</i> (created when <code>-v</code> is specified), is changed to <i>file_prefix.tab.c</i>, <i>file_prefix.tab.h</i>, and <i>file_prefix.output</i>, respectively. -d Generates the file <i>y.tab.h</i> with the <code>#define</code> statements that associate the yacc user-assigned “token codes” with the user-declared “token names”. This association allows source files other than <i>y.tab.c</i> to access the token codes. -l Specifies that the code produced in <i>y.tab.c</i> does not contain any <code>#line</code> constructs. This option should only be used after the grammar and the associated actions are fully debugged. -p <i>sym_prefix</i> Uses <i>sym_prefix</i> instead of <i>yy</i> as the prefix for all external names produced by yacc. The names affected include the functions <code>yyparse()</code>, <code>yylex()</code> and <code>yyerror()</code>, and the variables <i>yyval</i>, <i>yychar</i> and <i>yydebug</i>. (In the remainder of this section, the six symbols cited are referenced using their default names only as a notational convenience.) Local names can also be affected by the <code>-p</code> option. However, the <code>-p</code> option does not affect <code>#define</code> symbols generated by yacc. -P <i>parser</i> Allows you to specify the parser of your choice instead of <code>/usr/share/lib/ccs/yaccpar</code>. For example, you can specify: example% yacc -P ~/myparser parser.y -Q[y n] The <code>-Qy</code> option puts the version stamping information in <i>y.tab.c</i>. This allows you to know what version of yacc built the file. The <code>-Qn</code> option (the default) writes no version information. -t Compiles runtime debugging code by default. Runtime debugging code is always generated in <i>y.tab.c</i> under conditional compilation control. By default, this code is not included when <i>y.tab.c</i> is compiled. Whether or not

the `-t` option is used, the runtime debugging code is under the control of `YYDEBUG`, a preprocessor symbol. If `YYDEBUG` has a non-zero value, then the debugging code is included. If its value is `0`, then the code is not included. The size and execution time of a program produced without the runtime debugging code is smaller and slightly faster.

- `-v` Prepares the file `y.output`, which contains a description of the parsing tables and a report on conflicts generated by ambiguities in the grammar.
- `-V` Prints on the standard error output the version information for yacc.

操作数

The following operand is required:

file A path name of a file containing instructions for which a parser is to be created.

示例

示例 1 Accessing the yacc Library

Access to the yacc library is obtained with library search operands to `cc`. To use the yacc library `main`:

```
example% cc y.tab.c -ly
```

Both the `lex` library and the `yacc` library contain `main`. To access the `yacc main`:

```
example% cc y.tab.c lex.yy.c -ly -ll
```

This ensures that the `yacc` library is searched first, so that its `main` is used.

The historical yacc libraries have contained two simple functions that are normally coded by the application programmer. These library functions are similar to the following code:

```
#include <locale.h>
int main(void)
{
    extern int yyparse();

    setlocale(LC_ALL, "");

    /* If the following parser is one created by lex, the
       application must be careful to ensure that LC_CTYPE
       and LC_COLLATE are set to the POSIX locale. */
    (void) yyparse();
    return (0);
}

#include <stdio.h>

int yyerror(const char *msg)
{
    (void) fprintf(stderr, "%s\n", msg);
}
```

 示例 1 Accessing the yacc Library (续)

```

    return (0);
}

```

环境变量

See [environ\(5\)](#) for descriptions of the following environment variables that affect the execution of yacc: LANG, LC_ALL, LC_CTYPE, LC_MESSAGES, and NLSPATH.

yacc can handle characters from EUC primary and supplementary codesets as one-token symbols. EUC codes can only be single character quoted terminal symbols. yacc expects `yylex()` to return a wide character (`wchar_t`) value for these one-token symbols.

退出状态

The following exit values are returned:

```

0      Successful completion.
>0    An error occurred.

```

文件

```

y.output      state transitions of the generated parser
y.tab.c       source code of the generated parser
y.tab.h       header file for the generated parser
yacc.acts     temporary file
yacc.debug    temporary file
yacc.tmp      temporary file
yaccpar       parser prototype for C programs

```

属性

See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	developer/base-developer-utilities
Interface Stability	Committed
Standard	See standards(5) .

另请参见

[lex\(1\)](#), [attributes\(5\)](#), [environ\(5\)](#), [standards\(5\)](#)

诊断

The number of reduce-reduce and shift-reduce conflicts is reported on the standard error output. A more detailed report is found in the `y.output` file. Similarly, if some rules are not reachable from the start symbol, this instance is also reported.

附注

Because file names are fixed, at most one yacc process can be active in a given directory at a given time.

Users are encouraged to avoid using `$` as part of any identifier name.

引用名 yes – generate repetitive affirmative output

用法概要 yes [*term*]...

描述 The yes utility repeatedly outputs *y*, or if *term* is specified, *term* is output repeatedly. In the output, either *y* or *term* is followed by a NEWLINE. Multiple arguments are output separated by spaces and followed by a NEWLINE. To terminate yes, issue an interrupt character.

yes can be used to respond programatically to programs that require an interactive response.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/core-os

另请参见 [attributes\(5\)](#)

引用名 ypcat – print values in a NIS database

用法概要 ypcat [-kx] [-d *ypdomain*] *mname*

描述 The ypcat command prints out values in the NIS name service map specified by *mname*, which may be either a map name or a map nickname. Since ypcat uses the NIS network services, no NIS server is specified.

Refer to [ypfiles\(4\)](#) for an overview of the NIS name service.

选项

- k Display the keys for those maps in which the values are null or the key is not part of the value. None of the maps derived from files that have an ASCII version in /etc fall into this class.
- d *ypdomain* Specify a domain other than the default domain.
- x Display map nicknames.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

另请参见 [ypmatch\(1\)](#), [ypfiles\(4\)](#), [attributes\(5\)](#)

引用名 ypmatch – print the value of one or more keys from a NIS map

用法概要 ypmatch [-k] [-t] [-d *domain*] *key* [*key*]... *mname*

ypmatch -x

描述 ypmatch prints the values associated with one or more keys from the NIS's name services map specified by *mname*, which may be either a map name or a map nickname.

Multiple keys can be specified; all keys will be searched for in the same map. The keys must be the same case and length. No pattern matching is available. If a key is not matched, a diagnostic message is produced.

选项 The following options are supported:

-k Before printing the value of a key, print the key itself, followed by a colon (:).

-t Inhibit map nickname translation.

-d *domain* Specify a domain other than the default domain.

-x Display the map nickname table. This lists the nicknames the command knows of, and indicates the map name associated with each nickname.

操作数 The following operand is supported:

mname The NIS's name services map

退出状态 The following exit values are returned:

0 Successful operation.

1 An error occurred.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

另请参见 [ypcat\(1\)](#), [ypfiles\(4\)](#), [attributes\(5\)](#)

附注 ypmatch will fail with an RPC error message on yp operation if enough file descriptors are not available. The number of file descriptors should be increased if this occurs.

引用名 yppasswd – change your network password in the NIS database

用法概要 yppasswd [*username*]

描述 The yppasswd utility changes the network password associated with the user *username* in the Network Information Service (NIS) database. If the user has done a [keylogin\(1\)](#), and a publickey/secretkey pair exists for the user in the NIS publickey.byname map, yppasswd also re-encrypts the secretkey with the new password. The NIS password may be different from the local one on your own machine.

yppasswd prompts for the old NIS password, and then for the new one. You must type in the old password correctly for the change to take effect. The new password must be typed twice, to forestall mistakes.

New passwords must be at least four characters long, if they use a sufficiently rich alphabet, and at least six characters long if monospace. These rules are relaxed if you are insistent enough. Only the owner of the name or the super-user may change a password; superuser on the root master will not be prompted for the old password, and does not need to follow password construction requirements.

The NIS password daemon, rpc.yppasswdd must be running on your NIS server in order for the new password to take effect.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

另请参见 [keylogin\(1\)](#), [login\(1\)](#), [passwd\(1\)](#), [getpwnam\(3C\)](#), [getspnam\(3C\)](#), [secure_rpc\(3NSL\)](#), [nsswitch.conf\(4\)](#), [attributes\(5\)](#)

警告 Even after the user has successfully changed his or her password using this command, the subsequent [login\(1\)](#) using the new password will be successful only if the user's password and shadow information is obtained from NIS. See [getpwnam\(3C\)](#), [getspnam\(3C\)](#), and [nsswitch.conf\(4\)](#).

附注 The use of yppasswd is discouraged, as it is now only a wrapper around the [passwd\(1\)](#) command, which should be used instead. Using [passwd\(1\)](#) with the `-r nis` option will achieve the same results, and will be consistent across all the different name services available.

已知问题 The update protocol passes all the information to the server in one RPC call, without ever looking at it. Thus, if you type your old password incorrectly, you will not be notified until after you have entered your new password.

引用名 ypwhich – return name of NIS server or map master

用法概要 ypwhich [-d *domain*] [[-t] -m [*mname*] | [-Vn] *hostname*]

ypwhich -x

描述 ypwhich returns the name of the NIS server that supplies the NIS name services to a NIS client, or which is the master for a map. If invoked without arguments, it gives the NIS server for the local machine. If *hostname* is specified, that machine is queried to find out which NIS master it is using.

Refer to [ypfiles\(4\)](#) for an overview of the NIS name services.

选项

- d *domain* Use *domain* instead of the default domain.
- t This option inhibits map nickname translation.
- m *mname* Find the master NIS server for a map. No *hostname* can be specified with -m. *mname* can be a mapname, or a nickname for a map. When *mname* is omitted, produce a list of available maps.
- x Display the map nickname translation table.
- Vn Version of ypbind, V3 is default.

属性 See [attributes\(5\)](#) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Availability	system/network/nis

另请参见 [ypfiles\(4\)](#), [attributes\(5\)](#)

引用名	zlogin - 进入一个区域
用法概要	<pre>zlogin [-dCE] [-e c] [-l username] zonename zlogin [-ES] [-e c] [-l username] zonename utility [argument]...</pre>
描述	<p>zlogin 实用程序用于进入操作系统区域。只有在全局系统区域内进行操作的用户才能使用该实用程序，并且执行该实用程序时必须拥有所有特权。此外，用户必须先获得授权，然后才能使用“选项”部分中所述的特定选项。</p> <p>zlogin 检查前缀可为指定的区域名称并且以斜杠字符开头的授权字符串。如果省略区域名称，用户将有权进入任何区域。</p> <p>zlogin 采用以下三种模式之一运行：</p> <p>交互模式 如果未指定任何实用程序参数并且 zlogin 进程的标准输入文件描述符为 tty 设备，则 zlogin 的运行模式为交互模式。在该模式下，zlogin 会创建一个新的伪终端供在登录会话中使用。需要 tty 设备的程序（如 vi(1)）可以在该模式下正常运行。在该模式下，zlogin 会调用 login(1) 以提供合适的登录会话。</p> <p>非交互模式 如果指定了实用程序，则 zlogin 的运行模式为非交互模式。该模式对脚本作者很有用，因为 stdin、stdout 和 stderr 均被保留，并且一旦终止就会返回 utility 的退出状态。在该模式下，zlogin 会调用 su(1M) 以便设置用户环境并提供登录环境。</p> <p style="padding-left: 40px;">指定的命令以字符串形式进行传递，并由在非全局区域中运行的 shell 解释。请参见 rsh(1)。</p> <p>控制台模式 如果指定了 -c 选项，用户将连接到区域控制台设备并且 zlogin 会在控制台模式下运行。一旦区域处于已安装状态，便可使用区域控制台。在重新引导区域的过程中，将始终保持与控制台的连接。</p>
选项	<p>支持以下选项：</p> <p>-C 连接到区域控制台。访问区域控制台需具备 zone.manage/zonename 授权。</p> <p>-d 如果区域停止，则断开与控制台的连接。该选项只能和 -c 选项一起指定。</p> <p>-e c 指定另外一个转义符 c，作为用于访问扩展函数以及断开登录连接的键序。缺省的转义符为波浪号 (-)。</p> <p>-E 使用转义序列符禁止访问扩展函数或断开登录连接。</p> <p>-l username 为区域登录指定其他 username。如果不使用该选项，则使用的区域用户名为 root。如果指定了 -C 选项，该选项将无效。</p>

用户名必须在区域内有效。对于交互式登录，需具备 `solaris.zone.login/zonename` 授权并且在区域内进行口令验证。对于非交互式登录或要跳进口令验证，则需具备 `solaris.zone.manage/zonename` 授权。

-S 安全登录模式。zlogin 将进行最少处理，并且不调用 `login(1)` 或 `su(1M)`。区域用户名设置为 `root`。如果通过 `-l` 选项指定了用户名，而该用户名无法用于控制台登录，则 `-S` 选项将无效。只有当其他登录方式变得不可用时，才应使用该模式恢复受损的区域。

使用该选项需具备 `solaris.zone.manage/zonename` 授权。

转义序列

键入的以波浪号字符 (~) 开头的行即“转义序列”。转义符可通过 `-e` 选项更改。

~. 与区域断开连接。与区域断开连接和注销不同，因为本地主机在断开连接时不会向区域结尾发送警告信息。

安全

一旦将某进程置于全局区域以外的区域，该进程及其所有子进程便无法重新更改区域。

操作数

支持下列操作数：

zonename 要进入的区域的名称。

utility 要在指定区域中运行的实用程序。

argument... 传递给实用程序的参数。

退出状态

在交互模式和非交互模式下，当该命令或非全局区域中的 shell 退出时，zlogin 实用程序也会退出。在非交互模式下，将返回远程程序的退出状态作为 zlogin 的退出状态。在交互模式和控制台登录模式下，不会返回退出状态。只要没有出现与连接有关的错误，zlogin 便会返回 0 退出状态。

在所有模式下，如果无法建立与区域的连接，连接将会意外失败；或者如果用户没有足够的特权执行所请求的操作，zlogin 将会退出并返回状态 1。

综上所述，返回的退出值如下：

0 成功进入区域。

1 权限被拒绝或无法进入区域。

任何 来自实用程序或 `su(1M)` 的返回码（如果在非交互模式下运行）。

属性

有关下列属性的说明，请参见 `attributes(5)`：

属性类型	属性值
可用性	system/zones

属性类型	属性值
接口稳定性	Committed (已确定)

另请参见

[login\(1\)](#)、[rsh\(1\)](#)、[vi\(1\)](#)、[su\(1M\)](#)、[zoneadm\(1M\)](#)、[zonecfg\(1M\)](#)、[attributes\(5\)](#)、[zones\(5\)](#)

附注

如果 `zlogin` 的打开文件或其地址空间的任何部分与某个 NFS 文件对应，该命令将会失败。这包括该可执行文件本身或共享库。

- 引用名** zonename – 输出当前区域的名称
- 用法概要** zonename
- 描述** zonename 实用程序输出当前区域的名称。
- 退出状态** 将返回以下退出值：
- 0 成功完成。
 - >0 出现错误。
- 属性** 有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/core-os
接口稳定性	Committed（已确定）

- 另请参见** [zlogin\(1\)](#)、[zoneadm\(1M\)](#)、[zonecfg\(1M\)](#)、[attributes\(5\)](#)、[zones\(5\)](#)

引用名	zonestat – 报告活动区域的统计信息
用法概要	<pre>zonestat [-z zonelist] [-r reslist] [-n namelist] [-T u d i] [-R reports] [-q] [-x] [-p [-P lines]] [-S cols] interval [duration [report]]</pre>
描述	<p>zonestat 实用程序会报告当前正在运行的区域的 cpu、内存、网络和资源控制使用情况。对于每个区域的使用情况，都将报告系统资源百分比和区域所配置的限制两方面的内容。</p> <p>zonestat 实用程序按指定的时间间隔输出一系列的间隔报告。它还可以按指定的时间间隔输出一个或多个摘要报告。</p> <p>缺省输出是 cpu、物理内存、虚拟内存和网络使用情况的摘要。可使用 -r 选项为特定资源选择详细的输出。</p>
安全性	<p>在非全局区域 (NGZ) 内运行时，只会报告对 NGZ 可见的处理器集。NGZ 输出中包括所有其他系统资源，例如内存和限制。</p> <p>对于报告的所有资源，都会输出 NGZ 的使用情况。由系统、全局区域及所有其他区域使用的每个资源都会报告为由 [system] 使用。</p> <p>对于网络资源，只会输出 NGZ 的使用情况。NGZ 对其他区域的网络资源和统计信息不具有可见性。</p> <p>proc_info 特权是使用 zonestat 实用程序所必需的。该特权是基本特权集的成员。</p>
选项	<p>支持以下选项：</p> <p>-n name[,name] 指定要报告的资源名称的列表。对于 pset 资源，这是处理器集合的名称。对于 physical-memory、locked-memory 和 virtual-memory 资源，可指定的名称只有 mem_default 和 vm_default。对于网络资源，这是数据链路的名称。</p> <p>专用 CPU 处理器集可由其 pset 名称 (SUNWtmp_zonename) 指定或只由其 zonename 指定。</p> <p>由 psrset 创建的处理器集可由其池 pset 名称 (SUNWlegacy_pset id) 指定，或只由 pset id 指定。</p> <p>除了用逗号分隔的列表，还可以指定多个 -n 选项来报告一组资源。</p> <p>-p 可解析的输出。</p> <p>以计算机可解析的稳定格式显示输出。各个字段以冒号 (:) 进行分隔。行的格式为：</p> <pre>report type:resource:field[:field]*</pre> <p>如果指定了 -T，每行会以时间戳作为前缀：</p>

报告类型包括：report-total、report-average、report-high 和 interval。

资源类型包

括：header、footer、summary、physical-memory、virtual-memory、locked-memory、processor-s

header 资源是用于声明间隔或摘要报告的开始的一个特殊资源。位于 header 资源之间的所有输出行都属于同一个报告。每个 header 都有一个相匹配的 footer。

其余字段是特定于资源类型的。有关详细信息，请参见 zonestat 实用程序。

所有现有的输出字段都是稳定的。将来的版本可能会引入新的报告和资源类型。将来的版本还可能会在现有输出 N 行的末尾添加其他新字段。

-P line[,line]

对于可解析的输出，请指定要在可解析的输出中输出的行。可以选择下列行类型中的一个或多个：

header、footer	每个间隔和摘要报告都包含一个 header，它输出诸如间隔和计数信息等方面的详细信息。在每个报告后还会输出一个 footer。
resource	描述每个资源的行。
system	系统对每个资源的使用率。这包括内核和不是由特定区域消耗的任何资源。当 zonestat 在非全局区域内运行时，该值是系统和所有其他区域所消耗的总资源。不支持网络资源类型的 system 使用情况。
total	每个资源的总使用率。
zones	用于按区域详细列出每个资源的使用率的行。

-q

静默模式。只输出摘要报告（需要 -R 选项）。将忽略所有间隔报告。

-r resource[,resource]

指定要报告的资源类型。可用的资源包

括：physical-memory、virtual-memory、locked-memory、processor-set、processes、lwps、shm-ids、sem-ids、msg-ids、lofi 和 network。

summary	cpu、physical-memory、virtual memory 和 network 使用情况的摘要。
memory (内存)	physical-memory、virtual-memory 和 locked-memory。
psets	processor-set
default-pset	仅缺省的 pset。
limits	processes、lwps、lofi。
network	网络数据链路。

`sysv` `shm-memory`、`shm-ids`、`sem-ids` `msg-ids`。

`all` 所有资源类型。

缺省情况下将输出 `summary` 资源。

除了以逗号分隔的列表，还可以指定多个 `-r` 选项来报告一组资源类型。

系统的 `cpu` 可划分为处理器集 (`psets`)。缺省情况下，所有 `cpu` 都位于名为 `pset_default` 的单个 `pset` 中。

内存不能划分为不同的集。`zonestat` 实用程序针对这些资源的输出中将它们的名称显示为 `mem_default` 和 `vm_default`。

`all` 资源指定应该报告所有资源类型。

`-R report[,report]`

输出一个摘要报告。下面介绍了支持的报告类型。除了以逗号分隔的列表，还可以指定多个 `-R` 选项以输出一组摘要报告。

`total` 输出为每个资源详细列出以下内容的摘要报告：

`psets` 自启动命令调用以来所占用的总 `cpu` 时间。计算每个区域所占用的百分比时会考虑区域未运行的时间。例如，如果区域在运行时占用了 100% 的 `cpu`，但该区域运行时间间隔的一半后便停止，摘要报告便会显示该区域占用了 50% 的 `cpu` 时间。

`memory`、`limits`、`sysv` 自调用命令以来所报告的所有间隔的平均资源占用值。此平均值将区域未运行的时间间隔考虑在内。例如，如果某个区域在运行时占用的物理内存平均值为 100M，且只运行时间间隔的一半，摘要报告便会显示该区域平均占用了 50M 的物理内存。

`network` 由所有利用物理带宽的数据链路传输和接收的所有字节的总和。总和是自启动命令调用以来所计算的字节总数，并标准化为每秒的字节数。所使用的百分比基于总可用带宽。

`average` 与 `total` 类似，但只将区域运行的时间间隔考虑在内。例如，如果某个区域只在单个时间间隔内运行，且在该时间间隔内使用了 200M 虚拟内存，无论在该摘要报告之前报告了多少个时间间隔，其使用的平均虚拟内存都为 200M。

`high` 输出详细列出在调用 `zonestat` 实用程序的任何时间间隔内每个资源和区域的最高使用率的摘要报告。

-S col[,col]

对使用每个资源的区域进行排序。

可指定以下排序列：

name	按区域名称的字母数字顺序排序。
used	按所使用的资源数量排序。对于网络资源，这与按字节排序相同。 这是缺省值。
cap	按所配置的上限排序。
pcap	按所使用的上限的百分比排序。
shr	按所分配的份额排序。
pshru	按所使用的份额的百分比排序。
bytes	按传输和接收的总字节对网络资源进行排序。
prbyte	按通过线路接收的字节百分比对网络资源进行排序。
pobyte	按通过线路传输的字节百分比对网络资源进行排序。
maxbw	按使用的带宽百分比对网络资源进行排序。
cpu	在摘要中按 cpu 使用率排序。这是缺省值。
physical-memory	在摘要中按物理内存的使用率排序。
virtual-memory	在摘要中按虚拟内存的使用率排序。
network	在摘要中按网络使用率排序。
network	在摘要中按网络使用率排序。

-T u | d | i

包含每个报告的时间戳。支持下列格式：

- d** 标准日期格式。请参见 [date\(1\)](#)。此选项对 **--p** 无效。
- i** 使用符合 ISO 8601 格式设置的时间：
YYYYMMDDThhmmssZ
- u** 时间的内部表示形式的印刷表示形式。请参见 [time\(2\)](#)。这也称为 **unix** 时间。

-x

显示具有更多详细信息的扩展视图。例如，当与网络资源一起使用时，扩展视图会列出每个虚拟数据链路的详细信息。

-z zonename[,zonename]

指定要报告的区域的列表。缺省情况下会报告所有区域。

除了以逗号分隔的列表，还可以指定多个 `-z` 选项来报告一组区域。输出中包括指定的区域使用过的任何资源。

操作数

支持下列操作数：

interval

指定每个间隔报告之间暂停的时间长度（以秒为单位）。缺省时间间隔将使用为区域监视服务配置的时间间隔。请参见 `zonestatd(1M)`。

interval 是必需的。*interval* 不能为零。*interval* 可以指定为 `[nh][nm][n s]`，例如 `10s` 或 `1m`。

duration

指定要报告的时间间隔数。如果未指定，则缺省使用 `infinity`。命令持续时间为 $(interval * duration)$ 。*duration* 不能为零。还可以指定一个 `inf` 值以显式选择 `infinity`。

还可以将 *duration* 指定为 `[nh][nm][ns]`。在这种情况下，*duration* 被解释为执行时间的持续期。实际的 *duration* 会舍入到最接近的时间间隔倍数。

report

指定摘要报告周期。例如，*report* 为 4 时，每 4 个时间间隔生成一次报告。如果命令持续时间不是 *report* 的倍数，则最后一个报告为任何剩余时间间隔的内容。

还可以将 *report* 指定为 `[nh][nm][ns]`。在这种情况下，报告会按指定的时间段输出，并舍入到最近的时间间隔。如果命令 *duration* 不是 *report* 的倍数，则最后一个报告为任何剩余时间间隔的内容。

需要 `-R`。如果指定了 `-R` 但未指定 *report*，则报告周期为整个命令持续时间，即在命令执行结束时生成指定的报告。

输出

以下列表定义了命令输出的列标题：

SYSTEM-MEMORY

物理主机上可用的内存总量。

SYSTEM-LIMIT

物理主机上可用的最大资源量。

CPUS

分配给一个处理器集的 `cpu` 数量

ONLINE

在分配给一个处理器集的 `cpu` 中，可以执行进程的 `cpu` 的数量。

MIN/MAX

可由系统分配给处理器集的 `cpu` 的最小数量和最大数量。

ZONE

使用资源的区域。除了区域名称，此列还可以包含：

- [total] 系统范围内使用的资源总量。
- [system] 由内核使用的或以不与任何特定区域关联的方式使用的资源量。
- 当在非全局区域中使用 `zonestat` 时，[system] 指示由系统和所有其他区域使用的总体资源。
- 对于网络资源，不提供网络的系统使用情况。

USED

所使用的资源量。

%USED

所使用的资源量占总资源的百分比。

PCT

所使用的资源量占总资源的百分比。

%PART

所使用的 `cpu` 量占该区域绑定到的处理器集中的总 `cpu` 的百分比。如果某个区域是全局区域，或者如果使用了 `psrset(1M) psets`，则该区域只能将进程绑定到多个处理器集。如果为某个区域找到了多个绑定，则 **%PART** 就是所使用的 `cpu` 量占有所有绑定 `psets` 的百分比。对于 [total] 和 [system]，**%PART** 是所使用的 `cpu` 量占系统上所有 `cpu` 的百分比。

CAP

如果为某个区域在指定资源配置了上限，则会在此列中显示该上限。

%CAP

所使用的资源量占区域中配置的上限的百分比。

SHRS

分配给区域的份额数量。对于 [total] 行，这是分配给共享此资源的所有区域的份额总数。如果某个区域未配置为使用份额，并且与配置为使用份额的其他区域共享某个资源，则此列会针对该区域包含 `no-fss`。

%SHRS

分配给此区域的份额占总份额的百分比。例如，如果 2 个区域共享一个处理器集合，每个具有 10 个份额，则每个区域的 **%SHR** 为 50%。

%SHRU

在分配给区域的份额中，占资源的百分比为 100% 的份额。由于份额只会在存在资源争用时实施，因此区域可能具有超过 100% 的 **%SHRU**。

TOBYTES

由数据链路或虚拟链路传输和接收的字节数。

PRBYTE

消耗物理带宽的接收字节数。

POBYTE

消耗物理带宽的传输字节数。

%PRBYTE

用于接收 PRBYTE 的可用物理带宽百分比。

%POBYTE

用于传输 POBYTE 的可用物理带宽百分比。

%PUSE

PRBYTE 和 POBYTE 的总和占总可用物理带宽的百分比。

LINK

数据链路的名称。

MAXBW

在数据链路上配置的最大带宽。

%MAXBW

所有传输和接收的字节总和占配置的最大带宽的百分比。

示例

示例 1 使用 zonestat 显示 cpu 和内存使用情况的摘要

以下命令会每隔 5 秒钟显示一次 cpu 和内存使用情况的摘要：

```
# zonestat 5 1
SUMMARY Cpus/Online: 4/4 Physical: 8063M Virtual: 11.8G
---CPU--- --PHYSMEM-- ---VMEM--- ---NET---
ZONE USED %PART USED %USED USED %USED PBYTE %PUSE
[total] 0.23 5.76% 3211M 39.8% 4191M 34.6% 350M 18.7%
[system] 0.03 0.83% 2791M 34.6% 3890M 32.1% - -
global 0.19 4.86% 324M 4.01% 228M 1.89% 200M 10.7%
zoneA 0.00 0.03% 47.9M 0.59% 36.3M 0.30% 100M 5.3%
zoneB 0.00 0.02% 48.1M 0.59% 36.4M 0.30% 50M 2.7%
```

示例 2 使用 zonestat 生成可解析的输出

以下命令会生成可解析的输出。此命令以 5 秒的时间间隔为每个使用 pset 资源的区域输出一行：

```
# zonestat -p -P zones -r psets 5 1
```

示例 3 使用 zonestat 针对缺省的 pset 进行报告

以下命令在一分钟的时间内每一秒针对缺省的 pset 报告一次：

```
# zonestat -r default-pset 1 1m
```

示例 4 使用 zonestat 报告总使用率和最高使用率

以下命令在 24 个小时内以 10 秒的时间间隔静默监视，并每隔 1 小时生成总使用率和最高使用率报告：

示例4 使用 zonestat 报告总使用率和最高使用率 (续)

```
# zonestat -q -R total,high 10s 24h 1h
```

示例5 使用 zonestat 报告数据链路使用率

以下命令以 5 秒的时间间隔对名为 e1000g0 的数据链路报告 5 次：

```
# zonestat -r network -n e1000g0 5 5
```

退出状态

将返回以下退出值：

- 0 成功完成。
- 1 出现错误。
- 2 用法无效。
- 3 svc:system/zones_monitoring: 缺省情况下未运行或未响应。

属性

有关下列属性的说明，请参见 [attributes\(5\)](#)：

属性类型	属性值
可用性	system/zones
接口稳定性	请参见下文。

命令调用和可解析的输出是 "Committed"（已确定）。用户可读的输出（缺省输出）是 "Uncommitted"（未确定）。

另请参见

[date\(1\)](#)、[prctl\(1\)](#)、[pooladm\(1M\)](#)、[poolcfg\(1M\)](#)、[psrset\(1M\)](#)、[rcapadm\(1M\)](#)、[zoneadm\(1M\)](#)、[zonec](#)

附注

zonestat 实用程序依赖于区域监视服务：svc/system/zonestat:default。如果运行 zonestat 实用程序时 zonestat 服务停止，则 zonestat 命令调用将退出，且不再输出任何报告。如果在到达下一个报告周期之前，zonestat 被 (CTRL/c、SIGINT) 中断，则会输出报告 (-R)。