

# 在 Oracle® Solaris 11.2 中管理系统信息、进程和性能

ORACLE®

文件号码 E53838-02  
2014 年 9 月

版权所有 © 2014, 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

---

使用本文档 .....	7
<b>1 管理系统信息 .....</b>	<b>9</b>
显示系统信息 .....	9
用于显示系统信息的命令 .....	9
识别有关芯片多线程功能的信息 .....	18
更改系统信息 .....	20
更改系统信息任务列表 .....	20
▼ 如何手动设置系统的日期和时间 .....	20
▼ 如何设置每日消息 .....	21
▼ 如何更改系统标识 .....	21
<b>2 管理系统进程 .....</b>	<b>23</b>
无需管理的系统进程 .....	23
管理系统进程 .....	23
管理系统进程任务列表 .....	24
用于管理系统进程的命令 .....	24
显示和管理进程类信息 .....	32
显示进程类信息 .....	32
管理进程类信息任务列表 .....	34
更改进程的调度优先级 (priocntl) .....	35
▼ 如何指定进程优先级 (priocntl) .....	35
▼ 如何更改分时进程的调度参数 (priocntl) .....	36
▼ 如何更改进程的类 (priocntl) .....	37
更改分时进程的优先级 (nice) .....	37
更改进程优先级 (nice) .....	38
系统进程问题的故障排除 .....	39
<b>3 监视系统性能 .....</b>	<b>41</b>
在何处查找有关监视系统性能的信息 .....	41

使用 Oracle Enterprise Manager Ops Center 管理性能 .....	41
关于影响系统性能的系统资源 .....	42
关于进程和系统性能 .....	42
关于监视系统性能 .....	44
监视工具 .....	44
显示系统性能信息 .....	45
显示虚拟内存统计信息 .....	45
显示虚拟内存统计信息 (vmstat) .....	46
显示系统事件信息 (vmstat -s) .....	46
显示交换统计信息 (vmstat -S) .....	47
显示每台设备的中断次数 (vmstat -i) .....	47
显示磁盘使用率信息 .....	48
显示磁盘空间统计信息 (df) .....	50
监视系统活动 .....	51
监视系统活动 (sar) .....	51
自动收集系统活动数据 (sar) .....	67
4 调度系统任务 .....	71
自动执行系统任务的方法 .....	71
使用 crontab 调度重复性作业 .....	72
使用 at 调度单个作业 .....	72
调度系统任务 .....	73
创建和编辑 crontab 文件任务列表 .....	73
调度重复性系统任务 (cron) .....	73
创建和编辑 crontab 文件 .....	75
显示和验证 crontab 文件 .....	77
删除 crontab 文件 .....	78
控制对 crontab 命令的访问 .....	79
使用 at 命令来调度任务 .....	82
使用 at 命令 .....	82
调度单个系统任务 (at) .....	82
5 管理系统控制台、终端设备和电源服务 .....	89
管理系统控制台和本地连接的终端设备 .....	89
管理系统控制台和本地连接的终端设备的 SMF 服务 .....	89
管理系统电源服务 .....	91
▼ 如何在维护模式下从电源服务中恢复 .....	94

索引 ..... 95



## 使用本文档

---

- 概述 - 介绍了管理系统信息、进程和监视性能的任务。
- 目标读者 - 使用 Oracle Solaris 11 发行版的系统管理员
- 必备知识 - UNIX 系统管理经验

## 产品文档库

有关本产品的最新信息和已知问题均包含在文档库中，网址为：<http://www.oracle.com/pls/topic/lookup?ctx=E56344>。

## 获得 Oracle 支持。

Oracle 客户可通过 My Oracle Support 获得电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>；如果您听力受损，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>。

## 反馈

可以在 <http://www.oracle.com/goto/docfeedback> 上提供有本文档的反馈。



# ◆◆◆ 第 1 章

## 管理系统信息

---

本章介绍显示和更改基本系统信息所需执行的任务。

有关可用于灵活地分配、监视和控制系统资源的资源管理的信息，请参见《在 Oracle Solaris 11.2 中进行资源管理》中的第 1 章“资源管理介绍”。

以下列出本章所介绍的信息：

- “显示系统信息” [9]
- “更改系统信息” [20]

### 显示系统信息

本节介绍可用来显示常规系统信息的命令。

### 用于显示系统信息的命令

表 1-1 用于显示系统信息的命令

命令	显示的系统信息	手册页
date	日期和时间	<a href="#">date(1)</a>
hostid	主机 ID 号	<a href="#">hostid(1)</a>
isainfo	正在运行的系统中本机应用程序所支持的位数，该位数可以作为标记传递给脚本	<a href="#">isainfo(1)</a>
isalist	处理器类型	<a href="#">isalist(1)</a>
prtconf	系统配置信息，已安装的内存、设备属性和产品名称	<a href="#">prtconf(1M)</a>
prtdiag	系统配置和诊断信息，包括任何发生故障的现场可更换单元 (Field Replaceable Unit, FRU)	<a href="#">prtdiag(1M)</a>
psrinfo	处理器信息	<a href="#">psrinfo(1M)</a>
uname	操作系统名称、发行版、版本、节点名称、硬件名称和处理器类型	<a href="#">uname(1)</a>

## 显示系统的发行版信息

显示 `/etc/release` 文件的内容，以确定发行版的版本。

```
$ cat /etc/release
```

## 显示日期和时间

要根据系统时钟显示当前的日期和时间，请使用 `date` 命令。

以下示例显示 `date` 命令的样例输出。

```
$ date
Fri Jun 1 16:07:44 MDT 2012
$
```

## 显示系统的主机 ID 号

要以数字（十六进制）格式显示主机 ID 号，请使用 `hostid` 命令。

以下示例显示 `hostid` 命令的样例输出。

```
$ hostid
80a5d34c
```

## 显示系统的体系结构类型

使用 `isainfo` 命令显示如何为当前操作系统支持的应用程序显示本机指令集的体系结构类型和名称。

以下示例输出来自基于 x86 的系统：

```
$ isainfo
amd64 i386
```

以下示例输出来自基于 SPARC 的系统：

```
$ isainfo
sparcv9 sparc
```

`isainfo -v` 命令显示 32 位和 64 位应用程序支持。例如，以下示例输出来自基于 SPARC 的系统：

```
$ isainfo -v
64-bit sparcv9 applications
```

```

    asi_blk_init
32-bit sparc applications
    asi_blk_init v8plus div32 mul32
#

```

以下示例显示 `isainfo -v` 命令在基于 x86 的系统上的输出：

```

$ isainfo -v
64-bit amd64 applications
    sse4.1 ssse3 ahf cx16 sse3 sse2 sse fxsr mmx cmov amd_sysc cx8 tsc fpu
32-bit i386 applications
    sse4.1 ssse3 ahf cx16 sse3 sse2 sse fxsr mmx cmov sep cx8 tsc fpu

```

请参见 [isainfo\(1\)](#) 手册页。

有关更多信息，请参见 [isainfo\(1\)](#) 手册页。

## 显示系统的处理器类型

使用 `isalist` 命令可以显示有关系统的处理器类型的信息。

The following sample output is from an x86 based system:

```

$ isalist
pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86

```

以下示例输出来自基于 SPARC 的系统：

```

$ isalist
sparcv9 sparcv8plus sparcv8 sparcv8-fsmuld sparcv7 sparc sparcv9+vis sparcv9+vis2 \
sparcv8plus+vis sparcv8plus+vis2

```

请参见 [isalist\(1\)](#) 手册页。

## 显示系统的产品名称

要显示系统的产品名称，请将 `prtconf` 命令与 `-b` 选项结合使用：

```

$ prtconf -b

```

有关更多信息，请参见 [prtconf\(1M\)](#) 手册页。

该示例显示来自基于 SPARC 的系统上的 `prtconf -b` 命令示例输出。

```

$ prtconf -b
name: ORCL,SPARC-T4-2
banner-name: SPARC T4-2
compatible: 'sun4v'
$

```

该示例显示来自基于 SPARC 的系统上的 `prtconf -vb` 命令示例输出。添加的 `-v` 选项指定详细输出。

```
$ prtconf -vb
name: ORCL,SPARC-T3-4
banner-name: SPARC T3-4
compatible: 'sun4v'
idprom: 01840014.4fa02d28.00000000.a02d28de.00000000.00000000.00000000.00000000
openprom model: SUNW,4.33.0.b
openprom version: 'OBP 4.33.0.b 2011/05/16 16:26'
```

## 显示系统的已安装内存

要显示系统中已安装的内存量，请结合使用 `prtconf` 命令和 `grep Memory` 命令。以下示例显示示例输出，其中，`grep Memory` 命令选择 `prtconf` 命令的输出，以便只显示内存信息。

```
$ prtconf | grep Memory
Memory size: 523776 Megabytes
```

## 显示设备的缺省和定制属性值

要显示设备的缺省属性值和定制属性值，请将 `prtconf` 命令和 `-u` 选项结合使用。

```
$ prtconf -u
```

`prtconf -u` 命令的输出显示系统上所有驱动程序的缺省属性和定制属性。

有关此选项的更多信息，请参见 [prtconf\(1M\)](#) 手册页。

### 例 1-1 SPARC: 显示缺省和定制设备属性

此示例显示了 `bge.conf` 文件的缺省属性和定制属性。请注意，供应商提供的配置文件位于 `/kernel` 和 `/platform` 目录，而修改后的相应驱动程序配置文件则位于 `/etc/driver/drv` 目录。

```
$ prtconf -u
System Configuration: Oracle Corporation sun4v
Memory size: 523776 Megabytes
System Peripherals (Software Nodes):

ORCL,SPARC-T3-4
  scsi_vhci, instance #0
  disk, instance #4
  disk, instance #5
  disk, instance #6
  disk, instance #8
```

```

disk, instance #9
disk, instance #10
disk, instance #11
disk, instance #12
packages (driver not attached)
  SUNW,builtin-drivers (driver not attached)
  deblocker (driver not attached)
  disk-label (driver not attached)
  terminal-emulator (driver not attached)
  dropins (driver not attached)
  SUNW,asr (driver not attached)
  kbd-translator (driver not attached)
  obp-tftp (driver not attached)
  zfs-file-system (driver not attached)
  hsfs-file-system (driver not attached)
chosen (driver not attached)
openprom (driver not attached)
  client-services (driver not attached)
options, instance #0
aliases (driver not attached)
memory (driver not attached)
virtual-memory (driver not attached)
iscsi-hba (driver not attached)
  disk, instance #0 (driver not attached)
virtual-devices, instance #0
  flashprom (driver not attached)
  tpm, instance #0 (driver not attached)
  n2cp, instance #0
  ncp, instance #0
  random-number-generator, instance #0
  console, instance #0
  channel-devices, instance #0
    virtual-channel, instance #0
    virtual-channel, instance #1
    virtual-channel-client, instance #2
    virtual-channel-client, instance #3
    virtual-domain-service, instance #0
cpu (driver not attached)

```

例 1-2 x86: 显示缺省和定制设备属性

此示例显示了 bge.conf 文件的缺省属性和定制属性。请注意，供应商提供的配置文件位于 /kernel 和 /platform 目录，而修改后的相应驱动程序配置文件则位于 /etc/driver/drv 目录。

```

$ prtconf -u
System Configuration: Oracle Corporation i86pc

```

```
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
  scsi_vhci, instance #0
  pci, instance #0
    pci10de,5e (driver not attached)
    isa, instance #0
      asy, instance #0
      motherboard (driver not attached)
      pit_bEEP, instance #0
    pci10de,cb84 (driver not attached)
    pci108e,cb84, instance #0
      device, instance #0
        keyboard, instance #0
        mouse, instance #1
    pci108e,cb84, instance #0
    pci-ide, instance #0
      ide, instance #0
        sd, instance #0
        ide (driver not attached)
    pci10de,5c, instance #0
      display, instance #0
    pci10de,cb84, instance #0
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci1022,1100, instance #0
    pci1022,1101, instance #1
    pci1022,1102, instance #2
    pci1022,1103 (driver not attached)
    pci1022,1100, instance #3
    pci1022,1101, instance #4
    pci1022,1102, instance #5
    pci1022,1103 (driver not attached)
  pci, instance #1
    pci10de,5e (driver not attached)
    pci10de,cb84 (driver not attached)
    pci10de,cb84, instance #1
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci10de,5d (driver not attached)
    pci1022,7458, instance #1
    pci1022,7459 (driver not attached)
    pci1022,7458, instance #2
      pci8086,1011, instance #0
      pci8086,1011, instance #1
      pci1000,3060, instance #0
        sd, instance #1
        sd, instance #2
      pci1022,7459 (driver not attached)
    ioapics (driver not attached)
```

```

        ioapic, instance #0 (driver not attached)
        ioapic, instance #1 (driver not attached)
    fw, instance #0
        cpu (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
        cpu (driver not attached)
    sb, instance #1
used-resources (driver not attached)
iscsi, instance #0
fcoe, instance #0
pseudo, instance #0
options, instance #0
xsvc, instance #0
vga_arbiter, instance #0

```

### 例 1-3 x86: 显示系统配置信息

以下示例显示在基于 x86 的系统上如何使用带有 `-v` 选项的 `prtconf` 命令来确定与系统连接的磁盘、磁带和 DVD 设备。此命令的输出在不存在设备的设备实例旁显示了 "driver not attached" 消息。

```

$ prtconf -v | more
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

i86pc
  System properties:
    name='#size-cells' type=int items=1
      value=00000002
    name='#address-cells' type=int items=1
      value=00000003
    name='relative-addressing' type=int items=1
      value=00000001
    name='MMU_PAGEOFFSET' type=int items=1
      value=00000fff
    name='MMU_PAGESIZE' type=int items=1
      value=00001000
    name='PAGESIZE' type=int items=1
      value=00001000
    name='acpi-status' type=int items=1
      value=00000013
    name='biosdev-0x81' type=byte items=588
      value=01.38.74.0e.08.1e.db.e4.fe.00.d0.ed.fe.f8.6b.04.08.d3.db.e4.fe
    .
    .
    .

```

有关更多信息，请参见 [driver\(4\)](#)、[driver.conf\(4\)](#) 和 [prtconf\(1M\)](#) 手册页。

有关如何创建通过管理行为提供的配置文件的说明，请参见《在 Oracle Solaris 11.2 中管理设备》中的第 1 章“在 Oracle Solaris 中管理设备”。

## 显示系统诊断信息

使用 `prtdiag` 命令显示系统的配置和诊断信息。

```
$ prtdiag [-v] [-l]
```

`-v`                    详细模式。

`-l`                    记录输出。如果系统中存在故障或错误，则仅将此信息输出到 `syslogd(1M)`。

例 1-4                SPARC: 显示系统诊断信息

以下示例显示 `prtdiag -v` 命令在基于 SPARC 的系统上的示例输出。为简短起见，示例做了截断。

```
$ prtdiag -v | more
```

```
System Configuration: Oracle Corporation sun4v Sun Fire T200
Memory size: 16256 Megabytes
```

```
===== Virtual CPUs =====
```

CPU ID	Frequency	Implementation	Status
0	1200 MHz	SUNW,UltraSPARC-T1	on-line
1	1200 MHz	SUNW,UltraSPARC-T1	on-line
2	1200 MHz	SUNW,UltraSPARC-T1	on-line
3	1200 MHz	SUNW,UltraSPARC-T1	on-line
4	1200 MHz	SUNW,UltraSPARC-T1	on-line
5	1200 MHz	SUNW,UltraSPARC-T1	on-line
6	1200 MHz	SUNW,UltraSPARC-T1	on-line

```
.
.
.
```

```
===== Physical Memory Configuration =====
```

```
Segment Table:
```

Base Address	Segment Size	Interleave Factor	Bank Size	Contains Modules
0x0	16 GB	4	2 GB	MB/CMP0/CH0/R0/D0 MB/CMP0/CH0/R0/D1
			2 GB	MB/CMP0/CH0/R1/D0 MB/CMP0/CH0/R1/D1
			2 GB	MB/CMP0/CH1/R0/D0 MB/CMP0/CH1/R0/D1
			2 GB	MB/CMP0/CH1/R1/D0

```
.
.
```

```

System PROM revisions:
-----
OBP 4.30.4.d 2011/07/06 14:29

IO ASIC revisions:
-----
Location          Path          Device
          Revision
-----
IOBD/IO-BRIDGE          /pci@780      SUNW,sun4v-pci      0
.
.
.

```

例 1-5 x86: 显示系统诊断信息

以下示例显示 `prtdiag -l` 命令在基于 x86 的系统上的输出。

```

$ prtdiag -l
System Configuration: ... Sun Fire X4100 M2
BIOS Configuration: American Megatrends Inc. 0ABJX104 04/09/2009
BMC Configuration: IPMI 1.5 (KCS: Keyboard Controller Style)

==== Processor Sockets =====

Version          Location Tag
-----
Dual-Core AMD Opteron(tm) Processor 2220 CPU 1
Dual-Core AMD Opteron(tm) Processor 2220 CPU 2

==== Memory Device Sockets =====

Type      Status Set Device Locator      Bank Locator
-----
unknown   empty 0 DIMM0          NODE0
unknown   empty 0 DIMM1          NODE0
DDR2      in use 0 DIMM2          NODE0
DDR2      in use 0 DIMM3          NODE0
unknown   empty 0 DIMM0          NODE1
unknown   empty 0 DIMM1          NODE1
DDR2      in use 0 DIMM2          NODE1
DDR2      in use 0 DIMM3          NODE1

==== On-Board Devices =====
LSI serial-SCSI #1
Gigabit Ethernet #1
ATI Rage XL VGA

==== Upgradeable Slots =====

ID  Status  Type          Description
-----
1   available PCI Express      PCIExp SLOT0

```

```

2  available PCI Express      PCIExp SLOT1
3  available PCI-X           PCIExp SLOT2
4  available PCI Express      PCIExp SLOT3
5  available PCI Express      PCIExp SLOT4
$

```

## 识别有关芯片多线程功能的信息

`psrinfo` 命令进行了修改，不仅可以提供有关虚拟处理器的信息，还可以提供有关物理处理器的信息。添加此增强功能是为了识别芯片多线程 (chip multithreading, CMT) 功能。`-p` 选项可报告系统中物理处理器的总数。`-t` 选项显示系统处理器及其关联的套接字、核心和 `cpu id` 的树。

使用 `psrinfo -pv` 命令可列出系统中所有的物理处理器，以及与每个物理处理器关联的虚拟处理器。`psrinfo` 命令的缺省输出仍然显示系统的虚拟处理器信息。

有关更多信息，请参见 [psrinfo\(1M\)](#) 手册页。

## 显示系统的物理处理器类型

使用 `psrinfo -p` 命令可以显示系统中物理处理器的总数。

```

$ psrinfo -p
1

```

添加 `-v` 选项，可以同时显示有关每个物理处理器关联的虚拟处理器的信息。例如：

```

$ psrinfo -pv
The physical processor has 8 cores and 32 virtual processors (0-31)
The core has 4 virtual processors (0-3)
The core has 4 virtual processors (4-7)
The core has 4 virtual processors (8-11)
The core has 4 virtual processors (12-15)
The core has 4 virtual processors (16-19)
The core has 4 virtual processors (20-23)
The core has 4 virtual processors (24-27)
The core has 4 virtual processors (28-31)
UltraSPARC-T1 (chipid 0, clock 1000 MHz)

```

以下示例显示基于 x86 的系统上 `psrinfo -pv` 命令的示例输出。

```

$ psrinfo -pv
The physical processor has 2 virtual processors (0 1)
x86 (AuthenticAMD 40F13 family 15 model 65 step 3 clock 2793 MHz)
Dual-Core AMD Opteron(tm) Processor 2220 [ Socket: F(1207) ]
The physical processor has 2 virtual processors (2 3)

```

```
x86 (AuthenticAMD 40F13 family 15 model 65 step 3 clock 2793 MHz)
Dual-Core AMD Opteron(tm) Processor 2220 [ Socket: F(1207) ]
```

## 显示系统的虚拟处理器类型

使用 `psrinfo -v` 命令可显示有关系统虚拟处理器类型的信息。

```
$ psrinfo -v
```

在基于 x86 的系统中，使用 `isalist` 命令可以显示虚拟处理器类型。例如：

```
$ isalist
amd64 pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
```

### 例 1-6 SPARC: 显示系统的虚拟处理器类型

此示例说明如何显示有关 SPARC 系统的虚拟处理器类型的信息。

```
$ psrinfo -v
Status of virtual processor 28 as of: 09/13/2010 14:07:47
  on-line since 04/08/2010 21:27:56.
  The sparcv9 processor operates at 1400 MHz,
    and has a sparcv9 floating point processor.
Status of virtual processor 29 as of: 09/13/2010 14:07:47
  on-line since 04/08/2010 21:27:56.
  The sparcv9 processor operates at 1400 MHz,
    and has a sparcv9 floating point processor.
```

### 例 1-7 SPARC: 显示与系统上的每个物理处理器关联的虚拟处理器

以下示例显示在 Oracle SPARC T4-4 服务器上使用 `-pv` 选项运行 `psrinfo` 命令时的输出。输出显示有关线程位置的芯片（物理处理器）和核心信息。在确定线程位于哪个物理 CPU 上以及如何为核心级别映射时，这些信息可能很有用。

```
$ psrinfo -pv
The physical processor has 8 cores and 64 virtual processors (0-63)
  The core has 8 virtual processors (0-7)
  The core has 8 virtual processors (8-15)
  The core has 8 virtual processors (16-23)
  The core has 8 virtual processors (24-31)
  The core has 8 virtual processors (32-39)
  The core has 8 virtual processors (40-47)
  The core has 8 virtual processors (48-55)
  The core has 8 virtual processors (56-63)
  SPARC-T4 (chipid 0, clock 2998 MHz)
The physical processor has 8 cores and 64 virtual processors (64-127)
  The core has 8 virtual processors (64-71)
  The core has 8 virtual processors (72-79)
  The core has 8 virtual processors (80-87)
```

```

The core has 8 virtual processors (88-95)
The core has 8 virtual processors (96-103)
The core has 8 virtual processors (104-111)
The core has 8 virtual processors (112-119)
The core has 8 virtual processors (120-127)
SPARC-T4 (chipid 1, clock 2998 MHz)

```

## 更改系统信息

本节介绍可用来更改常规系统信息的命令。

### 更改系统信息任务列表

任务	方向	相关说明
手动设置系统的日期和时间。	使用 <code>date mmddHHMM[[cc]yy]</code> 命令行语法手动设置系统的日期和时间。	<a href="#">如何手动设置系统的日期和时间 [20]</a>
设置每日消息。	通过编辑 <code>/etc/motd</code> 文件在系统中设置每日消息。	<a href="#">如何设置每日消息 [21]</a>
更改系统的标识。	使用 <code>hostname</code> 命令更改系统标识。	<a href="#">如何更改系统标识 [21]</a>

## ▼ 如何手动设置系统的日期和时间

1. 成为管理员。

请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“使用所指定的管理权限”。

2. 提供新日期和时间。

```
$ date mmddHHMM[[cc]yy]
```

*mm*                    月份，使用两位数

*dd*                    月份中的某日，使用两位数

*HH*                    小时，使用两位数和 24 小时制

*MM*                    分钟，使用两位数

CC                                    世纪，使用两位数

YY                                    年份，使用两位数

有关更多信息，请参见 [date\(1\)](#) 手册页。

3. 使用不带选项的 `date` 命令验证是否正确重置了系统日期。

例 1-8 手动设置系统的日期和时间

以下示例显示如何使用 `date` 命令手动设置系统的日期和时间。

```
# date
Monday, September 13. 2010 02:00:16 PM MDT
# date 0921173404
Thu Sep 17:34:34 MST 2010
```

## ▼ 如何设置每日消息

您可以编辑每日消息文件 `/etc/motd`，以包含当系统的所有用户登录时对他们的声明和查询。请尽量少使用此功能，并定期编辑此文件，以删除过时的消息。

1. 承担指定有 **Administrator Message Edit** 配置文件的角色。  
请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“使用所指定的管理权限”。
2. 使用 `pfedit` 命令来编辑 `/etc/motd` 文件，并添加您选择的消息。

```
$ pfedit /etc/motd
```

编辑文本以包括要在用户登录期间显示的消息。包括空格、制表符和回车。

3. 通过显示 `/etc/motd` 文件的内容来验证更改。

```
$ cat /etc/motd
Welcome to the UNIX universe. Have a nice day.
```

## ▼ 如何更改系统标识

1. 成为管理员。  
请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“使用所指定的管理权限”。

## 2. 为系统设置主机名称。

```
# hostname name
```

`hostname` 以及 `domainname` 命令可用于永久地设置主机名和域名。当您使用这些命令时，还将自动更新相应的 SMF 属性和关联的 SMF 服务。

有关更多信息，请参见 [hostname\(1\)](#)、[domainname\(1M\)](#) 和 [nodename\(4\)](#) 手册页。

## 管理系统进程

---

本章介绍用于管理系统进程的过程。

本章包含以下主题：

- “无需管理的系统进程” [23]
- “管理系统进程” [23]
- “显示和管理进程类信息” [32]
- “系统进程问题的故障排除” [39]

### 无需管理的系统进程

Oracle Solaris 10 和 Oracle Solaris 11 发行版包含用于执行特定任务，但不需要进行任何管理的系统进程。

进程	描述
fsflush	将页面刷新到磁盘的系统守护进程
init	启动和重新启动其他进程和 SMF 组件的初始系统进程
intrd	由于中断而监视并平衡系统负载的系统进程
kmem_task	监视内存缓存大小的系统进程
pageout	控制到磁盘的内存分页的系统进程
sched	负责 OS 调度和进程交换的系统进程
vm_tasks	用于在各个 CPU 间平衡和分配与虚拟内存相关的工作负荷以提高性能的系统进程，该进程针对每个处理器有一个线程。
zpool- <i>pool-name</i>	针对每个 ZFS 存储池的系统进程，包含关联池的 I/O 任务线程

### 管理系统进程

本节介绍了管理系统进程的各种任务。

## 管理系统进程任务列表

任务	描述	相关说明
列出进程。	使用 <code>ps</code> 命令列出系统中的所有进程。	<a href="#">如何列出进程 [27]</a>
显示有关进程的信息。	使用 <code>pgrep</code> 命令获取要显示其详细信息的进程的进程 ID。	<a href="#">如何显示有关进程的信息 [28]</a>
控制进程。	使用 <code>pgrep</code> 命令找到进程。然后，使用相应的 <code>pcommand (/proc)</code> 来控制进程。有关 <code>(/proc)</code> 命令的说明，请参见表 2-2 “进程命令 ( <code>/proc</code> )”。	<a href="#">如何控制进程 [29]</a>
中止进程。	通过进程名称或进程 ID 找到进程。然后可以使用 <code>pkill</code> 或 <code>kill</code> 命令终止进程。	<a href="#">如何终止进程 (pkill) [29]</a> <a href="#">如何终止进程 (kill) [30]</a>

## 用于管理系统进程的命令

下表介绍了用于管理系统进程的命令。

表 2-1 用于管理进程的命令

命令	描述	手册页
<code>ps</code> 、 <code>pgrep</code> 、 <code>prstat</code> 、 <code>pkill</code>	检查系统中活动进程的状态，并显示有关这些进程的详细信息。	<a href="#">ps(1)</a> 、 <a href="#">pgrep(1)</a> 和 <a href="#">prstat(1M)</a>
<code>pkill</code>	功能与 <code>pgrep</code> 相同，但通过名称或其他属性来查找进程或向进程发送信号，然后终止进程。像 <code>kill</code> 命令一样向每个匹配的进程发送信号，而不列显进程 ID。	<a href="#">pgrep(1)</a> 和 <a href="#">pkill(1)</a> <a href="#">kill(1)</a>
<code>pargs</code> 、 <code>preap</code>	协助进行进程调试。	<a href="#">pargs(1)</a> 和 <a href="#">preap(1)</a>
<code>dispadmin</code>	列出缺省进程调度策略。	<a href="#">dispadmin(1M)</a>
<code>priocntl</code>	为优先级类指定进程并管理进程优先级。	<a href="#">priocntl(1)</a>
<code>nice</code>	更改分时进程的优先级。	<a href="#">nice(1)</a>
<code>psrset</code>	将特定进程组绑定到一组处理器而非一个处理器。	<a href="#">psrset(1M)</a>

## 使用 `ps` 命令

使用 `ps` 命令可以检查系统中活动进程的状态，并可显示有关进程的技术信息。此数据对于管理任务（例如，确定设置进程优先级的方式）很有用。

根据您所使用的选项，ps 命令会报告以下信息：

- 进程的当前状态
- 进程 ID
- 父进程 ID
- 用户 ID
- 调度类
- 优先级
- 进程的地址
- 已用内存
- 已用 CPU 时间

以下列表介绍了 ps 命令报告的一些字段。具体显示哪些字段取决于您选择的选项。有关所有可用选项的说明，请参见 [ps\(1\)](#) 手册页。

UID	进程所有者的有效用户 ID。
PID	进程 ID。
PPID	父进程 ID。
C	用于调度的处理器使用率。使用 -c 选项时，将不显示此字段。
CLS	进程所属的调度类，例如实时、系统或分时。只有 -c 选项包括此字段。
PRI	内核线程的调度优先级。数值越大，表示优先级越高。
NI	进程的 nice 数值，该数值对其调度优先级有影响。进程的 nice 值越高，意味着其优先级越低。
ADDR	proc 结构的地址。
SZ	进程的虚拟地址大小。
WCHAN	进程休眠的事件或锁定的地址。
STIME	以小时、分钟和秒表示的进程开始时间。
TTY	从中启动进程或其父级的终端。问号表示没有控制终端。
TIME	进程自开始以来使用的 CPU 时间总量。
CMD	生成进程的命令。

## 使用 /proc 文件系统和命令

使用进程命令可以显示有关 /proc 目录中所列进程的详细信息。下表列出了 /proc 进程命令。/proc 目录也称为进程文件系统 (Process File System, PROCFS)。活动进程的映像按进程 ID 号存储在 PROCFS 中。

表 2-2 进程命令 (/proc)

进程命令	描述
pcred	显示进程凭证信息
pfiles	为进程中打开的文件报告 fstat 和 fcntl 信息
pflags	显示 /proc 跟踪标志、暂挂信号和保留信号以及其他状态信息
pldd	列出链接至进程中的动态库
pmap	显示每个进程的地址空间图
psig	列出每个进程的信号操作和处理程序
prun	启动每个进程
pstack	显示每个进程中的每个轻量级进程的十六进制符号栈跟踪
pstop	停止每个进程
ptime	使用微状态记帐记录进程的时间
ptree	显示包含该进程的进程树
pwait	在进程终止后显示状态信息
pwdx	显示进程的当前工作目录

有关更多信息，请参见 [proc\(1\)](#) 手册页。

进程工具与 ps 命令的某些选项相似，区别在于这些命令提供的输出更详细。

进程命令可执行以下操作：

- 显示有关进程的更多信息，例如 fstat 和 fcntl、工作目录以及父进程和子进程树
- 通过允许用户停止或恢复进程来提供对进程的控制

## 使用进程命令 (/proc) 来管理进程

通过使用一些进程命令，可以显示有关进程的详细技术信息或者控制活动进程。[表 2-2 “进程命令 \(/proc\)”](#) 列出了部分 /proc 命令。

如果某个进程陷入无限循环或者执行时间过长，则可能需要停止（中止）该进程。有关使用 kill 或 pkill 命令停止进程的更多信息，请参见[第 2 章 管理系统进程](#)。

/proc 文件系统是一个目录分层结构，其中包括状态信息和控制功能的附加子目录。

/proc 文件系统还提供了 xwatchpoint 功能，用于在进程地址空间中的各页上重新映射读写权限。该工具没有限制，并且是 MT-safe（多线程安全）的。

为了能使用 `xwatchpoint` 功能，调试工具已进行了修改，这意味着整个 `xwatchpoint` 进程会更快。

使用 `dbx` 调试工具设置 `xwatchpoints` 时，以下限制不再适用：

- 由于基于 SPARC 的系统注册窗口而在栈的本地变量中设置 `xwatchpoints`。
- 在多线程进程中设置 `xwatchpoints`。

有关更多信息，请参见 [proc\(4\)](#) 和 [mdb\(1\)](#) 手册页。

## ▼ 如何列出进程

- 使用 `ps` 命令可列出系统中的所有进程。

```
$ ps [-efc]
```

`ps` 只显示与您的登录会话关联的进程。

`-ef` 显示系统中正在执行的所有进程的完整信息。

`-c` 显示进程调度程序信息。

### 例 2-1 列出进程

以下示例显示 `ps` 命令在不带任何选项时的输出。

```
$ ps
  PID TTY          TIME CMD
 1664 pts/4        0:06 csh
 2081 pts/4        0:00 ps
```

以下示例显示 `ps -ef` 命令的输出。此输出显示，系统启动时最先执行的进程是 `sched`（交换程序），然后是 `init` 进程、`pageout` 等。

```
$ ps -ef
UID    PID  PPID  C   STIME TTY          TIME CMD
root    0    0    0  18:04:04 ?        0:15 sched
root    5    0    0  18:04:03 ?        0:05 zpool-rpool
root    1    0    0  18:04:05 ?        0:00 /sbin/init
root    2    0    0  18:04:05 ?        0:00 pageout
root    3    0    0  18:04:05 ?        2:52 fsflush
root    6    0    0  18:04:05 ?        0:02 vmtasks
daemon 739  1    0  19:03:58 ?        0:00 /usr/lib/nfs/nfs4cbd
root    9    1    0  18:04:06 ?        0:14 /lib/svc/bin/svc.startd
root   11    1    0  18:04:06 ?        0:45 /lib/svc/bin/svc.configd
daemon 559  1    0  18:04:49 ?        0:00 /usr/sbin/rpcbind
netcfg 47    1    0  18:04:19 ?        0:01 /lib/inet/netcfgd
dladm  44    1    0  18:04:17 ?        0:00 /sbin/dlmgmt
netadm 51    1    0  18:04:22 ?        0:01 /lib/inet/ipmgmt
root   372  338  0  18:04:43 ?        0:00 /usr/lib/hal/hald-addon-cpufreq
```

```

root    67    1    0 18:04:30 ?          0:02 /lib/inet/in.mpathd
root    141   1    0 18:04:38 ?          0:00 /usr/lib/pfexecd
netadm  89    1    0 18:04:31 ?          0:03 /lib/inet/nwamd
root    602   1    0 18:04:50 ?          0:02 /usr/lib/inet/inetd start
root    131   1    0 18:04:35 ?          0:01 /sbin/dhcpagent
daemon  119   1    0 18:04:33 ?          0:00 /lib/crypto/kcfd
root    333   1    0 18:04:41 ?          0:07 /usr/lib/hal/hald --daemon=yes
root    370  338  0 18:04:43 ?          0:00 /usr/lib/hal/hald-addon-network-discovery
root    159   1    0 18:04:39 ?          0:00 /usr/lib/sysevent/syseventd
root    236   1    0 18:04:40 ?          0:00 /usr/lib/ldoms/drd
root    535   1    0 18:04:46 ?          0:09 /usr/sbin/nscd
root    305   1    0 18:04:40 ?          0:00 /usr/lib/zones/zonestatd
root    326   1    0 18:04:41 ?          0:03 /usr/lib/devfsadm/devfsadm
root    314   1    0 18:04:40 ?          0:00 /usr/lib/dbus-daemon --system
.
.

```

## ▼ 如何显示有关进程的信息

1. 获得要显示更多信息的进程的进程 ID。

```
# pgrep process
```

进程 ID 显示在输出的第一列中。

2. 显示进程信息。

```
# /usr/bin/pcommand PID
```

*pcommand*                    您希望运行的进程命令。表 2-2 “进程命令 (/proc)” 列出并介绍了这些命令。

*PID*                         标识进程 ID。

### 例 2-2 显示有关进程的信息

以下示例显示如何使用进程命令来显示有关 cron 进程的更多信息。

```

# pgrep cron        Obtains the process ID for the cron process
4780
# pwdx 4780        Displays the current working directory for the cron process
4780: /var/spool/cron/atjobs
# ptree 4780       Displays the process tree that contains the cron process
4780 /usr/sbin/cron
# pfiles 4780      Displays fstat and fcntl information
4780: /usr/sbin/cron
Current rlimit: 256 file descriptors
0: S_IFCHR mode:0666 dev:290,0 ino:6815752 uid:0 gid:3 rdev:13,2
O_RDONLY|O_LARGEFILE

```

```

/devices/pseudo/mm@0:null
1: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
   O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
   /var/cron/log
2: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
   O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
   /var/cron/log
3: S_IFIFO mode:0600 dev:32,128 ino:42049 uid:0 gid:0 size:0
   O_RDWR|O_LARGEFILE
   /etc/cron.d/FIFO
4: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
   O_RDWR|O_NONBLOCK
5: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
   O_RDWR

```

## ▼ 如何控制进程

1. 获得要控制的进程的进程 ID。

```
# pgrep process
```

进程 ID 显示在输出的第一列中。

2. 使用相应的进程命令来控制进程。

```
# /usr/bin/pcommand PID
```

*pcommand*            您希望运行的进程命令。表 2-2 “进程命令 (/proc)” 列出并介绍了这些命令。

*PID*                 标识进程 ID。

3. 验证进程状态。

```
# ps -ef | grep PID
```

## 终止进程 (pkill、kill)

有时，您可能需要停止（中止）无限循环进程，或在完成前停止大型作业。您可以中止属于您的任何进程。系统管理员可以中止系统中的任何进程，但进程 ID 为 0、1、2、3 和 4 的进程除外。中止这些进程很可能会使系统崩溃。

有关更多信息，请参见 [pgrep\(1\)](#)、[pkill\(1\)](#) 以及 [kill\(1\)](#) 手册页。

## ▼ 如何终止进程 (pkill)

1. 要终止其他用户的进程，请承担 root 角色。

2. 获得要终止的进程的进程 ID。

```
$ pgrep process
```

例如：

```
$ pgrep netscape
587
566
```

在输出中显示进程 ID。

---

注 - 要获取有关 Sun Ray™ 系统的进程的信息，请使用以下命令：

列出所有用户进程：

```
# ps -fu user
```

定位某位用户拥有的特定进程：

```
# ps -fu user | grep process
```

---

3. 终止进程。

```
$ pkill [signal] PID
```

*signal*                    当 `pkill` 命令行语法中不包含任何信号时，使用的缺省信号为 `-15` (`SIGKILL`)。将 `-9` 信号 (`SIGTERM`) 与 `pkill` 命令一起使用，可以确保进程快速终止。但是，不应使用 `-9` 信号来中止特定进程，例如数据库进程或 LDAP 服务器进程，因为数据可能已丢失。

*PID*                      要停止的进程的名称。

---

提示 - 使用 `pkill` 命令终止进程时，先尝试使用该命令本身，而不包括信号选项。如果几分钟后进程仍然没有终止，请使用带有 `-9` 信号的 `pkill` 命令。

---

4. 验证进程是否已终止。

```
$ pgrep process
```

`pgrep` 命令的输出中应不再列出已终止的进程。

## ▼ 如何终止进程 (kill)

1. 要终止其他用户的进程，请承担 `root` 角色。
2. 获得要终止的进程的进程 ID。

```
# ps -fu user
```

其中 *user* 是进程的所有者。  
进程 ID 显示在输出的第一列中。

### 3. 终止进程。

```
# kill [signal-number] PID
```

*signal*                    当 kill 命令行语法中不包括任何信号时，使用的缺省信号为 -15 (SIGKILL)。将 -9 信号 (SIGTERM) 与 kill 命令一起使用，可以确保进程快速终止。但是，不应使用 -9 信号来中止特定进程，例如数据库进程或 LDAP 服务器进程，因为数据可能已丢失。

*PID*                        是要终止的进程的进程 ID。

---

提示 - 使用 kill 命令停止进程时，先尝试使用该命令本身，而不包括信号选项。等待几分钟以确定进程是否终止，然后再使用带有 -9 信号的 kill 命令。

---

### 4. 验证进程是否已终止。

```
$ ps
```

ps 命令的输出中应不再列出已终止的进程。

## 调试进程 (pargs、preap)

pargs 命令和 preap 命令可以改进进程调试。pargs 命令可以列显与实时进程或核心文件关联的参数和环境变量。preap 命令可以删除不再存在 (僵停) 的进程。僵停进程尚不具有由父进程请求的退出状态。这些进程通常是无害的，但如果数量众多，则会占用系统资源。可以使用 pargs 和 preap 命令检查您有权检查的任何进程。如果您成为管理员，可以检查任何进程。

有关使用 preap 命令的信息，请参见 [preap\(1\)](#) 手册页。有关使用 pargs 命令的信息，请参见 [pargs\(1\)](#) 手册页。另请参见 [proc\(1\)](#) 手册页。

#### 例 2-3                    调试进程 (pargs)

pargs 命令可以解决一个长期存在的问题，即使用 ps 命令无法显示传递给进程的所有参数。以下示例显示如何通过结合使用 pargs 命令和 pgrep 命令来显示传递给进程的参数。

```
# pargs `pgrep ttymon`
579: /usr/lib/saf/ttymon -g -h -p system-name console login:
-T sun -d /dev/console -l
```

```
argv[0]: /usr/lib/saf/ttymon
argv[1]: -g
argv[2]: -h
argv[3]: -p
argv[4]: system-name console login:
argv[5]: -T
argv[6]: sun
argv[7]: -d
argv[8]: /dev/console
argv[9]: -l
argv[10]: console
argv[11]: -m
argv[12]: ldterm,ttcompat
548: /usr/lib/saf/ttymon
argv[0]: /usr/lib/saf/ttymon
```

以下示例显示如何使用 `pargs -e` 命令来显示与某一进程关联的环境变量。

```
$ pargs -e 6763
6763: tcsh
envp[0]: DISPLAY=:0.0
```

## 显示和管理进程类信息

您可以在您的系统上配置进程调度类以及分时类的用户优先级范围。

可能的进程调度类如下所示：

- 公平份额 (FSS)
- 固定 (FX)
- 系统 (SYS)
- 交互 (IA)
- 实时 (RT)
- 分时 (TS)
  - 用户提供的优先级范围为 -60 到 +60。
  - 进程的优先级是从父进程继承的。此优先级称为用户模式优先级。
  - 系统会在分时分发参数表中查找用户模式优先级。然后，系统将添加任何 `nice` 或 `prionctl` (用户提供的) 优先级并确保范围在 0–59 之间，以创建全局优先级。

## 显示进程类信息

本节包含以下主题：

[“显示进程优先级信息” \[33\]](#) 使用 `prionctl -l` 命令可以显示进程调度类和优先级范围。

“显示进程的全局优先级” [33] 使用 `ps -ecl` 命令可以显示进程的全局优先级。

## 显示进程优先级信息

使用 `prctl -l` 命令可显示进程调度类和优先级范围。

```
$ prctl -l
```

以下示例显示 `prctl -l` 命令的输出。

```
# prctl -l
CONFIGURED CLASSES
=====

SYS (System Class)

TS (Time Sharing)
    Configured TS User Priority Range: -60 through 60

FX (Fixed priority)
    Configured FX User Priority Range: 0 through 60

IA (Interactive)
    Configured IA User Priority Range: -60 through 60
```

## 显示进程的全局优先级

使用 `ps` 命令可显示进程的全局优先级。

```
$ ps -ecl
```

PRI 列下列出了全局优先级。

以下示例显示 `ps -ecl` 命令的输出。PRI 列中的值显示每个进程的优先级。

```
$ ps -ecl
 F S   UID   PID  PPID  CLS PRI   ADDR   SZ   WCHAN TTY          TIME CMD
 1 T    0     0    0  SYS  96    ?     0     ?     ?     0:11 sched
 1 S    0     5    0  SDC  99    ?     0     ? ?    ?     0:01 zpool-rp
 0 S    0     1    0   TS  59    ?    688    ? ?    ?     0:00 init
 1 S    0     2    0  SYS  98    ?     0     ? ?    ?     0:00 pageout
 1 S    0     3    0  SYS  60    ?     0     ? ?    ?     2:31 fsflush
 1 S    0     6    0  SDC  99    ?     0     ? ?    ?     0:00 vmtasks
 0 S   16    56    1   TS  59    ?   1026    ? ?    ?     0:01 ipmgmt
 0 S    0     9    1   TS  59    ?   3480    ? ?    ?     0:04 svc.star
 0 S    0    11    1   TS  59    ?   3480    ? ?    ?     0:13 svc.conf
 0 S    0    162   1   TS  59    ?    533    ? ?    ?     0:00 pfexecd
 0 S    0   1738  1730  TS  59    ?    817    ? pts/ 1    0:00 bash
 0 S    1    852    1   TS  59    ?    851    ? ?    ?     0:17 rpcbind
 0 S   17     43    1   TS  59    ?   1096    ? ?    ?     0:01 netcfgd
```

0 S	15	47	1	TS	59	?	765	??	0:00	dlmgmtd
0 S	0	68	1	TS	59	?	694	??	0:01	in.mpath
0 S	1	1220	1	FX	60	?	682	??	0:00	nfs4cbd
0 S	16	89	1	TS	59	?	1673	??	0:02	nwamd
0 S	0	146	1	TS	59	?	629	??	0:01	dhcpagen
0 S	1	129	1	TS	59	?	1843	??	0:00	kcfd
0 S	1	1215	1	FX	60	?	738	??	0:00	lockd
0 S	0	829	828	TS	59	?	968	??	0:00	hald-run
0 S	0	361	1	TS	59	?	1081	??	0:01	devfsadm
0 S	0	879	1	TS	59	?	1166	??	0:01	inetd
0 0	119764	1773	880	TS	59	?	557	cons ole	0:00	ps
0 S	0	844	829	TS	59	?	996	??	0:00	hald-add
0 S	0	895	866	TS	59	?	590	??	0:00	ttymon
0 S	0	840	1	TS	59	?	495	??	0:00	cron
0 S	0	874	1	TS	59	?	425	??	0:00	utmpd
0 S	0	1724	956	TS	59	?	2215	??	0:00	sshd
0 S	119764	880	9	TS	59	?	565	? cons ole	0:00	csch
0 S	0	210	1	TS	59	?	1622	??	0:00	sysevent
0 S	0	279	1	TS	59	?	472	??	0:00	iscsid
0 S	1	1221	1	TS	59	?	1349	??	0:00	nfsmapid
1 S	0	374	0	SDC	99	?	0	??	0:00	zpool-us
0 S	0	1207	1	TS	59	?	1063	??	0:00	rmvolmgr
0 S	0	828	1	TS	59	?	1776	??	0:03	hald
0 S	0	853	829	TS	59	?	896	??	0:02	hald-add
0 S	0	373	1	TS	59	?	985	??	0:00	picld
0 S	0	299	1	TS	59	?	836	??	0:00	dbus-dae
0 S	12524	1730	1725	TS	59	?	452	? pts/ 1	0:00	csch
0 S	0	370	1	TS	59	?	574	??	0:00	powerd
0 S	0	264	1	FX	60	?	637	??	0:00	zonestat
0 S	0	866	9	TS	59	?	555	??	0:00	sac
0 S	0	851	829	TS	59	?	998	??	0:00	hald-add
0 S	12524	1725	1724	TS	59	?	2732	??	0:00	sshd
0 S	1	1211	1	TS	59	?	783	??	0:00	statd
0 S	0	1046	1	TS	59	?	1770	??	0:13	intrd
0 S	0	889	1	TS	59	?	1063	??	0:00	syslogd
0 S	0	1209	1	TS	59	?	792	??	0:00	in.ndpd
0 S	0	1188	1186	TS	59	?	951	??	0:15	automoun
0 S	0	1172	829	TS	59	?	725	??	0:00	hald-add
0 S	0	1186	1	TS	59	?	692	??	0:00	automoun
0 S	101	1739	1738	TS	59	?	817	? pts/ 1	0:00	bash
0 S	0	1199	1	TS	59	?	1495	??	0:02	sendmail
0 S	0	956	1	TS	59	?	1729	??	0:00	sshd
0 S	25	1192	1	TS	59	?	1528	??	0:00	sendmail
0 S	0	934	1	TS	59	?	6897	??	0:14	fmd
0 S	0	1131	1	TS	59	?	1691	??	0:07	nscd
0 S	1	1181	1	TS	59	?	699	??	0:00	ypbind

## 管理进程类信息任务列表

使用以下过程管理您的进程类。

任务	描述	相关说明
指定进程优先级。	使用 <code>prioctl -e -c</code> 命令可以指定的优先级启动进程。	<a href="#">如何指定进程优先级 (prioctl) [35]</a>
更改分时进程的调度参数。	使用 <code>prioctl -s -m</code> 命令可以更改分时进程中的调度参数。	<a href="#">如何更改分时进程的调度参数 (prioctl) [36]</a>
更改进程的类。	使用 <code>prioctl -s -c</code> 命令可以更改进程的类。	<a href="#">如何更改进程的类 (prioctl) [37]</a>
更改进程的优先级。	使用 <code>/usr/bin/nice</code> 命令及相应选项可以降低或提高进程的优先级。	<a href="#">“更改进程优先级 (nice)” [38]</a>

## 更改进程的调度优先级 (prioctl)

进程的调度优先级是进程调度程序根据调度策略指定的优先级。dispadmin 命令可以列出缺省调度策略。有关更多信息，请参见 [dispadmin\(1M\)](#) 手册页。

可以使用 `prioctl` 命令将进程指定给优先级类并管理进程优先级，如下过程所示。

### ▼ 如何指定进程优先级 (prioctl)

1. 承担 `root` 角色。  
请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。
2. 以指定的优先级启动进程。

```
# prioctl -e -c class -m user-limit -p PRI command-name
```

<code>-e</code>	执行该命令。
<code>-c class</code>	指定要在其中运行进程的类。有效类包括 TS (timesharing, 分时)、RT (real time, 实时)、IA (interactive, 交互)、FSS (fair share, 公平份额) 和 FX (fixed priority, 固定优先级)。
<code>-m user-limit</code>	将此选项与 <code>-p</code> 选项一起使用时，还可指定可将优先级提高或降低的最大量。
<code>-p PRI</code>	允许您为实时线程指定在 RT 类中的相对优先级。对于分时进程而言，使用 <code>-p</code> 选项可以指定用户提供的优先级，该优先级范围是 -60 到 +60。
<code>command-name</code>	指定将执行的命令的名称。

3. 验证进程状态。

```
# ps -ecl | grep command-name
```

例 2-4 指定进程优先级 (priosctl)

以下示例显示如何以用户提供的最高优先级启动 find 命令。

```
# priosctl -e -c TS -m 60 -p 60 find . -name core -print
# ps -ecl | grep find
```

## ▼ 如何更改分时进程的调度参数 (priosctl)

1. 承担 root 角色。

请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 更改正在运行的分时进程的调度参数。

```
# priosctl -s -m user-limit [-p user-priority] -i ID type ID list
```

-s 允许设置用户优先级范围的上限并更改当前优先级。

-m user-limit 使用 -p 选项时，可以指定可将优先级提高或降低的最大量。

-p user-priority 允许指定优先级。

-i ID type ID list 结合使用 ID type 和 ID list 可以标识一个或多个进程。ID type 指定 ID 的类型，例如进程 ID 或用户 ID。ID list 标识进程 ID 或用户 ID 的列表。

3. 验证进程状态。

```
# ps -ecl | grep ID list
```

例 2-5 更改分时进程的调度参数 (priosctl)

以下示例显示如何以 500 毫秒的时间分片、RT 类中的优先级 20 以及全局优先级 120 来执行命令。

```
# priosctl -e -c RT -m 500 -p 20 myprog
# ps -ecl | grep myprog
```

## ▼ 如何更改进程的类 (priocntl)

1. (可选) 承担 root 角色。

---

注 - 您必须承担 root 角色，或在实时 shell 中工作，才能将实时进程更改为其他进程或将某个进程更改为实时进程。如果以 root 角色将用户进程改为实时类，则用户随后将无法使用 `priocntl -s` 命令来更改实时调度参数。

请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

---

2. 更改进程的类。

```
# priocntl -s -c class -i ID type ID list
```

-s 允许设置用户优先级范围的上限并更改当前优先级。

-c class 指定进程所要改用的类，分时则指定 TS，实时则指定 RT。

-i ID type ID list 结合使用 *ID type* 和 *ID list* 可以标识一个或多个进程。*ID type* 指定 ID 的类型，例如进程 ID 或用户 ID。*ID list* 标识进程 ID 或用户 ID 的列表。

3. 验证进程状态。

```
# ps -ecl | grep ID list
```

### 例 2-6 更改进程的类 (priocntl)

以下示例显示如何将属于用户 15249 的所有进程都更改为实时进程。

```
# priocntl -s -c RT -i uid 15249
# ps -ecl | grep 15249
```

## 更改分时进程的优先级 (nice)

支持 `nice` 命令只是为了向下兼容以前的发行版。`priocntl` 命令在管理进程方面提供了更大的灵活性。

进程的优先级由其调度类的策略和 `nice` 数值决定。每个分时进程都有全局优先级。全局优先级的计算方法是用户提供的优先级（该优先级可能受 `nice` 或 `priocntl` 命令影响）加上系统计算的优先级。

进程的执行优先级数由操作系统指定。优先级数由多个因素决定，其中包括进程的调度类、进程使用的 CPU 时间以及进程的 `nice` 数值（对于分时进程的情况）。

每个分时进程都以缺省 nice 数值启动，它从父进程中继承该数值。nice 数值显示在 ps 报告的 NI 列中。

用户可通过提高用户提供的优先级来降低进程的优先级。但是，只有管理员能够通过降低 nice 数值来提高进程的优先级。此限制使用户无法提高自己所拥有进程的优先级，因此会独占更大的 CPU 份额。

nice 数值的范围是从 0 到 +39，其中 0 表示最高优先级。每个分时进程的缺省 nice 值为 20。有两个可用的命令版本：标准版本 /usr/bin/nice 以及 C shell 内置命令。

## 更改进程优先级 (nice)

以用户身份，您可以降低进程的优先级。而成为管理员可以提高或降低进程的优先级。

- 以用户身份，您可以通过增大 nice 数值来降低命令的优先级。

以下 nice 命令使 nice 数值增大 5 个单位，从而以更低的优先级执行 *command-name*。

```
$ /usr/bin/nice -5 command-name
```

在此命令中，减号指定后面的内容是一个选项。也可以通过以下方式指定此命令：

```
$ /usr/bin/nice -n 5 command-name
```

以下 nice 命令使 nice 数值增大 10 个单位（缺省增量）但未超出最大值 39，从而降低了 *command-name* 的优先级。

```
$ /usr/bin/nice command-name
```

- 作为管理员，您可以通过更改 nice 数值来提高或降低命令的优先级。

以下 nice 命令使 nice 数值降低 10 个单位，从而提高了 *command-name* 的优先级。该数值不低于最小值 0。

```
# /usr/bin/nice --10 command-name
```

在此命令中，第一个减号指定后面的内容是一个选项。第二个减号指示一个负数。

以下 nice 命令使 nice 数值增加 5 个单位，从而降低了 *command-name* 的优先级。该数值不超过最大值 39。

```
# /usr/bin/nice -5 command-name
```

有关更多信息，请参见 [nice\(1\)](#) 手册页。

## 系统进程问题的故障排除

下面是您可能遇到的一些常见的系统进程问题：

- 查找同一用户所有的多个相同作业。出现此问题可能是由于正在运行的某个脚本启动了多个后台作业，而未等待任何作业完成。
- 查找累积了大量 CPU 时间的进程。通过检查 ps 输出中的 TIME 字段可以确定此问题。此值可能表明这个进程处于无限循环。
- 查找运行优先级过高的进程。使用 ps -c 命令检查 CLS 字段，其中显示每个进程的调度类。作为实时 (RT) 进程执行的进程会独占 CPU。或者，查找 nice 数值很大的分时 (TS) 进程。管理员可能提高了某一进程的优先级。系统管理员可以使用 nice 命令来降低优先级。
- 查找所占用的 CPU 时间逐步增加的失控进程。通过查看进程启动时的时间 (STIME) 并监视一会儿 CPU 时间累积 (TIME)，便可确定此问题。



# ◆◆◆ 第 3 章

## 监视系统性能

---

从计算机或网络中获取良好性能是系统管理的重要部分。本章介绍了与管理计算机系统性能有关的一些因素。此外，还介绍了使用 `vmstat`、`iostat`、`df` 和 `sar` 命令监视系统性能的过程。

本章包含以下主题：

- “在何处查找有关监视系统性能的信息” [41]
- “关于影响系统性能的系统资源” [42]
- “关于进程和系统性能” [42]
- “关于监视系统性能” [44]
- “显示系统性能信息” [45]
- “监视系统活动” [51]

### 在何处查找有关监视系统性能的信息

系统性能任务	更多信息
管理进程	<a href="#">第 2 章 管理系统进程</a>
监视系统性能	<a href="#">第 3 章 监视系统性能</a>
更改可调参数	<a href="#">《Oracle Solaris 11.2 可调参数参考手册》</a>
管理系统性能任务	<a href="#">《在 Oracle Solaris 11.2 中进行资源管理》</a> 中的 <a href="#">第 2 章 “关于项目和任务”</a>
使用 FX 和 FS 调度程序管理进程	<a href="#">《在 Oracle Solaris 11.2 中进行资源管理》</a> 中的 <a href="#">第 8 章 “公平份额调度器”</a>

### 使用 Oracle Enterprise Manager Ops Center 管理性能

如果您需要在数据中心内监视、分析和提高物理和虚拟操作系统、服务器和存储设备的性能，而不只是监视单个系统的性能，则可以使用 Oracle Enterprise Manager Ops Center 中提供的综合系统管理解决方案。

Enterprise Manager Ops Center 中的监视功能提供了有关数据中心内受监视的操作系统和区域的详细信息。您可以使用这些信息来评估性能、确定问题并执行调优。分析功能可用于 Oracle Solaris 操作系统、Linux 以及 OS 虚拟化技术，包括 Oracle Solaris Zones、Oracle VM Server for SPARC 以及 Oracle VM Server for x86 来宾。

有关信息，请参见 <http://www.oracle.com/pls/topic/lookup?ctx=oc122> (Oracle Enterprise Manager Ops Center 12c 发行版 2 文档库)。

## 关于影响系统性能的系统资源

计算机系统的性能取决于该系统使用和分配其资源的方式。定期监视系统的性能，以便了解系统在正常情况下的行为。应当明确了解预期情况，并能够在出现问题时进行识别。

影响性能的系统资源包括：

中央处理器 (central processing unit, CPU)	CPU 通过从内存中提取并执行指令来对指令进行处理。
输入/输出 (Input/Output, I/O) 设备	I/O 设备可向计算机传入信息，并可从中传出信息。此类设备可能是终端和键盘、磁盘驱动器或打印机。
内存	物理（或主）内存是系统中的随机访问存储器 (Random Access Memory, RAM) 量。

[第 3 章 监视系统性能](#)介绍了显示系统活动和性能相关统计信息的工具。

## 关于进程和系统性能

与进程相关的部分术语包括：

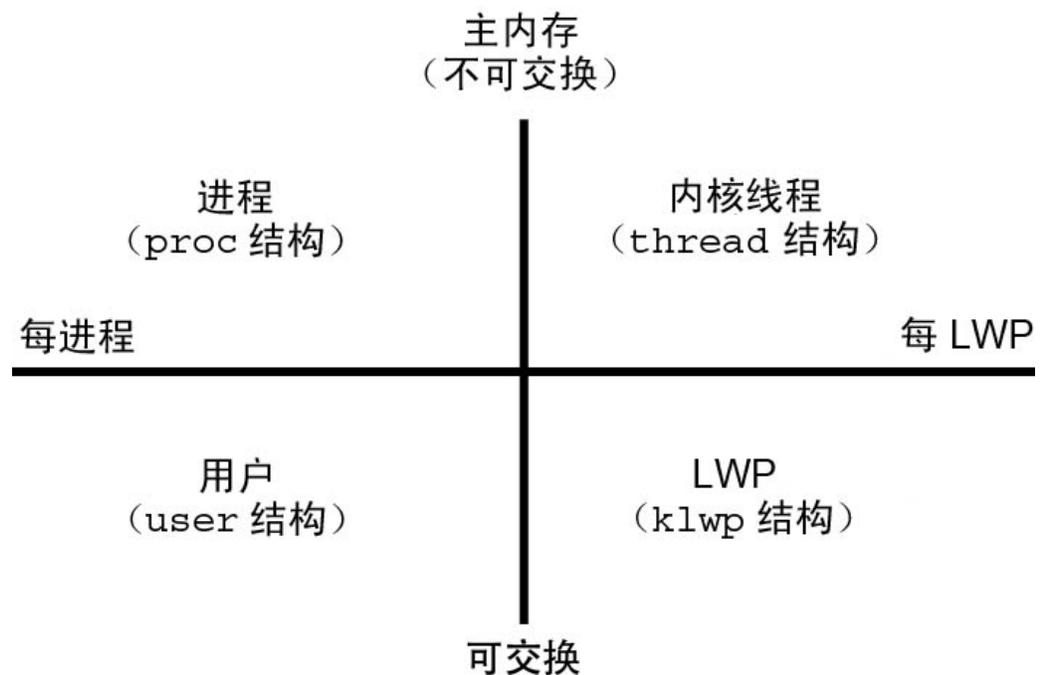
Process (进程)	任何系统活动或作业。每次引导系统、执行命令或启动应用程序时，系统都会激活一个或多个进程。
Lightweight Process, LWP (轻量级进程)	虚拟 CPU 或执行资源。LWP 由内核预定，根据其预定类和优先级来使用可用的 CPU 资源。LWP 包含可交换信息，以及包含必须始终位于内存中的信息的内核线程。
Application thread (应用程序线程)	具有单独栈的一系列指令，这些指令可在用户的地址空间中独立执行。可在 LWP 顶部复用应用程序线程。

一个进程可以由多个 LWP 和多个应用程序线程组成。内核预定内核线程结构，该结构是 Oracle Solaris 环境中的预定实体。各种进程结构如下所示：

proc	包含与整个进程有关的信息，该信息必须始终位于主内存中
kthread	包含与 LWP 有关的信息，该信息必须始终位于主内存中
user	包含可交换的“每进程”信息
klwp	包含可交换的“每 LWP 进程”信息

下图显示了这些进程结构之间的关系。

图 3-1 进程结构之间的关系



进程中的所有线程都可以访问大多数进程资源。几乎所有进程虚拟内存都是共享的。一个线程对共享数据的更改可供进程中的其他线程使用。

## 关于监视系统性能

计算机运行时，操作系统中的计数器会增加，以跟踪各种系统活动。跟踪的系统活动如下所示：

- 中央处理器 (Central Processing Unit, CPU) 使用率
- 缓冲区使用情况
- 磁盘和磁带输入/输出 (Input/Output, I/O) 活动
- 终端设备活动
- 系统调用活动
- 上下文切换
- 文件访问
- 队列活动
- 内核表
- 进程间通信
- 分页
- 可用内存和交换空间
- 内核内存分配 (Kernel Memory Allocation, KMA)

## 监视工具

Oracle Solaris 软件提供了多种工具，以帮助您跟踪系统的执行情况。

表 3-1 性能监视工具

命令	描述	更多信息
cpustat 和 cputrack 命令	使用 CPU 性能计数器监视系统或进程的性能。	<a href="#">cpustat(1M)</a> 和 <a href="#">cputrack(1)</a>
netstat 和 nfsstat 命令	显示有关网络性能的信息。	<a href="#">netstat(1M)</a> 和 <a href="#">nfsstat(1M)</a>
ps 和 prstat 命令	显示有关活动进程的信息。	<a href="#">第 2 章 管理系统进程</a>
sar 和 sadc 命令	收集并报告系统活动数据。	<a href="#">第 3 章 监视系统性能</a>
swap 命令	显示有关系统中的可用交换空间的信息。	<a href="#">《在 Oracle Solaris 11.2 中管理文件系统》</a> 中的 <a href="#">第 3 章 “配置附加交换空间”</a>
vmstat 和 iostat 命令	汇总系统活动数据，如虚拟内存统计信息、磁盘使用情况和 CPU 活动。	<a href="#">第 3 章 监视系统性能</a>
kstat 和 mpstat 命令	检查系统中可用的内核统计信息或 kstats，并报告与命令行上指定的条件匹配的统计信息。mpstat 命令以表格形式报告处理器统计信息。	<a href="#">kstat(1M)</a> 和 <a href="#">mpstat(1M)</a> 手册页。

## 显示系统性能信息

本节介绍了用于监视和显示系统性能信息的任务。

### 显示虚拟内存统计信息

可以使用 `vmstat` 命令报告虚拟内存统计信息，以及有关系统事件（例如 CPU 负载、分页、上下文切换数、设备中断和系统调用）的信息。`vmstat` 命令还可以显示有关交换、高速缓存刷新和中断的统计信息。

表 3-2 `vmstat` 命令的输出

类别	字段名	描述
procs	r	分发队列中的内核线程数
	b	正在等待资源的阻塞内核线程数
	w	正在等待处理中的资源完成的换出 LWP 数
memory		报告实际内存和虚拟内存的使用情况
	swap	可用交换空间
page	free	可用列表的大小
		以秒为单位报告缺页和分页活动
	re	回收的页面
	mf	次要错误和主要错误
	pi	页入的千字节数
	po	页出的千字节数
	fr	释放的千字节数
	de	最近换入的进程所需的预计内存
	sr	由 page 守护程序扫描的当前未使用的页数。如果 sr 不等于零，则 page 守护程序一直在运行。
	disk	报告每秒的磁盘操作数，最多显示四个磁盘中的数据
faults		报告每秒的陷阱/中断速率
	in	每秒的中断次数
	sy	每秒的系统调用数
	cs	CPU 上下文切换速率
cpu		报告 CPU 使用时间
	us	用户时间
	sy	系统时间
	id	空闲时间

有关此命令的更多详细说明，请参见 [vmstat\(1M\)](#) 手册页。

## 显示虚拟内存统计信息 (vmstat)

要显示虚拟内存统计信息，请以秒为时间间隔单位使用 `vmstat` 命令。

```
$ vmstat n
```

其中，*n* 是两次报告之间的间隔秒数。

以下示例显示了利用 `vmstat` 报告的以 5 秒为间隔收集的统计信息：

```
$ vmstat 5
kthr      memory          page        disk        faults        cpu
r  b  w    swap  free  re  mf  pi  po  fr  de  sr  dd  f0  s1  --  in  sy  cs  us  sy  id
0  0  0  863160 365680  0   3  1  0  0  0  0  0  0  0  0  406 378 209  1  0  99
0  0  0  765640 208568  0  36  0  0  0  0  0  0  0  0  0  479 4445 1378  3  3  94
0  0  0  765640 208568  0   0  0  0  0  0  0  0  0  0  0  423  214  235  0  0  100
0  0  0  765712 208640  0   0  0  0  0  0  0  0  3  0  0  0  412  158  181  0  0  100
0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  157  179  0  0  100
0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  403  153  182  0  0  100
0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  168  177  0  0  100
0  0  0  765832 208760  0   0  0  0  0  0  0  0  0  0  0  402  153  178  0  0  100
0  0  0  765832 208760  0  18  0  0  0  0  0  0  0  0  0  407  165  186  0  0  100
```

## 显示系统事件信息 (vmstat -s)

运行 `vmstat -s` 命令，以显示自上次引导系统以来发生的系统事件数。

```
$ vmstat -s
      0 swap ins
      0 swap outs
      0 pages swapped in
      0 pages swapped out
522586 total address trans. faults taken
17006 page ins
    25 page outs
23361 pages paged in
    28 pages paged out
45594 total reclaims
45592 reclaims from free list
      0 micro (hat) faults
522586 minor (as) faults
16189 major faults
 98241 copy-on-write faults
137280 zero fill page faults
 45052 pages examined by the clock daemon
      0 revolutions of the clock hand
    26 pages freed by the clock daemon
```

```

2857 forks
  78 vforks
1647 execs
34673885 cpu context switches
65943468 device interrupts
711250 traps
63957605 system calls
3523925 total name lookups (cache hits 99%)
 92590 user  cpu
 65952 system cpu
16085832 idle  cpu
 7450 wait  cpu

```

## 显示交换统计信息 (vmstat -S)

运行 `vmstat -S`，以显示交换统计信息。

```

$ vmstat -S
kthr      memory          page        disk        faults      cpu
 r  b  w  swap  free  si  so pi po fr de sr dd f0 s1 --  in  sy  cs us sy id
 0  0  0 862608 364792  0  0  1  0  0  0  0  0  0  0  406  394  213  1  0  99

```

以下列表介绍了交换统计信息字段。有关其他字段的说明，请参见表 3-2 “`vmstat` 命令的输出”。

si	每秒换入的平均 LWP 数
so	换出的完整进程数

---

注 - `vmstat` 命令会截断 `si` 和 `so` 字段的输出。应使用 `sar` 命令显示更精确的交换统计信息记录。

---

## 显示每台设备的中断次数 (vmstat -i)

运行 `vmstat -i` 命令，以显示每台设备的中断次数。

以下示例显示 `vmstat -i` 命令的输出。

```

$ vmstat -i
interrupt      total      rate
-----
clock          52163269    100
esp0           2600077     4
zsc0           25341       0
zsc1           48917       0
cgsixc0        459         0
lec0           400882      0
fdc0           14          0

```

```

bppc0          0          0
audiocs0       0          0
-----
Total          55238959      105
    
```

## 显示磁盘使用率信息

使用 `iostat` 命令可以报告有关磁盘输入和输出的统计信息，以及生成对吞吐量、使用率、队列长度、事务速率和服务时间的测量值。有关此命令的详细说明，请参阅 [iostat\(1M\)](#) 手册页。

### 显示磁盘使用率信息 (iostat)

使用 `iostat` 命令和时间间隔秒数可以显示磁盘使用率信息。

```

$ iostat 5
      tty          fd0          sd3          nfs1          nfs31          cpu
tin tout kps tps serv kps tps serv kps tps serv kps tps serv us sy wt id
  0   1   0   0  410   3   0  29   0   0   9   3   0  47   4  2  0  94
    
```

输出的第一行显示自上次引导系统以来的统计信息。后面的每行显示间隔的统计信息。缺省情况下会显示终端 (tty)、磁盘 (fd 以及 sd) 和 CPU (cpu) 的统计信息。

以下示例显示每隔 5 秒收集的磁盘统计信息。

```

$ iostat 5
      tty          sd0          sd6          nfs1          nfs49          cpu
tin tout kps tps serv kps tps serv kps tps serv kps tps serv us sy wt id
  0   0   1   0  49   0   0   0   0   0   0   0   0  15   0  0  0  100
  0  47   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16  44   6  132   0   0   0   0   0   0   0   0   0   0  0  0  1  99
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   3   1  23   0   0   0   0   0   0   0   0   0   0  0  0  1  99
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
  0  16   0   0   0   0   0   0   0   0   0   0   0   0   0  0  0  100
    
```

下表介绍了 `iostat n` 命令输出中的字段。

设备类型	字段名	描述
终端		

设备类型	字段名	描述
磁盘	tin	终端输入队列中的字符数
	tout	终端输出队列中的字符数
	bps	每秒块数
	tps	每秒事务数
	serv	平均服务时间 (毫秒)
CPU	us	在用户模式下
	sy	在系统模式下
	wt	等待 I/O
	id	空闲

## 显示扩展磁盘统计信息 (iostat -xtc)

运行 `iostat -xt` 命令，以显示扩展磁盘统计信息。

```
$ iostat -xt
device      r/s    w/s    kr/s    kw/s  wait  actv  wsvc_t  asvc_t  %w  %b  tin tout
blkdev0    0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0    0  0    0   1
sd0        0.1   19.3    1.4   92.4  0.0  0.0    0.2    1.6    0  1
sd1        0.0    0.0    0.0    0.0  0.0  0.0    0.0    0.0    0  0
nfs9       0.0    0.0    0.0    0.0  0.0  0.0    0.0    1.0    0  0
nfs10      0.0    0.0    0.0    0.0  0.0  0.0    0.0    7.6    0  0
nfs11      0.0    0.0    0.0    0.0  0.0  0.0    0.0   15.6    0  0
nfs12      0.3    0.0    1.9    0.0  0.0  0.0    0.0   30.5    0  1
```

`iostat -xt` 命令对每个磁盘显示一行输出。输出字段如下：

r/s	每秒读取次数
w/s	每秒写入次数
kr/s	每秒读取的千字节数
kw/s	每秒写入的千字节数
wait	等待服务的平均事务数 (队列长度)
actv	处于活动服务状态的平均事务数
svc_t	平均服务时间 (毫秒)
%w	队列不为空的时间百分比

%b 磁盘繁忙的时间百分比

## 显示磁盘空间统计信息 (df)

使用 `df` 命令可以显示挂载的每个磁盘中的可用磁盘空间量。`df` 报告的 *usable* 磁盘空间只反映全部容量的 90%，因为报告统计信息留出总可用空间的 10% 以上。此空间通常保持为空，以实现更好的性能。

`df` 命令实际报告的磁盘空间百分比是已用空间除以可用空间。

如果文件系统超出容量的 90%，则可使用 `cp` 命令将文件转移至具有可用空间的磁盘。或者，使用 `tar` 或 `cpio` 命令将文件转移至磁带。也可以删除文件。

有关此命令的详细说明，请参见 [df\(1M\)](#) 手册页。

## 显示磁盘空间信息 (df -k)

使用 `df -k` 命令显示磁盘空间信息（千字节）。

```
$ df -k
Filesystem          kbytes  used  avail capacity  Mounted on
/dev/dsk/c0t3d0s0   192807  40231 133296    24%    /
```

### 例 3-1 显示文件系统信息

以下示例显示 `df -k` 命令针对 SPARC 的系统的输出。

```
$ df -k
Filesystem          1024-blocks    Used  Available Capacity  Mounted on
rpool/ROOT/solaris-161 191987712    6004395  140577816    5%    /
/devices              0            0          0    0%    /devices
/dev                  0            0          0    0%    /dev
ctfs                  0            0          0    0%    /system/contract
proc                  0            0          0    0%    /proc
mnttab                0            0          0    0%    /etc/mnttab
swap                  4184236      496      4183740    1%    /system/volatile
objfs                  0            0          0    0%    /system/object
sharefs               0            0          0    0%    /etc/dfs/sharetab
/usr/lib/libc/libc_hwcapi.so.1 146582211    6004395  140577816    5%    /lib/libc.so.1
fd                    0            0          0    0%    /dev/fd
swap                  4183784      60       4183724    1%    /tmp
rpool/export          191987712     35     140577816    1%    /export
rpool/export/home    191987712     32     140577816    1%    /export/home
rpool/export/home/123 191987712   13108813  140577816    9%    /export/home/123
rpool/export/repo    191987712   11187204  140577816    8%    /export/repo
rpool/export/repo2010_11 191987712    31     140577816    1%    /export/repo2010_11
rpool                 191987712   5238974  140577816    4%    /rpool
```

```
/export/home/123      153686630   13108813   140577816   9%   /home/123
```

df -k 命令的输出字段如下所示：

1024-blocks	文件系统中可用空间的总大小
Used	已用空间量
Available	可用空间量
Capacity	已用空间量，表示为总容量的百分比
Mounted on	挂载点

例 3-2 使用不带任何选项的 df 命令显示文件系统信息

使用不带操作数或选项的 df 命令时，该命令将报告所有挂载的文件系统，如以下示例中所示。

```
$ df
/                (rpool/ROOT/solaris):100715496 blocks 100715496 files
/devices        (/devices       ):      0 blocks      0 files
/dev            (/dev           ):      0 blocks      0 files
/system/contract (ctfs           ):      0 blocks 2147483601 files
/proc          (proc           ):      0 blocks   29946 files
/etc/mnttab     (mnttab        ):      0 blocks      0 files
/system/volatile (swap          ):42257568 blocks 2276112 files
/system/object  (objfs         ):      0 blocks 2147483441 files
/etc/dfs/sharetab (sharefs       ):      0 blocks 2147483646 files
/dev/fd        (fd            ):      0 blocks      0 files
/tmp           (swap          ):42257568 blocks 2276112 files
/export        (rpool/export  ):100715496 blocks 100715496 files
/export/home   (rpool/export/home ):100715496 blocks 100715496 files
/export/home/admin (rpool/export/home/admin):100715496 blocks 100715496 files
/rpool        (rpool         ):100715496 blocks 100715496 files
/export/repo2010_11(rpool/export/repo2010_11):281155639 blocks 281155639 files
/rpool        (rpool         ):281155639 blocks 281155639 files
```

## 监视系统活动

本节介绍了监视系统活动的活动。

### 监视系统活动 (sar)

使用 sar 命令可执行以下任务：

- 组织并查看有关系统活动的数据。
- 根据特殊请求访问系统活动数据。
- 生成自动报告以测量和监视系统性能，并生成特殊请求报告以确定特定性能问题。有关如何设置要对系统运行的 `sar` 命令的信息以及相应工具的说明，请参见“[自动收集系统活动数据 \(sar\)](#)” [67]。

有关此命令的详细说明，请参见 [sar\(1\)](#) 手册页。

## 检查文件访问 (`sar -a`)

使用 `sar -a` 命令显示文件访问操作统计信息。

```
$ sar -a
```

```
SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v ...
```

```
00:00:00 iget/s namei/s dirbk/s
01:00:00      0       3       0
02:00:00      0       3       0
03:00:00      0       3       0
04:00:00      0       3       0
05:00:00      0       3       0
06:00:00      0       3       0
07:00:00      0       3       0
08:00:00      0       3       0
08:20:01      0       3       0
08:40:00      0       3       0
09:00:00      0       3       0
09:20:01      0      10       0
09:40:01      0       1       0
10:00:02      0       5       0

Average      0       4       0
```

`sar -a` 命令报告的操作系统例程如下所示：

<code>iget/s</code>	对不位于目录名称查找高速缓存 (Directory Name Look-up Cache, DNLC) 中的 inode 发出的请求数。
<code>namei/s</code>	每秒搜索的文件系统路径数。如果 <code>namei</code> 在 DNLC 中找不到目录名称，它会调用 <code>iget</code> 以获取文件或目录的 inode。因此，大多数 <code>igets</code> 都是 DNLC 遗漏的结果。
<code>dirbk/s</code>	每秒发出的目录块读取数。

这些操作系统例程的报告值越大，内核访问用户文件所用的时间就越多。时间的长短将反映程序和应用程序使用文件系统的程度。`-a` 选项有助于查看磁盘与应用程序的相关情况。

## 检查缓冲区活动 (sar -b)

使用 sar -b 命令可显示缓冲区活动统计信息。

缓冲区用于高速缓存元数据。元数据包括 inode、柱面组块和间接块。

```
$ sar -b
00:00:00 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0    100      0      0     55      0      0
```

下表介绍了 -b 选项显示的缓冲区活动。

字段名	描述
bread/s	从磁盘提交至高速缓存存储区的每秒平均读取数
lread/s	每秒从高速缓存存储区进行的平均逻辑读取数
%rcache	在高速缓存存储区中找到的逻辑读取的分数 (100 % 减去 bread/s 与 lread/s 之比)
bwrit/s	平均每秒从高速缓存存储区写入磁盘的物理块 (512 字节) 数
lwrit/s	每秒平均对高速缓存存储区进行的逻辑写入数
%wcache	在高速缓存存储区中找到的逻辑写入的分数 (100 % 减去 bwrit/s 与 lwrit/s 之比)
pread/s	每秒平均使用字符设备接口的物理读取数
pwrit/s	每秒平均使用字符设备接口的物理写入请求数

最重要的项是高速缓存命中率 %rcache 和 %wcache。这两项用于度量系统缓冲的有效性。如果 %rcache 低于 90% 或者 %wcache 低于 65%，则可通过增加缓冲区空间来改善性能。

### 例 3-3 检查缓冲区活动 (sar -b)

下列 sar -b 命令输出示例显示 %rcache 和 %wcache 缓冲区未引起速率下降。所有数据都在可接受的限制范围内。

```
$ sar -b
SunOS t2k-brm-24 5.10 Generic_144500-10 sun4v ...
00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0    100      0      0     94      0      0
02:00:01      0      0    100      0      0     94      0      0
03:00:00      0      0    100      0      0     92      0      0
04:00:00      0      1    100      0      1     94      0      0
05:00:00      0      0    100      0      0     93      0      0
06:00:00      0      0    100      0      0     93      0      0
07:00:00      0      0    100      0      0     93      0      0
```

08:00:00	0	0	100	0	0	93	0	0
08:20:00	0	1	100	0	1	94	0	0
08:40:01	0	1	100	0	1	93	0	0
09:00:00	0	1	100	0	1	93	0	0
09:20:00	0	1	100	0	1	93	0	0
09:40:00	0	2	100	0	1	89	0	0
10:00:00	0	9	100	0	5	92	0	0
10:20:00	0	0	100	0	0	68	0	0
10:40:00	0	1	98	0	1	70	0	0
11:00:00	0	1	100	0	1	75	0	0
Average	0	1	100	0	1	91	0	0

## 检查系统调用统计信息 (sar -c)

使用 sar -c 命令可显示系统调用统计信息。

```
$ sar -c
00:00:00 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
01:00:00 38 2 2 0.00 0.00 149 120
```

以下列表介绍了 -c 选项报告的系统调用类别。通常，读取和写入占系统调用总数的一半。但是，该百分比会因系统所执行的活动而产生极大的变化。

scall/s	每秒中所有类型的系统调用数，在具有 4 到 6 位用户的系统中，通常每秒大约有 30 个系统调用。
sread/s	每秒的 read 系统调用数。
swrit/s	每秒的 write 系统调用数。
fork/s	每秒的 fork 系统调用数，在具有 4 到 6 位用户的系统中，每秒中大约有 0.5 个该系统调用。如果正在运行 Shell 脚本，此数字会增加。
exec/s	每秒的 exec 系统调用数。如果 exec/s 除以 fork/s 的结果大于 3，请确定是否存在无效的 PATH 变量。
rchar/s	每秒由 read 系统调用传送的字符数（字节）。
wchar/s	每秒由 write 系统调用传送的字符数（字节）。

### 例 3-4 检查系统调用统计信息 (sar -c)

以下示例显示 sar -c 命令的输出。

```
$ sar -c
```

```

SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 scall/s sread/s swrit/s fork/s exec/s rchar/s wchar/s
01:00:00      89      14       9   0.01   0.00   2906   2394
02:00:01      89      14       9   0.01   0.00   2905   2393
03:00:00      89      14       9   0.01   0.00   2908   2393
04:00:00      90      14       9   0.01   0.00   2912   2393
05:00:00      89      14       9   0.01   0.00   2905   2393
06:00:00      89      14       9   0.01   0.00   2905   2393
07:00:00      89      14       9   0.01   0.00   2905   2393
08:00:00      89      14       9   0.01   0.00   2906   2393
08:20:00      90      14       9   0.01   0.01   2914   2395
08:40:01      90      14       9   0.01   0.00   2914   2396
09:00:00      90      14       9   0.01   0.01   2915   2396
09:20:00      90      14       9   0.01   0.01   2915   2396
09:40:00     880     207     156   0.08   0.08  26671   9290
10:00:00    2020     530     322   0.14   0.13  57675  36393
10:20:00     853     129      75   0.02   0.01  10500   8594
10:40:00    2061     524     450   0.08   0.08  579217  567072
11:00:00    1658     404     350   0.07   0.06 1152916 1144203

Average      302      66      49   0.02   0.01  57842  55544
    
```

## 检查磁盘活动 (sar -d)

使用 sar -d 命令可显示磁盘活动统计信息。

```

$ sar -d

00:00:00 device          %busy  avque  r+w/s  blks/s  await  avserv
    
```

以下列表介绍了 -d 选项报告的磁盘设备活动。

device	监视的磁盘设备的名称。
%busy	设备忙于为传送请求提供服务的时间份额。
avque	设备忙于为传送请求提供服务期间的平均请求数。
r+w/s	每秒对设备进行的读取和写入传送数。
blks/s	每秒传送给设备的 512 字节块的数量。
await	传送请求在队列中等待的平均时间（毫秒）。仅当队列被占用时才测量此时间。
avserv	设备完成传送请求所需的平均时间（毫秒）。对于磁盘而言，此值包括查找时间、旋转延迟时间和数据传送时间。

例 3-5 检查磁盘活动

此缩写示例演示了 `sar -d` 命令的输出。

```
$ sar -d

SunOS balmy 5.10 Generic_144500-10 sun4v ...

12:36:32 device      %busy  avque  r+w/s  blks/s  await  avserv
12:40:01 dad1        15     0.7    26     399    18.1   10.0
          dad1,a      15     0.7    26     398    18.1   10.0
          dad1,b      0     0.0    0       1     1.0    3.0
          dad1,c      0     0.0    0       0     0.0    0.0
          dad1,h      0     0.0    0       0     0.0    6.0
          fd0         0     0.0    0       0     0.0    0.0
          nfs1        0     0.0    0       0     0.0    0.0
          nfs2        1     0.0    1      12     0.0   13.2
          nfs3        0     0.0    0       2     0.0    1.9
          nfs4        0     0.0    0       0     0.0    7.0
          nfs5        0     0.0    0       0     0.0   57.1
          nfs6        1     0.0    6     125    4.3    3.2
          nfs7        0     0.0    0       0     0.0    6.0
          sd1         0     0.0    0       0     0.0    5.4
          ohci0,bu    0     0.0    0       0     0.0    0.0
          ohci0,ct    0     0.0    0       0     0.0    0.0
          ohci0,in    0     0.0    7       0     0.0    0.0
          ohci0,is    0     0.0    0       0     0.0    0.0
          ohci0,to    0     0.0    7       0     0.0    0.0
```

请注意，在队列不为空时测量队列长度和等待时间。当 `%busy` 很小时，如果队列和服务时间很大，则可能表示系统进行周期性的努力，以确保将警报块快速写入磁盘。

### 检查页出和内存 (sar -g)

使用 `sar -g` 命令可显示平均页出和内存释放活动。

```
$ sar -g
00:00:00 pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00 0.00 0.00 0.00 0.00 0.00
```

`sar -g` 命令的输出可以明确指示是否需要更多内存。使用 `ps -elf` 命令显示 `page` 守护程序使用的周期数。如果周期数很大，并且 `pgfree/s` 和 `pgscan/s` 字段的值也很大，则表明内存不足。

`sar -g` 命令还可表明是否回收 `inode` 的速率过快而引起可重用页丢失。

以下列表介绍了 `-g` 选项的输出。

`pgout/s`                    每秒的页出请求数。

ppgout/s	每秒调出的页的实际数量。单个页出请求可能涉及多个页的调出。
pgfree/s	每秒放置在可用列表中的页数。
pgscan/s	page 守护程序每秒扫描的页数。如果此值很大，则表明 page 守护程序花费大量时间来检查可用内存。此情况暗示，可能需要更多内存。
%ufs_ipf	具有关联的可重用页的 iget 从可用列表中取消的 ufs inode 的百分比。这些页面被刷新，并且不能由进程回收。因此，此字段表示具有页面刷新的 igets 的百分比。如果该值很大，则表明 inode 的可用列表页面密集，并且可能需要增加 ufs inode 的数量。

例 3-6 检查页出和内存 (sar -g)

以下示例显示 sar -g 命令的输出。

```
$ sar -g
SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:00 ppgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00 0.00 0.00 0.00 0.00 0.00
02:00:00 0.01 0.01 0.01 0.00 0.00
03:00:00 0.00 0.00 0.00 0.00 0.00
04:00:00 0.00 0.00 0.00 0.00 0.00
05:00:00 0.00 0.00 0.00 0.00 0.00
06:00:00 0.00 0.00 0.00 0.00 0.00
07:00:00 0.00 0.00 0.00 0.00 0.00
08:00:00 0.00 0.00 0.00 0.00 0.00
08:20:01 0.00 0.00 0.00 0.00 0.00
08:40:00 0.00 0.00 0.00 0.00 0.00
09:00:00 0.00 0.00 0.00 0.00 0.00
09:20:01 0.05 0.52 1.62 10.16 0.00
09:40:01 0.03 0.44 1.47 4.77 0.00
10:00:02 0.13 2.00 4.38 12.28 0.00
10:20:03 0.37 4.68 12.26 33.80 0.00
Average 0.02 0.25 0.64 1.97 0.00
```

## 检查内核内存分配

内核内存分配 (Kernel Memory Allocation, KMA) 允许内核子系统根据需要分配和释放内存。

KMA 并不是静态分配在峰值载荷下可能需要的最大内存量，而是将内存请求划分为三个类别：

- 小型 (少于 256 字节)

- 大型 (512 字节至 4 千字节)
- 超大型 (大于 4 千字节)

KMA 保留两个内存池，以满足小型和大型请求。超大型请求则通过从系统页面分配器中分配内存来满足。

如果您所检查的系统用来编写使用 KMA 资源的驱动程序或 STREAMS，则 `sar -k` 命令可能很有用。使用 KMA 资源但不一定在退出前返回资源的所有驱动程序或模块都可能产生内存泄漏。内存泄漏会导致 KMA 分配的内存量随事件而增加。因此，如果 `sar -k` 命令的 `alloc` 字段随时间稳定增加，则可能存在内存泄漏。表明存在内存泄漏的另一种情况是请求失败。如果出现此问题，内存泄漏很可能导致 KMA 无法保留和分配内存。

如果似乎存在内存泄漏，则应检查可能从 KMA 请求内存但未返回内存的所有驱动程序或 STREAMS。

### 检查内核内存分配 (`sar -k`)

使用 `sar -k` 命令可报告内核内存分配器 (Kernel Memory Allocator, KMA) 的活动。

```
$ sar -k
00:00:00 sml_mem  alloc  fail  lg_mem  alloc  fail  ovsz_alloc  fail
01:00:00 2523136 1866512    0 18939904 14762364    0    360448    0
02:00:02 2523136 1861724    0 18939904 14778748    0    360448    0
```

以下列表介绍了 `-k` 选项的输出。

<code>sml_mem</code>	KMA 在小型内存请求池中可用的内存量 (字节)。在此池中，小型请求小于 256 字节。
<code>alloc</code>	KMA 已从其小型内存请求池向小型内存请求分配的内存量 (字节)。
<code>fail</code>	请求少量内存并失败的请求数。
<code>lg_mem</code>	KMA 在大型内存请求池中可用的内存量 (字节)。在此池中，大型请求介于 512 字节到 4 千字节之间。
<code>alloc</code>	KMA 已从其大型内存请求池向大型内存请求分配的内存量 (字节)。
<code>fail</code>	请求大量内存并失败的请求数。
<code>ovsz_alloc</code>	为大于 4 千字节的超大型请求分配的内存量。这些请求可通过页面分配器来满足。因此，不存在池。
<code>fail</code>	因请求超大量内存而失败的请求数。

例 3-7 检查内核内存分配 (sar -k)

以下示例显示 sar -k 输出的缩写示例。

```
$ sar -k

SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:04 sml_mem alloc fail lg_mem alloc fail ovsz_alloc fail
01:00:00 6119744 4852865 0 60243968 54334808 156 9666560 0
02:00:01 6119744 4853057 0 60243968 54336088 156 9666560 0
03:00:00 6119744 4853297 0 60243968 54335760 156 9666560 0
04:00:00 6119744 4857673 0 60252160 54375280 156 9666560 0
05:00:00 6119744 4858097 0 60252160 54376240 156 9666560 0
06:00:00 6119744 4858289 0 60252160 54375608 156 9666560 0
07:00:00 6119744 4858793 0 60252160 54442424 156 9666560 0
08:00:00 6119744 4858985 0 60252160 54474552 156 9666560 0
08:20:00 6119744 4858169 0 60252160 54377400 156 9666560 0
08:40:01 6119744 4857345 0 60252160 54376880 156 9666560 0
09:00:00 6119744 4859433 0 60252160 54539752 156 9666560 0
09:20:00 6119744 4858633 0 60252160 54410920 156 9666560 0
09:40:00 6127936 5262064 0 60530688 55619816 156 9666560 0
10:00:00 6545728 5823137 0 62996480 58391136 156 9666560 0
10:20:00 6545728 5758997 0 62996480 57907400 156 9666560 0
10:40:00 6734144 6035759 0 64389120 59743064 156 10493952 0
11:00:00 6996288 6394872 0 65437696 60935936 156 10493952 0

Average 6258044 5150556 0 61138340 55609004 156 9763900 0
```

### 检查进程间通信 (sar -m)

使用 sar -m 命令可以报告进程间通信活动。

```
$ sar -m
00:00:00 msg/s sema/s
01:00:00 0.00 0.00
```

除非运行使用消息或信号的应用程序，否则这些数字通常都为零 (0.00)。

-m 选项的输出如下所示：

msg/s                   每秒的消息操作（发送和接收）数  
sema/s                   每秒的信号操作数

以下缩写示例显示 sar -m 命令的输出。

```
$ sar -m

SunOS balmy 5.10 Generic_144500-10 sun4v ...
00:00:00 msg/s sema/s
```

```

01:00:00  0.00  0.00
02:00:02  0.00  0.00
03:00:00  0.00  0.00
04:00:00  0.00  0.00
05:00:01  0.00  0.00
06:00:00  0.00  0.00

Average    0.00  0.00
    
```

## 检查页入活动 (sar -p)

使用 sar -p 命令可报告页入活动，其中包括保护和转换错误。

```

$ sar -p
00:00:00  atch/s  pgin/s  ppgin/s  pflt/s  vflt/s  slock/s
01:00:00   0.07   0.00   0.00   0.21   0.39   0.00
    
```

以下列表介绍了通过 -p 选项报告的统计信息。

atch/s	每秒通过回收当前在内存中的页来满足的缺页数（每秒附加数）。例如从可用列表中回收无效的页，以及共享其他进程当前正在使用的文本页。例如，两个或多个进程同时访问同一程序文本。
pgin/s	文件系统每秒接收页入请求的次数。
ppgin/s	每秒调进的页数。单个页入请求（例如软件锁定请求，请参见 slock/s）或块大小很大时可能涉及多个页的调进。
pflt/s	因保护错误引起的缺页数。保护错误实例表明非法访问页面和写复制。通常，此数目主要包含写复制。
vflt/s	每秒的地址转换缺页数。这些错误称为有效性错误，当给定虚拟地址的有效进程表项不存在时，会发生有效性错误。
slock/s	每秒内由要求物理 I/O 的软件锁定请求引起的错误数。从磁盘向内存传送数据时，就会出现软件锁定请求。系统锁定了要接收数据的页，因此其他进程无法请求和使用该页。

### 例 3-8 检查页入活动 (sar -p)

以下示例显示 sar -p 命令的输出。

```

$ sar -p

SunOS balmy 5.10 Generic_144500-10 sun4v      ...

00:00:04  atch/s  pgin/s  ppgin/s  pflt/s  vflt/s  slock/s
01:00:00   0.09   0.00   0.00   0.78   2.02   0.00
    
```

```

02:00:01  0.08  0.00  0.00  0.78  2.02  0.00
03:00:00  0.09  0.00  0.00  0.81  2.07  0.00
04:00:00  0.11  0.01  0.01  0.86  2.18  0.00
05:00:00  0.08  0.00  0.00  0.78  2.02  0.00
06:00:00  0.09  0.00  0.00  0.78  2.02  0.00
07:00:00  0.08  0.00  0.00  0.78  2.02  0.00
08:00:00  0.09  0.00  0.00  0.78  2.02  0.00
08:20:00  0.11  0.00  0.00  0.87  2.24  0.00
08:40:01  0.13  0.00  0.00  0.90  2.29  0.00
09:00:00  0.11  0.00  0.00  0.88  2.24  0.00
09:20:00  0.10  0.00  0.00  0.88  2.24  0.00
09:40:00  2.91  1.80  2.38  4.61  17.62  0.00
10:00:00  2.74  2.03  3.08  8.17  21.76  0.00
10:20:00  0.16  0.04  0.04  1.92  2.96  0.00
10:40:00  2.10  2.50  3.42  6.62  16.51  0.00
11:00:00  3.36  0.87  1.35  3.92  15.12  0.00

Average  0.42  0.22  0.31  1.45  4.00  0.00

```

## 检查队列活动 (sar -q)

使用 sar -q 命令可报告以下信息：

- 队列被占用时的平均队列长度。
- 队列处于占用状态的时间百分比。

```

$ sar -q
00:00:00 runq-sz %runocc swpq-sz %swpocc

```

-q 选项的输出如下所示。

runq-sz	内存中等待 CPU 以便运行的内核线程数。通常，此值应小于 2。如果此值持续偏高，则表明系统可能计算密集 (CPU-bound)。
%runocc	占用分发队列的时间百分比。
swpq-sz	换出进程的平均数量。
%swpocc	进程处于换出状态的时间百分比。

### 例 3-9 检查队列活动

以下示例显示 sar -q 命令的输出。如果 %runocc 值较大 (大于 90%) 并且 runq-sz 值大于 2，则表明 CPU 负载较大，并且响应变慢。在此情况下，可能需要提供附加的 CPU 容量，才能获得可接受的系统响应速度。

```

# sar -q
SunOS balmy 5.10 Generic_144500-10 sun4v ...

```

```
00:00:00 runq-sz %runocc swpq-sz %swpocc
01:00:00 1.0 7 0.0 0
02:00:00 1.0 7 0.0 0
03:00:00 1.0 7 0.0 0
04:00:00 1.0 7 0.0 0
05:00:00 1.0 6 0.0 0
06:00:00 1.0 7 0.0 0

Average 1.0 7 0.0 0
```

## 检查未使用的内存 (sar -r)

使用 sar -r 命令可报告当前未使用的内存页数和交换文件磁盘块数。

```
$ sar -r
00:00:00 freemem freeswap
01:00:00 2135 401922
```

-r 选项的输出如下所示：

freemem                    在该命令采样的时间间隔内可供用户进程使用的平均内存页数。页面大小与计算机有关。

freeswap                   可用于页交换的 512 字节磁盘块数。

### 例 3-10            检查未使用的内存 (sar -r)

以下示例显示 sar -r 命令的输出。

```
$ sar -r

SunOS balmy 5.10 Generic_144500-10 sun4v ...

00:00:04 freemem freeswap
01:00:00 44717 1715062
02:00:01 44733 1715496
03:00:00 44715 1714746
04:00:00 44751 1715403
05:00:00 44784 1714743
06:00:00 44794 1715186
07:00:00 44793 1715159
08:00:00 44786 1714914
08:20:00 44805 1715576
08:40:01 44797 1715347
09:00:00 44761 1713948
09:20:00 44802 1715478
09:40:00 41770 1682239
10:00:00 35401 1610833
10:20:00 34295 1599141
10:40:00 33943 1598425
```

```
11:00:00  30500  1561959
Average   43312  1699242
```

## 检查 CPU 使用率 (sar -u)

使用 sar -u 命令可显示 CPU 使用率统计信息。

```
$ sar -u
00:00:00  %usr  %sys  %wio  %idle
01:00:00      0     0     0    100
```

没有任何选项的 sar 命令与 sar -u 命令等效。在任意给定时刻，处理器都会处于繁忙或空闲状态。繁忙时，处理器可能处于用户模式或系统模式。空闲时，处理器可能在等待 I/O 完成，或没有需要执行的操作。

-u 选项的输出如下所示：

```
%usr          处理器处于用户模式的时间百分比
%sys          处理器处于系统模式的时间百分比
%wio          处理器空闲并等待 I/O 完成的时间百分比
%idle         处理器空闲并且未等待 I/O 的时间百分比
```

%wio 值越大，通常表示磁盘速率变慢。

### 例 3-11 检查 CPU 使用率 (sar -u)

以下示例显示 sar -u 命令的输出。

```
$ sar -u
00:00:04  %usr  %sys  %wio  %idle
01:00:00      0     0     0    100
02:00:01      0     0     0    100
03:00:00      0     0     0    100
04:00:00      0     0     0    100
05:00:00      0     0     0    100
06:00:00      0     0     0    100
07:00:00      0     0     0    100
08:00:00      0     0     0    100
08:20:00      0     0     0     99
08:40:01      0     0     0     99
09:00:00      0     0     0     99
09:20:00      0     0     0     99
09:40:00      4     1     0     95
10:00:00      4     2     0     94
```

```

10:20:00      1      1      0      98
10:40:00     18      3      0      79
11:00:00     25      3      0      72

Average        2      0      0      98
    
```

## 检查系统表状态 (sar -v)

使用 sar -v 命令可报告进程表、inode 表、文件表和共享内存记录表的状态。

```

$ sar -v
00:00:00  proc-sz   ov inod-sz   ov file-sz   ov  lock-sz
01:00:00  43/922    0 2984/4236  0 322/322    0   0/0
    
```

以下列表介绍了 -v 选项的输出。

- proc-sz                   内核中当前正在使用或已分配的进程项 (proc 结构) 数。
- inod-sz                   与内核中分配的最大 inode 数相比，内存中的 inode 总数。此数字不是严格的高水位标记。该数字可以溢出。
- file-sz                   开放式文件系统表的大小。由于文件表的空间是动态分配的，因此 sz 被给定为 0。
- ov                        在每个表的采样点之间发生的溢出。
- lock-sz                   内核中目前正在使用或已经分配的共享内存记录表项的数量。由于共享内存记录表的空间是动态分配的，因此 sz 被给定为 0。

### 例 3-12            检查系统表状态 (sar -v)

以下缩写示例显示 sar -v 命令的输出。此示例表明，所有表都足够大，因此没有溢出。这些表都基于物理内存量进行动态分配。

```

$ sar -v

00:00:04  proc-sz   ov inod-sz   ov file-sz   ov  lock-sz
01:00:00  69/8010   0 3476/34703  0  0/0      0  0/0
02:00:01  69/8010   0 3476/34703  0  0/0      0  0/0
03:00:00  69/8010   0 3476/34703  0  0/0      0  0/0
04:00:00  69/8010   0 3494/34703  0  0/0      0  0/0
05:00:00  69/8010   0 3494/34703  0  0/0      0  0/0
06:00:00  69/8010   0 3494/34703  0  0/0      0  0/0
07:00:00  69/8010   0 3494/34703  0  0/0      0  0/0
08:00:00  69/8010   0 3494/34703  0  0/0      0  0/0
08:20:00  69/8010   0 3494/34703  0  0/0      0  0/0
08:40:01  69/8010   0 3494/34703  0  0/0      0  0/0
09:00:00  69/8010   0 3494/34703  0  0/0      0  0/0
    
```

```
09:20:00 69/8010 0 3494/34703 0 0/0 0 0/0
09:40:00 74/8010 0 3494/34703 0 0/0 0 0/0
10:00:00 75/8010 0 4918/34703 0 0/0 0 0/0
10:20:00 72/8010 0 4918/34703 0 0/0 0 0/0
10:40:00 71/8010 0 5018/34703 0 0/0 0 0/0
11:00:00 77/8010 0 5018/34703 0 0/0 0 0/0
```

## 检查交换活动 (sar -w)

使用 sar -w 命令可报告交换和切换活动。

```
$ sar -w
00:00:00 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 22
```

以下列表介绍了 sar -w 命令输出的目标值和观测值。

swpin/s	每秒传入内存的 LWP 数。
bswin/s	每秒为换入传送的块数。/* (float)PGTOBLK(xx->cvmi.pgswapin) / sec_diff */.
swpot/s	每秒换出内存的平均进程数。如果该数字大于 1，则可能需要增大内存。
bswot/s	每秒为换出传送的块数。
pswch/s	每秒的内核线程切换数。

---

注 - 所有进程换入都包括进程初始化。

---

### 例 3-13 检查交换活动 (sar -w)

以下示例显示 sar -w 命令的输出。

```
$ sar -w
00:00:04 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 132
02:00:01 0.00 0.0 0.00 0.0 133
03:00:00 0.00 0.0 0.00 0.0 133
04:00:00 0.00 0.0 0.00 0.0 134
05:00:00 0.00 0.0 0.00 0.0 133
06:00:00 0.00 0.0 0.00 0.0 133
07:00:00 0.00 0.0 0.00 0.0 132
08:00:00 0.00 0.0 0.00 0.0 131
08:20:00 0.00 0.0 0.00 0.0 133
08:40:01 0.00 0.0 0.00 0.0 132
```

```

09:00:00  0.00  0.0  0.00  0.0  132
09:20:00  0.00  0.0  0.00  0.0  132
09:40:00  0.00  0.0  0.00  0.0  335
10:00:00  0.00  0.0  0.00  0.0  601
10:20:00  0.00  0.0  0.00  0.0  353
10:40:00  0.00  0.0  0.00  0.0  747
11:00:00  0.00  0.0  0.00  0.0  804

Average  0.00  0.0  0.00  0.0  198

```

## 检查终端活动 (sar -y)

使用 sar -y 命令可监视终端设备活动。

```

$ sar -y
00:00:00 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00      0      0      0      0      0      0

```

如果有多个终端 I/O，则可使用此报告来确定是否存在任何错误行。以下列表中定义了记录的活动。

rawch/s	每秒输入字符数 (原始队列)
canch/s	canon (规则队列) 每秒处理的输入字符数
outch/s	每秒输出字符数 (输出队列)
rcvin/s	每秒接收器硬件中断次数
xmtin/s	每秒传送器硬件中断次数
mdmin/s	每秒调制解调器中断次数

每秒调制解调器中断次数 (mdmin/s) 应接近于零。每秒的接收和传送中断次数 (rcvin/s 和 xmtin/s) 应分别小于或等于传入或传出字符数。否则，请检查是否存在错误行。

### 例 3-14 检查终端活动 (sar -y)

以下示例显示 sar -y 命令的输出。

```

$ sar -y
00:00:04 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00      0      0      0      0      0      0
02:00:01      0      0      0      0      0      0
03:00:00      0      0      0      0      0      0
04:00:00      0      0      0      0      0      0
05:00:00      0      0      0      0      0      0
06:00:00      0      0      0      0      0      0

```

07:00:00	0	0	0	0	0	0
08:00:00	0	0	0	0	0	0
08:20:00	0	0	0	0	0	0
08:40:01	0	0	0	0	0	0
09:00:00	0	0	0	0	0	0
09:20:00	0	0	0	0	0	0
09:40:00	0	0	1	0	0	0
10:00:00	0	0	37	0	0	0
10:20:00	0	0	0	0	0	0
10:40:00	0	0	3	0	0	0
11:00:00	0	0	3	0	0	0
Average	0	0	1	0	0	0

## 检查总体系统性能 (sar -A)

使用 `sar -A` 命令可显示所有选项的统计信息，以提供总体系统性能的综览。

此命令可提供更具全局性的透视。如果显示来自多个单时间段的数据，则该报告会包括平均值。

## 自动收集系统活动数据 (sar)

自动收集系统活动数据时需要使用三个命令：`sadc`、`sa1` 和 `sa2`。

`sadc` 数据收集实用程序定期收集系统数据，并以二进制格式的文件保存数据，每 24 小时保存一个文件。可以将 `sadc` 命令设置为定期运行（通常每小时一次），并在系统引导到多用户模式时运行。数据文件放置在 `/var/adm/sa` 目录中。每个文件都命名为 `sadd`，其中 `dd` 是当前日期。命令的格式如下：

```
/usr/lib/sa/sadc [t n] [ofile]
```

该命令以 `t` 秒为间隔采样 `n` 次，两次采样之间的间隔应大于 5 秒。然后，此命令将向二进制 `ofile` 文件或标准输出中写入数据。

## 引导时运行 `sadc` 命令

`sadc` 命令应在系统引导时运行，以记录自计数器重置为零以来的统计信息。为确保在引导时运行 `sadc` 命令，`svcadm enable system/sar:default` 命令会向每日数据文件中写入一条记录。

该命令项格式如下：

```
/usr/bin/su sys -c "/usr/lib/sa/sadc /var/adm/sa/sa`date +%d`"
```

## 使用 sa1 脚本定期运行 sadc 命令

为了生成定期记录，您需要定期运行 sadc 命令。最简单的方法是在 /var/spool/cron/crontabs/sys 文件中取消对下列行的注释：

```
# 0 * * * 0-6 /usr/lib/sa/sa1
# 20,40 8-17 * * 1-5 /usr/lib/sa/sa1
# 5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

缺省 sys crontab 项执行以下操作：

- 前两个 crontab 项可在星期一到星期五从上午 8 点到下午 5 点，每 20 分钟或每小时向 /var/adm/sa/sadd 文件写入一条记录。
- 第三项可从星期一到星期五每小时向 /var/adm/sa/saradd 文件写入一条记录，并且可以包括所有 sar 选项。

可以更改这些缺省设置，以满足您的需要。

## 使用 sa2 Shell 脚本生成报告

另一个 shell 脚本 sa2 可生成报告，而不是二进制数据文件。sa2 命令调用 sar 命令，并将 ASCII 输出写入报告文件。

## 设置自动数据收集 (sar)

sar 命令可用于自行收集系统活动数据，或报告 sadc 命令所创建的每日活动文件中收集的内容。

sar 命令格式如下：

```
sar [-aAbcdgkmpqruvw] [-o file] t [n]
sar [-aAbcdgkmpqruvw] [-s time] [-e time] [-i sec] [-f file]
```

第一种格式每隔 *t* 秒对操作系统中的累积活动计数器进行采样，共进行 *n* 次。*t* 应大于或等于 5 秒。否则，命令本身会对样本产生影响。必须指定采样的时间间隔。否则，命令将根据第二种格式运行。*n* 的缺省值为 1。

以下示例使用第二种格式，以 10 秒为间隔抽取两个样本。如果指定 -o 选项，则以二进制格式保存样本。

```
$ sar -u 10 2
```

使用第二种格式 sar 命令未指定采样间隔或采样次数，从以前记录的文件中提取数据。该文件是由 -f 选项为最近一天指定的文件，或对应于最近一天的标准每日活动文件 /var/adm/sa/sadd (缺省设置)。

-s 和 -e 选项定义报告的开始时间和结束时间。开始时间和结束时间的格式为 `hh[:mm[:ss]]`，其中 `hh`、`mm` 和 `ss` 表示小时、分钟和秒。

-i 选项指定记录选择之间的时间间隔（秒）。如果不包括 -i 选项，则报告在每日活动文件中找到的所有间隔。

sar 选项及其操作如下所示：

---

注 - 不使用任何选项等效于调用带 -u 选项的 sar 命令。

---

-a	检查文件访问操作
-b	检查缓冲区活动
-c	检查系统调用
-d	检查每个块设备的活动
-g	检查页出和内存释放
-k	检查内核内存分配
-m	检查进程间通信
-nv	检查系统表状态
-p	检查交换和分发活动
-q	检查队列活动
-r	检查未使用的内存
-u	检查 CPU 使用率
-w	检查交换和切换卷
-y	检查终端活动
-A	报告总体系统性能，这与输入所有选项等效

## ▼ 如何设置自动数据收集

### 1. 承担 root 角色。

请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 运行 `svcadm enable system/sar:default` 命令。

此版本的 `sadc` 命令会写入一条特殊记录，其中标记了将计数器重置为零的时间（引导时间）。

3. 编辑 `/var/spool/cron/crontabs/sys` `crontab` 文件。

---

注 - 不要直接编辑 `crontab` 文件。而要改用 `crontab -e` 命令对现有 `crontab` 文件进行更改。

---

```
# crontab -e sys
```

4. 取消对以下行的注释：

```
0 * * * 0-6 /usr/lib/sa/sa1
20,40 8-17 * * 1-5 /usr/lib/sa/sa1
5 18 * * 1-5 /usr/lib/sa/sa2 -s 8:00 -e 18:01 -i 1200 -A
```

有关更多信息，请参见 [crontab\(1\)](#) 手册页。

## 调度系统任务

---

本章介绍如何使用 `crontab` 和 `at` 命令来调度例程系统任务或单个（一次性）任务。本章还说明如何使用以下文件来控制对上述命令的访问：

- `cron.deny`
- `cron-allow`
- `at.deny`

本章包含以下主题：

- [“自动执行系统任务的方法” \[71\]](#)
- [“调度系统任务” \[73\]](#)
- [“使用 `at` 命令来调度任务” \[82\]](#)

### 自动执行系统任务的方法

可以设置多个要自动执行的系统任务。其中的某些任务应该以固定间隔执行。其他任务只需运行一次，可能是在晚间或周末等非高峰时间。

本节包含有关 `crontab` 和 `at` 这两个命令的概括性信息，使用这两个命令可以调度要自动执行的例行任务。`crontab` 命令用于调度重复性任务。`at` 命令则用于调度只执行一次的任务。

下表概括说明了 `crontab` 和 `at` 命令，以及可用来控制对这些命令的访问的文件。

表 4-1 命令摘要：调度系统任务

命令	调度对象	文件位置	控制访问的文件
<code>crontab</code>	按固定间隔执行的多个系统任务	<code>/var/spool/cron/crontabs</code>	<code>/etc/cron.d/cron.allow</code> 和 <code>/etc/cron.d/cron.deny</code>
<code>at</code>	单个系统任务	<code>/var/spool/cron/atjobs</code>	<code>/etc/cron.d/at.deny</code>

## 使用 crontab 调度重复性作业

可以使用 crontab 命令来调度例行系统管理任务，使其每日、每周或每月执行一次。

每日 crontab 系统管理任务可能包括以下内容：

- 从临时目录中删除几天前的文件
- 执行记帐摘要命令
- 使用 df 和 ps 命令捕获系统快照
- 执行每日安全监视
- 运行系统备份

每周 crontab 系统管理任务可包括以下内容：

- 重新生成 catman 数据库以供 man -k 命令使用
- 运行 fsck -n 命令以列出任何磁盘问题

每月 crontab 系统管理任务可能包括以下内容：

- 列出在特定月份中未使用的文件
- 生成每月记帐报告

此外，您还可以调度 crontab 命令执行其他例行系统任务，例如发送提醒和删除备份文件。

有关调度 crontab 作业的逐步说明，请参见[如何创建或编辑 crontab 文件 \[76\]](#)。

## 使用 at 调度单个作业

通过 at 命令可以调度要在以后执行的作业。该作业可由单个命令或脚本组成。

与 crontab 类似，使用 at 命令可以调度例行任务自动执行。但与 crontab 文件不同的是，at 文件只执行一次任务。然后，便从目录中删除这些文件。因此，at 命令非常适合于运行将输出定向到单独文件中（供以后检查）的简单命令或脚本。

提交 at 作业包括键入命令并按照 at 命令语法指定选项来调度执行作业的时间。有关提交 at 作业的更多信息，请参见[“提交 at 作业文件” \[83\]](#)。

at 命令在 /var/spool/cron/atjobs 目录中存储您运行的命令或脚本以及当前环境变量的副本。at 作业的文件名由一个长数字组成，用于指定该文件在 at 队列中的位置，后跟 .a 扩展名，例如 793962000.a。

cron 守护进程在启动时检查 at 作业并侦听是否提交了新作业。cron 守护进程执行 at 作业后，将从 atjobs 目录中删除 at 作业的文件。有关更多信息，请参见 [at\(1\)](#) 手册页。

有关调度 at 作业的逐步说明，请参见[如何创建 at 作业 \[83\]](#)。

## 调度系统任务

本节包含有关使用 crontab 文件来调度系统任务的任务。

### 创建和编辑 crontab 文件任务列表

任务	描述	相关说明
创建或编辑 crontab 文件。	使用 crontab -e 命令可创建或编辑 crontab 文件。	<a href="#">如何创建或编辑 crontab 文件 [76]</a>
验证 crontab 文件是否存在。	使用 ls -l 命令验证 /var/spool/cron/crontabs 文件的内容。	<a href="#">“验证 crontab 文件是否存在” [77]</a>
显示 crontab 文件。	使用 ls -l 命令显示 crontab 文件。	<a href="#">“显示 crontab 文件” [77]</a>
删除 crontab 文件。	crontab 文件设置时使用了受限权限。使用 crontab -r 命令而不是 rm 命令删除 crontab 文件。	<a href="#">如何删除 crontab 文件 [78]</a>
拒绝 crontab 访问。	要拒绝用户访问 crontab 命令，请将用户名添加到 /etc/cron.d/cron.deny 文件中。	<a href="#">如何拒绝 crontab 命令访问 [80]</a>
将 crontab 访问限定于指定的用户。	要允许用户访问 crontab 命令，请将用户名添加到 /etc/cron.d/cron.allow 文件中。	<a href="#">如何将 crontab 命令访问限制于指定的用户 [81]</a>

### 调度重复性系统任务 (cron)

以下各节介绍如何创建、编辑、显示和删除 crontab 文件，以及如何控制对这些文件的访问。

#### 在 crontab 文件内

cron 守护进程会根据在每个 crontab 文件中找到的命令来调度系统任务。crontab 文件由命令组成，每个命令占据一行，这些命令将以固定间隔执行。每行开头包含日期和时间信息，以告知 cron 守护进程何时执行命令。

例如，在 Oracle Solaris 软件安装期间将提供名为 root 的 crontab 文件。该文件的内容包括以下命令行：

```
10 3 * * * /usr/sbin/logadm          (1)
15 3 * * 0 /usr/lib/fs/nfs/nfsfind   (2)
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1    (3)
```

```
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean (4)
```

每个命令行的输出如下所示：

- 第一行在每天凌晨 3:10 运行 logadm 命令。
- 第二行在每个星期日凌晨 3:15 执行 nfsfind 脚本。
- 第三行在每天凌晨 2:10 运行用于检查夏时制时间（并根据需要进行更正）的脚本。如果没有 RTC 时区，也没有 /etc/rtc\_config 文件，则此项不执行任何操作。

---

仅限 x86 - /usr/sbin/rtc 脚本只能在基于 x86 的系统上运行。

---

- 第四行在每天凌晨 3:30 检查（并删除）通用安全服务表 /etc/gss/gsscred\_db 中的重复项。

有关 crontab 文件中各行语法的更多信息，请参见“[crontab 文件项的语法](#)” [75]。

crontab 文件存储在 /var/spool/cron/crontabs 目录中。在 Oracle Solaris 软件安装期间，会提供包括 root 在内的若干个 crontab 文件。

adm	记帐
root	一般系统功能和文件系统清除
sys	性能数据收集
uucp	一般 uucp 清除

除缺省 crontab 文件之外，您可以创建 crontab 文件来调度自己的系统任务。定制 crontab 文件按创建它们的用户帐户名称（如 bob、mary、smith 或 jones）命名。

要访问属于 root 或其他用户的 crontab 文件，需要具有超级用户特权。

## cron 守护进程处理调度的方法

cron 守护进程可管理 crontab 命令的自动调度。cron 守护进程将检查 /var/spool/cron/crontab 目录中是否存在 crontab 文件。

cron 守护进程在启动时执行以下任务：

- 检查新的 crontab 文件。
- 阅读文件中列出的执行时间。
- 在适当时间提交执行命令。
- 侦听来自 crontab 命令的有关更新的 crontab 文件的通知。

cron 守护进程以几乎相同的方式来控制 at 文件的调度。这些文件存储在 `/var/spool/cron/atjobs` 目录中。cron 守护进程还侦听来自 `crontab` 命令的有关已提交的 at 作业的通知。

## crontab 文件项的语法

crontab 文件由命令组成，每个命令占据一行，这些命令将按每个命令行的前五个字段（以空格分隔）指定的时间自动执行。

表 4-2 crontab 时间字段的可接受值

时间字段	值
Minute	0-59
Hour	0-23
Day of month	1-31
Month	1-12
Day of week	0-6 (0 = 星期日)

在 crontab 时间字段中使用特殊字符时请遵循以下规则：

- 使用空格分隔每个字段。
- 使用逗号分隔多个值。
- 使用连字符指定某一范围的值。
- 使用星号作为通配符来包括所有可能值。
- 在一行开头使用注释标记 (#) 来表示注释或空白行。

例如，以下 crontab 命令项将于每月第一天和第十五天下午 4 点在用户的控制台窗口中显示提醒。

```
0 16 1,15 * * echo Timesheets Due > /dev/console
```

crontab 文件中的每个命令必须只占据一行，即使这一行非常长也是如此。crontab 文件不识别额外的回车。有关 crontab 项和命令选项的更多详细信息，请参阅 [crontab\(1\)](#) 手册页。

## 创建和编辑 crontab 文件

创建 crontab 文件的最简单方法是使用 `crontab -e` 命令。此命令调用已使用 `EDITOR` 环境变量为您的系统环境定义的文本编辑器。如果尚未设置此变量，crontab 命令将使用缺省编辑器 `ed`。

以下示例说明如何确定是否已定义编辑器，以及如何将 `vi` 设置为缺省编辑器。

```
$ which $EDITOR
$
$ EDITOR=vi
$ export EDITOR
```

创建 crontab 文件时，该文件会自动放入 `/var/spool/cron/crontabs` 目录，并以您的用户名命名。可以为其他用户或 `root`（如果您有 `root` 用户特权）创建或编辑 crontab 文件。

## ▼ 如何创建或编辑 crontab 文件

开始之前 如果要创建或编辑属于其他用户的 crontab 文件，您必须承担 `root` 角色。请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

您无需承担 `root` 角色就可以编辑自己的 crontab 文件。

1. 创建新的 crontab 文件，或编辑现有文件。

```
# crontab -e [username]
```

其中，`username` 指定您要为其创建或编辑 crontab 文件的用户帐户的名称。无需具有超级用户特权便可创建自己的 crontab 文件，但如果要为 `root` 或其他用户创建或编辑 crontab 文件，则必须具有超级用户特权。



注意 - 如果意外键入了不带选项的 crontab 命令，请按下编辑器的中断字符，以便退出而不保存更改。如果保存了更改并退出文件，现有的 crontab 文件将被空文件覆盖。

2. 向 crontab 文件中添加命令行。  
按照“[crontab 文件项的语法](#)” [75] 中所述的语法操作。将 crontab 文件放入 `/var/spool/cron/crontabs` 目录。
3. 验证 crontab 文件更改。

```
# crontab -l [username]
```

### 例 4-1 创建 crontab 文件

以下示例说明如何为其他用户创建 crontab 文件。

```
# crontab -e mary
```

添加到新 crontab 文件中的以下命令项将在每个星期日的凌晨 1:00 自动删除 Mary 的起始目录中的所有日志文件。由于该命令项不重定向输出，因此将重定向字符添加到 `*.log` 之后的命令行中。这样可以确保正常执行命令。

```
# This command helps clean up user accounts.
1 0 * * 0 rm /home/mary/*.log > /dev/null 2>&1
```

## 显示和验证 crontab 文件

可以使用 `crontab -l` 命令显示和验证 crontab 文件的内容。

### 验证 crontab 文件是否存在

要验证某位用户是否存在 crontab 文件，请使用 `/var/spool/cron/crontabs` 目录中的 `ls -l` 命令。例如，以下示例输出说明系统上存在多个用户的 crontab 文件。

```
$ ls -l /var/spool/cron/crontabs
drwxr-xr-x  2 root    sys      12 Nov 26 16:55 ./
drwxr-xr-x  4 root    sys      4 Apr 28  2012 ../
-rw-----  1 root    sys      190 Jun 28  2011 adm
-rw-----  1 root    staff    0 Nov 13  2012 mary
-rw-----  1 root    un       437 Oct  8  2012 johndoe
-r-----  1 root    root     453 Apr 28  2012 lp
-rw-----  1 root    sparccad 63 Jul 17 10:39 mary2
-rw-----  1 root    sparccad 387 Oct 14 15:15 johndoe2
-rw-----  1 root    other    2467 Nov 26 16:55 root
-rw-----  1 root    sys      308 Jun 28  2011 sys
-rw-----  1 root    sietee   163 Nov 20 10:40 mary3
-r-----  1 root    sys      404 Jan 24  2013 uucp
```

### 显示 crontab 文件

`crontab -l` 命令显示 crontab 文件内容的方式与 `cat` 命令显示其他类型文件内容的方式非常相似。无需转到 `/var/spool/cron/crontabs` 目录（crontab 文件所在的目录），便可使用此命令。

缺省情况下，`crontab -l` 命令显示您自己的 crontab 文件。要显示属于其他用户的 crontab 文件，您必须承担 root 角色。

You can use the crontab command as follows:

```
# crontab -l [username]
```

其中，`username` 指定了要显示其 crontab 文件的用户帐户的名称。显示其他用户的 crontab 文件需要超级用户特权。



注意 - 如果意外键入了不带选项的 `crontab` 命令，请按下编辑器的中断字符，以便退出而不保存更改。如果保存了更改并退出文件，现有的 crontab 文件将被空文件覆盖。

例 4-2            显示 crontab 文件

此示例说明如何使用 `crontab -l` 命令来显示您的缺省 crontab 文件的内容。

```
$ crontab -l
13 13 * * * chmod g+w /home1/documents/*.book > /dev/null 2>&1
```

#### 例 4-3 显示缺省的 root crontab 文件

此示例说明如何显示缺省的 root crontab 文件。

```
$ su
Password:

# crontab -l
#ident "@(#)root 1.19 98/07/06 SMI" /* SVr4.0 1.1.3.1 */
#
# The root crontab should be used to perform accounting data collection.
#
#
10 3 * * * /usr/sbin/logadm
15 3 * * 0 /usr/lib/fs/nfs/nfsfind
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean
#10 3 * * * /usr/lib/krb5/kprop_script ___slave_kdcs___
```

#### 例 4-4 显示其他用户的 crontab 文件

此示例说明如何显示属于其他用户的 crontab 文件。

```
$ su
Password:
# crontab -l jones
13 13 * * * cp /home/jones/work_files /usr/backup/. > /dev/null 2>&1
```

## 删除 crontab 文件

缺省情况下，设置了 crontab 文件保护，以防止使用 `rm` 命令意外删除 crontab 文件。请改用 `crontab -r` 命令删除 crontab 文件。

缺省情况下，`crontab -r` 命令会删除您自己的 crontab 文件。

无需转到 `/var/spool/cron/crontabs` 目录（crontab 文件所在的目录），便可使用此命令。

### ▼ 如何删除 crontab 文件

**开始之前** 承担 root 角色以删除属于 root 或其他用户的 crontab 文件。角色包含授权和具有特权的命令。请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

您无需承担 root 角色即可删除自己的 crontab 文件。

### 1. 删除 crontab 文件。

```
# crontab -r [username]
```

其中 *username* 指定要为其删除 crontab 文件的用户帐户的名称。要删除其他用户的 crontab 文件，请承担 root 角色。



注意 - 如果意外键入了不带选项的 crontab 命令，请按下编辑器的中断字符，以便退出而不保存更改。如果保存了更改并退出文件，现有的 crontab 文件将被空文件覆盖。

### 2. 验证是否已删除 crontab 文件。

```
# ls /var/spool/cron/crontabs
```

#### 例 4-5 删除 crontab 文件

以下示例说明用户 smith 如何使用 crontab -r 命令删除他自己的 crontab 文件。

```
$ ls /var/spool/cron/crontabs
adm   jones  root   smith  sys    uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm   jones  root   sys    uucp
```

## 控制对 crontab 命令的访问

可以使用 `/etc/cron.d` 目录中的以下两个文件来控制对 crontab 命令的访问：`cron.deny` 和 `cron.allow`。这些文件只允许指定的用户执行 crontab 命令任务，例如创建、编辑、显示或删除自己的 crontab 文件。

`cron.deny` 和 `cron.allow` 文件包含用户名的列表，每行一个用户名。

这些访问控制文件按以下方式协同工作：

- 如果存在 `cron.allow`，则只有此文件中列出的用户可以创建、编辑、显示或删除 crontab 文件。
- 如果不存在 `cron.allow`，则所有用户都可以提交 crontab 文件（`cron.deny` 中列出的用户除外）。
- 如果既不存在 `cron.allow` 也不存在 `cron.deny`，您必须承担 root 角色才能运行 crontab 命令。
- 要编辑或创建 `cron.deny` 以及 `cron.allow` 文件，您必须承担 root 角色。

在 Oracle Solaris 软件安装期间创建的 `cron.deny` 文件包含以下用户名：

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

缺省 `cron.deny` 文件中的用户名都不能访问 `crontab` 命令。您可以编辑此文件，以添加被拒绝访问 `crontab` 命令的其他用户。

由于未提供缺省的 `cron.allow` 文件，因此，除缺省 `cron.deny` 文件中列出的用户之外的所有用户都可以使用 `crontab` 命令。如果创建 `cron.allow` 文件，则只有这些用户可以访问 `crontab` 命令。

## ▼ 如何拒绝 crontab 命令访问

1. 承担 `root` 角色。

请参见 [《在 Oracle Solaris 11.2 中确保用户和进程的安全》](#) 中的“使用所指定的管理权限”。

2. 编辑 `/etc/cron.d/cron.deny` 文件并添加将禁止其访问 `crontab` 命令的用户名，每行一个用户。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
username3
.
.
.
```

3. 验证 `/etc/cron.d/cron.deny` 文件是否包含新项。

```
# cat /etc/cron.d/cron.deny
daemon
bin
nuucp
listen
```

```
nobody
noaccess
```

## ▼ 如何将 crontab 命令访问限制于指定的用户

1. 承担 **root** 角色。  
请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。
2. 创建 `/etc/cron.d/cron.allow` 文件。
3. 将 **root** 角色添加到 `cron.allow` 文件。  
如果未将 **root** 添加到该文件中，则会拒绝 **root** 用户访问 `crontab` 命令。
4. 添加将允许其使用 `crontab` 命令的用户名，每行一个用户。

```
root
username1
username2
username3
.
.
.
```

### 例 4-6 将 crontab 命令访问限制于指定的用户

以下示例显示一个 `cron.deny` 文件，该文件用于禁止用户名 `jones`、`temp` 和 `visitor` 访问 `crontab` 命令。

```
$ cat /etc/cron.d/cron.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
temp
visitor
```

以下示例显示一个 `cron.allow` 文件。用户 `root`、`jones` 和 `smith` 是仅有的可以访问 `crontab` 命令的用户。

```
$ cat /etc/cron.d/cron.allow
root
jones
smith
```

## 如何验证受限的 crontab 命令访问

要验证特定用户是否可以访问 crontab 命令，请在使用该用户帐户登录后执行 crontab -l 命令。

```
$ crontab -l
```

如果用户可以访问 crontab 命令并已创建 crontab 文件，则会显示该文件。否则，如果用户可以访问 crontab 命令但不存在 crontab 文件，则会显示与以下消息类似的消息：

```
crontab: can't open your crontab file
```

此用户已列在 cron.allow 文件中（如果存在该文件），或者该用户未列在 cron.deny 文件中。

如果用户不能访问 crontab 命令，则无论是否存在以前的 crontab 文件，都会显示以下消息：

```
crontab: you are not authorized to use cron. Sorry.
```

此消息表明，该用户未列在 cron.allow 文件（如果该文件存在）中，或者该用户已列在 cron.deny 文件中。

## 使用 at 命令来调度任务

本节包含有关使用 at 命令来调度例行系统任务的任务。

### 使用 at 命令

使用以下任务在您的系统上创建和管理例行系统任务。

- [如何创建 at 作业](#) [83]
- [“显示 at 队列”](#) [84]
- [“验证 at 作业”](#) [84]
- [“显示 at 作业”](#) [85]
- [如何删除 at 作业](#) [85]
- [“拒绝访问 at 命令”](#) [86]

### 调度单个系统任务 (at)

以下各节介绍如何使用 at 命令来执行下列任务：

- 调度作业（命令和脚本）供以后执行
- 显示和删除作业
- 控制对 at 命令的访问

缺省情况下，用户可以创建、显示和删除自己的 at 作业文件。要访问属于 root 或其他用户的 at 文件，您必须承担 root 角色。

## 提交 at 作业文件

提交 at 作业时，会为该作业指定作业标识号和 .a 扩展名。此指定将成为该作业的文件名，以及其队列编号。

提交 at 作业文件需执行以下步骤：

1. 调用 at 实用程序并指定命令执行时间。
2. 键入以后要执行的命令或脚本。

---

注 - 如果此命令或脚本的输出很重要，请确保将输出定向到一个文件中，以便以后检查。

---

例如，以下 at 作业将在 7 月的最后一天接近午夜时删除用户帐户 smith 的 core 文件。

```
$ at 11:45pm July 31
at> rm /home/smith/*core*
at> Press Control-d
commands will be executed using /bin/csh
job 933486300.a at Tue Jul 31 23:45:00 2004
```

## 创建 at 作业

以下任务介绍如何创建和 at 作业。

### ▼ 如何创建 at 作业

1. 启动 at 实用程序，指定所需的作业执行时间。

```
$ at [-m] time [date]
```

-m 指定此选项将在作业完成之后向您发送一封电子邮件。

time 指定要调度作业的小时。如果不按 24 小时制指定小时，请添加 am 或 pm。可接受的关键字包括 midnight、noon 和 now。分钟是可选的选项。

`date` 指定月份的前三个或更多字母、一周中的某日或关键字 `today` 或 `tomorrow`。

2. 在 `at` 提示符下，键入要执行的命令或脚本，每行一个。  
通过在每行结尾处按回车键，可以键入多个命令。
3. 按 `Ctrl-D` 组合键，以退出 `at` 实用程序并保存 `at` 作业。  
`at` 作业将被指定一个队列编号，它也是该作业的文件名。退出 `at` 实用程序时将显示该编号。

#### 例 4-7 创建 at 作业

以下示例显示了用户 `jones` 创建的 `at` 作业，该作业用于在下午 7:30 删除其备份文件。由于她使用了 `-m` 选项，因此她会在该作业完成后收到电子邮件。

```
$ at -m 1930
at> rm /home/jones/*.backup
at> Press Control-D
job 897355800.a at Thu Jul 12 19:30:00 2004
```

她收到一封确认已执行 `at` 作业的电子邮件。

```
Your "at" job "rm /home/jones/*.backup"
completed.
```

以下示例说明 `jones` 如何调度在星期六凌晨 4:00 执行的大型 `at` 作业。该作业输出被定向到名为 `big.file` 的文件中。

```
$ at 4 am Saturday
at> sort -r /usr/dict/words > /export/home/jones/big.file
```

## 显示 at 队列

要检查在 `at` 队列中等待的作业，请使用 `atq` 命令。

```
$ atq
```

此命令可以显示您已创建的 `at` 作业的状态信息。

## 验证 at 作业

要验证您是否已创建了 `at` 作业，请使用 `atq` 命令。在以下示例中，`atq` 命令确认已将属于 `jones` 的 `at` 作业提交至队列。

```
$ atq
Rank  Execution Date      Owner   Job           Queue  Job Name
1st   Jul 12, 2004 19:30   jones  897355800.a   a      stdin
2nd   Jul 14, 2004 23:45   jones  897543900.a   a      stdin
3rd   Jul 17, 2004 04:00   jones  897732000.a   a      stdin
```

## 显示 at 作业

要显示有关 at 作业的执行时间信息，请使用 `at -l` 命令。

```
$ at -l [job-id]
```

其中，`-l job-id` 是您希望显示其状态的特定作业的可选标识号。如果没有 ID，命令将显示用户已提交的所有作业的状态。

例 4-8 显示 at 作业

以下示例显示 `at -l` 命令的示例输出，该输出提供有关用户已提交的所有作业的状态信息。

```
$ at -l
897543900.a Sat Jul 14 23:45:00 2004
897355800.a Thu Jul 12 19:30:00 2004
897732000.a Tue Jul 17 04:00:00 2004
```

以下示例显示使用 `at -l` 命令指定单个作业时显示的示例输出。

```
$ at -l 897732000.a
897732000.a Tue Jul 17 04:00:00 2004
```

## ▼ 如何删除 at 作业

开始之前 承担 root 角色以删除属于 root 或其他用户的 at 作业。请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

您无需承担 root 角色即可删除自己的 at 作业。

1. 在作业执行之前从队列中删除 at 作业。

```
# at -r [job-id]
```

其中，`-r job-id` 选项指定要删除的作业的标识号。

2. 使用 `at -l` (或 `atq`) 命令，验证是否已删除 at 作业。

`at -l` 命令显示 at 队列中剩余的作业。不应显示已指定标识号的作业。

```
$ at -l [job-id]
```

#### 例 4-9 删除 at 作业

在以下示例中，用户要删除计划在 7 月 17 日凌晨 4 点执行的 at 作业。首先，该用户显示 at 队列，以找到作业标识号。然后，用户从 at 队列中删除此作业。最后，该用户验证是否已从队列中删除此作业。

```
$ at -l
897543900.a Sat Jul 14 23:45:00 2003
897355800.a Thu Jul 12 19:30:00 2003
897732000.a Tue Jul 17 04:00:00 2003
$ at -r 897732000.a
$ at -l 897732000.a
at: 858142000.a: No such file or directory
```

## 控制对 at 命令的访问

您可以设置一个文件来控制对 at 命令的访问，只允许指定的用户创建、删除或显示有关 at 作业的队列信息。控制对 at 命令的访问的文件 `/etc/cron.d/at.deny` 由用户名列表构成，每个用户名占据一行。此文件中列出的用户不能访问 at 命令。

在 Oracle Solaris 软件安装期间创建的 `at.deny` 文件包含以下用户名：

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
```

使用超级用户特权，您可以编辑 `at.deny` 文件，以添加要限制其对 at 命令访问的其他用户名。

## 拒绝访问 at 命令

作为 root 身份编辑 `/etc/cron.d/at.deny` 文件，以添加希望禁止其使用 at 命令的用户名，每行一个用户名。

```
daemon
bin
smtp
nuucp
listen
nobody
noaccess
username1
username2
```

```
username3
.
```

#### 例 4-10 拒绝 at 访问

以下示例显示了一个 `at.deny` 文件，该文件已被编辑过，因此用户 `smith` 和 `jones` 无法访问 `at` 命令。

```
$ cat at.deny
daemon
bin
smtp
nuucp
listen
nobody
noaccess
jones
smith
```

#### 验证是否已拒绝访问 at 命令

要验证是否将用户名正确添加到 `/etc/cron.d/at.deny` 文件，请在作为该用户登录并使用 `at -l` 命令。例如，如果登录用户 `smith` 无法访问 `at` 命令，则显示以下消息：

```
# su smith
Password:
# at -l
at: you are not authorized to use at. Sorry.
```

类似地，如果该用户尝试提交 `at` 作业，则将显示以下消息：

```
# at 2:30pm
at: you are not authorized to use at. Sorry.
```

此消息确认该用户已列在 `at.deny` 文件中。

如果允许访问 `at` 命令，则 `at -l` 命令不会返回任何内容。



## 管理系统控制台、终端设备和电源服务

---

本章介绍如何通过 `ttymon` 程序以及系统电源服务管理系统控制台和本地连接的终端设备。

本章包含以下主题：

- “管理系统控制台和本地连接的终端设备” [89]
- “管理系统电源服务” [91]

### 管理系统控制台和本地连接的终端设备

系统控制台是具有特殊属性的终端，用于特定目的。例如：针对管理员的内核消息将发送到控制台，而不是其他终端。

终端是用于与 Oracle Solaris 交互的一种方式。您系统的位映射图形显示与字母数字终端不同。字母数字终端连接到串行端口，并仅显示文本。您不必执行任何特殊步骤即可管理图形显示。

终端也可以与计算机的物理显示器和键盘布局相关联。图形终端的不同之处在于，它必须与计算机的图形卡和显示器相关联。因此，字符放置到计算机图形卡的存储器中，而不是从串行端口传送出去。

### 管理系统控制台和本地连接的终端设备的 SMF 服务

系统控制台和本地连接的终端设备以 SMF 服务 (`svc:/system/console`) 的实例来表示。此服务可定义大部分行为，其每个实例都以特定值覆盖从服务继承的设置。使用 `ttymon` 程序提供这些终端的登录服务。每个终端都使用 `ttymon` 程序的独立实例。服务传递到 `ttymon` 程序的命令行参数管理其行为。

系统提供的服务实例如下所示：

- `svc:/system/console-login:default`  
该缺省实例始终表示 `ttymon` 程序提供系统硬件控制台登录。
- `svc:/system/console-login:{vt2, vt3, vt4, vt5, vt6}`

提供的其他服务实例用于系统的虚拟控制台。如果虚拟控制台不可用，则会自动禁用这些服务。有关更多信息，请参见 [vtdaemon\(1M\)](#) 手册页。

- `svc:/system/console-login:{terma, termb}`

为方便起见，还提供了 `svc:/system/console-login:terma` 和 `svc:/system/console-login:termb` 服务。这些服务可以帮助您设置其他 `/dev/term/a` 和 `/dev/term/b` 端口的登录服务。缺省情况下这两个服务处于禁用状态。

您可以将其他服务实例定义为 `svc:system/console-login` 服务的一部分。例如，如果您有需要支持的 `/dev/term/f` 设备，则可以实例化 `svc:/system/console-login:termf` 并进行相应的配置。

## ▼ 如何在辅助终端上设置登录服务

对于连接到系统的 `/dev/term/a` 或 `/dev/term/b` 串行端口的终端，提供了预定义的服务。

1. 承担 `root` 角色。

请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 启用服务实例。

例如，要启用 `/dev/term/a` 的登录服务：

```
# svcadm enable svc:/system/console-login:terma
```

3. 检查服务是否处于联机状态。

```
# svcs svc:/system/console-login:terma
```

输出应显示服务处于联机状态。如果服务处于维护模式，请查看服务的日志文件以获得进一步的详细信息。

## ▼ 如何在控制台上设置波特率速度

基于 x86 的系统上支持的控制台速度取决于具体的平台。

基于 SPARC 的系统上支持以下控制台速度：

- 9600 bps
- 19200 bps
- 38400 bps

1. 成为管理员。

请参见《在 Oracle Solaris 11.2 中确保用户和进程的安全》中的“使用所指定的管理权限”。

2. 使用 `eeeprom` 命令设置适合您的系统类型的波特率速度。

```
# eeeprom ttya-mode=baud-rate,8,n,1,-
```

例如，要将基于 x86 的系统控制台上的波特率更改为 38400，请键入：

```
# eeeprom ttya-mode=38400,8,n,1,-
```

3. 按如下所示在 `/etc/ttydefs` 文件中更改控制台一行：

```
console baud-rate hupcl opost onlcr:baud-rate::console
```

4. 根据您的系统类型，进行以下附加更改。

请注意，这些更改与平台相关。

- 在基于 SPARC 的系统上：在 `options.conf` 文件内（`/etc/driver/drv` 目录中的那一版本）更改波特率速度。例如：

要将波特率更改为 9600：

```
# 9600          :bd:
ttymodes="2502:1805:bd:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

要将波特率速度更改为 19200：

```
# 19200         :be:
ttymodes="2502:1805:be:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

要将波特率速度更改为 38400：

```
# 38400         :bf:
ttymodes="2502:1805:bf:8a3b:3:1c:7f:15:4:0:0:0:11:13:1a:19:12:f:17:16";
```

- 在基于 x86 的系统上：如果启用了 BIOS 串行重定向，请更改控制台速度。

## 管理系统电源服务

在 Oracle Solaris 11 操作系统中，电源管理配置已移动到 SMF 配置系统信息库中。使用新的 `poweradm` 命令来直接管理系统电源管理属性，而不再组合使用与电源相关的命令、守护进程和配置文件。这些变化是大范围变化中的一部分，用于使 Oracle Solaris 11 操作系统中的电源管理框架更为现代化。

以下电源管理功能已不再可用。

- `/etc/power.conf`

- pmconfig 和 powerd
- 设备电源管理

由以下属性描述电源管理组件：

- administrative-authority – 为 Oracle Solaris 电源管理定义管理控制源。此属性可设置为 none、platform (缺省值) 或 smf。  
 设置为 platform 时，将从平台的电源管理命令获取 time-to-full-capacity 和 time-to-minimum-responsiveness 的值。  
 设置为 smf 时，将从 SMF 获取 time-to-full-capacity 和 time-to-minimum-responsiveness 的值。  
 如果尝试从平台命令或 SMF 服务属性设置 time-to-full-capacity 或 time-to-minimum-responsiveness，而该值与所用的设置方式相反，则该值将被忽略。  
 当 administrative-authority 设置为 none 时，Oracle Solaris 实例中的电源管理将关闭。
- time-to-full-capacity – 定义系统处于活动状态时，允许系统从任何容量较低或响应较慢状态达到全部容量的最大时间（微秒）。最长时间包括它使用此边界中的任何或全部 PM 功能的时间。  
 缺省情况下，从平台（例如 i86pc）获取该值，因为 administrative-authority 的缺省设置为 platform。  
 或者，如果 administrative-authority 设置为 smf，则从 SMF 电源服务提供的定义中获取该值。安装时，该值未定义。如果您选择修改此属性，则应考虑与系统工作负荷或应用程序的需求相适应的值。
- time-to-minimum-responsiveness – 定义允许系统返回到活动状态的时间（毫秒）。此参数提供满足 time-to-full-capacity 约束所需的最小时间。由于缺省情况下会将 administrative-authority 设置为 platform，因此从平台获取该参数值，例如 i86pc。  
 或者，如果 administrative-authority 设置为 smf，则从 SMF 电源服务提供的定义中获取该值。安装时，该值未定义。如果您选择修改此属性，请使用与系统工作负荷或应用程序的需求相适应的值。  
 适中的值（例如几秒）允许将平台上的硬件组件或子系统置于响应较慢的不活动状态。较大的值（例如 30 秒到几分钟）允许使用诸如挂起到 RAM 等技术将整个系统暂停。
- suspend-enable – 缺省情况下，不允许任何运行 Oracle Solaris 的系统尝试暂停操作。将该属性设置为 true 可允许尝试暂停操作。administrative-authority 的值对此属性没有任何影响。
- platform-disabled – 当 platform-disabled 设置为 true 时，平台已禁用电源管理。设置为 false (缺省值) 时，电源管理由以上属性的值控制。

要显示电源管理状态的摘要，请发出以下命令：

```
$ /usr/sbin/poweradm show
Power management is enabled with the hardware platform as the authority:
```

```
time-to-full-capacity set to 250 microseconds
time-to-minimum-responsiveness set to 0 milliseconds
```

要显示电源管理属性，请发出以下命令：

```
$ /usr/sbin/poweradm list
active_config/time-to-full-capacity      current=250, platform=250
active_config/time-to-minimum-responsiveness current=0, platform=0
active_control/administrative-authority  current=platform, smf=platform
suspend/suspend-enable                   current=false
platform-disabled                         current=false
```

在此输出中，`active_control/administrative-authority` 表示配置源具有以下两个设置：

- `platform` – 电源管理配置来自于平台。这是缺省值。
- `smf` – 允许使用 `poweradm` 命令设置其他电源管理属性。

输出中的 `platform-disabled` 属性指示平台电源管理已启用：

```
platform-disabled          current=false
```

有关更多信息，请参见 [poweradm\(1M\)](#) 手册页。

#### 例 5-1 启用和禁用电源管理

如果之前在 `/etc/power.conf` 文件中启用了 S3 支持来暂停和恢复系统，可以使用类似下面的 `poweradm` 语法：

```
# poweradm set suspend-enable=true
```

缺省情况下，`suspend-enable` 属性设置为 `false`。

可使用以下语法禁用电源管理：

```
# poweradm set administrative-authority=none
```

禁用以下 SMF 电源管理服务不会禁用电源管理：

```
online      Sep_02   svc:/system/power:default
```

可使用以下语法禁用暂停和恢复：

```
# poweradm set suspend-enable=false
```

#### 例 5-2 设置和显示电源管理参数

以下示例显示如何将 `time-to-full-capacity` 设置为 300 微秒以及将 `time-to-minimum-responsiveness` 设置为 500 毫秒。最后，向 Oracle Solaris 实例通知新值。

```
# poweradm set time-to-full-capacity=300
```

```
# poweradm set time-to-minimum-responsiveness=500
# poweradm set administrative-authority=smf
```

以下命令显示当前的 time-to-full-capacity 值。

```
# poweradm get time-to-full-capacity
300
```

以下命令检索平台设置的 time-to-full-capacity 值。

```
# poweradm get -a platform time-to-full-capacity
```

请注意，只有将 administrative-authority 设置为 platform，则该值才会与当前值相同。有关更多信息，请参见以上 administrative-authority 属性说明。

## ▼ 如何在维护模式下从电源服务中恢复

如果在设置 time-to-full-capacity 和 time-to-minimum-responsiveness 之前将 administrative-authority 设置为 smf，则服务将进入维护模式。要从此状态恢复，请参见以下任务。

1. 成为管理员。  
请参见《[在 Oracle Solaris 11.2 中确保用户和进程的安全](#)》中的“使用所指定的管理权限”。
2. 将 administrative-authority 设置为 none。  

```
# poweradm set administrative-authority=none
```
3. 将 time-to-full-capacity 和 time-to-minimum-responsiveness 设置为所需的值。  

```
# poweradm set time-to-full-capacity=value
# poweradm set time-to-minimum-responsiveness=value
```
4. 清除服务。  

```
# svcadm clear power
```
5. 将 administrative-authority 设置为 smf。  

```
# poweradm set administrative-authority=smf
```

# 索引

---

## A

- 安全性, 80, 86
- at 命令, 82, 86, 87
  - 列出 at 作业, 85
  - 发送电子邮件确认, 83, 84
  - 控制访问, 71, 86, 86, 87
  - 概述, 71, 72, 82
  - 自动调度, 75
  - 错误消息, 87
- at 作业文件, 82, 86
  - 位置, 72
  - 创建, 83, 84
  - 删除, 86
  - 提交, 83
  - 显示, 85
  - 说明, 72
- at.deny 文件, 71, 86, 87
- atjobs 目录, 75

## B

- 编辑
  - crontab 文件, 75, 76, 76
- 波特率
  - 如何使用 eeprom 命令设置, 91
  - 如何在 ttymon 终端上设置, 90

## C

- 程序
  - 磁盘相关性, 52
- 创建
  - at 作业, 83
  - crontab 文件, 75, 76, 76
- 磁盘空间
  - 显示相关信息

- df 命令, 50
- 挂载点, 51
- 磁盘驱动器
  - 显示相关信息
  - 可用磁盘空间, 50
  - 查找并删除旧文件/非活动文件, 76
- 错误消息
  - at 命令, 87
  - crontab 命令, 82
- 重复性系统任务
  - 调度, 80
- 重新启动进程, 26
- CPU (central processing unit, 中央处理器)
  - 使用率高的进程, 39
  - 显示信息
    - 时间使用情况, 25, 39
- cron 守护进程, 72, 74
- cron.allow 文件, 79, 80, 81
- cron.deny 文件, 79, 80, 80
- crontab 命令, 71, 72, 80
  - cron 守护进程和, 74
  - 编辑 crontab 文件, 75, 76
  - 显示 crontab 文件, 77, 77, 77
  - 删除 crontab 文件, 78, 79
  - 使用的文件, 74, 74
  - 控制访问, 71, 79, 79, 79, 79, 80, 80, 80, 80, 81, 81
  - 每日任务, 72
  - 调度, 74
  - 退出而不保存更改, 76
  - 错误消息, 82
- crontab 文件
  - 位置, 74
  - 创建和编辑, 73, 75, 76, 76, 76, 76
  - 删除, 78, 78, 79
  - 拒绝访问, 80

- 显示, 77, 77
  - 缺省, 74
  - 语法, 75, 75
  - 说明, 74, 75
  - 验证, 77
- D**
- 地址空间映射
    - 显示, 26
  - 电源服务
    - 管理, 91
    - 问题的故障排除, 94
  - 调度, 72
    - 参见 crontab 命令、at 命令
    - 一次性系统任务, 72, 82
    - 重复性系统任务, 72, 73
  - 调度类, 32
    - 优先级和, 32, 36
    - 指定, 36
    - 显示信息, 33, 33
    - 显示相关信息, 25
    - 更改, 37
    - 更改优先级, 36, 38
  - 对进程进行故障排除, 39, 39
  - df 命令, 50, 50
    - k 选项 (千字节), 50
    - 概述, 50
    - 示例, 50
  - dispadmin 命令
    - 概述, 35
- E**
- EEPROM 命令
    - 用于在 ttymon 终端上设置波特率, 91
    - /etc/cron.d/at.deny 文件, 86, 87
    - /etc/cron.d/cron.allow 文件, 79, 80, 81
    - /etc/cron.d/cron.deny 文件, 79, 80
- F**
- 分时进程
    - 优先级
      - 更改, 36, 37, 38
- 概述, 32
  - 范围, 32
  - 更改调度参数, 36
  - fcntl 信息, 26, 26, 26, 28
  - fsck 命令, 72
  - fstat 信息, 26, 26, 26, 28
- G**
- 跟踪标志, 26
  - 更改
    - crontab 文件, 75
    - 优先级, 36, 38
    - 分时进程, 38
    - 更改, 37
    - 日期和时间, 20
    - 系统信息, 20
    - 系统标识, 21
    - 调度类, 37
  - 共享内存
    - 进程虚拟内存, 43
- H**
- 核心文件
    - 自动删除, 83
- I**
- iostat 命令, 48, 48
- J**
- 进程
    - nice 数值, 25, 37, 38, 39
    - proc 工具命令, 26
    - 中止, 26, 29
    - 优先级, 38
      - 指定, 36, 36
      - 显示信息, 33
      - 显示相关信息, 25
      - 更改, 36, 36, 37, 38, 38
      - 概述, 32, 38
      - 用户模式优先级, 32
      - 调度类和, 32, 36

- 进程类的全局优先级, 32, 33
- 信号操作, 26
- 失控, 39
- 已定义, 42
- 库链接到, 26, 26
- 应用程序线程和, 42
- 当前工作目录, 26, 26, 28
- 打开文件的 `fstat` 和 `fcntl` 信息, 26, 26, 26, 28
- 控制, 29
- 故障排除, 39, 39
- 显示地址空间映射, 26, 26
- 显示相关信息, 28
- 暂时停止, 26
- 术语, 42, 43
- 栈跟踪, 26
- 树, 26, 26, 28
- 用于管理的命令, 24
- 结构, 25, 43
- 调度类, 32, 33, 36
- 跟踪标志, 26, 26
- 重新启动, 26
- 进程类的全局优先级
  - 已定义, 32
  - 显示, 33
- 进程文件系统 (PROCFS), 26

## K

- 控制
  - 对 `at` 命令的访问, 71, 86, 87
  - 对 `crontab` 命令的访问, 80, 81
  - 进程, 29
- 控制台终端
  - 设置波特率, 91
- `klwp` 结构, 43
- `kthread` 结构, 43

## L

- 列出
  - `at` 作业, 84
  - 所执行的进程, 27
  - 进程, 27
- LWP (lightweight process, 轻量级进程)

- 已定义, 42
- 显示相关信息, 26
- 结构, 43
- 进程和, 42, 43

## M

- 每日消息 (message of the day, MOTD) 功能, 21
- 每月任务
  - 使用 `crontab` 调度, 72
- 目录
  - 进程的当前工作目录, 26, 26
- `motd` 文件, 21

## N

- 内存
  - 共享的进程虚拟内存, 43
  - 显示相关信息, 12
  - 虚拟进程, 43
  - 进程结构和, 43
- 内核线程
  - 结构, 25, 43
  - 调度和, 25
- `nice` 命令, 37, 38, 39
- `nice` 数值, 25, 38

## P

- `perf` 文件, 67
- `pfiles` 命令, 26, 26, 28
- `pflags` 命令, 26, 26
- `pkill` 命令, 26, 29
- `pldd` 命令, 26, 26
- `pmap` 命令, 26, 26
- `prionctl` 命令
  - 语法, 33
  - 概述, 35
  - 语法, 34
- `proc` 工具命令, 26
- `proc` 结构, 25, 43
- `/proc` 目录, 26
- PROCFS (process file system, 进程文件系统), 26
- `prtconf` 命令, 12

- 显示系统的产品名称, 11
  - ps 命令, 25, 27
    - 显示调度类的相关信息, 25
    - 显示全局优先级, 33
    - 显示有关进程的完整信息, 27
    - 报告的字段, 25
    - 显示有关调度类的信息, 39
    - 概述, 25
  - psig 命令, 26, 26
  - psrinfo 命令选项用于识别芯片多线程功能, 18
  - pstack 命令, 26, 26
  - ptime 命令, 26
  - ptree 命令, 26, 26, 28
  - pwait 命令, 26
  - pwdx 命令, 26, 26, 28
- R**
- 日志文件
    - 自动删除, 76
- S**
- 删除
    - at 作业, 86
    - crontab 文件, 78, 79
    - 日志文件, 76
    - 旧文件/非活动文件, 72
  - 删除 crontab 文件, 78
  - 失控进程, 39
  - 时间
    - CPU 使用量, 25, 39
    - 累积大量 CPU 时间的进程, 39
  - 实时进程
    - 更改类, 37
  - 使用 crontab 调度每日任务, 72
  - sa1 命令, 67
  - sa2 命令, 67, 68
  - sadc 命令
    - 引导时运行, 67
    - 自动系统数据收集, 67, 67
  - sadd 文件, 68
  - sar 命令, 52, 68
    - 所有选项, 68, 69
    - 概述, 51, 68
  - svcadm enable system/sar:default 命令, 67
  - sys crontab, 68
- U**
- user 结构, 43
  - /usr/proc/bin 目录, 26, 26
- V**
- /var/adm/sa/sadd 文件, 68
  - /var/spool/cron/atjobs 目录, 71, 72, 75, 75
  - /var/spool/cron/crontabs 目录, 74, 74
  - /var/spool/cron/crontabs/root 文件, 73
  - /var/spool/cron/crontabs/sys crontab, 68
  - vmstat 命令
    - 概述, 45
- W**
- 文件
    - fstat 和 fcntl 信息显示, 26, 26, 26, 28
    - 检查访问操作, 52, 52
  - 文件系统
    - 挂载点, 51
    - 磁盘空间使用情况, 50
- X**
- 系统的产品名称
    - 使用 prtconf 命令显示, 11
  - 系统活动
    - 手动收集数据, 68
    - 自动收集数据, 67, 67
    - 跟踪活动的列表, 44
  - 系统控制台
    - 管理, 89
    - 使用 SMF 服务, 89
  - 系统任务, 72
    - 参见 crontab 命令、at 命令
  - 调度
    - 一次性任务, 72, 82
    - 自动, 71
    - 重复性任务, 72, 73
  - 系统性能 见 性能

## 系统资源

- 概述, 42
- 监视, 86, 86

## 显示

- at 作业, 85
- crontab 文件, 77, 77
- LWP 信息, 26
- 主机 ID, 10
- 产品名称信息
  - prtconf, 11
- 优先级信息, 25, 33
- 体系结构类型, 10
- 发行版信息, 10
- 地址空间映射, 26
- 处理器类型, 11
- 所执行的进程的信息, 27
- 扩展磁盘统计信息, 49
- 日期和时间, 10
- 有关进程的信息, 27
- 物理处理器类型
  - psrinfo 命令, 18
- 磁盘使用率信息, 48
- 磁盘空间统计信息, 50
- 系统信息
  - 命令, 9
- 系统已安装内存, 12
- 系统活动信息, 51, 68
- 虚拟处理器类型, 19
- 设备的属性值, 12
- 诊断信息, 16
- 调度类信息, 25, 33, 33
- 进程信息, 26, 26, 28
- 链接的库, 26, 26

## 新增功能

- svcadm enable system/sar:default 命令, 67

## 性能

- 使用 Ops Center 监视, 41
- 手动收集活动数据和, 52, 68
- 报告, 51
- 文件访问和, 52, 52
- 监视工具, 44
- 自动收集活动数据, 67, 67
- 跟踪的活动, 44
- 进程管理和, 26, 38, 42

## Y

## 验证

- crontab 文件, 77
- 应用程序线程, 42, 43
- 用户进程
  - 优先级, 32
  - 更改优先级, 37, 38
- 用户模式优先级, 32
- 优先级 (进程)
  - 全局
    - 已定义, 32
    - 显示, 33
  - 指定, 36, 36
  - 显示相关信息, 25, 33
  - 更改, 36, 38
  - 更改分时进程, 36, 37, 38
  - 概述, 32, 38
  - 用户模式优先级, 32
  - 调度类和, 36

## Z

- 在 ttymon 控制台终端上设置波特率, 90
- 暂停进程, 26
- 中止进程, 26, 29
- 终端
  - 进程控制, 25
- 终端设备
  - 管理, 89
    - 使用 SMF 服务, 89
    - 设置登录服务, 90
- 自动系统活动
  - 报告, 67, 68
  - 数据收集, 67, 67
- 自动执行例行任务, 71
- 自动执行系统任务, 71
  - 单个任务, 82, 86, 87
  - 重复性任务, 80, 81

