

在 Oracle® Solaris 11.2 中排除系统管理问题

ORACLE®

文件号码 E53853-02
2014 年 9 月

版权所有 © 1998, 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	5
1 系统崩溃故障排除	7
Oracle Solaris 11.2 中的故障转储文件新增功能	7
重新构造了故障转储文件	7
故障转储文件保存在 /var/crash 目录中	7
关于系统崩溃	7
系统故障转储文件	8
重新构造的文件	8
dumpadm 和 savecore 命令	9
针对故障转储配置系统	9
显示当前的故障转储配置	10
修改故障转储配置	10
禁用或启用故障转储的保存	12
系统崩溃后进行故障排除	13
系统崩溃时应执行的操作	13
检查故障转储信息	13
系统崩溃故障排除核对表	15
在故障转储目录已满时保存数据	16
使用 Oracle Enterprise Manager Ops Center 管理意外事件	16
2 系统挂起或重新引导失败时进行故障排除	19
重新引导失败时应执行的操作	19
忘记 Root 口令或无法引导系统时应执行的操作	20
系统挂起时应执行的操作	20
3 文件系统问题故障排除	21
文件系统占满时应执行的操作	21
由于创建了大文件或目录导致文件系统占满	21
由于系统内存不足导致 TMPFS 文件系统变满	21

复制或恢复后文件 ACL 丢失时应执行的操作	22
文件访问问题故障排除	22
解决搜索路径的问题 (Command not found)	22
更改文件和组的所有权	24
解决文件访问问题	24
识别网络访问问题	24
4 通过使用核心文件为可能的进程故障做准备	25
关于进程故障和核心文件	25
核心文件的创建参数	25
管理核心文件规范	27
显示当前的核心转储配置	27
设置核心文件名称模式	27
启用文件路径	28
启用 setuid 程序以生成核心文件	29
恢复为缺省核心文件设置	29
更正过时的核心文件参数	30
在进程失败后检查核心文件	30
5 管理系统日志和消息	31
使用 rsyslogd 扩展的系统日志记录	31
▼ 安装和启用 rsyslog	31
管理系统消息	32
查看系统消息	32
系统日志轮转	34
定制系统消息日志	35
启用远程控制台消息传送	37
索引	41

使用本文档

- 概述 - 介绍如何排除 SPARC 和 x86 平台上的问题
- 目标读者 - 使用 Oracle Solaris 11 发行版的系统管理员
- 必备知识 - 具有管理 UNIX 系统的经验

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> 上提供有关本文档的反馈。

◆◆◆ 第 1 章

系统崩溃故障排除

本章介绍了如何在 Oracle Solaris OS 中为系统崩溃做准备以及进行故障排除。
本章涵盖以下内容：

- [“Oracle Solaris 11.2 中的故障转储文件新增功能” \[7\]](#)
- [“关于系统崩溃” \[7\]](#)
- [“针对故障转储配置系统” \[9\]](#)

Oracle Solaris 11.2 中的故障转储文件新增功能

本节介绍了本 Oracle Solaris 发行版中在管理故障转储文件方面的更改。

重新构造了故障转储文件

在本发行版中，内核故障转储文件根据其内容分为多个新文件，以便更快地执行初始分析并在配置中启用更高粒度。有关信息，请参见[“重新构造的文件” \[8\]](#)。

故障转储文件保存在 `/var/crash` 目录中

在 Oracle Solaris 11 操作系统中，故障转储文件保存在 `/var/crash` 目录中。此更改最初是在 Oracle Solaris 10 1/13 发行版中引入的，现在包括在所有 Oracle Solaris 11 发行版中。

关于系统崩溃

系统崩溃可能是由于存在硬件故障、I/O 问题和软件错误而引起的。如果系统崩溃，则会在控制台中显示一条错误消息，然后向转储设备中写入物理内存的副本。系统随后进

行自动重新引导。重新引导系统时，将执行 `savecore` 命令，以从转储设备中检索数据，并将保存的故障转储文件写入 `savecore` 目录。这些保存的文件提供了有助于诊断问题的非常有价值的信息。

注 - 故障崩溃一词是指此进程的整体结果，包括故障崩溃文件集、这些文件的位置以及如何组织这些文件和设置其格式。

系统故障转储文件

`savecore` 命令在系统崩溃后会自动运行，以从转储设备检索故障转储信息并将该信息写入一组文件。然后，可以在同一系统或其他系统上调用 `savecore` 命令来扩展压缩的故障转储文件。

注 - 故障转储文件有时容易与核心转储文件混淆，后者是在应用程序异常终止时写入的用户应用程序的映像。

故障转储文件保存在预先确定的目录中，该目录缺省为 `/var/crash/`。在先前的发行版中，除非手动使系统将物理内存的映像保存到故障转储文件中，否则系统重新引导时会覆盖故障转储文件。现在，缺省情况下便可保存故障转储文件。

重新构造的文件

在 Oracle Solaris 11.2 发行版中，重新构造了内核故障转储文件。这些文件根据其内容分为多个新文件，以便更快地执行初始分析并在配置中启用更高粒度。这些文件更便于访问和研究。

内核故障转储以前存储在以下文件中：

- `vmdump.N`
- `unix.N`
- `vmcore.N`

`vmdump.N` 和 `vmcore.N` 分别以压缩和未压缩的格式存储内核页元数据和数据。

从 Oracle Solaris 11.2 发行版开始，故障转储信息写入一组 `vmdump-section.n` 文件。`section` 的值是包含特定种类转储信息的文件段的名称。`n` 值为正数，每次运行 `savecore` 来复制故障转储以及在转储设备上发现新的故障转储时，该值将递增。可能的文件包括：

<code>vmdump-proc.N</code>	包含压缩的进程页的转储文件
<code>vmdump-zfs.N</code>	包含压缩的 ZFS 元数据的转储文件

vmdump-other.N 包含其他页的转储文件

有关详细信息，请参见 `dumpadm(1M)` 和 `savecore(1M)` 手册页。

dumpadm 和 savecore 命令

`dumpadm` 和 `savecore` 实用程序按如下所示配置和管理故障转储的创建：

在系统启动过程中，`svc:/system/dumpadm:default` 服务调用 `dumpadm` 命令以配置故障转储参数。它通过 `/dev/dump` 接口来初始化转储设备和转储内容。

注 - 在 Oracle Solaris 11.2 发行版中，`dumpadm` 命令具有新的选项，可用于指定转储内容、输出磁盘空间估计值以及生成可解析的输出。请参见“[修改故障转储配置](#)” [10]。

完成转储配置后，`savecore` 脚本会查找故障转储文件目录的位置。然后，会调用 `savecore` 来检查故障转储，并检查故障转储目录中 `minfree` 文件的内容。缺省情况下，会保存 `savecore` 命令生成的系统故障转储文件。

转储数据以压缩格式存储在转储设备上。内核故障转储映像的大小可以为 4 GB 或更多。压缩数据意味着转储速度更快且转储设备需要的磁盘空间更少。

专用转储设备（而不是交换区域）是转储配置的一部分时，对故障转储文件的保存是在后台运行的。正在引导的系统不会等待 `savecore` 命令完成，便转到下一步。在较大的内存系统中，可在 `savecore` 完成前使用系统。

通过 `savecore -L` 命令，管理员可以获取当前正在运行的 Oracle Solaris OS 的故障转储。此命令用于通过在某些发生故障的状态下（例如瞬态性能问题或服务故障）捕获内存快照，对正在运行的系统进行故障排除。如果系统已启动且您仍可以运行某些命令，则可以执行 `savecore -L` 命令将系统快照保存到转储设备，然后立即将故障转储文件写出到 `savecore` 目录。由于系统仍在运行，因此仅当您已配置了专用转储设备时才能使用 `savecore -L` 命令。

有关更多信息，请参见“[修改故障转储配置](#)” [10]、`dumpadm(1M)` 和 `savecore(1M)` 手册页。

针对故障转储配置系统

本节介绍了用于管理系统的故障转储过程的任务。

在处理系统故障转储信息时，请记住以下几点内容：

- 必须成为 `root` 角色才可以访问和管理系统故障转储信息。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”。

- 请勿禁用系统上用于保存系统故障转储文件的选项。系统故障转储文件提供了用于确定系统崩溃原因的非常有价值的方法。
- 将专用的 ZFS 卷用于交换区域和转储区域。安装后，可能需要调整交换设备和转储设备的大小或者可能需要重新创建交换卷和转储卷。有关说明，请参见《在 Oracle Solaris 11.2 中管理 ZFS 文件系统》中的“管理 ZFS 交换和转储设备”。

要在发生系统崩溃之前定制系统的故障转储过程，请参见以下内容：

- “显示当前的故障转储配置” [10]
- “修改故障转储配置” [10]
- “禁用或启用故障转储的保存” [12]

显示当前的故障转储配置

要显示当前的故障转储配置，需要成为 root 角色并发出不带参数的 dumpadm 命令。

```
# dumpadm
Dump content: kernel pages
Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash
Savecore enabled: yes
Save compressed: on
```

此示例显示以下配置：

- 转储内容是内核内存页面。
- 内核内存将在专用转储设备 /dev/zvol/dsk/rpool/dump 上进行转储。
- 系统故障转储文件将保存在 /var/crash 目录中。
- 已启用故障转储文件的保存。
- 以压缩格式保存故障转储。

修改故障转储配置

要修改故障转储配置，需要成为 root 角色并使用 dumpadm 命令。

dumpadm 命令的语法如下：

```
# /usr/sbin/dumpadm [-nuy] [-c content-type] [-d dump-device] [-m mink | minm | min%]
[-s savecore-dir] [-r root-dir] [-z on | off]
```

-c content-type 指定要转储的数据类型。对于 Oracle Solaris 11.2 发行版，此选项的值已更改。使用 kernel 仅转储内核内存页，all 转储所有内存页，curproc 转储内核内存以及在发生崩溃时其线程正在执行的进程的内存页，allproc 转储内核内存页和所有进程页，zfs 转储用于存储 ZFS 元数据的内核页。缺省转储内容是内核内存。

请参见 -c 选项的以下示例：

```
# dumpadm -c kernel
# dumpadm -c +zfs
# dumpadm -c -zfs
# dumpadm -c curproc+zfs
```

- d *dump-device* 指定在系统崩溃时临时存储转储数据的设备。主转储设备即为缺省转储设备。当转储设备不是交换区域时，`savecore` 将在后台运行，这样可以加快引导过程进行的速度。
- e 输出存储压缩的故障转储所需的磁盘空间估计值。该值是使用当前配置和当前运行的系统计算的。
- m *mink* | *minm* | *min%* 通过在当前的 `savecore` 目录中创建 `minfree` 文件，指定用于保存故障转储文件的最小空闲磁盘空间。可以 KB (*mink*)、MB (*minm*) 或文件系统大小百分比 (*min%*) 的形式指定此参数。如果未配置最小空闲空间，则缺省值为 1 MB。

`savecore` 命令会在写入故障转储文件之前访问此文件。如果写入故障转储文件会由于其大小而减少空闲空间量并使其低于 `minfree` 阈值，则不写入转储文件，并记录一条错误消息。有关从此情况进行恢复的信息，请参见“[在故障转储目录已满时保存数据](#)” [16]。
- n 指定重新引导系统时不应运行 `savecore`。不推荐使用此转储配置。如果已将系统故障转储信息写入交换设备并且未启用 `savecore`，则系统开始交换时将覆盖故障转储信息。
- p 生成计算机可解析的输出。
- s *savecore-dir* 指定用于存储故障转储文件的备用目录。在 Oracle Solaris 11 中，缺省目录为 `/var/crash`。
- u 强制更新基于 `/etc/dumpadm.conf` 文件内容的内核转储配置。
- y 修改转储配置以在重新引导时自动执行 `savecore` 命令，即此转储设置的缺省值。
- z *on* | *off* 修改转储配置以控制重新引导时 `savecore` 命令的操作。`on` 设置允许以压缩格式保存核心文件。`off` 设置会自动解压缩故障转储文件。由于故障转储文件可能非常大，而以压缩格式保存这些文件将需要较少的文件系统空间，因此缺省值为 `on`。

例 1-1 修改故障转储配置

在此示例中，所有内存都将转储到专用转储设备 `/dev/zvol/dsk/rpool/dump`，在保存故障转储文件后可用的最小空闲空间必须为 10% 的文件系统空间。

```
# dumpadm
  Dump content: kernel pages
  Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash
  Savecore enabled: yes
  Save compressed: on

# dumpadm -c all -d /dev/zvol/dsk/rpool/dump -m 10%
  Dump content: all pages
  Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash (minfree = 5697105KB)
  Savecore enabled: yes
  Save compressed: on
```

禁用或启用故障转储的保存



注意 - Oracle Solaris 强烈建议您不要禁用故障转储的保存。故障转储提供了用于确定系统崩溃原因的非常有价值的方法。

作为 root 角色，您可以禁用或启用故障转储的保存：

```
# dumpadm -n | -y
```

-n 禁用故障转储的保存

-y 启用故障转储的保存

例 1-2 禁用故障转储的保存

此示例说明如何在系统中禁用故障转储的保存。

```
# Dump content: all pages
  Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash (minfree = 5697105KB)
  Savecore enabled: no
  Save compressed: on
```

例 1-3 启用故障转储的保存

此示例说明如何在系统中启用故障转储的保存。

```
# dumpadm -y
  Dump content: all pages
  Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash (minfree = 5697105KB)
  Savecore enabled: yes
```

Save compressed: on

系统崩溃后进行故障排除

如果 Oracle Solaris 系统崩溃，您将需要为您的服务提供商提供尽可能多的信息，包括常规系统信息以及故障转储文件中提供的特定信息。

要对现有故障转储进行故障排除，请参见以下内容：

- [“系统崩溃时应执行的操作” \[13\]](#)
- [“检查故障转储信息” \[13\]](#)
- [“系统崩溃故障排除核对表” \[15\]](#)
- [“在故障转储目录已满时保存数据” \[16\]](#)

系统崩溃时应执行的操作

以下列表介绍了在系统崩溃时需要注意的最重要信息：

1. 记录系统控制台消息。
 - 如果系统崩溃，则使其重新运行可能是最紧迫的事情。但是，在重新引导系统之前，请先检查控制台屏幕上的消息。这些消息可能有助于了解导致崩溃的原因。即使系统自动重新引导且控制台消息已从屏幕中消失，您也可能能够通过查看系统错误日志（/var/adm/messages 文件）来检查这些消息。有关查看系统错误日志文件的更多信息，请参见[如何查看系统消息 \[33\]](#)。
 - 如果您频繁遇到崩溃且无法确定原因，请从系统控制台或 /var/adm/messages 文件中收集可以收集的任何信息，并将其准备就绪以供客户服务代表检查。有关要为服务提供商收集的故障排除信息的完整列表，请参见[“系统崩溃故障排除核对表” \[15\]](#)。
2. 查看系统崩溃后是否生成了系统故障转储。



注意 - 在将重要的系统故障转储信息发送给客户服务代表之前，不要删除这些信息。

3. 如果系统在崩溃后无法引导，请参见 [《引导和关闭 Oracle Solaris 11.2 系统》](#) 中的[“为进行恢复而关闭和引导系统”](#)了解进一步的说明。

检查故障转储信息

使用 mdb 实用程序可以检查控制结构、活动表、正常运行或崩溃的系统内核的内存映像，以及有关内核运行的其他信息，如以下过程中所述。

注 - 以下过程仅提供了如何使用 `mdb` 实用程序的有限示例。充分发挥 `mdb` 实用程序的潜能需要对内核有详细的了解，这不在本手册的讨论范围之内。有关使用此实用程序的详细信息，请参见 [mdb\(1\)](#) 手册页。

▼ 如何检查故障转储信息

1. 成为 `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. 转至保存故障转储信息的目录。

例如：

```
# cd /var/crash
```

如果您不确定故障转储的位置，请使用 `dumpadm` 命令来确定系统中配置为存储内核故障转储文件的位置。例如：

```
# /usr/sbin/dumpadm
Dump content: kernel pages
Dump device: /dev/zvol/dsk/rpool/dump (dedicated)
Savecore directory: /var/crash
Savecore enabled: yes
Save compressed: on
```

3. 通过使用模块调试器实用程序 (`mdb`) 来检查故障转储。

```
# /usr/bin/mdb [-k] crashdump-file
```

`-k` 通过假定文件为操作系统故障转储文件来指定内核调试模式。

`crashdump-file` 指定操作系统故障转储文件。

例如：

```
# /usr/bin/mdb -K vmcore.0
```

也可以通过以下方式指定此命令：

```
# /usr/bin/mdb -k 0
```

4. 显示系统崩溃状态。

```
> ::status
.
.
.
> ::system
```

·
·
·

要在检查内核故障转储时使用 `::system dcmd` 命令，核心文件必须为内核故障转储文件，且必须在启动 `mdb` 实用程序时指定了 `-k` 选项。

5. 退出 `mdb` 实用程序。

```
> $quit
```

例 1-4 检查故障转储信息

此示例显示 `mdb` 实用程序的样例输出，其中包括系统信息，并列在此系统的 `/etc/system` 文件中设置的可调参数。

```
# cd /var/crash
# /usr/bin/mdb -k unix.0
Loading modules: [ unix krtld genunix ip nfs ipc ptm ]
> ::status
debugging crash dump /dev/mem (64-bit) from ozlo
operating system: 5.10 Generic sun4v
> ::system
set ufs_ninode=0x9c40 [0t40000]
set ncsz=0x4e20 [0t20000]
set pt_cnt=0x400 [0t1024]
> $q
```

系统崩溃故障排除核对表

回答以下核对表中的问题可以帮助隔离问题以及准备咨询您的支持提供商。

项目	数据
系统故障转储是否可用？	
确定操作系统发行版以及相应软件应用程序的发行版级别。	
确定系统硬件。	
包括 SPARC 系统的 <code>prtdiag</code> 输出。包括其他系统的资源管理器输出。	
是否安装了修补程序？如果已安装，请包括 <code>showrev -p</code> 输出。	
问题是否可重现？	
这一点很重要，因为在调试很难的问题时，可重现的测试案例是必不可少的。通过重现问题，服务提供商可以使用特殊设备构造内核，以触发、诊断和更正错误。	

项目	数据
系统中是否有任何第三方驱动程序？	具有相同特权的多个驱动程序在内核所在的同一地址空间中运行，如果这些驱动程序存在已知问题，则会导致系统崩溃。
系统在崩溃前正在执行什么操作？	如果系统在执行异常的操作，例如运行新的负荷测试或遇到特别高的负荷，则可能导致系统崩溃。
在系统崩溃前是否有任何异常的控制台消息？	有时，系统会在实际崩溃前显示故障信号，此信息通常很有用。
是否向 <code>/etc/system</code> 文件中添加了任何参数？	有时，调优参数（如增大共享内存段，以使系统尝试分配比实际拥有内存更多的内存）会导致系统崩溃。
问题是在最近开始的吗？	如果是这样，问题是否在对系统的更改同时出现？例如新的驱动程序、新软件、不同工作负荷、CPU 升级或内存升级。

在故障转储目录已满时保存数据

如果系统崩溃，但 `savecore` 目录中无剩余空间，并且您需要保存一些关键的系统故障转储信息，则使用以下方法之一。

系统重新引导后，以 `root` 角色身份登录。从 `savecore` 目录中删除已经发送给服务提供商的现有故障转储文件。

注 - `savecore` 目录通常为 `/var/crash`。

或者，系统重新引导后，以 `root` 角色身份登录。手动运行 `savecore` 命令并指定拥有足够磁盘空间的备用目录：

```
# savecore directory
```

使用 Oracle Enterprise Manager Ops Center 管理意外事件

如果需要管理数据中心内物理和虚拟操作系统、服务器和存储设备的意外事件，而不只是监视各个系统内的意外事件，则可以使用 Oracle Enterprise Manager Ops Center 中的可用综合系统管理解决方案。

使用 Enterprise Manager Ops Center，您可以设置警报以便在数据中心的某些部分未按预期运行时通知您、管理这些意外事件报告以及尝试进行修复。

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 库)。

系统挂起或重新引导失败时进行故障排除

本章介绍了系统挂起或者您重新引导系统遇到问题时可以执行的操作。

本章包含以下信息：

- “重新引导失败时应执行的操作” [19]
- “忘记 Root 口令或无法引导系统时应执行的操作” [20]
- “系统挂起时应执行的操作” [20]

重新引导失败时应执行的操作

如果系统未完全重新引导，或者重新引导后再次崩溃，则可能存在使系统无法成功引导的软件或硬件问题。

系统未引导的原因	解决该问题的办法
系统找不到 <code>/platform/`uname -m`/kernel/sparcv9/unix</code> 。	您可能需要在基于 SPARC 的系统上更改 PROM 中的 <code>boot-device</code> 设置。有关更改缺省引导设备的信息，请参见《 引导和关闭 Oracle Solaris 11.2 系统 》中的“ 显示和设置引导属性 ”。
Oracle Solaris 引导归档文件已损坏。或者，SMF 引导归档文件服务失败。如果运行 <code>svcs -x</code> 命令，则会显示一条错误消息。	创建另一个引导环境，该环境是主引导环境的备份。当主引导环境无法引导时，可从此备份引导环境引导。也可以从 Live CD 或 USB 介质引导。
<code>/etc/passwd</code> 文件中有一个无效条目。	有关从无效的 <code>passwd</code> 文件进行恢复的信息，请参见《 引导和关闭 Oracle Solaris 11.2 系统 》中的“ 如何从介质进行引导以解决 root 口令未知问题 ”。
x86 引导装载程序 (GRUB) 已损坏。或者，GRUB 菜单缺失或受损。	有关从损坏的 x86 引导装载程序或者缺失或损坏的 GRUB 菜单进行恢复的信息，请参见《 引导和关闭 Oracle Solaris 11.2 系统 》中的“ 如何从介质进行引导以解决阻止系统引导的 GRUB 配置问题 ”。
磁盘或其他设备存在硬件问题。	检查硬件连接： <ul style="list-style-type: none">■ 确保已插入设备。■ 确保所有交换机都已正确设置。■ 查看所有连接器和电缆，包括以太网电缆。■ 如果所有这些步骤都失败，请关闭系统电源，等待 10 到 20 秒钟，然后再次打开电源。

如果采用上述建议仍无法解决问题，请与当地服务提供商联系。

忘记 Root 口令或无法引导系统时应执行的操作

如果您忘记了 root 口令，或遇到了阻止系统进行引导的其他问题，请执行以下操作：

- 停止系统。
- 按照《[引导和关闭 Oracle Solaris 11.2 系统](#)》中的“[如何从介质进行引导以解决 root 口令未知问题](#)”中的指示操作。
- 如果问题在于 root 口令，请将其从 /etc/shado 文件中删除。
- 重新引导系统。
- 登录并设置 root 口令。

系统挂起时应执行的操作

如果某些软件进程出现问题，系统可能会冻结或挂起，而不是完全崩溃。遵循以下步骤可以从挂起的系统中进行恢复。

1. 确定系统是否正在运行窗口环境并遵循以下建议。如果这些建议无法解决问题，请转到步骤 2。
 - 确保指针位于正在键入命令的窗口中。
 - 如果用户意外按下了可冻结屏幕的 Ctrl-s，请按下 Ctrl-q。Ctrl-s 仅会冻结窗口，而不是整个屏幕。如果一个窗口被冻结，请尝试使用其他窗口。
 - 如果可能，请从网络中的其他系统中远程登录。使用 pgrep 命令查找挂起的进程。如果窗口系统看似已挂起，请标识进程并将其中止。
2. 按 Ctrl-\ 组合键强制退出运行的程序并（可能）写出 core 文件。
3. 按 Ctrl-C 组合键中断可能正在运行的程序。
4. 远程登录并尝试确定和中止使系统挂起的进程。
5. 远程登录，成为 root 角色，然后重新引导系统。
6. 如果系统仍然无法响应，请强制进行故障转储并重新引导。有关强制故障转储并引导的信息，请参见《[引导和关闭 Oracle Solaris 11.2 系统](#)》中的“[强制实施系统故障转储和重新引导](#)”。
7. 如果系统仍然无法响应，请关闭电源，等待一分钟左右，然后重新打开电源。
8. 如果始终无法使系统响应，请联系当地的服务提供商获取帮助。

文件系统问题故障排除

本章介绍了如何修复文件系统问题，包括以下内容：

- “文件系统占满时应执行的操作” [21]
- “复制或恢复后文件 ACL 丢失时应执行的操作” [22]
- “文件访问问题故障排除” [22]

文件系统占满时应执行的操作

当根 (/) 文件系统或任何其他文件系统占满时，您将在控制台窗口中看到以下消息：

```
.... file system full
```

系统占满的原因可能有多种。以下各节介绍了从已满文件系统中进行恢复的几种方案。

由于创建了大量文件或目录导致文件系统占满

出现错误的原因	解决该问题的办法
有人意外地将文件或目录复制到错误位置。当应用程序崩溃并将大型 core 文件写入到文件系统时，也会发生这种情况。	登录并成为 root 角色，然后在特定文件系统中使用 <code>ls -tl</code> 命令识别新创建的大型文件，随后将其删除。

由于系统内存不足导致 TMPFS 文件系统变满

出现错误的原因	解决该问题的办法
如果 TMPFS 尝试写入的内容比允许量多或者有些当前进程使用了大量内存，则可能出现此情况。	有关从与 tmpfs 相关的错误消息进行恢复的信息，请参见 tmpfs(7FS) 手册页。

复制或恢复后文件 ACL 丢失时应执行的操作

出现错误的原因	解决该问题的办法
如果将具有 ACL 的文件或目录复制或恢复到 /tmp 目录，则会丢失 ACL 属性。/tmp 目录通常作为临时文件系统挂载，而临时文件系统不支持 ACL 等 UFS 文件系统属性。	将文件复制或恢复到 /var/tmp 目录中。

文件访问问题故障排除

用户经常由于无法访问以前可用的程序、文件或目录等问题，而向系统管理员寻求帮助。

当您遇到此类问题时，请检查是否是由以下三个方面的某一问题引起：

- 用户的搜索路径可能已更改，或者搜索路径中的目录顺序不正确。
- 文件或目录可能不具有正确的权限或所有权。
- 通过网络访问的系统配置可能已更改。

本章简要介绍如何识别上述每个方面的问题，并提出可能的解决方案。

解决搜索路径的问题 (Command not found)

"Command not found" 消息表示存在以下情况之一：

- 命令在系统中不可用。
- 命令目录不在搜索路径中。

要解决搜索路径问题，您需要知道存储命令的目录的路径名。

如果找到了该命令的错误版本，则一个包含同名命令的目录会出现在搜索路径中。在这种情况下，正确的目录可能稍后会出现在搜索路径中，或者根本不会出现。

使用 `echo $PATH` 命令可以显示当前的搜索路径。

使用 `type` 命令可以确定您所运行的命令是否是错误版本的命令。例如：

```
$ type acroread
acroread is /usr/bin/acroread
```

▼ 如何诊断和更正搜索路径问题

1. 显示当前搜索路径，以验证命令的目录是否不在路径中，或验证该路径是否有拼写错误。

```
$ echo $PATH
```

2. 请检查以下内容：

- 搜索路径是否正确？
- 该搜索路径是否在找到了该命令另一个版本的其他搜索路径之前列出？
- 命令是否位于其中一个搜索路径中？

如果需要更正路径，请转到步骤 3。否则，请转到步骤 4。

3. 将路径添加到相应文件，如下表所示。

Shell	文件	语法	备注
bash 和 ksh93	\$HOME/.profile	\$ PATH=\$HOME/bin:/sbin:/usr/local/bin ... \$ export PATH	用冒号分隔路径名。

4. 按以下方式激活新路径：

Shell	路径位置	激活路径的命令
bash 和 ksh93	.profile	. \$HOME/.profile
	.login	hostname\$ source \$HOME/.login

5. 验证新路径。

```
$ which command
```

例 3-1 诊断和更正搜索路径问题

此示例显示执行 type 命令后发现 mytool 可执行文件不在搜索路径中的任何目录内。

```
$ mytool
-bash: mytool: command not found
$ type mytool
-bash: type: mytool: not found
$ echo $PATH
/usr/bin:
$ vi $HOME/.profile
(Add appropriate command directory to the search path)
$ . $HOME/.profile
$ mytool
```

如果找不到某一命令，请在手册页中查看其目录路径。

更改文件和组的所有权

通常，文件和目录的所有权会因为有人以管理员的身份编辑文件而发生变化。在为新用户创建起始目录时，务必要使用该用户成为起始目录中的点 (.) 文件的所有者。如果用户不是 "." 的所有者，他们将无法在自己的起始目录中创建文件。

当组所有权发生变化或从 `/etc/group` 数据库中删除了某用户所属的组时，也可能发生访问问题。

有关如何更改存在访问问题的文件的权限或所有权的信息，请参见《[在 Oracle Solaris 11.2 中确保文件的安全和确认文件完整性](#)》中的第 1 章“控制对文件的访问”。

解决文件访问问题

当用户无法访问以前可以访问的文件或目录时，可能是这些文件或目录的权限或所有权已经更改。

识别网络访问问题

如果用户在使用 `rcp` 远程复制命令通过网络复制文件时遇到问题，则可能是通过设置权限对远程系统中的目录和文件进行了访问限制。另一个可能的原因是未将远程系统和本地系统配置为允许访问。

有关网络访问问题以及通过 AutoFS 访问系统的问题的信息，请参见《[在 Oracle Solaris 11.2 中管理网络文件系统](#)》中的“NFS 故障排除的策略”。

通过使用核心文件为可能的进程故障做准备

本章介绍了如何为进程出现故障时系统所生成的核心文件设置规范以及在出现故障后如何检查这些核心文件。

下面列出了本章所介绍的信息：

- [“关于进程故障和核心文件” \[25\]](#)
- [“核心文件的创建参数” \[25\]](#)
- [“管理核心文件规范” \[27\]](#)
- [“在进程失败后检查核心文件” \[30\]](#)

关于进程故障和核心文件

进程或应用程序异常终止时，系统自动生成一组文件。此过程可以描述为核心转储。创建的文件为核心文件。核心文件是进程终止时进程地址空间的内容的磁盘副本以及有关进程状态的其他信息。通常，在由于相应应用程序中的错误而导致进程终止后会生成核心文件。核心文件为您提供非常有价值的信息，用于诊断导致进程故障的问题。请参见[“核心文件的创建参数” \[25\]](#)。

作为您的持续性系统管理的一部分，可以使用 `coreadm` 命令控制核心文件的创建规范。例如，可以使用 `coreadm` 命令来配置系统，以便将所有进程核心文件都放在一个系统目录中，从而您可以更容易跟踪问题。请参见[“管理核心文件规范” \[27\]](#)。

进程异常终止时，您可以使用 `mdb` 等调试器或使用 `proc` 工具检查创建的核心文件。请参见[“在进程失败后检查核心文件” \[30\]](#)。

核心文件的创建参数

进程失败时，系统尝试为每个失败的进程创建至多两个核心文件，使用全局核心文件名称模式和每进程核心文件名称模式来创建每个核心文件名。`coreadm` 命令控制这些名称模式并指定核心文件的位置。本节介绍了一些文件路径和文件名参数。有关核心转储进程的完整说明，请参见 `core(4)` 手册页。有关 `coreadm` 选项的完整说明，请参见 `coreadm(1M)` 手册页。

可配置的核心文件路径

当进程异常终止时，缺省情况下会在当前目录中生成一个核心文件。如果启用了全局核心文件，则每个异常终止的进程可能会生成两个文件，一个在当前工作目录中，另一个在全局核心文件位置。使用的文件路径是可配置参数。

可以按如下所示独立启用或禁用两个可配置 core 文件路径：

- 每进程核心文件路径，缺省为 core，缺省情况下处于启用状态。如果启用，则每进程核心文件路径会导致在进程异常终止时生成 core 文件。每进程路径由新进程从其父进程处继承。
生成每进程核心文件时，该文件由具有所有者读/写权限的进程所有者所有。只有所有者用户可以查看此文件。
- 全局核心文件路径，缺省为 core，缺省情况下处于禁用状态。如果启用，则会使用全局核心文件路径生成内容与每进程核心文件相同的附加核心文件。
全局核心文件在生成时归 root 用户所有，只有 root 用户具有读/写权限。非特权用户不能查看此文件。

注 - 缺省情况下，setuid 进程不使用全局或每进程路径生成核心文件。

扩展的核心文件名

核心文件的名称包含具有关于失败进程的信息的字段。有关核心文件名字段的完整说明，请参见 `coreadm(1M)` 手册页。本节重点介绍全局变量。

如果启用了全局 core 文件目录，则可以使用下表中介绍的变量来区分各个 core 文件。

%d	可执行文件目录名，最多包含 MAXPATHLEN 个字符
%f	可执行文件名，最多包含 MAXCOMLEN 个字符
%g	有效组 ID
%m	计算机名 (<code>uname -m</code>)
%n	系统节点名 (<code>uname -n</code>)
%p	进程 ID
%t	时间的十进制值 (2)
%u	有效用户 ID
%z	在其中执行进程的区域的名称 (<code>zonename</code>)
%%	字面值 %

例如，假定 `/var/core/core.%f.%p` 设置为全局核心文件路径。如果 PID 为 12345 的 `sendmail` 进程异常终止，它将生成 `/var/core/core.sendmail.12345` 作为 core 文件。

改进核心文件转储参数

通过从核心转储中排除进程的二进制映像的某些部分，可以改进在系统上转储的核心文件的性能。输入 `coreadm` 命令定制核心转储规范时，应该指定从核心转储排除的 DISM 映射、ISM 映射或 System V 共享内存等。有关说明，请参见 `coreadm(1M)` 手册页。

管理核心文件规范

可以按如下所示管理核心文件：

- “显示当前的核心转储配置” [27]
- “设置核心文件名称模式” [27]
- “启用文件路径” [28]
- “启用 `setuid` 程序以生成核心文件” [29]
- “恢复为缺省核心文件设置” [29]
- “更正过时的核心文件参数” [30]

显示当前的核心转储配置

使用不带任何选项的 `coreadm` 命令可以显示当前的核心转储配置。

```
$ coreadm
      global core file pattern:
global core file content: default
      init core file pattern: core
      init core file content: default
      global core dumps: disabled
      per-process core dumps: enabled
      global setid core dumps: disabled
per-process setid core dumps: disabled
      global core dump logging: disabled
```

设置核心文件名称模式

可以按全局、区域或进程设置核心文件名称模式。此外，可以设置在系统重新引导后保持不变的每进程缺省值。

例如，可以使用以下 `coreadm` 命令为 `init` 进程启动的所有进程设置缺省每进程核心文件模式。此设置适用于未显式覆盖缺省核心文件模式的所有进程。此设置在系统重新引导后继续存在。

```
# coreadm -i /var/core/core.%f.%p
```

可以使用以下 `coreadm` 命令为任意进程设置每进程核心文件名称模式：

```
# coreadm -p /var/core/core.%f.%p $$
```

`$$` 符号表示当前正在运行的 Shell 的进程 ID 的占位符。所有子进程都会继承每进程核心文件名称模式。

Here's another example:

```
$ coreadm -p $HOME/corefiles/%f.%p $$
```

或者，成为 `root` 角色并设置全局文件名称模式：

```
# coreadm -g /var/corefiles/%f.%p
```

设置全局或每进程核心文件名称模式之后，必须使用 `coreadm -e` 命令启用该名称模式。

通过将该命令放入用户的初始化文件（如 `.profile`）中，可以为用户登录会话期间运行的所有进程设置核心文件名称模式。

启用文件路径

可以启用每进程或全局核心文件路径。

- 要启用每进程核心文件路径，需要成为 `root` 角色并发出以下命令：

```
# coreadm -e process
```

如果要验证配置，请显示当前进程核心文件路径：

```
# coreadm $$
1180: /home/kryten/corefiles/%f.%p
```

- 要启用全局核心文件路径，需要成为 `root` 角色并发出以下命令：

```
# coreadm -e global -g /var/core/core.%f.%p
```

如果要验证配置，请显示当前进程核心文件路径：

```
# coreadm
global core file pattern: /var/core/core.%f.%p
global core file content: default
init core file pattern: core
```

```
init core file content: default
global core dumps: enabled
per-process core dumps: enabled
global setid core dumps: disabled
per-process setid core dumps: disabled
global core dump logging: disabled
```

启用 `setuid` 程序以生成核心文件

可以使用 `coreadm` 命令启用或禁用 `setuid` 程序，以便通过设置以下路径来为所有系统进程或每个进程生成核心文件：

- 如果启用了全局 `setuid` 选项，则全局核心文件路径允许系统中的所有 `setuid` 程序生成 `core` 文件。
- 如果启用了每进程 `setuid` 选项，每进程核心文件路径将允许特定 `setuid` 进程生成 `core` 文件。

缺省情况下，这两个标志都被禁用。由于安全原因，全局核心文件路径必须为全路径名，以斜杠 / 开头。如果 `root` 用户禁用了每进程核心文件，个人用户将无法获取核心文件。

`setuid` 核心文件归 `root` 用户所有，只有 `root` 用户具有读/写权限。一般用户无法访问这些文件，即使生成 `setuid` 核心文件的进程由普通用户所拥有也是如此。

有关更多信息，请参见 [coreadm\(1M\)](#) 手册页。

恢复为缺省核心文件设置

以 `root` 身份，运行以下命令之一来禁用核心文件路径并删除核心文件名称模式：

- 对于全局核心文件设置：

```
# coreadm -d global -g ""
```

注 - "" 是不包含空格的空字符串。

- 对于每进程核心文件设置：

```
# coreadm -d process -g ""
```

-d 选项可禁用核心文件路径。具有空字符串变量的 -g 选项可删除核心文件名称模式。核心文件路径和核心文件名称模式将恢复为原始缺省设置。

更正过时的核心文件参数

错误消息

```
NOTICE: 'set allow_setid_core = 1' in /etc/system is obsolete
NOTICE: Use the coreadm command instead of 'allow_setid_core'
```

原因

存在一个过时的参数，它允许 setuid 核心文件位于 /etc/system 文件中。

解决方法

从 /etc/system 文件中删除 allow_setid_core=1。然后使用 coreadm 命令启用全局 setuid 核心文件路径。

在进程失败后检查核心文件

通过 proc 工具，您可以检查进程核心文件以及实时进程。这些 proc 工具是可以处理 /proc 文件系统功能的实用程序。

可以将 /usr/proc/bin/pstack、pmap、pldd、pflags 以及 pcred 工具应用于核心文件，方法是在命令行中指定核心文件的名称，方式类似于指定这些命令的进程 ID。

有关使用 proc 工具检查核心文件的更多信息，请参见 [proc\(1\)](#)。

例 4-1 使用 proc 工具检查核心文件

```
$ ./a.out
Segmentation Fault(coredump)
$ /usr/proc/bin/pstack ./core
core './core' of 19305: ./a.out
000108c4 main      (1, ffbef5cc, ffbef5d4, 20800, 0, 0) + 1c
00010880 _start    (0, 0, 0, 0, 0, 0) + b8
```

管理系统日志和消息

本章讲述了如何查看和管理系统日志和系统消息。

本章包含以下信息：

- “使用 `rsyslogd` 扩展的系统日志记录” [31]
- “查看系统消息” [32]
- “系统日志轮转” [34]
- “定制系统消息日志” [35]
- “启用远程控制台消息传送” [37]

使用 `rsyslogd` 扩展的系统日志记录

此 Oracle Solaris 发行版包括用于安装和使用 `rsyslog` 服务来管理系统日志的选项。`rsyslog` 是一种可靠的扩展 `syslog` 守护进程实现，采用模块化的设计，支持多种功能，例如过滤、TCP、加密、高精度时间戳以及输出控制等。

系统日志 SMF 服务 `svc:/system/system-log:default` 仍然是缺省日志记录服务。要使用 `rsyslog` 服务，需要安装 `rsyslog` 软件包并启用 `rsyslog` 服务。

▼ 安装和启用 `rsyslog`

1. 通过按如下所示尝试启用服务，可以检查是否已在系统上安装了 `rsyslog` 软件包：

```
root@pcclone: ~# svcadm enable svc:/system/system-log:rsyslog
```

如果未安装 `rsyslog` 软件包，则显示以下消息：

```
svcadm: Pattern 'svc:/system/system-log:rsyslog' doesn't match any instance.
```

2. 如果未安装 `rsyslog` 软件包，请安装该软件包。

```
root@pcclone:~# pkg install rsyslog
```

```

Packages to install: 3
Services to change: 1
Create boot environment: No
Create backup boot environment: No

```

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
Completed	3/3	68/68	1.7/1.7	354k/s

PHASE	ITEMS
Installing new actions	147/147
Updating package state database	Done
Updating package cache	0/0
Updating image state	Done
Creating fast lookup database	Done

3. 确认存在 rsyslog 的实例。

```

root@pcclone:~# svcs -a | grep "system-log"
disabled      18:27:16 svc:/system/system-log:rsyslog
online        18:27:21 svc:/system/system-log:default

```

此输出确认 rsyslog 实例已存在，但是该实例被禁用。

4. 切换到 rsyslog 服务。

```

root@pcclone:~# svcadm disable svc:/system/system-log:default
root@pcclone:~# svcadm enable svc:/system/system-log:rsyslog
root@pcclone:~# svcs -xv

```

These commands disable the default service, enable rsyslog and report on status.

接下来的步骤 安装和启用 rsyslog 后，可以在 /etc/rsyslog.conf 中配置 syslog。

管理系统消息

以下各节介绍了 Oracle Solaris 中的系统消息传送功能。

查看系统消息

系统消息显示在控制台设备中。大多数系统消息的文本如下所示：

```
[ID msgid facility.]
```

例如：

```
[ID 672855 kern.notice] syncing file systems...
```

如果消息来自内核，则会显示内核模块名称。例如：

```
Oct 1 14:07:24 mars ufs: [ID 845546 kern.notice] alloc: /: file system full
```

当系统崩溃时，系统控制台可能会显示如下消息：

```
panic: error message
```

少数情况下，可能会显示以下消息而非故障消息：

```
Watchdog reset !
```

错误日志守护进程 `syslogd` 可在消息文件中自动记录各种系统警告和错误。缺省情况下，其中许多系统消息都会在系统控制台中显示，并存储在 `/var/adm` 目录中。通过设置系统消息日志可以指示这些消息的存储位置。有关更多信息，请参见[“定制系统消息日志” \[35\]](#)。这些消息可以提醒您系统出现问题，例如设备将要出现故障。

`/var/adm` 目录中包含若干个消息文件。最新消息位于 `/var/adm/messages` 文件中（和 `messages.*` 中），而最旧的消息位于 `messages.3` 文件中。经过一段时间后（通常为每隔十天），会创建一个新的 `messages` 文件。`messages.0` 文件被重命名为 `messages.1`，`messages.1` 被重命名为 `messages.2`，而 `messages.2` 被重命名为 `messages.3`。当前的 `/var/adm/messages.3` 文件将被删除。

由于 `/var/adm` 目录存储包含消息、故障转储和其他数据的大型文件，因此该目录可能会占用许多磁盘空间。为防止 `/var/adm` 目录变得过大，并确保可以保存将来的故障转储，应定期删除不需要的文件。可以使用 `crontab` 文件自动执行此任务。有关使此任务自动化的更多信息，请参见《[在 Oracle Solaris 11.2 中管理设备](#)》中的“删除转储文件”和《[在 Oracle Solaris 11.2 中管理系统信息、进程和性能](#)》中的第 4 章“调度系统任务”。

▼ 如何查看系统消息

- 使用 `dmesg` 命令显示由系统崩溃或重新引导生成的最新消息。

```
$ dmesg
```

或者，使用 `more` 命令逐屏显示消息。

```
$ more /var/adm/messages
```

例 5-1 查看系统消息

以下示例显示了 Oracle Solaris 10 系统中 `dmesg` 命令的输出。

```
$ dmesg
Mon Sep 13 14:33:04 MDT 2010
Sep 13 11:06:16 sr1-ubrm-41 svc.startd[7]: [ID 122153 daemon.warning] ...
Sep 13 11:12:55 sr1-ubrm-41 last message repeated 398 times
Sep 13 11:12:56 sr1-ubrm-41 svc.startd[7]: [ID 122153 daemon.warning] ...
```

```
Sep 13 11:15:16 sr1-ubrm-41 last message repeated 139 times
Sep 13 11:15:16 sr1-ubrm-41 xscreensaver[25520]: ,,,
Sep 13 11:15:16 sr1-ubrm-41 xscreensaver[25520]: ...
Sep 13 11:15:17 sr1-ubrm-41 svc.startd[7]: [ID 122153 daemon.warning]...
.
.
.
```

另请参见 有关更多信息，请参见 [dmesg\(1M\)](#) 手册页。

系统日志轮转

使用 root crontab 中的一个项内的 `logadm` 命令，可以轮转系统日志文件。不再使用 `/usr/lib/newsyslog` 脚本。

系统日志轮转在 `/etc/logadm.conf` 文件中定义。此文件包含用于 `syslogd` 等进程的日志轮转项。例如，`/etc/logadm.conf` 文件中的一个项指定，除非 `/var/log/syslog` 文件为空，否则该文件每周轮转一次。最新的 `syslog` 文件成为 `syslog.0`，下一个最新的文件成为 `syslog.1`，依此类推。会保留八个以前的 `syslog` 日志文件。

`/etc/logadm.conf` 文件还包含记录最后一次日志轮转发生时间的戳。

可以使用 `logadm` 命令来定制系统日志，并可根据需要在 `/etc/logadm.conf` 文件中添加其他日志。

例如，要轮转 Apache 访问和错误日志，请使用以下命令：

```
# logadm -w /var/apache/logs/access_log -s 100m
# logadm -w /var/apache/logs/error_log -s 10m
```

在此示例中，Apache `access_log` 文件会在大小达到 100 MB 时进行轮转，以 `.0`、`.1` 等作为后缀，并保留旧 `access_log` 文件的 10 个副本。`error_log` 会在大小达到 10 MB 时进行轮转，后缀和副本数与 `access_log` 文件相同。

用于上述 Apache 日志轮转示例的 `/etc/logadm.conf` 项与以下示例类似：

```
# cat /etc/logadm.conf
.
.
.
/var/apache/logs/error_log -s 10m
/var/apache/logs/access_log -s 100m
```

有关更多信息，请参见 [logadm\(1M\)](#)。

如果要向非 root 用户授予维护日志文件的特权，可以授予 "Log Management"（日志管理）作为该用户的权限配置文件。可以将 `-P` 选项与 `useradd` 命令（对于新用

户) 或 `usermod` 命令 (对于现有用户) 配合使用来执行该授予。有关说明, 请参见 `useradd(1M)` 和 `usermod(1M)` 手册页。

定制系统消息日志

通过修改 `/etc/syslog.conf` 文件, 可以捕获各个系统进程生成的其他错误消息。缺省情况下, `/etc/syslog.conf` 文件会将许多系统进程消息定向到 `/var/adm/messages` 文件。崩溃和引导消息也存储在这些文件中。要查看 `/var/adm` 消息, 请参见[如何查看系统消息 \[33\]](#)。

`/etc/syslog.conf` 文件有两个通过制表符分隔的列:

facility.level ... action

facility.level 消息或情况的工具或系统源。可能是由逗号分隔的工具列表。[表 5-1 “syslog.conf 消息的源工具”](#) 中列出了工具值。*level*, 表示所记录情况的严重程度或优先级。[表 5-2 “syslog.conf 消息的优先级”](#) 中列出了优先级。

如果同一工具的两个项用于不同优先级, 则不要将这两个项放在同一行中。在 `syslog` 文件中放置优先级表示将至少记录该优先级的所有消息, 最后一条消息优先。对于给定的工具和级别, `syslogd` 将匹配该级别以及所有更高级别的所有消息。

action 操作字段表示将消息转发到的位置。

以下示例显示缺省的 `/etc/syslog.conf` 文件中的样例行。

```
user.err                                    /dev/sysmsg
user.err                                    /var/adm/messages
user.alert                                 `root, operator'
user.emerg                                 *
```

这意味着将自动记录以下用户消息:

- 将用户错误列显到控制台, 同时将其记录到 `/var/adm/messages` 文件中。
- 将需要立即操作的用户消息 (`alert`) 发送给 `root` 用户和 `operator` 用户。
- 将用户紧急消息发送给各用户。

注 - 如果在 `/etc/syslog.conf` 文件中多次指定一个日志目标, 则将各项分别放置在不同的行中可能会导致消息的记录顺序混乱。请注意, 可在一个行项中指定多个选择器, 每个选择器之间用分号分隔。

下表中显示了最常见的错误情况源。最常见的优先级将按严重性顺序显示在[表 5-2 “syslog.conf 消息的优先级”](#) 中。

表 5-1 syslog.conf 消息的源工具

来源	说明
kern	内核
auth	验证
daemon	所有守护进程
mail	邮件系统
lp	假脱机系统
user	用户进程

注 - 可在 /etc/syslog.conf 文件中激活的 syslog 工具数没有限制。

表 5-2 syslog.conf 消息的优先级

优先级	说明
emerg	系统紧急情况
alert	需要立即更正的错误
crit	严重错误
err	其他错误
info	信息性消息
debug	用于调试的输出
none	此设置不记录输出

▼ 如何定制系统消息日志

1. 成为 root 角色或指定有 `solaris.admin.edit/etc/syslog.conf` 授权的角色。
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` 命令编辑 `/etc/syslog.conf` 文件，按照 `syslog.conf(4)` 中所述的语法添加或更改消息源、优先级以及消息位置。

```
$ pfedit /etc/syslog.conf
```
3. 保存更改。

例 5-2 定制系统消息日志

此样例 `/etc/syslog.conf` `user.emerg` 工具将用户紧急消息发送给 root 和个人用户。

```
user.emerg                `root, *'
```

启用远程控制台消息传送

以下新增控制台功能可以提高您对远程系统进行故障排除的能力：

- 利用 `consadm` 命令，您可以选择串行设备作为辅助（或远程）控制台。使用 `consadm` 命令，系统管理员可以配置一个或多个串行端口，以便当系统在不同运行级之间转换时显示重定向的控制台消息，并托管 `sulogin` 会话。借助此功能，您可用调制解调器拨入并连接到串行端口，以监视控制台消息并参与 `init` 状态转换。有关更多信息，请参见 `sulogin(1M)` 和后面的逐步过程。

使用配置为辅助控制台的端口登录系统时，它主要用作输出设备，其中显示的信息也在缺省控制台中显示。如果引导脚本或其他应用程序从缺省控制台中读取内容或向其中写入内容，则写入输出将在所有辅助控制台上显示，但输入只从缺省控制台中读取。有关在交互式登录会话期间使用 `consadm` 命令的更多信息，请参见“[在交互式登录会话期间使用 `consadm` 命令的准则](#)” [38]。

- 控制台输出包括内核和写入新的伪设备 `/dev/sysmsg` 的 `syslog` 消息。此外，`rc` 脚本启动消息也将写入 `/dev/msglog`。以前，所有这些消息都写入 `/dev/console`。
如果希望看到显示在辅助控制台中的脚本消息，那么需要把将控制台输出定向到 `/dev/console` 的脚本更改为 `/dev/msglog`。如果希望将消息重定向到辅助设备，则应该将引用 `/dev/console` 的程序显式修改为使用 `syslog()` 或 `strlog()`。
- `consadm` 命令运行守护进程来监视辅助控制台设备。指定为辅助控制台并且已断开、挂起或失去载体的任何显示设备都将从辅助控制台设备列表中删除，并且不再处于活动状态。启用一个或多个辅助控制台不会禁用缺省控制台上的消息显示，消息将继续在 `/dev/console` 中显示。

在运行级转换期间使用辅助控制台消息传递

在运行级转换期间使用辅助控制台消息传递时，请记住以下几点：

- 如果在系统引导时运行的 `rc` 脚本期望用户输入，则输入不能来自辅助控制台。输入必须来自缺省控制台。
- 由 `init` 调用以用于在运行级之间转换时提示输入 `root` 口令的 `sulogin` 程序已被修改，除了缺省控制台设备之外，还可以向每个辅助设备发送 `root` 口令提示。
- 用户永远不应直接调用 `sulogin`。用户必须具有 `solaris.system.maintenance` 授权才能使用此实用程序。
- 当系统处于单用户模式并且使用 `consadm` 命令启用一个或多个辅助控制台时，将在第一个设备上运行控制台登录会话，以便为 `sulogin` 提示符提供正确的 `root` 口令。从控制台设备收到正确口令时，`sulogin` 将禁用来自所有其他控制台设备的输入。
- 如果其中一个控制台承担了单用户特权，将在缺省控制台和其他辅助控制台上显示一条消息。此消息指出已通过接受正确的 `root` 口令而成为控制台的设备。如果运行单用户 `shell` 的辅助控制台中丢失载体，则会执行以下两种操作之一：
 - 如果辅助控制台代表一个处于运行级 1 的系统，则系统会继续到缺省运行级。

- 如果辅助控制台代表一个处于运行级 S 的系统，则系统会显示已通过 Shell 输入 `init s` 或 `shutdown` 命令的设备中的 `ENTER RUN LEVEL (0-6, s or S):` 消息。如果该设备中也没有任何载体，则必须重新建立载体并输入正确的运行级。`init` 或 `shutdown` 命令不再重新显示运行级提示。
- 如果使用串行端口登录系统，并发出 `init` 或 `shutdown` 命令以转换到其他运行级，则无论此设备是否为辅助控制台，登录会话都将丢失。此情况与没有辅助控制台功能的发行版相同。
- 一旦使用 `consadm` 命令将设备选作辅助控制台，该设备将一直用作辅助控制台，直到重新引导系统或取消选中辅助控制台为止。但是，`consadm` 命令有一个选项，可在系统重新引导期间将设备设置为辅助控制台。（有关逐步说明，请参见以下过程。）

在交互式登录会话期间使用 `consadm` 命令的准则

如果要通过使用与串行端口连接的终端登录系统，再使用 `consadm` 命令查看终端的控制台消息的方式来运行交互式登录会话，请注意以下行为。

- 如果在辅助控制台处于活动状态时将终端用于交互式登录会话，则会向 `/dev/sysmsg` 或 `/dev/msglog` 设备发送控制台消息。
- 在终端发出命令时，输入将转到交互式会话而非缺省控制台 (`/dev/console`)。
- 如果运行 `init` 命令更改运行级，远程控制台软件将中止交互式会话并运行 `sulogin` 程序。此时，只接受来自终端的输入，并将其视为来自控制台设备的输入。这样您便可以按“[在运行级转换期间使用辅助控制台消息传递](#)” [37]中所述为 `sulogin` 程序输入口令。

然后，如果您在（辅助）终端中输入正确口令，辅助控制台将运行交互式 `sulogin` 会话，并锁定缺省控制台和任何竞争性的辅助控制台。这意味着，终端基本上可用作系统控制台。

- 此时，您可以更改到运行级 3 或转到其他运行级。如果更改运行级，`sulogin` 将在所有控制台设备中再次运行。如果您退出或指定系统应达到运行级 3，则所有辅助控制台都将丧失提供输入的能力。它们将恢复为控制台消息的显示设备。

随着系统的提升，您必须为缺省控制台设备中的 `rc` 脚本提供信息。在系统恢复启动后，`login` 程序将在串行端口上运行，您可以重新登录到其他交互式会话中。如果已将该设备指定为辅助控制台，您将继续在终端中获得控制台消息，但来自该终端的所有输入都将转至交互式会话。

▼ 如何启用辅助（远程）控制台

在您使用 `consadm` 命令添加辅助控制台之前，`consadm` 守护进程不会开始监视端口。作为一种安全功能，在载体脱机或取消选择辅助控制台设备之前，控制台消息只能重定向。这意味着必须在端口中建立载体，才能成功使用 `consadm` 命令。

有关启用辅助控制台的更多信息，请参见 [consadm\(1m\)](#) 手册页。

1. 登录到系统并成为 **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. 启用辅助控制台。

```
# consadm -a devicename
```

3. 验证当前连接是否为辅助控制台。

```
# consadm
```

例 5-3 启用辅助（远程）控制台

```
# consadm -a /dev/term/a
# consadm
/dev/term/a
```

▼ 如何显示辅助控制台的列表

1. 登录到系统并成为 **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. 选择以下步骤之一：

- a. 显示辅助控制台的列表。

```
# consadm
/dev/term/a
```

- b. 显示持久性辅助控制台的列表。

```
# consadm -p
/dev/term/b
```

▼ 如何在系统重新引导期间启用辅助（远程）控制台

1. 登录到系统并成为 **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. 在系统重新引导期间启用辅助控制台。

```
# consadm -a -p devicename
```

这将会向持久性辅助控制台列表中添加设备。

3. 验证设备是否已添加至持久性辅助控制台的列表中。

```
# consadm
```

例 5-4 在系统重新引导期间启用辅助（远程）控制台

```
# consadm -a -p /dev/term/a
# consadm
/dev/term/a
```

▼ 如何禁用辅助（远程）控制台

1. 登录到系统并成为 `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. 选择以下步骤之一：

- a. 禁用辅助控制台。

```
# consadm -d devicename
```

或

- b. 禁用辅助控制台并从持久性辅助控制台列表中将其删除。

```
# consadm -p -d devicename
```

3. 验证是否已禁用辅助控制台。

```
# consadm
```

例 5-5 禁用辅助（远程）控制台

```
# consadm -d /dev/term/a
# consadm
```

索引

B

- 崩溃, 35
 - 以下过程, 13
 - 保存其他系统信息, 33
 - 保存故障转储信息, 8
 - 客户服务和, 13
 - 显示生成的系统信息, 15, 33
 - 检查故障转储, 14, 15
 - 概述, 7, 9
 - 重新引导失败, 之后, 19

C

- 错误消息
 - 与崩溃相关的, 33
 - 优先级, 36, 36
 - 定制日志, 35, 35
 - 崩溃消息, 33
 - 指定存储位置, 33, 35, 35
 - 日志文件, 13, 33
 - 源, 35, 35
- Command not found 错误消息, 22
- consadm 命令, 38
 - 启用辅助控制台, 38
 - 在系统重新引导期间, 39
 - 显示辅助控制台的列表, 39
 - 禁用辅助控制台, 40
- coreadm 命令, 25
 - 显示核心转储配置, 27
 - 管理核心文件, 25
 - 设置核心文件名称模式, 27
- crontab 命令
 - /var/adm 维护和, 33, 33

D

- 定制
 - 系统消息日志, 35, 36
- dmesg 命令, 33, 33
- dumpadm 命令, 9

E

- /etc/syslog.conf 文件, 35, 35

F

- 辅助 (远程) 控制台, 37

G

- 故障
 - 故障排除, 9
 - 故障消息, 33
 - 故障转储
 - 保存在 /var/crash 目录中, 7
 - 修改配置, 10
 - 在目录已满的情况下保存数据, 16
 - 文件, 8
 - 更改, 7, 8
 - 显示当前配置, 10
 - 检查故障转储信息, 13
 - 禁用或启用保存, 12
 - 重新构造的文件, 7

H

- 核心文件
 - 使用 coreadm 管理, 25

- 使用 proc 工具检查，30
- 管理，27
- 核心文件名称模式
 - 设置使用 coreadm，27
- 核心转储配置
 - 使用 coreadm 显示，27

J

- 技术支持
 - 发送崩溃信息，13
- 检查核心文件
 - 使用 proc 工具，30
- 进程故障
 - 故障排除，25
- 禁用
 - 辅助控制台，使用 consadm 命令，40
- 警报信息优先级（对于 syslogd），36

K

- 客户服务
 - 发送崩溃信息，13
- 控制台
 - 辅助
 - 在系统重新引导期间启用，39

M

- 每进程核心文件路径
 - 设置使用 coreadm，26
- mdb 实用程序，13，14，15
- messages 文件，13，35
- messages.*n* 文件，33

O

- Ops Center，16

P

- proc tools
 - 检查核心文件，30

Q

- 启用
 - 辅助控制台，使用 consadm 命令，38
 - 辅助控制台，在系统重新引导期间，39
- 全局核心文件路径
 - 设置使用 coreadm，26

R

- rsyslog 服务，31

S

- 设置
 - 核心文件名称模式，使用 coreadm，27
- 识别网络访问问题，24
- 搜索路径
 - 用于设置的文件，23
- savecore 命令，9
 - 更改目录，16
- syslog.conf 文件，35，35
- syslogd 守护进程，33

U

- UNIX 系统（故障转储信息），8
 - /usr/adm/messages 文件，13
 - /usr/bin/mdb 实用程序，14

V

- /var/adm/messages 文件，13，35
- /var/adm/messages.*n* 文件，33

W

- 网络
 - 识别访问问题，24
- 文件
 - 用于设置搜索路径，23
- 文件或组的所有权
 - 解决文件访问问题，24
- Watchdog reset! 消息，33

X

系统崩溃 见 崩溃

系统消息

定制日志, 35, 36

指定存储位置, 33

显示文件系统已满, 21

显示

崩溃信息, 15, 33

引导消息, 33, 33

核心转储配置, 使用 `coreadm`, 27

Y

引导

故障排除, 20

系统挂起, 20

显示生成的消息, 33, 33

Z

在故障转储目录已满时保存数据, 16

重新引导

崩溃后失败, 19

