

解决 Oracle® Solaris 11.1 中的典型问题

版权所有 © 1998, 2012, Oracle 和/或其附属公司。保留所有权利。

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

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

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

U.S. GOVERNMENT END USERS:

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

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

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

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

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

目录

前言	5
1 管理系统故障转储信息（任务）	7
管理系统故障转储信息中的新增内容	7
savecore 行为的更改	7
系统崩溃（概述）	8
系统故障转储文件	8
保存故障转储	9
使用 dumpadm 命令管理系统故障转储信息	9
dumpadm 命令的工作原理	10
管理系统故障转储信息	10
管理系统故障转储信息（任务列表）	10
▼如何显示当前的故障转储配置	10
▼如何修改故障转储配置	11
▼如何检查故障转储信息	12
▼如何从完整的故障转储目录中恢复（可选的）	14
▼如何禁用或启用故障转储的保存	14
2 管理核心文件（任务）	17
管理核心文件	17
配置核心文件路径	17
扩展的核心文件名	18
设置核心文件名称模式	18
启用 setuid 程序以生成核心文件	19
管理核心文件（任务列表）	19
显示当前的核心转储配置	20
▼如何设置核心文件名称模式	20

▼ 如何启用每进程核心文件路径	20
▼ 如何启用全局核心文件路径	21
核心文件问题故障排除	21
检查核心文件	22
3 系统和软件问题的故障排除 (任务)	23
系统崩溃故障排除	23
系统崩溃时应执行的操作	23
收集故障排除数据	24
系统崩溃故障排除核对表	24
管理系统消息	25
查看系统消息	25
系统日志轮转	26
定制系统消息日志	27
启用远程控制台消息传送	30
文件访问问题故障排除	33
解决搜索路径的问题 (Command not found)	34
更改文件和组的所有权	35
解决文件访问问题	35
识别网络访问问题	35
4 其他系统和软件问题的故障排除 (任务)	37
重新引导失败时应执行的操作	37
忘记 root 口令或遇到阻止系统进行引导的问题时应执行的操作	38
系统挂起时应执行的操作	38
文件系统占满时应执行的操作	39
由于创建了大文件或目录导致文件系统占满	39
由于系统内存不足导致 TMPFS 文件系统变满	39
复制或恢复后文件 ACL 丢失时应执行的操作	40
索引	41

前言

《解决 Oracle Solaris 11.1 中的典型问题》是文档集的一部分，该文档集包含 Oracle Solaris 系统管理信息中的重要部分。该指南包含基于 SPARC 和基于 x86 的系统的信息。

本书假设您已经完成以下任务：

- 安装了 Oracle Solaris 软件
- 设置了任何计划使用的网络软件

对于 Oracle Solaris，系统管理员可能感兴趣的新增功能在相应各章中名为“... 的新增功能”的各节中进行了介绍。

注 - 此 Oracle Solaris 发行版支持使用 SPARC 和 x86 系列处理器体系结构的系统。支持的系统可以在 Oracle Solaris OS: Hardware Compatibility Lists (Oracle Solaris OS: 硬件兼容性列表) 中找到。本文档列举了在不同类型的平台上进行实现时的所有差别。

有关支持的系统，请参见 [Oracle Solaris OS: Hardware Compatibility Lists](#) (Oracle Solaris OS: 硬件兼容性列表)。

目标读者

本书适用于所有负责对运行 Oracle Solaris 11 发行版的一个或多个系统进行管理的人员。要使用本书，您应当具备 1 到 2 年的 UNIX 系统管理经验。参加 UNIX 系统管理培训课程可能会对您有所帮助。

获取 Oracle 支持

Oracle 客户可以通过 My Oracle Support 获取电子支持。有关信息，请访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>，或访问 <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> (如果您听力受损)。

印刷约定

下表介绍了本书中的印刷约定。

表 P-1 印刷约定

字体或符号	含义	示例
AaBbCc123	命令、文件和目录的名称；计算机屏幕输出	编辑 .login 文件。 使用 <code>ls -a</code> 列出所有文件。 machine_name% you have mail.
AaBbCc123	用户键入的内容，与计算机屏幕输出的显示不同	machine_name% su Password:
<i>aabbcc123</i>	要使用实名或值替换的命令行占位符	删除文件的命令为 <code>rm filename</code> 。
<i>AaBbCc123</i>	保留未译的新词或术语以及要强调的词	这些称为 <i>Class</i> 选项。 注意 ：有些强调的项目在联机时以粗体显示。
新词术语强调	新词或术语以及要强调的词	高速缓存 是存储在本地的副本。 请勿保存文件。
《书名》	书名	阅读《用户指南》的第 6 章。

命令中的 shell 提示符示例

下表显示了 Oracle Solaris OS 中包含的缺省 UNIX shell 系统提示符和超级用户提示符。请注意，在命令示例中显示的缺省系统提示符可能会有所不同，具体取决于 Oracle Solaris 发行版。

表 P-2 shell 提示符

shell	提示符
Bash shell、Korn shell 和 Bourne shell	\$
Bash shell、Korn shell 和 Bourne shell 超级用户	#
C shell	machine_name%
C shell 超级用户	machine_name#

管理系统故障转储信息（任务）

本章介绍如何在 Oracle Solaris OS 中管理系统故障转储信息。

以下列出本章所介绍的信息：

- 第 7 页中的“管理系统故障转储信息中的新增内容”
- 第 8 页中的“系统崩溃（概述）”
- 第 10 页中的“管理系统故障转储信息”

管理系统故障转储信息中的新增内容

本节介绍此 Oracle Solaris 发行版中用于管理系统资源的新增功能或已更改的功能。

savecore 行为的更改

现在，`savecore` 命令首先使用 `.partial` 后缀创建文件，该后缀将附加到文件中。文件完全写入之后，系统将会对其重命名并删除该后缀。潜在问题可能会阻止对文件进行重命名以及删除后缀，例如，如果 `savecore` 命令仍处于繁忙状态。另一个示例是 `savecore` 命令由于引导后系统短暂崩溃而中断。

如果该命令处于繁忙状态，可以使用 `ps` 命令搜索正在运行的 `savecore` 进程的进程 ID (process ID, PID)，然后等待进程完成。如果该进程中断，可以手动删除剩余文件，然后通过运行带有 `-d` 选项的 `savecore` 命令重新创建文件。

有关更多信息，请参见 [savecore\(1M\)](#) 手册页。

系统崩溃（概述）

在处理系统故障转储信息时，请记住以下几点重要内容：

- 必须承担 `root` 角色才可以访问和管理系统故障转储信息。请参见《[Oracle Solaris 11.1 管理：安全服务](#)》中的“如何使用指定给您的管理权限”。
- 请勿禁用系统上用于保存系统故障转储的选项。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。
- 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。

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。然后，将自动重新引导系统。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。保存的故障转储文件提供了有助于诊断问题的非常有价值的信息。

故障转储信息以压缩格式写入 `vmdump.n` 文件中，其中 `n` 是标识故障转储的一个整数。随后，可在同一系统或其他系统中调用 `savecore` 命令以将压缩的故障转储文件扩展为名为 `unix.n` 和 `vmcore.n` 的文件对。还可以使用 `dumpadm` 命令配置重新引导时保存故障转储文件的目录。

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《[Oracle Solaris 11.1 管理：ZFS 文件系统](#)》中的“管理 ZFS 交换和转储设备”。

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写入一对文件，名称分别为 `unix.x` 和 `vmcore.x`，其中 `x` 标识转储序列号。这些文件共同表示保存的系统故障转储信息。

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。

使用 `dumpadm` 命令可以管理系统故障转储信息。有关更多信息，请参见第 9 页中的“[使用 dumpadm 命令管理系统故障转储信息](#)”。

保存故障转储

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。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信息，请参见 [mdb\(1\)](#) 手册页。

使用 `dumpadm` 命令管理系统故障转储信息

使用 `dumpadm` 命令可以管理 Oracle Solaris OS 中的系统故障转储信息。

- 使用 `dumpadm` 命令可以配置操作系统的故障转储。`dumpadm` 配置参数包括转储内容、转储设备以及保存故障转储文件的目录。
- 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。
- 当专用转储设备（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。这意味着正在引导的系统不等 `savecore` 命令完成就进入下一步骤。在较大的内存系统中，可在 `savecore` 完成前使用系统。有关潜在问题，请参见 [第 7 页中的“savecore 行为的更改”](#)。
- 缺省情况下，会保存 `savecore` 命令生成的系统故障转储文件。
- 通过 `savecore -L` 命令，您可以获取正在运行的 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，因此您只能使用 `savecore -L` 命令（如果已配置专用转储设备）。

转储配置参数由 `dumpadm` 命令管理。下表介绍了 `dumpadm` 的配置参数。

转储参数	说明
转储设备	在系统崩溃时临时存储转储数据的设备。当转储设备不是交换区域时， <code>savecore</code> 将在后台运行，这样可以加快引导过程进行的速度。
<code>savecore</code> 目录	存储系统故障转储文件的目录。
转储内容	要转储的内存数据的类型。
最小空闲空间	保存故障转储文件后 <code>savecore</code> 目录中所需的最小空闲空间量。如果未配置最小空闲空间，则缺省值为 1 MB。

有关更多信息，请参见 [dumpadm\(1M\)](#)。

dumpadm 命令的工作原理

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。

具体地说，`dumpadm` 通过 `/dev/dump` 接口来初始化转储设备和转储内容。

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。

管理系统故障转储信息

本节介绍有关管理系统故障转储信息的任务。

管理系统故障转储信息（任务列表）

任务	说明	参考
1. 显示当前的故障转储配置。	使用 <code>dumpadm</code> 命令显示当前的故障转储配置。	第 10 页中的“如何显示当前的故障转储配置”
2. 修改故障转储配置。	使用 <code>dumpadm</code> 命令指定要转储的数据类型、系统是否使用专用转储设备和用于保存故障转储文件的目录，以及在写入故障转储文件后必须保持可用的空间量。	第 11 页中的“如何修改故障转储配置”
3. 检查故障转储文件。	使用 <code>mdb</code> 命令查看故障转储文件。	第 12 页中的“如何检查故障转储信息”
4.（可选的）从完整的故障转储目录中恢复。	系统崩溃，但 <code>savecore</code> 目录中没有可用空间，并且您需要保存一些关键的系统故障转储信息。	第 14 页中的“如何从完整的故障转储目录中恢复（可选的）”
5.（可选的）禁用或启用故障转储文件的保存。	使用 <code>dumpadm</code> 命令禁用或启用故障转储文件的保存。缺省情况下，会启用故障转储文件的保存。	第 14 页中的“如何禁用或启用故障转储的保存”

▼ 如何显示当前的故障转储配置

1 承担 root 角色。

请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

2 显示当前的故障转储配置。

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

前面的示例输出表明：

- 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写入 `/var/crash/` 目录。
- 已启用故障转储文件的保存。
- 以压缩格式保存故障转储。

▼ 如何修改故障转储配置

1 承担 root 角色。

请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

2 确定当前的故障转储配置。

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

此输出标识了运行 Oracle Solaris 11 发行版的系统的缺省转储配置。

3 修改故障转储配置。

```
# /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* 指定要转储的数据类型。使用 `kernel` 转储所有内核内存，使用 `all` 转储所有内存，或使用 `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面。缺省转储内容是内核内存。
- 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。
- m *nnnk* | *nnnm* | *nnn%* 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 (`nnnk`)、MB (`nnnm`) 或文件系统大小百分比 (`nnn%`) 的形式指定此参数。`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（根据大小）会减少空闲空间量

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。有关从此情况中恢复的信息，请参见第 14 页中的“如何从完整的故障转储目录中恢复（可选的）”。

<code>-n</code>	指定重新引导系统时不应运行 <code>savecore</code> 。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 <code>savecore</code> ，则系统开始交换时将覆盖故障转储信息。
<code>-s</code>	指定用于存储故障转储文件的备用目录。在 Oracle Solaris 11 中，缺省目录是 <code>/var/crash</code> 。
<code>-u</code>	强制更新基于 <code>/etc/dumpadm.conf</code> 文件内容的内核转储配置。
<code>-y</code>	修改转储配置以在重新引导时自动执行 <code>savecore</code> 命令，即此转储设置的缺省值。
<code>-z on off</code>	修改转储配置以控制重新引导时 <code>savecore</code> 命令的操作。 <code>on</code> 设置允许以压缩格式保存核心文件。 <code>off</code> 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，而以压缩格式保存需要的文件系统空间较少，因此缺省值为 <code>on</code> 。

示例 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
```

▼ 如何检查故障转储信息

1 承担 root 角色。

请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

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
或者，可以按如下所示指定命令：
```

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

4 显示系统崩溃状态，如下所示：

```
> ::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
```

示例 1-2 检查故障转储信息

以下示例显示 `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 ncsiz=0x4e20 [0t20000]
set pt_cnt=0x400 [0t1024]
> $q
```

▼ 如何从完整的故障转储目录中恢复（可选的）

在此情况下，系统崩溃，但 `savecore` 目录中没有任何可用空间，并且您要保存一些关键的系统故障转储信息。

- 1 系统重新引导后，以 `root` 角色身份登录。
- 2 通过删除已发送给服务提供商的现有故障转储文件来清除 `savecore` 目录（通常为 `/var/crash/`）。
 - 或者，手动运行 `savecore` 命令以指定拥有足够磁盘空间的备用目录。


```
# savecore [ directory ]
```

▼ 如何禁用或启用故障转储的保存

- 1 承担 `root` 角色。
请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。
- 2 在系统中禁用或启用故障转储的保存。

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

示例 1-3 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
```

示例 1-4 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
```


管理核心文件（任务）

本章介绍如何使用 `coreadm` 命令管理核心文件。

以下列出本章所介绍的信息：

- 第 17 页中的“管理核心文件”
- 第 21 页中的“核心文件问题故障排除”
- 第 22 页中的“检查核心文件”

管理核心文件

核心文件是在异常终止进程或应用程序时生成的。使用 `coreadm` 命令可以管理核心文件。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。这意味着，当进程或守护进程异常终止时，可通过检查特定目录中的核心文件来跟踪问题。

配置核心文件路径

以下两个可配置的 `core` 文件路径可以独立于彼此启用或禁用：

- 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。

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。

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

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。

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

例如，如果全局核心文件路径设置为：

```
/var/core/core.%f.%p
```

并且 PID 为 12345 的 `sendmail` 进程异常终止，则会生成以下 `core` 文件：

```
/var/core/core.sendmail.12345
```

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，以下 `coreadm` 命令将设置缺省的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。例如，以下 `coreadm` 命令为 `init` 进程启动的所有进程设置全局核心文件模式。此模式在系统重新引导后继续存在。

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

以下 `coreadm` 命令可为任何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。

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。有关更多信息，请参见以下过程。

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 `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。

管理核心文件（任务列表）

任务	说明	参考
1. 显示当前的核心转储配置。	使用 <code>coreadm</code> 命令显示当前的核心转储配置。	第 20 页中的“显示当前的核心转储配置”

任务	说明	参考
2. 修改核心转储配置。	修改核心转储配置，以执行下列操作之一： <ul style="list-style-type: none"> ■ 设置核心文件名称模式。 ■ 启用每进程核心文件路径。 ■ 启用全局核心文件路径。 	第 20 页中的“如何设置核心文件名称模式” 第 20 页中的“如何启用每进程核心文件路径” 第 21 页中的“如何启用全局核心文件路径”
3. 检查核心转储文件。	使用 <code>proc</code> 工具查看核心转储文件。	第 22 页中的“检查核心文件”

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
```

▼ 如何设置核心文件名称模式

- 确定要设置每进程核心文件还是全局核心文件，并选择下列项之一：

- a. 设置每进程文件名称模式。

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

- b. 承担 `root` 角色。

- c. 设置全局文件名称模式。

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

▼ 如何启用每进程核心文件路径

- 1 承担 `root` 角色。

请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- 2 启用每进程核心文件路径。

```
# coreadm -e process
```

- 3 显示当前的进程核心文件路径，以验证配置。

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

▼ 如何启用全局核心文件路径

- 1 承担 root 角色。

请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。

- 2 启用全局核心文件路径。

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

- 3 显示当前的进程核心文件路径，以验证配置。

```
# 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
```

核心文件问题故障排除

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。

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 `pcrd` 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\(1\)](#)。

示例 2-1 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
```

系统和软件问题的故障排除（任务）

本章概述了软件问题的故障排除，包括系统崩溃故障排除、管理故障转储信息以及查看和管理系统消息方面的信息。

下面列出了本章中的信息。

- 第 23 页中的“系统崩溃故障排除”
- 第 25 页中的“管理系统消息”
- 第 33 页中的“文件访问问题故障排除”

系统崩溃故障排除

如果正在运行 Oracle Solaris 的系统崩溃，请为您的服务提供商提供尽可能多的信息，包括故障转储文件。

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，则使其再次运行可能会看起来像是您最紧迫的问题。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，请参见第 26 页中的“如何查看系统消息”。
 - 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，请参见第 23 页中的“系统崩溃故障排除”。

2. 查看系统崩溃后是否生成了系统故障转储。缺省情况下，会保存系统故障转储。有关故障转储的信息，请参见第 1 章，[管理系统故障转储信息（任务）](#)。
3. 如果系统在崩溃后无法引导，请参见《[引导和关闭 Oracle Solaris 11.1 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。

收集故障排除数据

请回答以下问题，以帮助查出系统问题。有关收集崩溃系统的故障排除数据的信息，请参见第 24 页中的“[系统崩溃故障排除核对表](#)”。

表 3-1 确定系统崩溃数据

问题	说明
是否可以重现问题？	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。
是否使用了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。
系统在崩溃之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。
在系统崩溃之前，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 /etc/system 文件中添加了任何调优参数？	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，问题是否与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。

系统崩溃故障排除核对表

可在为崩溃的系统收集系统数据时使用此核对表。

项	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。	

项	数据
	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？
	系统中是否有任何第三方驱动程序？
	系统在崩溃前正在执行什么操作？
	在系统崩溃前是否有任何异常的控制台消息？
	是否向 <code>/etc/system</code> 文件中添加了任何参数？
	问题是在最近开始的吗？

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.priority]
```

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，请参见第 27 页中的“定制系统消息日志”。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《Oracle Solaris 11.1 管理：设备和文件系统》中的“如何删除故障转储文件”和《在 Oracle Solaris 11.1 中管理系统信息、进程和性能》中的第 4 章“调度系统任务（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
```

示例 3-1 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时间戳。

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\(1M\)](#)。

可以超级用户身份或通过承担等效角色（具有日志管理权限）来使用 `logadm` 命令。通过 RBAC（role-based access control，基于角色的访问控制），您可以通过提供对 `logadm` 命令的访问权限来授予非 `root` 用户维护日志文件的特权。

例如，可通过向 `/etc/user_attr` 文件添加以下项，授予用户 `andy` 使用 `logadm` 命令的权限：

```
andy:::profiles=Log Management
```

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，请参见第 26 页中的“如何查看系统消息”。

`/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 3-2 中列出了工具值。*level* 表示所记录情况的严重程度或优先级。优先级级别列在表 3-3 中。

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记录该优先级或更高优先级的所有消息，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。最常见的优先级按严重性顺序显示在表 3-3 中。

表 3-2 `syslog.conf` 消息的源工具

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 3-3 `syslog.conf` 消息的优先级

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

▼ 如何定制系统消息日志

- 1 承担 `root` 角色或指定有 `solaris.admin.edit/etc/syslog.conf` 授权的角色。
请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。
- 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。

示例 3-2 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，请参见第 31 页中的“[在交互式登录会话期间使用 consadm 命令的准则](#)”。

- 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超级用户口令的 `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超级用户口令提示。
- 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` 提示提供正确的超级用户口令。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超级用户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。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。这样您就可以按第 30 页中的“在运行级转换期间使用辅助控制台消息传递”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 登录到系统并承担 **root** 角色。
请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。
- 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
```

示例 3-3 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 登录到系统并承担 **root** 角色。
请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。
- 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 登录到系统并承担 **root** 角色。
请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。
- 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
```

示例 3-4 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 登录到系统并承担 root 角色。
请参见《Oracle Solaris 11.1 管理：安全服务》中的“如何使用指定给您的管理权限”。
- 2 选择以下步骤之一：
 - a. 禁用辅助控制台。

```
# consadm -d devicename
```


或者
 - 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
```

示例 3-5 禁用辅助（远程）控制台

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

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 检查以下各项：

- 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	附注
bash 和 ksh93	<code>\$HOME/.profile</code>	<code>\$ PATH=\$HOME/bin:./sbin:/usr/local/bin ...</code> <code>\$ export PATH</code>	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	<code>.profile</code>	<code>.\$HOME/.profile</code>

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

5 验证新路径。

```
$ which command
```

示例 3-6 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超级用户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《[Oracle Solaris 11.1 管理：安全服务](#)》中的第 7 章“控制对文件的访问（任务）”。

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 11.1 中管理网络文件系统》中的“NFS 故障排除的策略”。

其他系统和软件问题的故障排除（任务）

本章介绍可能偶然发生并且相对易于修复的其他系统和软件问题。此故障排除过程通常包括解决与特定软件应用程序或主题不相关的问题，如重新引导和整个文件系统失败。

下面列出了本章中的信息。

- 第 37 页中的“重新引导失败时应执行的操作”
- 第 38 页中的“系统挂起时应执行的操作”
- 第 39 页中的“文件系统占满时应执行的操作”
- 第 40 页中的“复制或恢复后文件 ACL 丢失时应执行的操作”

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 11.1 系统》中的“显示和设置引导属性”。
Oracle Solaris 引导归档文件受损。或者，SMF 引导归档文件服务失败。如果您运行 <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> 文件恢复的信息，请参见《引导和关闭 Oracle Solaris 11.1 系统》中的“如何从介质进行引导以解决 <code>root</code> 口令未知问题”。

系统未引导的原因	解决该问题的办法
x86 引导装载程序 (GRUB) 受损。或者，GRUB 菜单缺失或受损。	有关从受损的 x86 引导装载程序或者缺失或受损的 GRUB 菜单恢复的信息，请参见《引导和关闭 Oracle Solaris 11.1 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 口令或遇到阻止系统进行引导的问题时应执行的操作

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 11.1 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/shadow 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，承担 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 11.1 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。	登录并承担 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。	有关利用与 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。

索引

C

重新引导, 崩溃后失败, 37–38
Command not found 错误消息, 34
consadm 命令, 31–32

- 禁用辅助控制台, 33
- 启用辅助控制台, 31–32
 - 在系统重新引导期间, 32–33
- 显示辅助控制台列表 (如何), 32

core 文件, 使用 proc 工具检查, 22
coreadm 命令, 17

- 管理核心文件, 17
- 设置核心文件名称模式, 20
- 显示核心转储配置, 20

crontab 命令

- /var/adm 维护和, 26

D

dmesg 命令, 26

E

/etc/syslog.conf 文件, 27

M

mdb 实用程序, 13
messages.*n* 文件, 26
messages 文件, 23, 27

P

proc 工具, 检查核心文件, 22

S

syslog.conf 文件, 27
syslogd 守护进程, 25

U

UNIX 系统 (故障转储信息), 8
/usr/adm/messages 文件, 23
/usr/bin/mdb 实用程序, 13

V

/var/adm/messages 文件, 23, 27
/var/adm/messages.*n* 文件, 26

W

Watchdog reset! 消息, 25

崩

崩溃, 27

- 保存故障转储信息, 8
- 保存其他系统信息, 26

崩溃 (续)

- 后续过程, 23
- 检查故障转储, 13
- 客户服务和, 23
- 显示生成的系统信息, 13, 25
- 重新引导失败, 之后, 37-38

从

- 从完整的故障转储目录中恢复, 14

错

错误消息

- 崩溃消息, 26
- 定制日志, 27
- 日志文件, 23, 25
- 优先级, 29
- 与崩溃相关的, 25
- 源, 27, 28
- 指定存储位置, 25, 27, 28

定

定制

- 系统消息日志, 27
- 系统消息日志 (如何), 29

辅

- 辅助 (远程) 控制台, 30

故

- 故障消息, 25
- 故障转储目录, 从完整目录中恢复, 14

核

- 核心文件, 使用 coreadm 管理, 17
- 核心文件名称模式, 使用 coreadm 设置, 18
- 核心转储配置, 使用 coreadm 显示, 20

技

- 技术支持, 发送崩溃信息, 23

检

- 检查核心文件, 使用 proc 工具, 22

禁

- 禁用, 辅助控制台, 使用 consadm 命令, 33

警

- 警报信息优先级 (对于 syslogd), 29

客

- 客户服务, 发送崩溃信息, 23

控

控制台

辅助

- 在系统重新引导期间启用, 32-33

每

- 每进程核心文件路径, 使用 coreadm 设置, 17

启

启用

- 辅助控制台, 使用 `consadm` 命令, 31–32
- 辅助控制台, 在系统重新引导期间, 32–33

全

- 全局核心文件路径, 使用 `coreadm` 设置, 17

设

- 设置, 核心文件名称模式, 使用 `coreadm`, 20

搜

- 搜索路径, 用于设置的文件, 34

网

- 网络, 识别访问问题, 35–36

文

- 文件, 用于设置搜索路径, 34
- 文件或组的所有权, 解决文件访问问题, 35

系

系统消息

- 定制日志 (如何), 29
- 指定存储位置, 25

- 系统消息日志 (定制), 27

系统资源

监视

- 崩溃, 27

显

显示

- 崩溃信息, 25
- 故障转储信息, 13
- 核心转储配置, 使用 `coreadm`, 20
- 引导消息, 26

引

引导

- 显示生成的消息, 26

识

- 识别网络访问问题, 35–36

