



Netra™ 240 服务器系统管理指南

Sun Microsystems, Inc.
www.sun.com

部件号 817-5016-11
2004 年 7 月, 修订版 A

请将您对本文档的意见提交到: <http://www.sun.com/hwdocs/feedback>

版权所有 2004 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. 保留所有权利。

Sun Microsystems, Inc. 拥有与本文档中所述技术有关的知识产权。重点来讲（但不限于此），这些知识产权包括在 <http://www.sun.com/patents> 网站上列出的一种或多种美国专利，以及在美国和其它国家/地区注册的一种或多种其它专利或正在申请中的专利。

本文档及其所述产品的发行受限制其使用、复制、发行和反编译的许可证的制约。未经 Sun 及其许可证发行者（如果有）事先书面授权，不得以任何形式、任何方式复制本产品或文档的任何部分。

第三方软件，包括字体技术，均已从 Sun 供应商获得版权和许可。

产品的某些部件可能源于 Berkeley BSD 系统，Sun 已从 University of California 获得使用许可。UNIX 是在美国及其它国家/地区的注册商标，Sun 已从 X/Open Company, Ltd. 获得独家使用授权。

Sun、Sun Microsystems、Sun 徽标、AnswerBook2、docs.sun.com、OpenBoot、Netra、SunVTS、Sun Enterprise Authentication Mechanism 以及 Solaris 是 Sun Microsystems, Inc. 在美国和其它国家/地区的商标或注册商标。

所有 SPARC 商标都是 SPARC International, Inc. 在美国以及其它国家/地区的商标或注册商标，必须根据许可证条款使用它们。带有 SPARC 商标的产品均以 Sun Microsystems, Inc. 开发的体系结构为基础。

OPEN LOOK 和 Sun™ Graphical User Interface（图形用户界面）是 Sun Microsystems, Inc. 为其用户和许可证持有者开发的。Sun 感谢 Xerox 在用户界面形象化和图形化研发方面为计算机行业所做的先导性贡献。Sun 公司持有由 Xerox 公司颁发的对 Xerox Graphical User Interface 的非独有许可证，其许可对象还包括实施 OPEN LOOK GUI 或遵守 Sun 书面许可协议的 Sun 许可证持有者。

美国政府权利—商用。政府用户受 Sun Microsystems, Inc. 标准许可协议限制，并遵守 FAR 及其补充条款中适用的规定。

本资料按“现有形式”提供，不承担明确或隐含的条件、陈述和保证，包括对特定目的的商业活动和适用性或非侵害性的任何隐含保证，除非这种不承担责任的声明是不合法的。



Adobe PostScript

目录

序言 xi

1. 故障排除工具 1

诊断工具概述 2

系统提示符 2

Advanced Lights Out Manager 4

服务器状态指示灯 4

▼ 显示定位器 LED 的状态 5

▼ 打开定位器 LED 5

▼ 关闭定位器 LED 5

警报状态指示灯 6

加电自检诊断程序 7

控制 POST 诊断程序 8

▼ 启动 POST 诊断程序 10

OpenBoot 命令 11

probe-scsi 命令和 probe-scsi-all 命令 11

probe-ide 命令 12

show-devs 命令 13

▼ 运行 OpenBoot 命令 13

OpenBoot 诊断程序	14
▼ 启动 OpenBoot 诊断程序	14
控制 OpenBoot 诊断程序测试	15
test 命令和 test-all 命令	16
OpenBoot 诊断程序错误消息	16
操作系统诊断工具	17
错误和系统消息日志文件	17
Solaris 软件系统信息命令	18
prtconf 命令	18
prtdiag 命令	20
prtfru 命令	22
psrinfo 命令	23
showrev 命令	24
▼ 运行 Solaris 平台系统信息命令	25
最新诊断程序测试结果	25
▼ 查看最新测试结果	25
OpenBoot 配置变量	26
▼ 查看和设置 OpenBoot 配置变量	26
使用 watch-net 和 watch-net-all 命令检查网络连接情况	27
系统自动恢复	28
Auto-Boot (自动引导) 选项	28
错误处理概要	29
重置情况	30
▼ 启用 ASR	30
▼ 禁用 ASR	31
2. SunVTS 软件	33
SunVTS 软件概述	33
SunVTS 测试程序	34

SunVTS 软件和安全	35
▼ 确定是否安装了 SunVTS 软件	35
安装 SunVTS 软件	35
查看 SunVTS 软件文档	36
3. Advanced Lights Out Manager	37
Advanced Lights Out Manager 概述	37
ALOM 端口	38
设置 admin 口令	39
ALOM 基本功能	39
▼ 切换到 ALOM 提示符	40
▼ 切换到服务器控制台提示符	40
▼ 取消另一用户的控制台写入权限	40
Automatic Server Restart	41
环境监视与控制	41
A. 警报继电器输出应用编程接口	45
索引	51

图

- 图 1-1 系统提示符流程 3
- 图 1-2 前面板指示灯的位置 4

表

表 1-1	故障排除工具概述	2
表 1-2	服务器状态指示灯（前面和后面）	4
表 1-3	警报指示灯和干接点警报状态	6
表 1-4	OpenBoot 配置变量	8
表 1-5	test-args OpenBoot 配置变量的关键字	15
表 1-6	Solaris 平台信息显示命令	25
表 2-1	SunVTS 软件测试	34
表 3-1	ALOM 监视的组件	38
表 3-2	Netra 240 服务器机箱温度阈值	42

序言

《Netra 240 服务器系统管理指南》的读者对象是有经验的系统管理员。它提供了有关 Netra™ 240 服务器诊断工具的一般说明，并介绍了各种服务器管理任务。

要使用本手册中介绍的信息，用户必须熟悉有关计算机网络概念和术语方面的应用知识，而且还应精通 Solaris™ 操作系统 (Solaris OS)。

阅读本书之前

本书不包括有关服务器安装和机架安装的说明。有关这些主题的详细信息，请参阅《Netra 240 服务器安装指南》（部件号 817-4999-11）。

执行本书介绍的任何步骤之前，请确保已阅读《Important Safety Information for Sun Hardware Systems》（部件号 816-7190-10）。

使用 UNIX 命令

本节用于提示读者文中并没有提供所有的 UNIX 命令。

例如：

本文档可能不包括有关基本 UNIX® 命令和过程（如关闭系统、引导系统和配置设备等）的信息。有关这些信息，请参阅以下文档：

- 系统附带的软件文档
- Solaris™ 操作系统文档
<http://docs.sun.com>

Shell 提示符

Shell	提示符
C shell	机器名 %
C shell 超级用户	机器名 #
Bourne shell 和 Korn shell	\$
Bourne shell 和 Korn shell 超级用户	#

印刷约定

字体 ⁱ	含义	示例
AaBbCc123	命令、文件以及目录的名称；计算机屏幕输出	编辑 <code>.login</code> 文件。 使用 <code>ls -a</code> 可列出所有文件。 % You have mail.
AaBbCc123	键入的内容（相对于计算机的屏幕输出）	% su Password:
<i>AaBbCc123</i>	书名、新词或术语、需要强调的词使用真名或实际值替换命令行变量。	请阅读 <i>用户指南</i> 的第6章。 这些被称为 <i>类选项</i> 。 要执行该操作，您 <i>必须是</i> 超级用户。 要删除文件，请键入 <code>rm 文件名</code> 。

ⁱ 您的浏览器中的设置可能与此处的设置不同。

相关文档

应用	标题	部件号
安装概述	<i>Netra 240 Server Quick Start Guide</i>	817-3904-xx
最新产品更新	<i>Netra 240 Server Release Notes</i>	817-3142-xx
符合标准和安全信息	<i>Important Safety Information for Sun Hardware Systems</i>	816-7190-10
	<i>Netra 240 Server Safety and Compliance Manual</i>	817-5018-11
文档 Web 站点位置	<i>Sun Netra 240 Server Product Documentation</i>	817-2697-10
安装	<i>Netra 240 服务器安装指南</i>	817-4999-11
Lights-out 管理	<i>Sun Advanced Lights Out Manager 软件用户指南 (适用于 Netra 240 服务器)</i>	817-5008-11
维修	<i>Netra 240 Server Service Manual</i>	817-2699-xx

访问 Sun 文档

用户可通过以下网站查看、打印或订购 Sun 提供的各类文档（包括本地化版本）：

<http://www.sun.com/documentation>

第三方 Web 站点

Sun 不保证本文档中所提及的第三方 Web 站点一定可供访问。对于这些站点或资源提供的任何内容、广告、产品或其它资料，Sun 既不担保也不承担任何责任和义务。对于因使用或依据这些站点或资源提供的任何内容、产品或服务而造成的、或与这些内容相关的任何实际或声称的损害或损失，Sun 均不承担任何责任。

联系 Sun 技术支持部门

如果通过阅读本文档不能解决您在使用本产品时遇到的技术问题，请转至：

<http://www.sun.com/service/contacting>

Sun 欢迎您提出意见

Sun 十分注重改进自身文档的质量，并欢迎您提出宝贵的意见和建议。您可以通过以下地址来提交意见：

<http://www.sun.com/hwdocs/feedback>

请在您的反馈信息中标明文档的名称和部件号：

Netra 240 服务器系统管理指南，部件号 817-5016-11

故障排除工具

本章介绍可供 Netra 240 服务器使用的诊断工具。本章包括以下几节：

- 第 2 页上的 “诊断工具概述”
- 第 2 页上的 “系统提示符”
- 第 4 页上的 “Advanced Lights Out Manager”
- 第 7 页上的 “加电自检诊断程序”
- 第 11 页上的 “OpenBoot 命令”
- 第 14 页上的 “OpenBoot 诊断程序”
- 第 17 页上的 “操作系统诊断工具”
- 第 25 页上的 “最新诊断程序测试结果”
- 第 26 页上的 “OpenBoot 配置变量”
- 第 28 页上的 “系统自动恢复”

诊断工具概述

Sun 提供了多种可供 Netra 240 服务器使用的诊断工具，下表对它们进行了汇总。

表 1-1 故障排除工具概述

诊断工具	键入:	说明	可访问性和可用性	远程能力
ALOM	硬件和软件	监控环境状况，隔离基本故障，并可访问远程控制台。	可以使用备用电源运行，且无需操作系统。	可以远程访问。
LED	硬件	指示整个系统和特定组件的状态。	从系统机箱上查看。只要接通电源便可工作。	本地，但可利用 ALOM 查看。
加电自检 (POST)	固件	测试系统的核心组件。	在系统启动时自动运行。无需运行操作系统即可使用。	本地，但可利用 ALOM 查看。
OpenBoot 命令	固件	显示各种系统信息。	无需运行操作系统即可使用。	本地，可利用 ALOM 访问。
OpenBoot 诊断程序	固件	测试系统组件，侧重点放在外围设备和 I/O 设备上。	自动或交互运行。无需运行操作系统即可使用。	本地，但可利用 ALOM 查看。
Solaris 软件命令	软件	显示各种系统信息。	需要运行操作系统	本地，可利用 ALOM 访问。
SunVTS™ 软件	软件	通过同时运行多个测试来检查系统并对系统进行负荷测试。	需要运行操作系统可选软件包。	可通过网络查看和控制。

系统提示符

Netra 240 服务器使用以下默认服务器提示符：

- `ok` — OpenBoot PROM 提示符
- `sc>` — Advanced Lights Out Manager (ALOM) 提示符
- `#` — Solaris 软件超级用户（Bourne 和 Korn shell）提示符

图 1-1 显示了这三种提示符之间的关系以及如何切换它们。

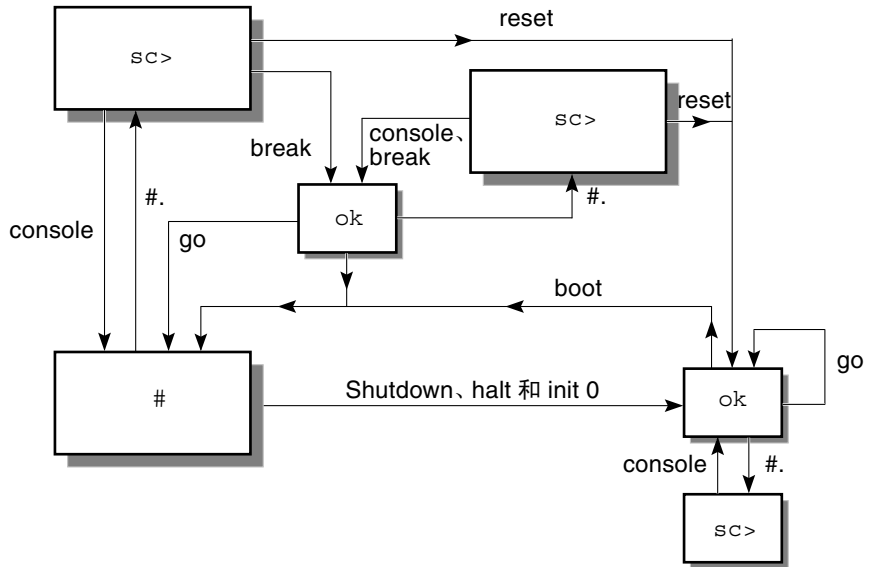


图 1-1 系统提示符流程

图 1-1 的流程图中显示了以下命令：

- ALOM 命令：console、reset 和 break
- 转义符序列：#.
- Solaris 软件命令：shutdown、halt 和 init 0
- OpenBoot 命令：go 和 boot

Advanced Lights Out Manager

Netra 240 服务器中的 Sun™ Advanced Lights Out Manager (ALOM) 提供了一系列 LED 状态指示灯。本节详细介绍了 LED 状态的含义，以及如何打开和关闭这些指示灯。有关 ALOM 的详细信息，请参阅第 3 章。

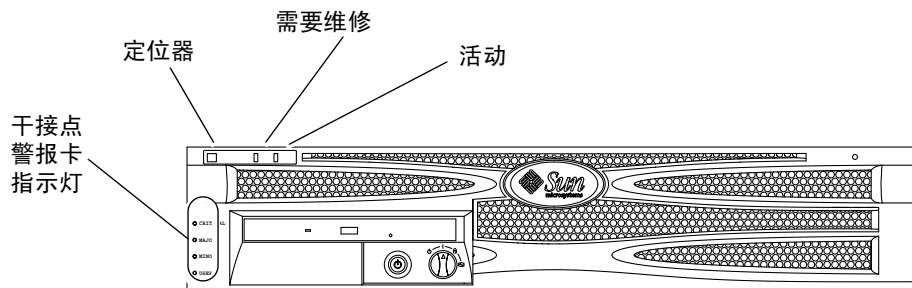


图 1-2 前面板指示灯的位置

服务器状态指示灯

服务器上有三个 LED 状态指示灯。它们位于前面板上（图 1-2）；后面板上也有三个同样的指示灯。表 1-2 简要介绍了这些指示灯。

表 1-2 服务器状态指示灯（前面和后面）

指示灯	LED 颜色	LED 状态	含义
活动	绿色	亮起	服务器接通电源且正在运行 Solaris 操作系统。
		熄灭	未接通电源，或未运行 Solaris 操作系统。
需要维修	黄色	亮起	服务器已检测到问题并要求维修人员进行检查。
		熄灭	服务器未检测到故障。
定位器	白色	亮起	使用 <code>setlocator</code> 命令时，指示灯会持续发光，从而可以从机架中识别出服务器。

您可以从系统控制台或 ALOM 命令行界面 (CLI) 打开或关闭定位器 LED。

▼ 显示定位器 LED 的状态

- 执行以下步骤之一：
 - 以超级用户的身份键入：

```
# /usr/sbin/locator
```

- 在 ALOM 命令行界面中键入：

```
sc> showlocator
```

▼ 打开定位器 LED

- 执行以下步骤之一：
 - 以超级用户的身份键入：

```
# /usr/sbin/locator -n
```

- 在 ALOM 命令行界面中键入：

```
sc> setlocator on
```

▼ 关闭定位器 LED

- 执行以下步骤之一：
 - 以超级用户的身份键入：

```
# /usr/sbin/locator -f
```

- 在 ALOM 命令行界面中键入：

```
sc> setlocator off
```

警报状态指示灯

干接点警报卡上有四个 ALOM 支持的 LED 状态指示灯。它们垂直分布在前面板上（图 1-2）。表 1-3 中提供了警报指示灯和干接点警报状态的相关信息。有关警报指示灯的详细信息，请参阅《Sun Advanced Lights Out Manager 软件用户指南（适用于 Netra 240 服务器）》（部件号 817-5008-11）。有关通过 API 控制警报指示灯的详细信息，请参阅附录 A。

表 1-3 警报指示灯和干接点警报状态

指示灯和继电器标签	指示灯颜色	应用程序或服务器状态	运行状况或操作	系统指示灯状态	警报指示灯状态	继电器 NC ^{iv} 状态	继电器 NO ^v 状态	备注	
紧急 (警报 0)	红色	服务器状态 (接通/断开电源和 Solaris 操作系统运行正常/运行不正常)	无电源。	熄灭	熄灭	断开	接通	默认状态。	
			系统电源关闭。	熄灭	熄灭 ⁱⁱⁱ	断开	接通	电源已接通。	
			系统电源接通；Solaris 操作系统未完全加载。	熄灭	熄灭 ⁱⁱⁱ	断开	接通	瞬态。	
			Solaris 操作系统已成功加载。	亮起	熄灭	接通	断开	正常操作状态。	
			软件狗超时。	熄灭	亮起	断开	接通	瞬态；重新引导 Solaris 操作系统。	
			用户关闭了 Solaris 操作系统 ⁱ 。	熄灭	熄灭 ⁱⁱⁱ	断开	接通	瞬态。	
			电源断电。	熄灭	熄灭	断开	接通	默认状态。	
			用户关闭系统电源。	熄灭	熄灭 ⁱⁱⁱ	断开	接通	瞬态。	
			应用程序状态	用户打开“发出紧急警报”功能 ⁱⁱ 。	—	亮起	断开	接通	检测到紧急故障。
			用户关闭“发出紧急警报”功能 ⁱⁱ 。	—	熄灭	接通	断开	紧急故障已清除。	
重要 (警报 1)	红色	应用程序状态	用户打开“发出重要警报”功能 ⁱⁱ 。	—	亮起	接通	断开	检测到重要故障。	
			用户关闭“发出重要警报”功能 ⁱⁱ 。	—	熄灭	断开	接通	重要故障已清除。	

表 1-3 警报指示灯和干接点警报状态 (续)

指示灯和继电器标签	指示灯颜色	应用程序或服务器状态	运行状况或操作	系统指示灯状态	警报指示灯状态	继电器 NC ^{iv} 状态	继电器 NO ^v 状态	备注
次要 (警报 2)	琥珀色	应用程序状态	用户打开“发出次要警报”功能 ii 。	—	亮起	接通	断开	检测到次要故障。
			用户关闭“发出次要警报”功能 ii 。	—	熄灭	断开	接通	次要故障已清除。
用户 (警报 3)	琥珀色	应用程序状态	用户打开“发出用户警报”功能 ii 。	—	亮起	接通	断开	检测到用户故障。
			用户关闭“发出用户警报”功能 ii 。	—	熄灭	断开	接通	用户故障已清除。

i 用户可以使用 `init0` 和 `init6` 之类的命令来关闭系统。该操作不会关闭系统电源。

ii 根据对故障情况的判断，用户可以通过使用 Solaris 平台警报 API 或 ALOM CLI 来打开警报功能。有关警报 API 的详细信息，请参阅附录 A，有关 ALOM CLI 的详细信息，请参阅《Sun Advanced Lights Out Manager 软件用户指南 (适用于 Netra 240 服务器)》(部件号 817-5008-11)。

iii 此警报指示灯状态的指示情况可能会改变。

iv NC 状态指的是常闭状态。该状态代表常闭继电器接点的默认模式。

v NO 状态指的是常开状态。该状态代表常开继电器接点的默认模式。

如果用户设置了警报，则所有情况下，控制台上都会显示消息。例如，设置紧急警报后，控制台上会显示以下消息：

```
SC Alert:CRITICAL ALARM is set
```

注意，在某些情况下，尽管设置了紧急警报，相关警报指示灯也不会发亮。这种情况在以后的版本中会有所改变。(请参阅表 1-3 的脚注 [iii](#))。

加电自检诊断程序

加电自检 (POST) 是一个固件程序，可用于确定系统中的某个部件是否出现故障。POST 可以验证系统的核心功能，包括 CPU 模块、主板、内存和某些板载 I/O 设备。该软件还可以生成用于确定硬件故障类型的消息。即使系统不能引导，POST 也可运行。

POST 驻留在主板的 OpenBoot PROM 中，可以检测大多数系统故障。通过设置两个环境变量 `diag-switch?` 和 `diag-level` 标记，您可以安排 OpenBoot 软件在开机时运行 POST。这两个变量存储在系统配置卡内。

如果满足以下所有条件，POST 会在系统开机时或在系统自动重置之后自动运行：

- `diag-switch?` 设为 `true`（默认值为 `false`）。
- `diag-level` 设为 `min`、`max` 或 `menus`（默认值为 `min`）。
- `post-trigger` 与重置类型相符（默认值为 `power-on-reset`）。

如果 `diag-level` 设为 `min` 或 `max`，POST 会相应执行简化或更全面的测试。

如果 `diag-level` 设为 `menus`，则会显示一个菜单，其中列出可在开机时执行的所有测试。

POST 诊断和错误消息报告显示在控制台上。

控制 POST 诊断程序

您可以通过设置 OpenBoot 配置变量来控制 POST 诊断程序的运行（和引导过程的其他方面）。只有在重新启动系统后，对 OpenBoot 配置变量所作的更改才会生效。表 1-4 列出了最重要并且最有用的那些变量。第 26 页上的“[查看和设置 OpenBoot 配置变量](#)”介绍了如何更改 OpenBoot 配置变量。

表 1-4 OpenBoot 配置变量

OpenBoot 配置变量	说明和关键字
<code>auto-boot</code>	确定操作系统是否自动启动。默认值为 <code>true</code> 。 <ul style="list-style-type: none">• <code>true</code> — 固件测试完成后操作系统自动启动。• <code>false</code> — 除非键入 <code>boot</code> 命令，否则系统将停留在 <code>ok</code> 提示符状态下。
<code>diag-level</code>	确定所执行的诊断程序的级别和类型。默认值为 <code>min</code> 。 <ul style="list-style-type: none">• <code>off</code> — 不运行测试。• <code>min</code> — 仅运行基本测试。• <code>max</code> — 运行更全面的测试，具体取决于设备。• <code>menus</code> — 可以单独运行 POST 级别的菜单驱动测试。
<code>diag-script</code>	确定 OpenBoot 诊断程序测试哪些设备。默认值为 <code>none</code> 。 <ul style="list-style-type: none">• <code>none</code> — 不测试任何设备。• <code>normal</code> — 测试那些带有自测程序的板载（基于主板的）设备。• <code>all</code> — 测试所有带有自测程序的设备。
<code>diag-switch?</code>	使系统进入或退出诊断模式。默认值为 <code>false</code> 。 <ul style="list-style-type: none">• <code>true</code> — 诊断模式：运行 POST 诊断程序和 OpenBoot 诊断程序测试。• <code>false</code> — 默认模式：不运行 POST 或 OpenBoot 诊断程序测试。

表 1-4 OpenBoot 配置变量 (续)

OpenBoot 配置变量	说明和关键字
post-trigger obdiag-trigger	<p>这两个变量指定可以导致 POST (或 OpenBoot 诊断程序测试) 运行的重置事件的类型。这些变量可以接受单个关键字, 也可接受前三个关键字的组合 (由空格隔开)。有关详细信息, 请参阅第 26 页上的“查看和设置 OpenBoot 配置变量”。</p> <ul style="list-style-type: none"> • error-reset — 由某些不可恢复的硬件错误情况引起的重置。一般情况下, 当硬件问题导致系统状态数据被损坏时, 就会发生由于错误而导致的重置。此类示例包括 CPU 和系统软件狗重置、致命错误以及某些特定的 CPU 重置事件 (默认)。 • power-on-reset — 按 On/Standby (开机/待机) 按钮时导致的重置 (默认)。 • user-reset — 由用户或操作系统启动的重置。 • all-resets — 任何类型的系统重置。 • none — 不运行 POST (或 OpenBoot 诊断程序测试)。
input-device	<p>选择从何处获取控制台输入。默认值为 ttya。</p> <ul style="list-style-type: none"> • ttya — 从内置 SERIAL MGT 端口。 • ttyb — 从内置通用串行端口 (10101)。 • keyboard — 从所连接的属于图形终端一部分的键盘。
output-device	<p>选择在何处显示诊断程序和其它控制台输出。默认值为 ttya。</p> <ul style="list-style-type: none"> • ttya — 输出到内置 SERIAL MGT 端口。 • ttyb — 输出到内置通用串行端口 (10101)。 • screen — 输出到所连接的属于图形终端一部分的屏幕。ⁱ

i POST 消息不能在图形终端上显示。即使将 output-device 设置为 screen, 它们也会被发送到 ttya。

注 – 这些变量既会影响 OpenBoot 诊断程序测试, 也会影响 POST 诊断程序。

一旦 POST 诊断程序运行完毕, 就会向 OpenBoot 固件报告它所运行的每一个测试的状态。随后, 控制权交回至 OpenBoot 固件代码。

如果 POST 诊断程序没有发现故障, 而您的服务器仍不能启动, 请运行 OpenBoot 诊断程序测试。

▼ 启动 POST 诊断程序

1. 切换到 `ok` 提示符下。
2. 键入以下命令：

```
ok setenv diag-switch? true
```

3. 键入以下命令：

```
ok setenv diag-level value
```

其中，*value* 既可以是 `min`、`max`，也可以是 `menus`，具体设置取决于您要查看的诊断信息量。

4. 键入以下命令：

```
ok reset-all
```

如果 `post-trigger` 设为 `user-reset`，则系统会运行 POST 诊断程序，并在控制台窗口中显示状态和错误消息。如果 POST 检测到错误，它会显示一则描述故障的错误消息。

5. 运行 POST 之后，键入以下命令，将 `diag-switch?` 的值改回 `false`：

```
ok setenv diag-switch? false
```

将 `diag-switch?` 的值改回 `false` 可以最大程度地缩短引导时间。

OpenBoot 命令

OpenBoot 命令是指您在 ok 提示符下键入的命令。可以提供有用诊断信息的 OpenBoot 命令包括：

- probe-scsi 和 probe-scsi-all
- probe-ide
- show-devs

probe-scsi 命令和 probe-scsi-all 命令

probe-scsi 和 probe-scsi-all 命令用于诊断 SCSI 设备问题。



注意 – 如果曾使用 halt 命令或 Stop-A 组合键来进入 ok 提示符状态，那么发出 probe-scsi 或 probe-scsi-all 命令会使系统挂起。

probe-scsi 命令可与连接到板载 SCSI 控制器的所有 SCSI 设备通信。probe-scsi-all 命令还可以访问与 PCI 插槽中所安装的任何主适配器相连的设备。

对于任何连接的且处于活动状态的 SCSI 设备，probe-scsi 和 probe-scsi-all 命令均会显示它的环路 ID、主机适配器、逻辑单元号、唯一的全球通用名 (WWN) 以及设备说明（包括设备类型和制造商）。

以下是 probe-scsi 命令的输出示例。

代码示例 1-1 probe-scsi 命令的输出

```
{1} ok probe-scsi
Target 0
  Unit 0   Disk       SEAGATE ST373307LSUN72G 0207
Target 1
  Unit 0   Disk       SEAGATE ST336607LSUN36G 0207
{1} ok
```

以下是 probe-scsi-all 命令的输出示例。

代码示例 1-2 probe-scsi-all 命令的输出

```
{1} ok probe-scsi-all
/pci@1c,600000/scsi@2,1

/pci@1c,600000/scsi@2
Target 0
  Unit 0   Disk      SEAGATE ST373307LSUN72G 0207
Target 1
  Unit 0   Disk      SEAGATE ST336607LSUN36G 0207

{1} ok
```

probe-ide 命令

probe-ide 命令与连接到 IDE 总线的所有集成电路 (IDE) 设备进行通信。该总线是用于介质设备（例如 DVD 驱动器）的内部系统总线。



注意 – 如果曾使用 halt 命令或 Stop-A 组合键来进入 ok 提示符状态，那么发出 probe-ide 命令会使系统挂起。

以下是 probe-ide 命令的输出示例。

代码示例 1-3 probe-ide 命令的输出

```
{1} ok probe-ide
Device 0 ( Primary Master )
      Not Present

Device 1 ( Primary Slave )
      Not Present

Device 2 ( Secondary Master )
      Not Present

Device 3 ( Secondary Slave )
      Not Present

{1} ok
```

show-devs 命令

show-devs 命令列出固件设备树中各设备的硬件设备路径。[代码示例 1-4](#) 显示了部分输出示例。

代码示例 1-4 show-devs 命令的输出

```
/pci@1d,700000
/pci@1c,600000
/pci@1e,600000
/pci@1f,700000
/memory-controller@1,0
/SUNW,UltraSPARC-IIIi@1,0
/memory-controller@0,0
/SUNW,UltraSPARC-IIIi@0,0
/virtual-memory
/memory@m0,0
/aliases
/options
/openprom
/chosen
/packages
/pci@1d,700000/network@2,1
/pci@1d,700000/network@2
/pci@1c,600000/scsi@2,1
/pci@1c,600000/scsi@2
/pci@1c,600000/scsi@2,1/tape
/pci@1c,600000/scsi@2,1/disk
/pci@1c,600000/scsi@2/tape
/pci@1c,600000/scsi@2/disk
/pci@1e,600000/ide@d
/pci@1e,600000/usb@a
/pci@1e,600000/pmu@6
/pci@1e,600000/isa@7
/pci@1e,600000/ide@d/cdrom
/pci@1e,600000/ide@d/disk.....
```

▼ 运行 OpenBoot 命令

1. 中止系统以切换到 ok 提示符下。
应在关闭系统之前通知用户。
2. 在控制台提示符下，键入所需的命令。

OpenBoot 诊断程序

与 POST 诊断程序类似，OpenBoot 诊断程序代码也基于固件，且驻留在 Boot PROM 中。

▼ 启动 OpenBoot 诊断程序

1. 键入：

```
ok setenv diag-switch? true
ok setenv auto-boot? false
ok reset-all
```

2. 键入：

```
ok obdiag
```

此命令用于显示 OpenBoot 诊断程序菜单。

```
ok obdiag
```

```
o b d i a g
```

1 i2c@0,320	2 ide@d	3 network@2
4 network@2,1	5 rtc@0,70	6 scsi@2
7 scsi@2,1	8 serial@0,2e8	9 serial@0,3f8
10 usb@a	11 usb@b	12 flashprom@2,0

```
Commands: test test-all except help what printenvs setenv versions exit
```

注 – 如果您的服务器内安装了 PCI 卡，则 obdiag 菜单中还会显示其它测试。

3. 键入以下命令：

```
obdiag> test n
```

其中，*n* 表示您要运行的测试所对应的编号。

系统提供了各个测试的简要说明。在 obdiag> 提示符下键入：

```
obdiag> help
```

控制 OpenBoot 诊断程序测试

您用来控制 POST 的大多数 OpenBoot 配置变量（请参阅表 1-4）也可用于控制 OpenBoot 诊断程序测试。

- 使用 `diag-level` 变量可以控制 OpenBoot 诊断程序的测试级别。
- 使用 `test-args` 变量可以自定义测试的运行方式。

默认情况下，`test-args` 设置为包含一个空的字符串。您可以使用表 1-5 中列出的一个或多个保留关键字来修改 `test-args`。

表 1-5 test-args OpenBoot 配置变量的关键字

关键字	说明
bist	调用外部设备和外围设备上的内置自测程序 (BIST)。
debug	显示所有调试消息。
iopath	验证总线 and 互连完整性。
loopback	检查设备的外部环路。
media	验证外部设备和外围设备介质的可访问性。
restore	尝试恢复上次未通过测试的设备的原始状态。
silent	仅显示错误而不显示每个测试的状态。
subtests	显示主测试程序和所调用的每一个子测试程序。
verbose	显示所有测试的详细状态消息。
callers= <i>n</i>	显示出错之前的 <i>N</i> 个调用程序的历史记录。 callers=0 — 显示对出现错误前的所有调用的追踪信息。
errors= <i>n</i>	继续执行测试，直到出现 <i>N</i> 个错误。 errors=0 — 显示所有错误报告，但不终止测试。

如果要对 OpenBoot 诊断程序的测试进行自定义，可以将 `test-args` 设置为一个以逗号分隔的关键字列表，如下例所示：

```
ok setenv test-args debug,loopback,media
```

test 命令和 test-all 命令

您还可以在 `ok` 提示符下直接运行 OpenBoot 诊断程序测试。要执行此操作，请键入 `test` 命令，然后键入所要测试设备（或一组设备）的完整硬件路径。例如：

```
ok test /pci@x,y/SUNW,qlc@2
```

要自定义单个测试，请按以下方法使用 `test-args`：

```
ok test /usb@1,3:test-args={verbose,debug}
```

此语法只影响当前测试，而不会更改 `test-args` OpenBoot 配置变量的值。

使用 `test-all` 命令可以测试设备树中的所有设备：

```
ok test-all
```

如果为 `test-all` 指定了一个路径参数，则将只测试指定的设备及其子设备。下面的命令示例用于测试 USB 总线以及所有连接到 USB 总线的带有自测程序的设备：

```
ok test-all /pci@9,700000/usb@1,3
```

OpenBoot 诊断程序错误消息

OpenBoot 诊断程序的错误结果以表格的形式出现，其中包含问题简要说明、受影响的硬件设备、失败的子测试以及其它诊断信息。[代码示例 1-5](#) 显示了 OpenBoot 诊断程序错误消息的一个示例。

代码示例 1-5 OpenBoot 诊断程序错误消息

```
Testing /pci@1e,600000/isa@7/flashprom@2,0

ERROR   : FLASHPROM CRC-32 is incorrect
SUMMARY : Obs=0x729f6392 Exp=0x3d6cdf53 XOR=0x4ff3bcc1 Addr=0xfeebbfcc
DEVICE  : /pci@1e,600000/isa@7/flashprom@2,0
SUBTEST : selftest:crc-subtest
MACHINE : Netra 240
SERIAL#  : 52965531
DATE    : 03/05/2003 01:33:59 GMT
CONTROLS: diag-level=max test-args=

Error: /pci@1e,600000/isa@7/flashprom@2,0 selftest failed, return code = 1
Selftest at /pci@1e,600000/isa@7/flashprom@2,0 (errors=1) .....
failed
Pass:1 (of 1) Errors:1 (of 1) Tests Failed:1 Elapsed Time: 0:0:0:27
```

操作系统诊断工具

通过 OpenBoot 诊断程序测试后，系统会尝试引导 Solaris 操作系统。一旦服务器以多用户模式运行，您即可访问基于软件的诊断工具和 SunVTS 软件。您可以使用这些工具监控服务器、检查服务器以及确定服务器的故障。

注 – 如果您将 OpenBoot 配置变量 auto-boot? 设为 false，则在完成基于固件的测试之后，操作系统不会进行引导。

除以上提及的工具之外，您还可以参考错误和系统消息日志文件以及 Solaris 软件信息命令。

错误和系统消息日志文件

错误消息和其它系统消息保存在 /var/adm/messages 文件中。此文件中记录的消息来源于多个方面，包括操作系统、环境控制子系统和各种软件应用程序。

Solaris 软件系统信息命令

您可以使用以下 Solaris 软件系统信息命令所显示的数据来评估 Netra 240 服务器的运行状况：

- prtconf
- prtdiag
- prtfru
- psrinfo
- showrev

本节介绍这些命令提供的信息。有关使用这些命令的详细信息，请参阅相应的手册页。

prtconf 命令

prtconf 命令显示 Solaris 软件设备树。此设备树包括由 OpenBoot 固件检测到的所有设备，以及其它只能由操作系统软件识别的设备，如单个磁盘等。prtconf 的输出中还包括系统的总内存量。[代码示例 1-6](#) 显示了 prtconf 输出的部分摘要。

代码示例 1-6 prtconf 命令的输出

```
# prtconf

System Configuration: Sun Microsystems sun4u
Memory size: 5120 Megabytes
System Peripherals (Software Nodes):

SUNW,Netra-240
  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)
    kbd-translator (driver not attached)
    obp-tftp (driver not attached)
    SUNW,i2c-ram-device (driver not attached)
    SUNW,fru-device (driver not attached)
    ufs-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)
  SUNW,UltraSPARC-IIIi (driver not attached)
  memory-controller, instance #0
  SUNW,UltraSPARC-IIIi (driver not attached)
  memory-controller, instance #1
  pci, instance #0.....
```

prtconf 命令的 -p 选项生成的输出类似于 OpenBoot show-devs 命令的输出。此输出只列出了系统固件所统计的设备。

prtdiag 命令

prtdiag 命令显示一个诊断信息表，其中概述了系统各组件的状态。prtdiag 命令采用的显示格式因系统中运行的 Solaris 操作系统版本而异。以下代码示例是在正常运行 Solaris 软件的 Netra 240 服务器上执行 prtdiag 命令后所生成的输出的部分摘要。

代码示例 1-7 prtdiag 命令输出

```
# prtdiag
System Configuration: Sun Microsystems sun4u Netra 240
System clock frequency: 160 MHz
Memory size: 2GB

===== CPUs =====
      CPU  Freq      E$      CPU      CPU      Temperature      Fan
      CPU  Freq      Size      Impl.  Mask      Die      Ambient      Speed  Unit
      ---  ---      ---      ---      ---      ---      ---      ---      ---
      MB/P0 1280 MHz  1MB      US-IIIi  2.3      -      -
      MB/P1 1280 MHz  1MB      US-IIIi  2.3      -      -

===== IO Devices =====
      Bus  Freq
      Brd  Type  MHz  Slot      Name
      ---  ---  ---  ---      ---
      0    pci   66    2    network-pci14e4,1648.108e.16+
      0    pci   66    2    network-pci14e4,1648.108e.16+
      0    pci   66    2    scsi-pci1000,21.1000.1000.1 +
      0    pci   66    2    scsi-pci1000,21.1000.1000.1 +
      0    pci   66    2    network-pci14e4,1648.108e.16+
      0    pci   66    2    network-pci14e4,1648.108e.16+
      0    pci   33    7    isa/serial-su16550 (serial)
      0    pci   33    7    isa/serial-su16550 (serial)
      0    pci   33    7    isa/rmc-comm-rmc_comm (seria+
      0    pci   33    13   ide-pci10b9,5229.c4 (ide)

===== Memory Configuration =====
Segment Table:
-----
Base Address      Size      Interleave Factor  Contains
-----
0x0               1GB      1                  GroupID 0
0x1000000000      1GB      1                  GroupID 0
```

代码示例 1-7 prtdiag 命令输出 (续)

```
Memory Module Groups:
-----
ControllerID  GroupID  Labels
-----
0              0        MB/P0/B0/D0,MB/P0/B0/D1
Memory Module Groups:
-----
ControllerID  GroupID  Labels
-----
1              0        MB/P1/B0/D0,MB/P1/B0/D1
```

除代码示例 1-7 中的信息之外，带有 verbose 选项 (-v) 的 prtdiag 命令还报告前面板状态、磁盘状态、风扇状态、电源、硬件版本和系统温度（请参阅代码示例 1-8）。

代码示例 1-8 带有冗余选项的 prtdiag 的输出

```
-----
Location      Sensor      Temperature  Lo LoWarn HiWarn  Hi Status
-----
MB            T_ENC       22C         -7C  -5C   55C   58C  okay
MB/P0        T_CORE      57C         -    -    110C  115C  okay
MB/P1        T_CORE      54C         -    -    110C  115C  okay
PS0          FF_OT       -           -    -    -     -    okay
PS1          FF_OT       -           -    -    -     -    okay
```

如果发生温度过高的情况，prtdiag 会在“Status”（状态）栏内报告错误消息（代码示例 1-9）。

代码示例 1-9 prtdiag 温度过高指示输出

```
-----
Location      Sensor      Temperature  Lo LoWarn HiWarn  Hi Status
-----
MB            T_ENC       22C         -7C  -5C   55C   58C  okay
MB/P0        T_CORE      118C        -    -    110C  115C  failed
MB/P1        T_CORE      112C        -    -    110C  115C  warning
PS0          FF_OT       -           -    -    -     -    okay
PS1          FF_OT       -           -    -    -     -    okay
```

同样，如果某个特定组件出现故障，prtdiag 也会在相应的“Status”（状态）栏内报告故障消息（代码示例 1-10）。

代码示例 1-10 prtdiag 的输出指示出现了故障

```
Fan Speeds:
-----
Location      Sensor      Status      Speed
-----
MB/P0/F0      RS          failed      0 rpm
MB/P0/F1      RS          okay        3994 rpm
F2            RS          okay        2896 rpm
PS0           FF_FAN      okay
F3            RS          okay        2576 rpm
PS1           FF_FAN      okay
-----
```

prtfru 命令

Netra 240 服务器保留了有关系统中所有现场可换部件 (FRU) 的层次列表, 以及各个 FRU 的特定信息。

prtfru 命令可以显示此分层列表, 以及许多 FRU 上的“串行、可电子擦除、可编程、只读存储器”(SEEPROM) 设备中包含的数据。代码示例 1-11 显示了带有 -l 选项的 prtfru 命令所生成的 FRU 分层列表的摘要。

代码示例 1-11 prtf -l 命令输出

```
# prtf -l
/frutree
/frutree/chassis (fru)
/frutree/chassis/MB?Label=MB
/frutree/chassis/MB?Label=MB/system-board (container)
/frutree/chassis/MB?Label=MB/system-board/SC?Label=SC
/frutree/chassis/MB?Label=MB/system-board/SC?Label=SC/sc (fru)
/frutree/chassis/MB?Label=MB/system-board/BAT?Label=BAT
/frutree/chassis/MB?Label=MB/system-board/BAT?Label=BAT/battery (fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu (fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F0?Label=F0
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F0?Label=F0/fan-unit
(fru)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F1?Label=F1
/frutree/chassis/MB?Label=MB/system-board/P0?Label=P0/cpu/F1?Label=F1/fan-unit
(fru).....
```

代码示例 1-12 显示了带有 `-c` 选项的 `prtfriu` 命令所生成的 SEEPRO 数据的摘要。此输出只显示容器及其数据，并不会显示 FRU 树状分层结构。

代码示例 1-12 `prtfriu -c` 命令输出

```
# prtfriu -c
/frutree/chassis/MB?Label=MB/system-board (container)
  SEGMENT: SD
    /ManR
    /ManR/UNIX_Stamp32: Mon Dec  2 19:47:38 PST 2002
    /ManR/Fru_Description: FRUID, INSTR, M'BD, 2X1.28GHZ, CPU
    /ManR/Manufacture_Loc: Hsinchu, Taiwan
    /ManR/Sun_Part_No: 3753120
    /ManR/Sun_Serial_No: 000615
    /ManR/Vendor_Name: Mitac International
    /ManR/Initial_HW_Dash_Level: 02
    /ManR/Initial_HW_Rev_Level: 0E
    /ManR/Fru_Shortname: MOTHERBOARD
    /SpecPartNo: 885-0076-11
/frutree/chassis/MB?Label=MB/system-board/P0?Label=
P0/cpu/B0?Label=B0/bank/D0?Label=D0/mem-module (container)
/frutree/chassis/MB?Label=MB/system-board/P0?Label=
P0/cpu/B0?Label=B0/bank/D1?Label=D1/mem-module (container).....
```

`prtfriu` 命令所显示的数据会因 FRU 的类型而异。通常包括以下内容：

- FRU 说明
- 制造商名称和地址
- 部件号和系列号
- 硬件版本级别

psrinfo 命令

`psrinfo` 命令显示各个 CPU 的联机日期和时间。如果带有 `verbose (-v)` 选项，该命令将显示有关 CPU 的其它信息，其中包括时钟速度。代码示例 1-13 显示了带 `-v` 选项的 `psrinfo` 命令的输出示例。

代码示例 1-13 psrinfo -v 命令的输出

```
# psrinfo -v
Status of processor 0 as of: 07/28/2003 14:43:29
  Processor has been on-line since 07/21/2003 18:43:37.
  The sparcv9 processor operates at 1280 MHz,
    and has a sparcv9 floating point processor.
Status of processor 1 as of: 07/28/2003 14:43:29
  Processor has been on-line since 07/21/2003 18:43:36.
  The sparcv9 processor operates at 1280 MHz,
    and has a sparcv9 floating point processor
```

showrev 命令

showrev 命令显示当前硬件和软件的版本信息。代码示例 1-14 显示了 showrev 命令的输出示例。

代码示例 1-14 showrev 命令的输出

```
# showrev
Hostname: vsp78-36
Hostid: 8328c87b
Release: 5.8
Kernel architecture: sun4u
Application architecture: sparc
Hardware provider: Sun_Microsystems
Domain: vsplab.SFBay.Sun.COM
Kernel version: SunOS 5.8 Generic 108528-18 November 2002
```

如果 showrev 命令带有 -p 选项，则还会显示已安装的增补程序。代码示例 1-15 显示了带有 -p 选项的 showrev 命令的部分输出示例。

代码示例 1-15 showrev -p 命令的输出

```
Patch: 109729-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109783-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109807-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 109809-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110905-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110910-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 110914-01 Obsoletes: Requires: Incompatibles: Packages: SUNWcsu
Patch: 108964-04 Obsoletes: Requires: Incompatibles: Packages: SUNWcsr
```


▼ 运行 Solaris 平台系统信息命令

- 在命令提示符下，键入要显示系统信息类型的命令。

有关详细信息，请参阅第 18 页上的“Solaris 软件系统信息命令”。表 1-6 中汇总了这些命令。

表 1-6 Solaris 平台信息显示命令

命令	显示内容	键入内容	注释
prtconf	系统配置信息	/usr/sbin/prtconf	—
prtdiag	诊断和配置信息	/usr/platform/sun4u/sbin/prtdiag	使用 -v 选项可获取更详细的信息。
prtfru	FRU 分层结构和 SEEPROM 内存设备内容	/usr/sbin/prtfru	使用 -l 选项可显示分层结构。使用 -c 选项可显示 SEEPROM 数据。
psrinfo	每一个 CPU 的联机日期和时间；处理器时钟频率	/usr/sbin/psrinfo	使用 -v 选项可获取时钟频率及其它数据。
showrev	硬件和软件版本信息	/usr/bin/showrev	使用 -p 选项可显示软件增补程序。

最新诊断程序测试结果

通过关闭并打开系统电源，可以保存由开机自检 (POST) 和 OpenBoot 诊断程序测试生成的最新结果汇总。

▼ 查看最新测试结果

1. 切换到 ok 提示符下。
2. 执行下列一种步骤：
 - 要查看 POST 的最新结果汇总，请键入以下命令：

```
ok show-post-results
```

- 要查看 OpenBoot 诊断程序测试的最新结果汇总，请键入以下命令：

```
ok show-obdiag-results
```

此命令将生成一个硬件组件列表（因系统而异），并显示哪些组件已通过和未通过 POST 或 OpenBoot 诊断程序测试。

OpenBoot 配置变量

IDPROM 中存储的参数开关和诊断配置变量决定了如何以及何时执行 POST 诊断程序和 OpenBoot 诊断程序测试。本节介绍如何访问和修改 OpenBoot 配置变量。有关重要的 OpenBoot 配置变量的列表，请参见表 1-4。

对 OpenBoot 配置变量所做的更改会在系统下一次引导后生效。

▼ 查看和设置 OpenBoot 配置变量

- 中止服务器以显示 ok 提示符。
 - 要显示所有 OpenBoot 配置变量的当前值，可使用 `printenv` 命令。
下面是此命令输出的部分摘要示例。

```
ok printenv
Variable Name      Value      Default Value
diag-level         min        min
diag-switch?      false      false
```

- 要设置或更改 OpenBoot 配置变量的值，可使用 `setenv` 命令：

```
ok setenv diag-level max
diag-level =          max
```

- 要设置可以接受多个关键字的 OpenBoot 配置变量，请用空格隔开这些关键字。

使用 watch-net 和 watch-net-all 命令检查网络连接情况

watch-net 诊断程序测试用于监控主网络接口上的以太网信息包。watch-net-all 诊断程序测试不仅监控主网络接口上的以太网信息包，而且还监控连接到系统板的其它网络接口上的以太网信息包。系统收到的完好信息包由句点 (.) 表示。系统收到的错误信息包，如组帧错误和循环冗余校验 (CRC) 错误，由 X 表示，且附带错误说明。

- 要启动 watch-net 诊断程序测试，请在 ok 提示符下键入 watch-net 命令（代码示例 1-16）。

代码示例 1-16 watch-net 诊断程序的输出消息

```
{0} ok watch-net
Internal loopback test -- succeeded.
Link is -- up
Looking for Ethernet Packets.
`.` is a Good Packet.`X` is a Bad Packet.
Type any key to stop.....
```

- 要启动 watch-net-all 诊断程序测试，请在 ok 提示符下键入 watch-net-all 命令（代码示例 1-17）。

代码示例 1-17 watch-net-all 诊断程序的输出消息

```
{0} ok watch-net-all
/pci@1f,0/pci@1,1/network@c,1
Internal loopback test -- succeeded.
Link is -- up
Looking for Ethernet Packets.
`.` is a Good Packet.`X` is a Bad Packet.
Type any key to stop.
```

系统自动恢复

注 – Automatic System Recovery (ASR) 不同于 Netra 240 服务器同时支持的 Automatic Server Restart。有关 Automatic Server Restart 的信息，请参阅[第 3 章](#)。

Automatic System Recovery (ASR) 包含自检功能和自动配置功能，可以检测到出现故障的硬件组件并禁用它们。通过启用 ASR，服务器可以在出现非致命的硬件故障时继续运行。

如果某个组件由 ASR 监控，且服务器在没有此组件时仍可运行，则在该组件出现故障时，服务器将会自动重新引导。此功能可以防止出现故障的硬件组件重复造成整个系统关闭或无法运行。

如果在开机过程中检测到某个故障，则会禁用出现故障的组件。如果系统仍可正常操作，则引导过程会继续进行。

为支持此类降级的引导功能，OpenBoot 固件使用 1275 客户机接口（通过设备树）在设备树节点处生成相应的状态属性，从而将设备标记为 *failed*（故障）或 *disabled*（禁用）。Solaris 操作系统不会激活带有此类标记的子系统的驱动程序。

只要出现故障的组件不会产生电力中断问题（例如，不会导致随机总线错误或信噪），系统就会在收到服务请求时自动重新引导并恢复操作。

只要用新设备更换了出现故障或禁用的设备，OpenBoot 固件会在重新引导时自动修改该设备的状态。

注 – 此时，您需要激活 ASR 才能启用 ASR（请参阅[第 30 页上的“启用 ASR”](#)）。

Auto-Boot（自动引导）选项

`auto-boot?` 设置用于控制固件是否在每次重置后均自动引导操作系统。默认设置为 `true`。

`auto-boot-on-error?` 设置用于控制系统是否在检测到子系统故障时执行降级引导。必须将 `auto-boot?` 和 `auto-boot-on-error?` 都设为 `true` 后，才能执行自动降级引导。

- 要设置参数，请键入以下命令：

```
ok setenv auto-boot? true
ok setenv auto-boot-on-error? true
```

注 – auto-boot-on-error? 的默认设置为 false。因此，只有将该设置更改为 true 后，系统才会执行降级引导。此外，对于不可恢复的致命错误，即使启用降级引导功能，系统也不会执行降级引导。有关不可恢复的致命错误示例，请参阅第 29 页上的“[错误处理概要](#)”。

错误处理概要

开机期间所执行的错误处理分成以下三种情况：

- 如果 POST 或 OpenBoot 诊断程序未检测出任何错误，并且 auto-boot? 设置为 true，则系统将执行引导。
- 如果 POST 或 OpenBoot 诊断程序检测到非致命性错误，并且 auto-boot? 和 auto-boot-on-error? 均设为 true，则系统将执行引导。

注 – 如果 POST 或 OpenBoot 诊断程序检测出常用引导设备发生了非致命错误，OpenBoot 固件将自动取消该设备的配置，并尝试使用 boot-device 配置变量所指定的下一个引导设备。

- 如果 POST 或 OpenBoot 诊断程序检测到致命错误，则不论 auto-boot? 或 auto-boot-on-error? 的设置如何，系统均不会进行引导。不可恢复的致命错误包括以下几项：
 - 所有 CPU 均出现故障
 - 所有逻辑内存组均出现故障
 - 闪存 RAM 循环冗余校验 (CRC) 失败
 - 关键现场可更换部件 (FRU) PROM 配置数据错误
 - 关键专用集成电路 (ASIC) 出现故障

重置情况

diag-switch?、obdiag-trigger 和 post-trigger 这三个 OpenBoot 配置变量用于控制在发生系统重置事件时，系统如何运行固件诊断程序。

标准系统重置方案将会跳过 POST 和 OpenBoot 诊断程序，除非您将 diag-switch? 变量设为 true。该变量的默认设置为 false。由于 ASR 依靠固件诊断程序来检测那些出现故障的设备，因此必须将 diag-switch? 变量设为 true 才能运行 ASR。有关说明，请参阅第 30 页上的“启用 ASR”。

要控制哪些重置事件可以自动启动固件诊断程序（如果存在），请使用 obdiag-trigger 和 post-trigger 变量。有关这些变量及其用法的详细说明，请参阅第 8 页上的“控制 POST 诊断程序”和第 15 页上的“控制 OpenBoot 诊断程序测试”。

▼ 启用 ASR

1. 在系统 ok 提示符下键入：

```
ok setenv diag-switch? true
ok setenv auto-boot? true
ok setenv auto-boot-on-error? true
```

2. 将 obdiag-trigger 变量设置为 power-on-reset、error-reset 或 user-reset。

例如，键入以下命令：

```
ok setenv obdiag-trigger user-reset
```

3. 键入以下命令：

```
ok reset-all
```

系统会永久存储对参数所做的更改，并且在 OpenBoot 变量 auto-boot? 设为 true（默认值）的情况下自动执行引导。

注 – 此外，您还可使用前面板上的 On/Standby（开机/待机）按钮关闭然后打开系统电源，以此来存储参数更改。

▼ 禁用 ASR

1. 在系统 `ok` 提示符下键入：

```
ok setenv diag-switch? false
```

2. 键入以下命令：

```
ok reset-all
```

系统会永久存储参数更改。

注 – 此外，您还可使用前面板上的 On/Standby（开机/待机）按钮关闭并打开系统电源，以此来存储参数更改。

SunVTS 软件

本章介绍 SunVTS。本章包括以下各节：

- 第 33 页上的 “SunVTS 软件概述”
- 第 34 页上的 “SunVTS 测试程序”
- 第 35 页上的 “SunVTS 软件和安全”
- 第 35 页上的 “安装 SunVTS 软件”
- 第 36 页上的 “查看 SunVTS 软件文档”

SunVTS 软件概述

Netra 240 服务器支持 SunVTS 5.1 Patch Set 5 (PS5) 软件及将来的兼容版本。

SunVTS 软件（Sun 验证和测试套件）是一个联机诊断工具，用于验证硬件控制器、设备和平台的配置与功能。此工具在 Solaris 操作系统中运行并具有以下界面：

- 命令行界面 (CLI)
- 串行 (tty) 界面

SunVTS 软件套件执行系统和外围设备的负荷测试。您可以通过网络查看和控制 SunVTS 软件会话。通过远程计算机，您可以查看网络中另一台计算机上的测试进程，更改该计算机上的测试选项，以及控制该计算机上的所有测试功能。

SunVTS 软件可在三种测试模式下运行：

- *连接模式*，验证设备控制器是否存在。一般情况下，此操作只需要花费几分钟的时间，因此不失为一种对系统连接进行“健康检查”的好方法。
- *功能模式*，只检查您选定的特定子系统。这是系统默认的测试模式。
- *自动配置模式*，自动检测所有子系统，然后对它们进行下列一种测试：
 - *可靠性测试* — 对所有子系统都执行一次测试，然后停止运行。对于典型系统配置，此过程大约需要一到两个小时。
 - *综合测试* — 24 小时重复测试所有子系统。

由于 SunVTS 软件可以同时运行多个测试程序，因此会耗用大量系统资源。在工作系统上运行此软件时，应认真考虑这一情况。如果您使用 SunVTS 软件的综合测试模式对某个系统进行负荷测试，则应在测试期间关闭该系统上的所有应用程序。

服务器必须运行 Solaris 操作系统，才能使用 SunVTS 软件进行测试。由于 SunVTS 软件包是可选件，因此您的系统可能并未安装这些软件包。有关说明，请参阅第 35 页上的“[确定是否安装了 SunVTS 软件](#)”。

SunVTS 测试程序

通过 SunVTS 软件，您可以查看和控制远程连接的服务器上的测试会话。[表 2-1](#) 列出了一些可用的测试。

表 2-1 SunVTS 软件测试

SunVTS 软件测试	说明
<code>cputest</code>	测试 CPU。
<code>disktest</code>	测试本地磁盘驱动器。
<code>dvdtest</code>	测试 DVD-ROM 驱动器。
<code>n240atest</code>	测试警报卡以了解警报继电器、LED 和 FRU ID 的情况。
<code>fputest</code>	测试浮点运算单元。
<code>nettest</code>	测试系统板上的以太网硬件和任何 PCI 卡选件上的联网硬件。
<code>netlbttest</code>	执行环路测试，以确定以太网适配器是否可以收发数据包。
<code>pmem</code>	测试物理内存（只读）。
<code>sutest</code>	测试服务器的板载串行端口。
<code>vmem</code>	测试虚拟内存（包括交换分区和物理内存）。
<code>env6test</code>	测试环境设备。
<code>ssptest</code>	测试 ALOM 硬件设备。
<code>i2c2test</code>	测试 I ² C 设备以确定操作是否正常。

SunVTS 软件和安全

安装 SunVTS 软件时，您必须确定是选择基本安全方案，还是选择 Sun Enterprise Authentication Mechanism™ (SEAM) 安全方案。基本安全方案使用 SunVTS 软件安装目录中的本地安全方案文件来确定是否允许用户、用户组和主机使用 SunVTS 软件。SEAM 安全方案基于标准网络验证协议 Kerberos，可用于为通过网络进行的事务处理提供安全用户验证、数据完整性和保密机制。

如果站点使用 SEAM 安全方案，则必须在联网环境中安装 SEAM 客户端和服务端软件，并在 Solaris 软件和 SunVTS 软件中正确进行配置。如果您的站点不使用 SEAM 安全方案，则在安装 SunVTS 软件时不要选择 SEAM 选项。

如果您在安装期间选择了错误的安全方案，或者未正确配置选定的安全方案，则可能无法运行 SunVTS 软件测试。有关详细信息，请参阅《SunVTS User's Guide》和 SEAM 软件所附带的说明。

▼ 确定是否安装了 SunVTS 软件

- 键入以下命令：

```
# pkginfo -l SUNWvts
```

- 如果已装载了 SunVTS 软件，则会显示有关其软件包的信息。
- 如果系统未装载 SunVTS 软件，则会显示以下错误消息：

```
ERROR:information for "SUNWvts" was not found
```

安装 SunVTS 软件

默认情况下，Netra 240 服务器上不安装 SunVTS 软件。不过，您可以从 Solaris OS 增补 CD 中找到其安装程序，并且可以从以下 Web 站点下载其最新版本：

<http://www.sun.com/oem/products/vts/>

注 – Netra 240 服务器支持 SunVTS 5.1 Patch Set 5 (PS5) 软件及将来的兼容版本。

有关 SunVTS 软件的详细使用说明，请参阅与您所运行的 Solaris 软件版本对应的 SunVTS 文档。从上述 Web 站点上，还可以找到有关 SunVTS 软件以及安装说明的其它信息。

查看 SunVTS 软件文档

Solaris 介质套件附带的软件增补 CD 中提供了 SunVTS 软件文档。另外，也可以从 <http://docs.sun.com> 获得这些文档。

您还可参阅以下 SunVTS 文档来了解有关 SunVTS 软件的详细信息：

- *SunVTS User's Guide*，介绍如何安装、配置和运行 SunVTS 诊断软件。
- *SunVTS Quick Reference Card*，简要介绍如何使用 SunVTS 界面。
- *SunVTS Test Reference Manual*，详细介绍各项 SunVTS 测试。

Advanced Lights Out Manager

本章简要介绍 Sun™ Advanced Lights Out Manager (ALOM) 软件。本章包括以下主题：

- 第 37 页上的 “Advanced Lights Out Manager 概述”
 - 第 38 页上的 “ALOM 端口”
 - 第 39 页上的 “设置 admin 口令”
 - 第 39 页上的 “ALOM 基本功能”
 - 第 41 页上的 “Automatic Server Restart”
 - 第 41 页上的 “环境监视与控制”
-

Advanced Lights Out Manager 概述

Netra 240 服务器上随附安装了 Sun Advanced Lights Out Manager。默认情况下，系统控制台指向 ALOM，并配置为在启动时显示服务器控制台信息。

通过 ALOM，您可以使用串行连接（使用 SERIAL MGT 端口）或以太网连接（使用 NET MGT 端口）来监视和控制服务器。有关配置以太网连接的信息，请参阅《*Sun Advanced Lights Out Manager 软件用户指南（适用于 Netra 240 服务器）*》(817-5008-11)。

注 – 标为 “SERIAL MGT” 的 ALOM 串行端口只能用于管理服务器。如果需要通用串行端口，请使用标为 “10101” 的串行端口。

可以将 ALOM 配置为向管理员发送电子邮件，以通知与服务器或 ALOM 相关的硬件故障和其它事件。

ALOM 电路使用服务器提供的备用电源，这样：

- 一旦服务器接通电源，ALOM 便会激活，直到拔下电源电缆切断服务器的电源。

- 服务器操作系统处于脱机状态后，ALOM 固件和软件仍处于激活状态。

表 3-1 列出了 ALOM 监视的组件以及该软件针对每个组件提供的信息。

表 3-1 ALOM 监视的组件

组件	提供的信息
硬盘驱动器	存在和状态
系统和 CPU 风扇	速度和状态
CPU	存在、温度以及一些与热量有关的警报或故障状态
电源设备	存在和状态
系统温度	环境温度以及一些与热量有关的警报或故障状态
服务器前面板	旋转式开关的位置和 LED 状态
电压	状态和阈值
SCSI 和 USB 断路器	状态
干接点中继器警报	状态

ALOM 端口

默认情况下，ALOM 使用标为“SERIAL MGT”的端口来管理服务器。此端口使用 RJ-45 连接器且仅用于服务器管理；它只支持与外部控制台之间的 ASCII 连接。首次运行服务器时，需要使用此端口。

另一个串行端口 — 标为“10101” — 可用于传输一般的串行数据。此端口使用 DB-9 连接器。有关管脚引线的信息，请参阅《Netra 240 服务器安装指南》（部件号 817-4999-11）。

另外，服务器还配有一个 10Base-T 以太网管理域接口（标为“NET MGT”）。要使用此端口，需要进行 ALOM 配置。有关信息，请参阅《Sun Advanced Lights Out Manager 软件用户指南（适用于 Netra 240 服务器）》（部件号 817-5008-11）。

设置 admin 口令

如果在初次加电之后切换到 ALOM 软件，则会看到 `sc>` 提示符。此时，可以执行不需要用户权限的命令。（有关命令的列表，请参阅《*Sun Advanced Lights Out Manager 软件用户指南（适用于 Netra 240 服务器）*》，部件号 817-5008-11）。如果试图执行任何需要用户权限的命令，则系统将提示您设置用户 admin 的口令。

- 如果系统提示您设置口令，请设置 admin 用户的口令。

口令必须包含以下内容：

- 至少两个字母
- 至少一个数字或一个特殊字符
- 六到八个字符

一旦设置了口令，admin 用户将拥有完全访问权限，且可执行所有的 ALOM CLI 命令。当用户随后切换到 ALOM 时，系统将提示用户使用 admin 口令登录。

ALOM 基本功能

本节介绍 ALOM 的一些基本功能。有关全面的文档，请参阅《*Sun Advanced Lights Out Manager 软件用户指南（适用于 Netra 240 服务器）*》（部件号 817-5008-11）和《*Netra 240 Server Release Notes*》（817-3142-xx）。

▼ 切换到 ALOM 提示符

- 在命令提示符下，键入 #. 换码序列：

```
# #.
```

注 – 切换到 ALOM 提示符时，您的登录用户 ID 为 admin。请参阅第 39 页上的“[设置 admin 口令](#)”。

▼ 切换到服务器控制台提示符

- 键入以下命令：

```
sc> console
```

多个 ALOM 用户可以同时连接到服务器控制台，但只允许一个用户向控制台键入输入字符。

如果另一用户已登录且具有写入权限，则在键入 console 命令后，您会看到以下消息：

```
sc> Console session already in use.[view mode]
```

▼ 取消另一用户的控制台写入权限

- 键入以下命令：

```
sc> console -f
```

Automatic Server Restart

注 – Automatic System Recovery (ASR) 不同于 Netra 240 服务器同时支持的 Automatic Server Restart。

Automatic Server Restart 是 ALOM 的一个组件。它用于监视处于运行状态的 Solaris 操作系统，在默认情况下，能够同步文件系统并在服务器失败时重新启动服务器。

ALOM 使用“软件狗”进程来专门监视内核。如果某个进程挂起，但内核仍在运行，则 ALOM 不会重新启动服务器。用户不能配置用于控制“软件狗”查询间隔和“软件狗”超时的 ALOM “软件狗”参数。

如果内核挂起且“软件狗”超时，则 ALOM 会报告并记录相关事件，然后执行下列三种用户可配置的操作之一：

- **xir** — 此操作为默认操作，使服务器同步文件系统并重新启动。如果系统挂起，则 ALOM 将在 15 分钟之后还原到硬重置。
- **Reset**（重置）— 即硬重置过程，使系统可以快速恢复运行，但不会存储与挂起有关的诊断数据，且可能会造成重大损坏。
- **None**（无）— 这使系统在“软件狗”超时之后，无限期地保持挂起状态。

有关详细信息，请参阅《*Sun Advanced Lights Out Manager 软件用户指南（适用于 Netra 240 服务器）*》（部件号 817-5008-11）的 `sys_autorestart` 部分。

有关使用 Automatic System Recovery（自动系统恢复，ASR）的说明，请参阅第 1 章。

环境监视与控制

Netra 240 服务器提供了一个环境监视子系统，可防止服务器及其组件出现以下情况：

- 温度过高
- 系统内部空气流通不畅
- 运行时缺少组件或组件配置不当
- 电源故障
- 内部硬件故障

监视和控制功能由 ALOM 固件负责处理，这可确保即使系统中止或无法引导，仍能正常实施监视功能。另外，使用 ALOM 固件监视系统可以释放 CPU 和内存资源，使之专用于操作系统和应用程序软件。

环境监视子系统使用符合行业标准的 I²C 总线。I²C 总线是由两条线组成的简单串行总线。它遍布于整个系统，以帮助对温度传感器、风扇、电源、状态 LED 和前面板系统控制旋转式开关等进行监控。

服务器包含三个温度传感器，可用于监视服务器的环境温度以及两个 CPU 的内核温度。监视子系统轮询每个传感器，使用取样温度来报告任何温度过高或过低的情况，并对其做出响应。附加的 I²C 设备则用于检测组件是否存在以及组件故障。

硬件和软件共同发挥作用，可确保机箱内的温度不会超出预定义的“安全运行”范围。如果传感器检测到的温度低于低温报警阈值或高于高温报警阈值，监视子系统软件将使前后面板上的系统“需要维修”LED 发光。如果温度条件持续不好转，且达到软关机温度的最高阈值或最低阈值，则系统将启动从容关机方式。如果温度达到硬关机温度的最高阈值或最低阈值，则系统将启动强制关机方式。

错误和警告消息将发送到系统控制台，并记录在 `/var/adm/messages` 文件中；“需要维修”LED 在系统自动关闭后仍然发光，以帮助诊断问题之所在。

发送到系统控制台并记录在 `/var/adm/messages` 文件中的消息的类型取决于设置 `sc_clieventlevel` 和 `sys_eventlevel` ALOM 用户变量的方式。有关设置这些变量的信息，请参阅《*Sun Advanced Lights Out Manager 软件用户指南（适用于 Netra 240 服务器）*》(817-5008-11)。

表 3-2 Netra 240 服务器机箱温度阈值

温度阈值	温度	服务器操作
低温，硬关机	-11° C	服务器启动强制关机方式。
低温，软关机	-9° C	服务器执行从容关机。
低温警告	-7° C	服务器将亮起前后面板上的系统“需要维修”LED 指示灯。
高温警告	57° C	服务器将亮起前后面板上的系统“需要维修”LED 指示灯。
高温，软关机	60° C	服务器执行从容关机。
高温，硬关机	63° C	服务器启动强制关机方式。

监视子系统还可用于检测四个系统风扇的故障。如果任何一个风扇出现故障，监视子系统将检测到该故障，生成错误消息并发送到系统控制台上，同时将消息记录到 `/var/adm/messages` 文件中，同时使“需要维修”LED 发光。

监视电源子系统的方式同上。通过定期轮询电源状态，监视子系统可指示每个电源的输出、输入和存在状态。

如果检测到电源问题，则将向系统控制台发送一条错误消息，而且将该消息记录到 `/var/adm/messages` 文件中。此外，每个电源上的 LED 都会发光，以表明发生了故障。系统“需要维修”LED 将发光，以表明发生了系统故障。ALOM 控制台警报将记录电源故障。

可以使用 `showenvironment ALOM` 命令来查看电源子系统的警告阈值以及风扇速度。有关使用此命令的说明，请参阅《*Sun Advanced Lights Out Manager 软件用户指南（适用于 Netra 240 服务器）*》（部件号 817-5008-11）。

警报继电器输出应用编程接口

本附录提供一个示例程序（[代码示例 A-1](#)），说明如何使用 `get/set` 命令获取/设置警报的状态。该应用程序可以使用 `LOMIOCALSTATE ioctl` 函数来获取每个警报的状态，并使用 `LOMIOCALCTL ioctl` 函数来分别设置警报的状态。有关警报指示灯的详细信息，请参阅《*Netra 240 Server Service Manual*》(817-2699-xx)。

代码示例 A-1 使用 `get/set` 命令获取/设置状态的程序示例

```
#include <sys/types.h>
#include <string.h>
#include <stdlib.h>
#include <sys/unistd.h>
#include <fcntl.h>
#include "lom_io.h"

#define ALARM_INVALID    -1
#define LOM_DEVICE      "/dev/lom"

static void usage();
static void get_alarm(const char *alarm);
static int set_alarm(const char *alarm, const char *alarmval);
static int parse_alarm(const char *alarm);
static int lom_ioctl(int ioc, char *buf);
static char *get_alarmval(int state);
static void get_alarmvals();

main(int argc, char *argv[])
{
    if (argc < 3) {
        usage();
        if (argc == 1)
            get_alarmvals();
        exit(1);
    }
}
```

代码示例 A-1 使用 get/set 命令获取/设置状态的程序示例 (续)

```
#include <sys/types.h>

    if (strcmp(argv[1], "get") == 0) {
        if (argc != 3) {
            usage();
            exit (1);
        }
        get_alarm(argv[2]);
    }
    else
    if (strcmp(argv[1], "set") == 0) {
        if (argc != 4) {
            usage();
            exit (1);
        }
        set_alarm(argv[2], argv[3]);
    } else {
        usage();
        exit (1);
    }
}

static void
usage()
{
    printf("usage: alarm [get|set] [crit|major|minor|user] [on|off]\n");
}

static void
get_alarm(const char *alarm)
{
    ts_aldata_t    ald;
    int altype = parse_alarm(alarm);
    char *val;

    if (altype == ALARM_INVALID) {
        usage();
        exit (1);
    }

    ald.alarm_no = altype;
    ald.alarm_state = ALARM_OFF;

    lom_ioctl(LOMIOCALSTATE, (char *)&ald);

    if ((ald.alarm_state != ALARM_OFF) &&
        (ald.alarm_state != ALARM_ON)) {
```

代码示例 A-1 使用 get/set 命令获取/设置状态的程序示例 (续)

```
#include <sys/types.h>
    printf("Invalid value returned: %d\n", ald.alarm_state);
    exit(1);
}

printf("ALARM.%s = %s\n", alarm, get_alarmval(ald.alarm_state));
}

static int
set_alarm(const char *alarm, const char *alarmstate)
{
    ts_aldata_t    ald;
    int alarmval = ALARM_OFF, altype = parse_alarm(alarm);

    if (altype == ALARM_INVALID) {
        usage();
        exit (1);
    }

    if (strcmp(alarmstate, "on") == 0)
        alarmval = ALARM_ON;
    else
    if (strcmp(alarmstate, "off") == 0)
        alarmval = ALARM_OFF;
    else {
        usage();
        exit (1);
    }

    ald.alarm_no = altype;
    ald.alarm_state = alarmval;

    if (lom_ioctl(LOMIOCALCTL, (char *)&ald) != 0) {
        printf("Setting ALARM.%s to %s failed\n", alarm, alarmstate);
        return (1);
    } else {
        printf("Setting ALARM.%s successfully set to %s\n", alarm,
alarmstate);
        return (1);
    }
}

static int
parse_alarm(const char *alarm)
{
    int altype;
```

代码示例 A-1 使用 get/set 命令获取/设置状态的程序示例 (续)

```
#include <sys/types.h>
if (strcmp(alarm, "crit") == 0)
    altype = ALARM_CRITICAL;
else
if (strcmp(alarm, "major") == 0)
    altype = ALARM_MAJOR;
else
if (strcmp(alarm, "minor") == 0)
    altype = ALARM_MINOR;
else
if (strcmp(alarm, "user") == 0)
    altype = ALARM_USER;
else {
    printf("invalid alarm value: %s\n", alarm);
    altype = ALARM_INVALID;
}

return (altype);
}

static int
lom_ioctl(int ioc, char *buf)
{
    int fd, ret;

    fd = open(LOM_DEVICE, O_RDWR);

    if (fd == -1) {
        printf("Error opening device: %s\n", LOM_DEVICE);
        exit (1);
    }

    ret = ioctl(fd, ioc, (void *)buf);

    close (fd);

    return (ret);
}

static char *
get_alarmval(int state)
{
    if (state == ALARM_OFF)
        return ("off");
    else
        if (state == ALARM_ON)
```


代码示例 A-1 使用 get/set 命令获取/设置状态的程序示例 (续)

```
#include <sys/types.h>
        return ("on");
    else
        return (NULL);
}
static void
get_alarmvals()
{
    get_alarm("crit");
    get_alarm("major");
    get_alarm("minor");
    get_alarm("user");
}
```


索引

符号

“需要维修” LED, 42

英文

Advanced Lights Out Manager

参阅 ALOM

ALOM

Automatic Server Restart, 41

LED 状态指示灯, 4

端口, 38

概述, 37

环境监视子系统, 41

基本功能, 39

设置口令, 39

诊断工具, 2

ASR, 28

auto-boot? 变量, 8

Automatic Server Restart, 41

Automatic System Recovery 另请参阅 ASR, 28

BIST, 参阅 内部自检

CPU

时钟频率, 23

显示信息, 23

diag-level 变量, 8, 15

diag-script 变量, 8

diag-switch? 变量, 7, 8

FRU, 22 至 23

I²C 传感器, 42

I²C 总线, 42

IDE 总线, 12

input-device 变量, 9

LED, 诊断工具, 2

obdiag-trigger 变量, 9

OpenBoot PROM 参数, diag-level 变量, 7

OpenBoot 命令

probe-ide, 12

probe-scsi 和 probe-scsi-all, 11

show-devs, 13

运行, 13

诊断工具, 2

OpenBoot 配置变量

关键字, 8

说明, 8

OpenBoot 诊断程序, 14

控制测试, 15

启动, 14

诊断工具, 2

OpenBoot 诊断程序测试

test-all 命令, 16

test 命令, 16

错误消息, 解释, 16

硬件设备路径, 16

在 ok 提示符下, 16

output-device 变量, 9

POST

错误消息, 8

启动诊断程序, 10

诊断程序, 控制, 8

- 诊断工具, 2
- post-trigger 变量, 9
- probe-ide 命令 (OpenBoot), 12
- probe-scsi 命令和 probe-scsi-all 命令 (OpenBoot), 11
- prtconf 命令, Solaris, 18
- prtdiag 命令, Solaris, 20
- prtfru 命令, Solaris, 22
- psrinfo 命令, Solaris, 23
- SCSI 设备, 诊断问题, 11
- SEAM, 35
- show-devs 命令, OpenBoot, 13
- showrev 命令, Solaris, 24
- Solaris 操作系统
 - SunVTS, 34
 - 设备树, 18
- Solaris 命令
 - prtconf, 18
 - prtdiag, 20
 - prtfru, 22
 - psrinfo, 23
 - showrev, 24
 - 诊断工具, 2
- SunVTS, 33 至 36
 - SEAM 安全方案, 35
 - Solaris 操作系统, 34
 - 安装, 35
 - 概述, 33
 - 基本安全方案, 35
 - 兼容版本, 33, 35
 - 界面, 33
 - 可选软件包, 34
 - 可用的测试程序, 34
 - 确定安装, 35
 - 软件, 测试模式, 33
 - 文档, 36
 - 诊断工具, 2
- Sun 企业验证机制, 请参阅 SEAM
- Sun 验证测试套件
 - 请参阅 SunVTS
- test-all 命令 (OpenBoot 诊断程序测试), 16
- test-args 变量, 15

- 关键字, 15
- 自定义测试, 15
- test 命令 (OpenBoot 诊断程序测试), 16
- USB 设备, OpenBoot 诊断程序自检, 16
- watch-net, 27

B

- 版本, 硬件和软件, 显示 showrev, 24

C

- 常闭 (NC), 继电器状态, 7
- 常开 (NO), 继电器状态, 7
- 重置事件, 类型, 9
- 处理器速度, 显示, 23
- 次要, 警报指示灯, 7
- 错误消息
 - OpenBoot 诊断程序, 解释, 16
 - OpenBoot 诊断程序测试, 16
 - 日志文件, 42
 - 与电源相关, 42

D

- 电源, 故障监视, 42
- 定位器 LED, 4
 - 打开, 5
 - 关闭, 5
 - 状态, 5

F

- 服务器提示符
 - Advanced Lights Out Manager 提示符, 2
 - OpenBoot 提示符, 2
 - Solaris 软件超级用户提示符, 2
- 服务器状态指示灯, 前面和后面, 4

G

- 故障排除工具, 2

H

- 环境监视子系统, 41
- 环路 ID(probe-scsi), 11

J

集成电路设备 (IDE), 参阅 IDE 总线

继电器状态

常闭 (NC), 7

常开 (NO), 7

加电自检

参阅 POST

监视子系统

温度过低, 42

温度过高, 42

紧急, 警报指示灯, 6

警报

get (获取) 状态, 45 至 49

set (设置) 状态, 45 至 49

继电器输出 API, 45 至 49

警报板

警报指示灯, 6

警报状态, 6

警报指示灯, 6

次要, 7

紧急, 6

用户, 7

重要, 6

警报状态, 干接点, 6

L

逻辑单元号 (probe-scsi), 11

N

内置自测程序, test-args 变量, 15

Q

全球通用名称 (probe-scsi), 11

R

日志文件, 17

错误消息, 17

系统消息, 17

软件版本, 显示 showrev, 24

S

设备路径, 硬件, 13, 16

设备树, Solaris 软件, 显示, 18

时钟频率, CPU, 23

W

温度传感器, 42

温度过低, 监视子系统, 42

温度过高, 监视子系统, 42

温度过高情况, 21

X

系统内存, 确定大小, 18

系统配置卡, 7

系统状态 LED

环境故障指示器, 42

请参阅 LED

消息

POST, 错误, 8

分析错误, 16

Y

硬件版本, 显示 showrev, 24

硬件设备路径, 13, 16

用户, 警报指示灯, 7

Z

增补程序, 已安装, showrev, 24

诊断程序

OpenBoot, 14

POST, 8

Solaris 操作系统, 17

SunVTS, 33

诊断程序测试, 绕过, 9

诊断工具

ALOM, 2

LED, 2

OpenBoot 命令, 2

OpenBoot 诊断程序, 2

Solaris 软件命令, 2

SunVTS, 2

加电自检, 2

中央处理器, 参阅 CPU

重要, 警报指示灯, 6

主适配器 (probe-scsi), 11

