System Administration Guide: Advanced Administration

Part No: 821–1452–01
November 2010

ORACLE

110425@25097

Contents

Preface

System Administration Guide: Advanced Administration is part of a documentation set that covers a significant part of the Oracle Solaris system administration information. This guide includes information for both SPARC and x86 based systems.

This book assumes that you have installed the Oracle Solaris operating system (OS). It also assumes that you have set up any networking software that you plan to use.

For the Oracle Solaris release, new features that are interesting to system administrators are covered in sections called *What's New in ... ?* in the appropriate chapters.

Note – This Oracle Solaris release supports systems that use the SPARC and x86 families of processor architectures. The supported systems appear in the Oracle Solaris OS: Hardware Compatibility Lists (`http://www.sun.com/bigadmin/hcl`). This document cites any implementation differences between the platform types.

In this document these x86 related terms mean the following:

- "x86" refers to the larger family of 64-bit and 32-bit x86 compatible products.
- "x64" relates specifically to 64-bit x86 compatible CPUs.
- "32-bit x86" points out specific 32-bit information about x86 based systems.

For supported systems, see the *Oracle Solaris OS: Hardware Compatibility Lists*.

Who Should Use This Book

This book is intended for anyone responsible for administering one or more systems that are running Oracle Solaris 10. To use this book, you should have 1-2 years of UNIX system administration experience. Attending UNIX system administration training courses might be helpful.

How the System Administration Guides Are Organized

Here is a list of the topics that are covered by the System Administration Guides.

Book Title	Topics
System Administration Guide: Basic Administration	User accounts and groups, shutting down and booting a system, and managing services
System Administration Guide: Advanced Administration	Terminals and modems, system resources, system processes, and troubleshooting Oracle Solaris software problems
System Administration Guide: Devices and File Systems	Removable media, disks and devices, file systems, and backing up and restoring data
System Administration Guide: IP Services	TCP/IP network administration, IPv4 and IPv6 address administration, DHCP, IPsec, IKE, IP filter, Mobile IP, and IPQoS
System Administration Guide: Naming and Directory Services (DNS, NIS, and LDAP)	DNS, NIS, and LDAP naming and directory services, including transitioning from NIS to LDAP
System Administration Guide: Network Interfaces and Network Virtualization	Networking stack, NIC driver property configuration, NWAM configuration, manual network interface configuration, administration of VLANs and link aggregations, IP network multipathing (IPMP), WiFi wireless networking configuration, virtual NICs (VNICs), and network resource management
System Administration Guide: Network Services	Web cache servers, time-related services, network file systems (NFS and autofs), mail, SLP, and PPP
System Administration Guide: Printing	Printing topics and tasks, using services, tools, protocols, and technologies to set up and administer printing services and printers
System Administration Guide: Security Services	Auditing, device management, file security, BART, Kerberos services, PAM, Oracle Solaris Cryptographic Framework, privileges, RBAC, SASL, and Oracle Solaris Secure Shell
System Administration Guide: Oracle Solaris Zones, Oracle Solaris 10 Containers, and Resource Management	Resource management features, which enable you to control how applications use available system resources; Oracle Solaris Zones software partitioning technology, which virtualizes operating system services to create an isolated environment for running applications; and Oracle Solaris 10 Containers, which host Oracle Solaris 10 environments running on the Oracle Solaris 11 Express kernel

Book Title	Topics
Oracle Solaris SMB and Windows Interoperability Administration Guide	Oracle Solaris SMB service, which enables you to configure an Oracle Solaris system to make SMB shares available to SMB clients; Oracle Solaris SMB client, which enables you to access SMB shares; and native identity mapping service, which enables you to map user and group identities between Oracle Solaris systems and Windows systems
Oracle Solaris Trusted Extensions Configuration and Administration	System installation, configuration, and administration that is specific to the Oracle Solaris' Trusted Extensions feature
Oracle Solaris ZFS Administration Guide	ZFS storage pool and file system creation and management, snapshots, clones, backups, using access control lists (ACLs) to protect ZFS files, using ZFS on an Oracle Solaris system with zones installed, emulated volumes, and troubleshooting and data recovery

Related Third-Party Web Site References

Note – Oracle is not responsible for the availability of third-party web sites mentioned in this document. Oracle does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Oracle will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Documentation, Support, and Training

See the following web sites for additional resources:

- Documentation (`http://docs.sun.com`)
- Support (`http://www.oracle.com/us/support/systems/index.html`)
- Training (`http://education.oracle.com`) – Click the Sun link in the left navigation bar.

Oracle Software Resources

Oracle Technology Network (`http://www.oracle.com/technetwork/index.html`) offers a range of resources related to Oracle software:

- Discuss technical problems and solutions on the Discussion Forums (`http://forums.oracle.com`).
- Get hands-on step-by-step tutorials with Oracle By Example (`http://www.oracle.com/technetwork/tutorials/index.html`).
- Download Sample Code (`http://www.oracle.com/technology/sample_code/index.html`).

Typographic Conventions

The following table describes the typographic conventions that are used in this book.

TABLE P–1 Typographic Conventions

Typeface	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`
`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`** `Password:`
aabbcc123	Placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. A *cache* is a copy that is stored locally. Do *not* save the file. **Note:** Some emphasized items appear bold online.

Shell Prompts in Command Examples

The following table shows the default UNIX system prompt and superuser prompt for shells that are included in the Oracle Solaris OS. Note that the default system prompt that is displayed in command examples varies, depending on the Oracle Solaris release.

TABLE P–2 Shell Prompts

Shell	Prompt
Bash shell, Korn shell, and Bourne shell	`$`
Bash shell, Korn shell, and Bourne shell for superuser	`#`
C shell	`machine_name%`
C shell for superuser	`machine_name#`

General Conventions

Be aware of the following conventions that are used in this book.

- When following steps or using examples, be sure to type double-quotes ("), left single-quotes (‘), and right single-quotes (’) exactly as shown.
- The key referred to as Return is labeled Enter on some keyboards.
- It is assumed that the root path includes the `/sbin`, `/usr/sbin`, `/usr/bin`, and `/etc` directories, so the steps in this book show the commands in these directories without absolute path names. Steps that use commands in other, less common, directories show the absolute path in the example.

◆ ◆ ◆ CHAPTER 1

Managing Terminals, Modems and Serial Port Services (Tasks)

This chapter describes how to manage terminals and modems and how to use the Service Access Facility (SAF) to manage serial port services. Also included in this chapter is information about how to perform console administration by using the Service Management Facility (SMF).

Note – The SAF and SMF are two different tools in the Oracle Solaris OS. Starting with the Oracle Solaris 10, `ttymon` invocations on the system console are now managed by SMF. The SAF tool is still used to administer terminals, modems, and other network devices.

This is a list of the information that is in this chapter.

For step-by-step instructions on managing serial ports , see the following:

For reference information about the SAF, see "Service Access Facility Administration (Reference)" on page 36.

Terminals, Modems, Ports, and Services

Terminals and modems provide both local and remote access to system and network resources. Setting up terminals and modem access is an important responsibility of a system administrator. This section explains some of the concepts behind modem and terminal management in the Oracle Solaris operating system.

Terminal Description

Your system's bitmapped graphics display is not the same as an alphanumeric terminal. An alphanumeric terminal connects to a serial port and displays only text. You do not have to perform any special steps to administer the graphics display.

Modem Description

Modems can be set up in three basic configurations:

- Dial-out
- Dial-in
- Bidirectional

A modem that is connected to your home computer might be set up to provide *dial-out* service. With dial-out service, you can access other computers from your own home. However, nobody outside can gain access to your machine.

Dial-in service is just the opposite. Dial-in service enables you to access a system from remote sites. However, dial—in service does not permit calls to the outside world.

Bidirectional access, as the name implies, provides both dial-in and dial-out capabilities.

Ports Description

A *port* is a channel through which a device communicates with the operating system. From a hardware perspective, a port is a "receptacle" into which a terminal or modem cable might be physically connected.

However, a port is not strictly a physical receptacle, but an entity with hardware (pins and connectors) and software (a device driver) components. A single physical receptacle often provides multiple ports, allowing connection of two or more devices.

Common types of ports include serial, parallel, small computer systems interface (SCSI), and Ethernet.

A *serial port*, using a standard communications protocol, transmits a byte of information bit-by-bit over a single line.

Devices that have been designed according to RS-232-C or RS-423 standards, including most modems, alphanumeric terminals, plotters, and some printers, can be connected interchangeably by using standard cables into the serial ports of computers that are similarly designed.

When many serial port devices must be connected to a single computer, you might need to add an *adapter board* to the system. The adapter board, with its driver software, provides additional serial ports for connecting more devices than could otherwise be accommodated.

Services Description

Modems and terminals gain access to computing resources by using serial port software. Serial port software must be set up to provide a particular "service" for the device attached to the port. For example, you can set up a serial port to provide bidirectional service for a modem.

Port Monitors

The main mechanism for gaining access to a service is through a port monitor. A *port monitor* is a program that continuously monitors for requests to log in or access printers or files.

When a port monitor detects a request, it sets whatever parameters are required to establish communication between the operating system and the device requesting service. Then, the port monitor transfers control to other processes that provide the services needed.

The following table describes the two types of port monitors included in the Oracle Solaris release.

TABLE 1–1 Port Monitor Types

Man Page	Port Monitor	Description
`listen(1M)`	`listen`	Controls access to network services, such as handling remote print requests prior to the Solaris 2.6 release. The default operating system no longer uses this port monitor type.
`ttymon(1M)`	`ttymon`	Provides access to the login services needed by modems and alphanumeric terminals. The Serial Ports tool automatically sets up a `ttymon` port monitor to process login requests from these devices.

You might be familiar with an older port monitor, called a `getty`. The `ttymon` port monitor is more powerful. A single `ttymon` port monitor can replace multiple occurrences of `getty`. Otherwise, these two programs serve the same function. For more information, see the `ttymon(1M)` man page.

Overview of the Service Access Facility

The SAF is the tool used for administering terminals, modems, and other network devices.

In particular, the SAF enables you to set up the following:

- `ttymon` and `listen` port monitors by using the `sacadm` command
- `ttymon` port monitor services by using the `pmadm` and `ttyadm` commands
- `listen` port monitor services by using the `pmadm` and `nlsadmin` commands
- Troubleshoot `tty` devices
- Troubleshoot incoming network requests for printing service
- Troubleshoot the Service Access Controller by using the `sacadm` command

The SAF is an open-systems solution that controls access to system and network resources through `tty` devices and local-area networks (LANs). The SAF is not a program, but a hierarchy of background processes and administrative commands.

Using the Service Access Facility

You can set up terminals and modems with the SAF commands.

The SAF is a tool that is used to administer terminals, modems, and other network devices. The top-level SAF program is the Service Access Controller (SAC). The SAC controls port monitors that you administer through the `sacadm` command. Each port monitor can manage one or more ports.

You administer the services that are associated with ports through the `pmadm` command. While services provided through the SAC can differ from network to network, the SAC and its administrative commands, `sacadm` and `pmadm`, are network independent.

The services of `ttymon` and `listen` are in turn controlled by the `pmadm` command. One instance of `ttymon` can service multiple ports. One instance of `listen` can provide multiple services on a network interface.

The following table describes the SAF control hierarchy. The `sacadm` command is used to administer the SAC, which controls the `ttymon` and `listen` port monitors.

TABLE 1–2 SAF Control Hierarchy

Function	Program	Description
Overall administration	`sacadm`	Command for adding and removing port monitors
Service Access Controller	`sac`	SAF's master program
Port monitors	`ttymon`	Monitors serial port login requests
	`listen`	Monitors requests for network services

TABLE 1–2 SAF Control Hierarchy *(Continued)*

Function	Program	Description
Port monitor service administrator	`pmadm`	Command for controlling port monitors services
Services	logins, remote procedure calls	Services to which the SAF provides access
Console administration	console login	Console services are managed by the SMF service, `svc:/system/console-login:default`. This service invokes the `ttymon` port monitor. Do not use the `pmadm` or the `sacadm` command to manage the console. For more information, see "`ttymon` and the Console Port" on page 25, "How to Set the `ttymon` Console Terminal Type" on page 27, and "How to Set the Baud Rate Speed on the `ttymon` Console Terminal" on page 27.

Managing Serial Ports (Task Map)

Task	Description	For Instructions
Perform console administration.	You might need to perform the following console administration tasks: ■ Set the `ttymon` console terminal type. Starting with Oracle Solaris 10, you must use the `svccfg` command to specify the `ttymon` console terminal type. ■ Set the `ttymon` console terminal baud rate speed.	"How to Set the `ttymon` Console Terminal Type" on page 27 "How to Set the Baud Rate Speed on the `ttymon` Console Terminal" on page 27
Add a `ttymon` port monitor.	Use the `sacadm` command to add a `ttymon` port monitor.	"How to Add a `ttymon` Port Monitor" on page 28
View a `ttymon` port monitor status.	Use the `sacadm` command to view `ttymon` port monitor status.	"How to View `ttymon` Port Monitor Status" on page 29
Stop a `ttymon` port monitor.	Use the `sacadm` command to stop a `ttymon` port monitor.	"How to Stop a `ttymon` Port Monitor" on page 30
Start a `ttymon` port monitor.	Use the `sacadm` command to start a `ttymon` port monitor.	"How to Start a `ttymon` Port Monitor" on page 30

Task	Description	For Instructions
Disable a ttymon port monitor.	Use the sacadm command to disable a ttymon port monitor.	"How to Disable a ttymon Port Monitor" on page 30
Enable a ttymonport monitor.	Use the sacadm command to enable a ttymon port monitor.	"How to Enable a ttymon Port Monitor" on page 31
Remove a ttymon port monitor.	Use the sacadm command to remove a ttymon port monitor.	"How to Remove a ttymon Port Monitor" on page 31

Overall SAF Administration (sacadm)

The sacadm command is the top level of the SAF. The sacadm command primarily is used to add and remove port monitors such as ttymon and listen. Other sacadm functions include listing the current status of port monitors and administering port monitor configuration scripts.

Service Access Controller (SAC Program)

The Service Access Controller program (SAC) oversees all port monitors. A system automatically starts the SAC upon entering multiuser mode.

When the SAC program is invoked, it first looks for, and interprets, each system's configuration script. You can use the configuration script to customize the SAC program environment. This script is empty by default. The modifications made to the SAC environment are inherited by all the "children" of the SAC. This inherited environment might be modified by the children.

After the SAC program has interpreted the per-system configuration script, the SAC program reads its administrative file and starts the specified port monitors. For each port monitor, the SAC program runs a copy of itself, forking a child process. Each child process then interprets its per-port monitor configuration script, if such a script exists.

Any modifications to the environment specified in the per-port monitor configuration script affect the port monitor and will be inherited by all its children. Finally, the child process runs the port monitor program by using the command found in the SAC program administrative file.

SAC Initialization Process

The following steps summarize what happens when SAC is first started:

1. The SAC program is started by the SMF service, svc:/system/sac:default.
2. The SAC program reads /etc/saf/_sysconfig, the per-system configuration script.
3. The SAC program reads /etc/saf/_sactab, the SAC administrative file.

4. The SAC program forks a child process for each port monitor it starts.
5. Each port monitor reads `/etc/saf/pmtag/_config`, the per-port monitor configuration script.

Port Monitor Service Administration (`pmadm`)

The `pmadm` command enables you to administer port monitors' services. In particular, you use the `pmadm` command to add or remove a service and to enable or disable a service. You can also install or replace per-service configuration scripts, or print information about a service.

Each instance of a service must be uniquely identified by a port monitor and a port. When you use the `pmadm` command to administer a service, you specify a particular port monitor with the *pmtag* argument, and a particular port with the *svctag* argument.

For each port monitor type, the SAF requires a specialized command to format port monitor-specific configuration data. This data is used by the `pmadm` command. For `ttymon` and `listen` type port monitors, these specialized commands are `ttyadm` and `nlsadmin`, respectively.

`ttymon` Port Monitor Process

Whenever you attempt to log in by using a directly connected modem or alphanumeric terminal, `ttymon` goes to work. First, the SAC process is started by SMF. Then, the SAC automatically starts the port monitors that are designated in its administrative file, `/etc/saf/_sactab`. After the `ttymon` port monitor has been started, it monitors the serial port lines for service requests.

When someone attempts to log in by using an alphanumeric terminal or a modem, the serial port driver passes the activity to the operating system. The `ttymon` port monitor notes the serial port activity, and attempts to establish a communications link. The `ttymon` port monitor determines which data transfer rate, line discipline, and handshaking protocol are required to communicate with the device.

After the proper parameters for communication with the modem or terminal are established, the `ttymon` port monitor passes these parameters to the login program and transfers control to it. For more information, see "TTY Monitor and Network Listener Port Monitors" on page 24.

Port Initialization Process

When an instance of the `ttymon` port monitor is invoked by the SAC, `ttymon` starts to monitor its ports. For each port, the `ttymon` port monitor first initializes the line disciplines, if they are specified, and the speed and terminal settings. The values used for initialization are taken from the appropriate entry in the `/etc/ttydefs` file.

The ttymon port monitor then writes the prompt and waits for user input. If the user indicates that the speed is inappropriate by pressing the Break key, the ttymon port monitor tries the next speed and writes the prompt again.

If *autobaud* is enabled for a port, the ttymon port monitor tries to determine the baud rate on the port automatically. Users must press Return before the ttymon port monitor can recognize the baud rate and print the prompt.

When valid input is received, the ttymon port monitor does the following tasks:

- Interprets the per-service configuration file for the port
- Creates an /etc/utmpx entry, if required
- Establishes the service environment
- Invokes the service that is associated with the port

After the service terminates, the ttymon port monitor cleans up the /etc/utmpx entry, if this entry exists, and returns the port to its initial state.

Bidirectional Service

If a port is configured for bidirectional service, the ttymon port monitor does the following:

- Allows users to connect to a service.
- Allows the uucico, cu, or ct commands to use the port for dialing out, if the port is free.
- Waits to read a character before printing a prompt.
- Invokes the port's associated service, without sending the prompt message, when a connection is requested, if the connect-on-carrier flag is set.

TTY Monitor and Network Listener Port Monitors

Though the SAF provides a generic means for administering any future or third-party port monitors, only two port monitors are implemented in the Oracle Solaris release: ttymon and listen.

TTY Port Monitor (ttymon)

The ttymon port monitor is STREAMS-based and does the following:

- Monitors ports
- Sets terminal modes, baud rates, and line disciplines
- Invokes the login process

The ttymon port monitor provides users the same services that the getty port monitor did under previous versions of SunOS 4.1 software.

The ttymon port monitor runs under the SAC program and is configured with the sacadm command. Each instance of ttymon can monitor multiple ports. These ports are specified in the port monitor's administrative file. The administrative file is configured by using the pmadm and ttyadm commands.

ttymon and the Console Port

Console services are not managed by the Service Access Controller (SAC), nor by any explicit ttymon administration file. ttymon invocations are managed by SMF. As a result, you can no longer invoke ttymon by adding an entry to the /etc/inittab file. A property group with the type, application, and the name ttymon, has been added to the SMF service, svc:/system/console-login:default. The properties within this property group are used by the method script, /lib/svc/method/console-login. This script uses the property values as arguments to the ttymon invocation. Usually, if the values are empty, or if the values are not defined for any of the properties, then the value is not used for ttymon. However, if the ttymon device value is empty, or not set, then /dev/console is used as the default to enable ttymon to run.

The following properties are available under the SMF service, svc:/system/console-login:default:

ttymon/nohangup	Specifies the nohangup property. If set to true, do not force a line hang up by setting the line speed to zero before setting the default or specified speed.
ttymon/prompt	Specifies the prompt string for the console port.
ttymon/terminal_type	Specifies the default terminal type for the console.
ttymon/device	Specifies the console device.
ttymon/label	Specifies the TTY label in the /etc/ttydefs line.

ttymon-Specific Administrative Command (ttyadm)

The ttymon administrative file is updated by the sacadm and pmadm commands, as well as by the ttyadm command. The ttyadm command formats ttymon-specific information and writes it to standard output, providing a means for presenting formatted ttymon-specific data to the sacadm and pmadm commands.

Thus, the ttyadm command does not administer ttymon directly. The ttyadm command complements the generic administrative commands, sacadm and pmadm. For more information, see the ttyadm(1M) man page.

Network Listener Service (`listen`)

The `listen` port monitor runs under the SAC and does the following:

- Monitors the network for service requests
- Accepts requests when they arrive
- Invokes servers in response to those service requests

The `listen` port monitor is configured by using the `sacadm` command. Each instance of `listen` can provide multiple services. These services are specified in the port monitor's administrative file. This administrative file is configured by using the `pmadm` and `nlsadmin` commands.

The network listener process can be used with any connection-oriented transport provider that conforms to the Transport Layer Interface (TLI) specification. In the Oracle Solaris OS, `listen` port monitors can provide additional network services not provided by the `inetd` service.

Special `listen`-Specific Administrative Command (`nlsadmin`)

The `listen` port monitor's administrative file is updated by the `sacadm` and `pmadm` commands, as well as by the `nlsadmin` command. The `nlsadmin` command formats `listen`-specific information and writes it to standard output, providing a means of presenting formatted `listen`-specific data to the `sacadm` and `pmadm` commands.

Thus, the `nlsadmin` command does not administer `listen` directly. The command complements the generic administrative commands, `sacadm` and `pmadm`.

Each network, configured separately, can have at least one instance of the network listener process that is associated with it. The `nlsadmin` command controls the operational states of `listen` port monitors.

The `nlsadmin` command can establish a `listen` port monitor for a given network, configure the specific attributes of that port monitor, and *start* and *kill* the monitor. The `nlsadmin` command can also report on the `listen` port monitors on a machine.

For more information, see the `nlsadmin(1M)` man page.

Administering ttymon Port Monitors

Console administration for ttymon is now managed by SMF. Use the svccfg command to set ttymon system console properties. Continue to use the SAF command, sacadm, to add, list, remove, kill, start, enable, disable, enable, and remove ttymon port monitors.

How to Set the ttymon Console Terminal Type

This procedure shows how to change the console terminal type by using the svccfg command.

1. **Become the root user.**

```
$ su -
Password:
#
```

Note – This method works whether root is a user or a role.

2. **Run the svccfg command to set the property for the service instance that you want to change.**

```
# svccfg -s console-login setprop ttymon/terminal_type = "xterm"
```

where xterm is an example of a terminal type that you might want to use.

3. **(Optional) Restart the service instance.**

```
# svcadm restart svc:/system/console-login:default
```

Caution – If you choose to restart the service instance immediately, you are logged out of the console. If you do not restart the service instance immediately, the property changes apply at the next login prompt on the console.

How to Set the Baud Rate Speed on the ttymon Console Terminal

This procedure shows how to set the baud rate speed on the ttymon console terminal. Support for console speeds on x86 based systems are dependent on the specific platform.

The following are supported console speeds for SPARC based systems:

- 9600 bps
- 19200 bps
- 38400 bps

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2. **Use the eeprom command to set a baud rate speed that is appropriate for your system type.**

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

   For example, to change the baud rate on an x86 based system's console to 38400, type:

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

3. **Change the console line in the /etc/ttydefs file as follows:**

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

4. **Make the following additional changes for your system type.**

   Note that these changes are platform-dependent.

   - **On SPARC based systems:** Change the baud rate speed in the /kernel/drv/options.conf file.

     Use the following command to change the baud rate to 9600:

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

     Use the following command to change the baud rate speed to 19200.

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

     Use the following command to change the baud rate speed to 38400:

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

   - **On x86 based systems:** Change the console speed if the BIOS serial redirection is enabled. The method that you use to change the console speed is platform-dependent.

▼ How to Add a ttymon Port Monitor

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2. **Add a ttymon port monitor.**

   ```
   # sacadm -a -p mbmon -t ttymon -c /usr/lib/saf/ttymon -v 'ttyadm
   -V' -y "TTY Ports a & b"
   ```

`-a` Specifies the *add* port monitor option.

`-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

`-t` Specifies the port monitor *type* as `ttymon`.

`-c` Defines the *command* string used to start the port monitor.

`-v` Specifies the *version* number of the port monitor.

`-y` Defines a comment to describe this instance of the port monitor.

▼ How to View `ttymon` Port Monitor Status

1 Become an administrator.

For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2 View the status of a `ttymon` port monitor.

```
# sacadm -l -p mbmon
```

`-l` Specifies the *list* port monitor status flag.

`-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

Example 1–1 Viewing `ttymon` Port Monitor Status

This example shows how to view a port monitor named, `mbmon`.

```
# sacadm -l -p mbmon
PMTAG  PMTYPE  FLGS RCNT STATUS    COMMAND
mbmon  ttymon  -    0    STARTING  /usr/lib/saf/ttymon #TTY Ports a & b
```

`PMTAG` Identifies the port monitor name, `mbmon`.

`PMTYPE` Identifies the port monitor type, `ttymon`.

`FLGS` Indicates whether the following flags are set:

- d — Do not enable the new port monitor.
- x — Do not start the new port monitor.
- dash (-) — No flags are set.

`RCNT` Indicates the return count value. A return count of `0` indicates that the port monitor is not to be restarted if it fails.

`STATUS` Indicates the current status of the port monitor.

COMMAND	Identifies the command used to start the port monitor.
#TTY Ports a & b	Identifies any comment used to describe the port monitor.

How to Stop a ttymon Port Monitor

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2. **Stop a ttymon port monitor.**

   ```
   # sacadm -k -p mbmon
   ```

-k	Specifies the *kill* port monitor status flag.
-p	Specifies the *pmtag* mbmon as the port monitor tag.

How to Start a ttymon Port Monitor

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2. **Start a killed ttymon port monitor.**

   ```
   # sacadm -s -p mbmon
   ```

-s	Specifies the *start* port monitor status flag.
-p	Specifies the *pmtag* mbmon as the port monitor tag.

How to Disable a ttymon Port Monitor

Disabling a port monitor prevents new services from starting, without affecting existing services.

1. **Become an administrator..**

   For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2. **Disable a ttymon port monitor.**

   ```
   # sacadm -d -p mbmon
   ```

`-d` Specifies the *disable* port monitor status flag.

`-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

How to Enable a `ttymon` Port Monitor

Enabling a `ttymon` port monitor allows it to service new requests.

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2. **Enable a `ttymon` port monitor.**

   ```
   # sacadm -e -p mbmon
   ```

   `-e` Specifies the *enable* port monitor status flag.

   `-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

How to Remove a `ttymon` Port Monitor

Removing a port monitor deletes all the configuration files that are associated with it.

Note – Port monitor configuration files cannot be updated or changed by using the `sacadm` command. To reconfigure a port monitor, *remove* it and then *add* a new one.

---

1. **Become an administrator.**

   For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2. **Remove a `ttymon` port monitor.**

   ```
   # sacadm -r -p mbmon
   ```

   `-r` Specifies the *remove* port monitor status flag.

   `-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

Administering ttymon services (Task Map)

Task	Description	For Instructions
Add a ttymon service.	Use the pmadm command to add a service.	"How to Add a Service" on page 32
View the Status of a TTY Port Service.	Use the pmadmcommand to view the status of a TTY port.	"How to View the Status of a TTY Port Service" on page 33
Enable a port monitor service.	Use the pmadm command with the -e option to enable a port monitor.	"How to Enable a Port Monitor Service" on page 35
Disable a port monitor service.	Use the pmadm command with the -d option to disable a port monitor.	"How to Disable a Port Monitor Service" on page 35

Administering ttymon Services

Use the pmadm command to add services, list the services of one or more ports that are associated with a port monitor, and enable or disable a service.

▼ How to Add a Service

1. **Become the root user..**

2. **Add a standard terminal service to the port monitor.**

```
# pmadm -a -p monitor-name -s a -i root -v 'ttyadm -V' -m "'ttyadm -i 'Terminal
 disabled' -l contty -m ldterm,ttcompat -S y -d /dev/term/a
-s /usr/bin/login'"
```

Note – In this example, the input wraps automatically to the next line. Do not use a Return key or line feed.

`-a` Specifies the *add* port monitor status flag.

`-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

`-s` Specifies the *svctag* `a` as the port monitor *service* tag.

`-i` Specifies the *identity* to be assigned to *svctag* when the service runs.

`-v` Specifies the *version* number of the port monitor.

`-m` Specifies the `ttymon`-specific configuration data formatted by `ttyadm`.

The preceding pmadm command contains an embedded ttyadm command. The options in this embedded command are as follows:

-b Specifies the *bidirectional* port flag.

-i Specifies the *inactive* (disabled) response message.

-l Specifies which TTY *label* in the /etc/ttydefs file to use.

-m Specifies the STREAMS *modules* to push before invoking this service.

-d Specifies the full path name to the *device* to use for the TTY port.

-s Specifies the full path name of the *service* to invoke when a connection request is received. If arguments are required, enclose the command and its arguments in quotation marks (“).

▼ How to View the Status of a TTY Port Service

Use the pmadm command as shown in this procedure to list the status of a TTY port or all the ports that are associated with a port monitor.

1 Become the root user.

2 List one service of a port monitor.

```
# pmadm -l -p monitor-name -s a
```

-l Lists service information about the system.

-p Specifies the *pmtag* mbmon as the port monitor tag.

-s Specifies the *svctag* a as the port monitor *service* tag.

Example 1–2 Viewing the Status of a TTY Port Monitor Service

This example lists all services of the port monitor mbmon.

```
# pmadm -l -p mbmon
PMTAG  PMTYPE  SVCTAG FLAGS ID    <PMSPECIFIC>
mbmon  ttymon  a      -     root  /dev/term/a - - /usr/bin/login - contty
ldterm,ttcompat login:  Terminal disabled tvi925 y  #
```

PMTAG Identifies the port monitor name, mbmon, that is set by using the pmadm -p command.

PMTYPE Identifies the port monitor type, ttymon.

SVCTAG Indicates the service tag value that is set by using the pmadm -s command.

FLAGS	Identifies whether the following flags are set by using the `pmadm -f` command. ■ x — Do not enable the service. ■ u — Create a `utmpx` entry for the service. ■ dash (-) — No flags are set.
ID	Indicates the identity assigned to the service when it is started. This value is set by using the `pmadm -i` command.
<PMSPECIFIC>	*Information*
/dev/term/a	Indicates the TTY port path name that is set by using the `ttyadm -d` command.
-	Indicates whether the following flags are set by using the `ttyadm -c -b -h -I -r` command. ■ `c` — Sets the connect on carrier flag for the port. ■ `b` — Sets the port as bidirectional, allowing both incoming and outgoing traffic. ■ `h` — Suppresses an automatic hangup immediately after an incoming call is received. ■ `I` — Initializes the port. ■ `r`— Forces `ttymon` to wait until it receives a character from the port before it prints the `login:` message. ■ dash (-) — No flags are set.
-	Indicates a value that is set by using the `ttyadm -r` *count* option. This option determines when `ttymon` displays a prompt after receiving data from a port. If *count* is 0, `ttymon` waits until it receives any character. If *count* is greater than 0, `ttymon` waits until *count* new lines have been received. No value is set in this example.
/usr/bin/login	Identifies the full path name of the service to be invoked when a connection is received. This value is set by using the `ttyadm -s` command.
-	Identifies the `ttyadm -t` command's time-out value. This option specifies that `ttymon` should close a port if the open on the port succeeds, and no input data is received in *timeout* seconds. There is no time-out value in this example.
contty	Identifies the TTY label in the `/etc/ttydefs` file. This value is set by using the `ttyadm -l` command.

`ldterm,ttcompat`	Identifies the STREAMS modules to be pushed. These modules are set by using the `ttyadmin -m` command.
`login: Terminal disabled`	Identifies an inactive message to be displayed when the port is disabled. This message is set by using the `ttyadm -i` command.
`tvi925`	Identifies the terminal type, if set, by using the `ttyadm -T` command. The terminal type is `tvi925` in this example.
`y`	Identifies the software carrier value that is set by using the `ttyadm -S` command. `n` turns the software carrier off. `y` turns the software carrier on. The software carrier is turned on in this example.
`#`	Identifies any comment specified with the `pmadm -y` command. There is no comment in this example.

How to Enable a Port Monitor Service

1. **Become the root user.**
2. **Enable a disabled port monitor service.**

   `# pmadm -e -p` *monitor-name* `-s a`

   `-e` Specifies the *enable* flag.

   `-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

   `-s` Specifies the *svctag* `a` as the port monitor *service* tag.

How to Disable a Port Monitor Service

1. **Become the root user.**

   For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2. **Disable a port monitor service.**

   `# pmadm -d -p` *monitor-name* `-s a`

   `-d` Specifies the *disable* flag.

   `-p` Specifies the *pmtag* `mbmon` as the port monitor tag.

   `-s` Specifies the *svctag* `a` as the port monitor *service* tag.

Service Access Facility Administration (Reference)

This section includes reference information for administration of the Service Access Facility.

Files That Are Associated With the SAF

The SAF uses configuration files that can be modified by using the `sacadm` and `pmadm` commands. You should not need to manually edit the configuration files.

File Name	Description
`/etc/saf/_sysconfig`	Per-system configuration script
`/etc/saf/_sactab`	The SAC's administrative file that contains configuration data for the port monitors that the SAC controls
`/etc/saf/`*pmtag*	Home directory for port monitor *pmtag*
`/etc/saf/`*pmtag*`/_config`	Per-port monitor configuration script for port monitor *pmtag* if it exists
`/etc/saf/`*pmtag*`/_pmtab`	Port monitor *pmtag*'s administrative file that contains port monitor-specific configuration data for the services *pmtag* provides
`/etc/saf/`*pmtag*`/`*svctag*	Per-service configuration script for service *svctag*
`/var/saf/log`	The SAC's log file
`/var/saf/`*pmtag*	Directory for files created by *pmtag*, for example, log files

`/etc/saf/_sactab` File

The information in the `/etc/saf/_sactab` file is as follows:

```
# VERSION=1
zsmon:ttymon::0:/usr/lib/saf/ttymon
#
```

`# VERSION=1` Indicates the Service Access Facility version number.

`zsmon` Is the name of the port monitor.

`ttymon` Is the type of port monitor.

`::` Indicates whether the following two flags are set:

d Do not enable the port monitor.

x Do not start the port monitor. No flags are set in this example.

`0` Indicates the return code value. A return count of `0` indicates that the port monitor is not be restarted if the port monitor fails.

`/usr/lib/saf/ttymon` Indicates the port monitor path name.

`/etc/saf/`*pmtab*`/_pmtab` File

The `/etc/saf/`*pmtab*`/_pmtab` file, such as `/etc/saf/zsmon/_pmtab`, is similar to the following:

```
# VERSION=1
ttya:u:root:reserved:reserved:reserved:/dev/term/a:I::/usr/bin/login::9600:
ldterm,ttcompat:ttya login\: ::tvi925:y:#
```

`# VERSION=1` Indicates the Service Access Facility version number.

`ttya` Indicates the service tag.

`x,u` Identifies whether the following flags are set:

x Do not enable the service.

u Create a `utmpx` entry for the service.

`root` Indicates the identity assigned to the service tag.

`reserved` This field is reserved for future use.

`reserved` This field is reserved for future use.

`reserved` This field is reserved for future use.

`/dev/term/a` Indicates the TTY port path name.

`/usr/bin/login` Identifies the full path name of the service to be invoked when a connection is received.

`:c,b,h,I,r:` Indicates whether the following flags are set:

c Sets the connect on carrier flag for the port.

b Sets the port as bidirectional, allowing both incoming and outgoing traffic.

h Suppresses an automatic hand-up immediately after an incoming call is received.

I Initializes the port.

	r Forces `ttymon` to wait until it receives a character from the port before `ttymon` prints the `login:` message.
`9600`	Identifies the TTY label defined in the `/etc/ttydefs` file.
`ldterm,ttcompat`	Identifies the STREAMS modules to be pushed.
`ttya login\:`	Identifies the prompt to be displayed.
`:y/n:`	Indicates yes or no response.
message	Identifies any inactive (disabled) response message.
`tvi925`	Identifies the terminal type.
`y`	Indicates whether the software carrier is set (`y/n`).

Service States

The `sacadm` command controls the states of services. The following list describes the possible states of services.

Enabled *Default state*. When the port monitor is added, the service operates.

Disabled *Default state*. When the port monitor is removed, the service stops.

To determine the state of any particular service, use the following:

```
# pmadm -l -p portmon-name -ssvctag
```

Port Monitor States

The `sacadm` command controls the states of the `ttymon` and `listen` port monitors. The following table describes the possible port monitor states.

State	Description
`Started`	*Default state* – When the port monitor is added, it is automatically started.
`Enabled`	*Default state* – When the port monitor is added, it is automatically ready to accept requests for service.
`Stopped`	*Default state* – When the port monitor is removed, it is automatically stopped.
`Disabled`	*Default state* – When the port monitor is removed, it automatically continues existing services and refuses to add new services.
`Starting`	*Intermediate state* – The port monitor is in the process of starting.

State	Description
`Stopping`	*Intermediate state* – The port monitor has been manually terminated, but it has not completed its shutdown procedure. The port monitor is on the way to becoming stopped.
`Notrunning`	*Inactive state* – The port monitor has been killed. All ports previously monitored are inaccessible. An external user cannot tell whether a port is `disabled` or `notrunning`.
`Failed`	*Inactive state* – The port monitor is unable to start and remain running.

To determine the state of any particular port monitor, use the following command:

```
# sacadm -l -p portmon-name
```

Port States

Ports can be enabled or disabled depending on the state of the port monitor that controls the ports.

State	Description
Serial (`ttymon`) port states	
`Enabled`	The `ttymon` port monitor sends a prompt message to the port and provides login service to it.
`Disabled`	Default state of all ports if `ttymon` is killed or disabled. If you specify this state, `ttymon` sends out the `disabled` message when it receives a connection request.

◆ ◆ ◆ CHAPTER 2

Displaying and Changing System Information (Tasks)

This chapter describes the tasks that are required to display and change the most common system information.

For information about the procedures that are associated with displaying and changing system information, see the following:

- "Displaying System Information (Task Map)" on page 41
- "Changing System Information (Task Map)" on page 46

Using these features, you can display general system information, monitor disk space, set disk quotas and use accounting programs. You can also schedule the `cron` and `at` commands to automatically run routine commands.

Using these features, you can display general system information, monitor disk space, and use the `cron` and `at` commands to automatically run routine commands.

This chapterdoes not cover information about resource management that enables you to allocate, monitor, and control system resources in a flexible way.

For information about managing system resources with resource management, see Chapter 1, "Introduction to Resource Management," in *System Administration Guide: Oracle Solaris Zones, Oracle Solaris 10 Containers, and Resource Management*.

Displaying System Information (Task Map)

Task	Description	For Instructions
Display a system's release information.	Display the contents of the `/etc/release` file to identify the Oracle Solaris release version.	"How to Display a System's Release Information" on page 43

Task	Description	For Instructions
Display a system's host ID number.	Use the `hostid` command to display your system's host id.	"How to Display a System's Host ID Number" on page 43
Display a system's product name.	You can use the `prtconf -b` command to display the product name of a system.	"How to Display a System's Product Name" on page 43
Display a system's installed memory.	Use the `prtconf` command to display information about your system's installed memory.	"How to Display a System's Installed Memory" on page 44
Display a system's date and time.	Use the `date` command to display your system's date and time.	"How to Display the Date and Time" on page 44
Display a system's physical processor type.	Use the `psrinfo -p` command to list the total number of physical processors on a system. Use the `psrinfo -pv` command to list all physical processors on a system and the virtual processors that are associated with each physical processor.	"How to Display a System's Physical Processor Type" on page 45
Display a system's logical processor type.	Use the `psrinfo -v` command to display a system's logical processor type.	"How to Display a System's Logical Processor Type" on page 45

Displaying System Information

Displaying System Information

The following table describes commands that enable you to display general system information.

TABLE 2–1 Commands for Displaying System Information

Command	System Information Displayed	Man Page
`date`	Date and time	`date(1)`
`hostid`	Host ID number	`hostid(1)`
`isainfo`	The number of bits supported by *native* applications on the running system, which can be passed as a token to scripts	`isainfo(1)`
`isalist`	Processor type for x86 based systems	`psrinfo(1M)`

TABLE 2–1 Commands for Displaying System Information *(Continued)*

Command	System Information Displayed	Man Page
`prtconf`	System configuration information, installed memory, and product name	`prtconf(1M)`
`psrinfo`	Processor type	`psrinfo(1M)`
`uname`	Operating system name, release, version, node name, hardware name, and processor type	`uname(1)`

How to Display a System's Release Information

- **Display the contents of the `/etc/release` file to identify your release version.**

```
% cat /etc/release
                            Oracle Solaris Nevada Next Development snv_146 x86
          Copyright (c) 2010, Oracle and/or its affiliates.  All Rights Reserved.
                     Assembled 29 July 2010
```

How to Display a System's Host ID Number

- **To display the host ID number in hexadecimal format, use the `hostid` command.**

Example 2–1 Displaying a System's Host ID Number

The following example shows sample output from the `hostid` command.

```
$ hostid
80a5d34c
```

How to Display a System's Product Name

The `-b` option to the `prtconf` command enables you to display a system's product name. For more information about this feature, see the `prtconf(1M)` man page.

- **To display the product name for your system, use the `prtconf` command with the `-b` option, as follows:**

```
% prtconf -b
```

Example 2–2 Displaying a System's Product Name

This example shows sample output from the `prtconf -b` command.

```
$ prtconf -b
name:  SUNW,Ultra-5_10
model:  SUNW,375-0066
banner-name:  Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 333MHz)
```

This example shows sample output from the `prtconf -vb` command.

```
% prtconf -vb
name:  SUNW,Sun-Fire-T200
banner-name:  Sun Fire T200
compatible: 'sun4v'
idprom:  01840014.4f7e5e84.00000000.7e5e84de.00000000.00000000.00000000.00000000
openprom model:  SUNW,4.30.4.a
openprom version: 'OBP 4.30.4.a 2010/01/06 14:56'
```

▼ How to Display a System's Installed Memory

- **To display the amount of memory that is installed on your system, use the `prtconf` command.**

Example 2–3 Displaying a System's Installed Memory

The following example shows sample output from the `prtconf` command. The `grep Memory` command selects output from the `prtconf` command to display memory information only.

```
$ prtconf | grep Memory
Memory size: 65408 Megabytes
```

▼ How to Display the Date and Time

- **To display the current date and time according to your system clock, use the `date` command.**

Example 2–4 Displaying the Date and Time

The following example shows sample output from the `date` command.

```
$ date
Mon Sep 13 17:32:59 MST 2010
$
```

Command Option to Identify Chip Multithreading Features

The `psrinfo` command has been modified to provide information about physical processors, in addition to information about virtual processors. This enhanced functionality has been added to identify chip multithreading (CMT) features. The new `-p` option reports the total number of physical processors that are in a system. Using the `psrinfo -pv` command will list all the physical processors that are in the system, as well as the virtual processors that are associated with each physical processor. The default output of the `psrinfo` command continues to display the virtual processor information for a system.

For more information, see the `psrinfo(1M)` man page.

For information about the procedures that are associated with this feature, see "How to Display a System's Physical Processor Type" on page 45.

▼ How to Display a System's Physical Processor Type

- **Use the `psrinfo -p` command to display the total number of physical processors on a system.**

```
$ psrinfo -p
1
```

Use the `psrinfo -pv` command to display information about each physical processor on a system, and the virtual processor that is associated with each physical processor.

```
$ psrinfo -pv
The UltraSPARC-IV physical processor has 2 virtual processors (8, 520)
The UltraSPARC-IV physical processor has 2 virtual processors (9, 521)
The UltraSPARC-IV physical processor has 2 virtual processors (10, 522)
The UltraSPARC-IV physical processor has 2 virtual processors (11, 523)
The UltraSPARC-III+ physical processor has 1 virtual processor (16)
The UltraSPARC-III+ physical processor has 1 virtual processor (17)
The UltraSPARC-III+ physical processor has 1 virtual processor (18)
The UltraSPARC-III+ physical processor has 1 virtual processor (19)
```

When you use the `psrinfo -pv` command on an x86 based system, the following output is displayed:

```
$ psrinfo -pv
The i386 physical processor has 2 virtual processors (0, 2)
The i386 physical processor has 2 virtual processors (1, 3)
```

▼ How to Display a System's Logical Processor Type

- **Use the `psrinfo -v` command to display information about a system's processor type.**

```
$ psrinfo -v
```

On an x86 based system, use the `isalist` command to display the virtual processor type.

```
$ isalist
```

Example 2–5 SPARC: Displaying a System's Processor Type

This example shows how to display information about a SPARC based system's processor type.

```
% psrinfo -v
Status of virtual processor 28 as of: 09/13/2010 14:07:47
  on-line since 04/08/2010 21:27:56.
```

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

Example 2–6 x86: Displaying a System's Processor Type

This example shows how to display information about an x86 based system's processor type.

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

Changing System Information (Task Map)

Task	Directions	For Instructions
Manually set a system's date and time.	Manually set your system's date and time by using the `date` *mmddHHMM[[cc]yy]* command-line syntax.	"How to Manually Set a System's Date and Time" on page 46
Set up a message-of-the-day.	Set up a message-of-the-day on your system by editing the `/etc/motd` file.	"How to Set Up a Message-Of-The-Day" on page 47
Change a system's host name.	Change your system's host name by editing the following files: ■ `/etc/nodename` ■ `/etc/hostname.`**host-name* ■ `/etc/inet/hosts`	"How to Change a System's Host Name" on page 48

Changing System Information

This section describes commands that enable you to change general system information.

▼ How to Manually Set a System's Date and Time

1 Become an administrator.

For more information, see "How to Obtain Administrative Rights" in *System Administration Guide: Security Services*.

2 Enter the new date and time.

```
# date mmddHHMM[[cc]yy]
```

mm Month, using two digits.

dd Day of the month, using two digits.

HH Hour, using two digits and a 24-hour clock.

MM Minutes, using two digits.

cc Century, using two digits.

yy Year, using two digits.

See the `date(1)` man page for more information.

3 Verify that you have reset your system's date correctly by using the `date` command with no options.

Example 2–7 Manually Setting a System's Date and Time

The following example shows how to use the `date` command to manually set a system's date and time.

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

▼ How to Set Up a Message-Of-The-Day

Edit the message-of-the-day file, `/etc/motd`, to include announcements or inquiries to all users of a system when they log in. Use this feature sparingly, and edit this file regularly to remove obsolete messages.

1 Become the root user.

```
$ su -
Password:
#
```

Note – This method works whether `root` is a user or a role.

---

2 Edit the `/etc/motd` file and add a message of your choice.

Edit the text to include the message that will be displayed during user login. Include spaces, tabs, and carriage returns.

3 **Verify the changes by displaying the contents of the `/etc/motd` file.**

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

Example 2–8 Setting Up a Message-Of-The-Day

The default message-of-the-day, which is provided when you install Oracle Solaris software, contains version information. The following example shows an edited `/etc/motd` file that provides information about system availability to each user who logs in.

```
$ cat /etc/motd
The system will be down from 7:00 a.m to 2:00 p.m. on
Saturday, July 7, for upgrades and maintenance.
Do not try to access the system during those hours.
Thank you.
```

▼ How to Change a System's Host Name

A system's host name is specified in several different locations.

Remember to update your name service database to reflect the new host name.

Use the following procedure to change or rename a system's host name.

You can also use the `sys-unconfig` command to reconfigure a system, including the host name. For more information, see the `sys-unconfig(1M)` man page.

1 **Become the root user.**

2 **Change the system's host name in the following files:**

- `/etc/nodename`
- `/etc/hostname.`**interface*
- `/etc/inet/hosts`
- `/etc/inet/ipnodes` (Applies *only* to some releases)

3 **(Optional) If you are using a name service, change the system's host name in the `hosts` file.**

4 **Rename the host name directory within the `/var/crash` directory.**

```
# cd /var/crash
# mv old-host-name  new-host-name
```

5 **Reboot the system to activate the new host name.**

```
# init 6
```

◆ ◆ ◆ CHAPTER 3

Scheduling System Tasks (Tasks)

This chapter describes how to schedule routine or single (one-time) system tasks by using the `crontab` and `at` commands.

This chapter also explains how to control access to these commands by using the following files:

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

For information about the procedures that are associated with scheduling system tasks, see the following:

- "Creating and Editing `crontab` Files (Task Map)" on page 49
- "Using the `at` Command (Task Map)" on page 61

Creating and Editing `crontab` Files (Task Map)

Task	Description	For Instructions
Create or edit a `crontab` file.	Use the `crontab -e` command to create or edit a `crontab` file.	"How to Create or Edit a `crontab` File" on page 55
Verify that a `crontab` file exists.	Use the `ls -l` command to verify the contents of the `/var/spool/cron/crontabs` file.	"How to Verify That a `crontab` File Exists" on page 56
Display a `crontab`file.	Use the `ls -l` command to display the `crontab` file.	"How to Display a `crontab` File" on page 56

Task	Description	For Instructions
Remove a `crontab` file.	The `crontab` file is set up with restrictive permissions Use the `crontab -r` command, rather than the `rm` command to remove a `crontab` file.	"How to Remove a `crontab` File" on page 57
Deny `crontab` access.	To deny users access to `crontab` commands, add user names to the `/etc/cron.d/cron.deny` file by editing this file.	"How to Deny `crontab` Command Access" on page 59
Limit `crontab` access to specified users.	To allow users access to the `crontab` command, add user names to the `/etc/cron.d/cron.allow` file.	"How to Limit `crontab` Command Access to Specified Users" on page 60

Ways to Automatically Execute System Tasks

You can set up many system tasks to execute automatically. Some of these tasks should occur at regular intervals. Other tasks need to run only once, perhaps during off hours such as evenings or weekends.

This section contains overview information about two commands, `crontab` and `at`, which enable you to schedule routine tasks to execute automatically. The `crontab` command schedules repetitive commands. The `at` command schedules tasks that execute once.

The following table summarizes `crontab` and `at` commands, as well as the files that enable you to control access to these commands.

TABLE 3–1 Command Summary: Scheduling System Tasks

Command	What It Schedules	Location of Files	Files That Control Access
`crontab`	Multiple system tasks at regular intervals	`/var/spool/cron/crontabs`	`/etc/cron.d/cron.allow` and `/etc/cron.d/cron.deny`
`at`	A single system task	`/var/spool/cron/atjobs`	`/etc/cron.d/at.deny`

For Scheduling Repetitive Jobs: `crontab`

You can schedule routine system administration tasks to execute daily, weekly, or monthly by using the `crontab` command.

Daily crontab system administration tasks might include the following:

- Removing files more than a few days old from temporary directories
- Executing accounting summary commands
- Taking snapshots of the system by using the df and ps commands
- Performing daily security monitoring
- Running system backups

Weekly crontab system administration tasks might include the following:

- Rebuilding the catman database for use by the man -k command
- Running the fsck -n command to list any disk problems

Monthly crontab system administration tasks might include the following:

- Listing files not used during a specific month
- Producing monthly accounting reports

Additionally, users can schedule crontab commands to execute other routine system tasks, such as sending reminders and removing backup files.

For step-by-step instructions on scheduling crontab jobs, see "How to Create or Edit a crontab File" on page 55.

For Scheduling a Single Job: at

The at command allows you to schedule a job for execution at a later time. The job can consist of a single command or a script.

Similar to crontab, the at command allows you to schedule the automatic execution of routine tasks. However, unlike crontab files, at files execute their tasks once. Then, they are removed from their directory. Therefore, the at command is most useful for running simple commands or scripts that direct output into separate files for later examination.

Submitting an at job involves typing a command and following the at command syntax to specify options to schedule the time your job will be executed. For more information about submitting at jobs, see "Description of the at Command" on page 62.

The at command stores the command or script you ran, along with a copy of your current environment variable, in the /var/spool/cron/atjobs directory. Your at job file name is given a long number that specifies its location in the at queue, followed by the .a extension, such as 793962000.a.

The cron daemon checks for at jobs at startup and listens for new jobs that are submitted. After the cron daemon executes an at job, the at job's file is removed from the atjobs directory. For more information, see the at(1) man page.

For step-by-step instructions on scheduling at jobs, see "How to Create an at Job" on page 63.

Scheduling a Repetitive System Task (cron)

The following sections describe how to create, edit, display, and remove crontab files, as well as how to control access to them.

Inside a crontab File

The cron daemon schedules system tasks according to commands found within each crontab file. A crontab file consists of commands, one command per line, that will be executed at regular intervals. The beginning of each line contains date and time information that tells the cron daemon when to execute the command.

For example, a crontab file named root is supplied during SunOS software installation. The file's contents include these command lines:

```
10 3 * * * /usr/sbin/logadm      (1)
15 3 * * 0 /usr/lib/fs/nfs/nfsfind      (2)
1 2 * * * [ -x /usr/sbin/rtc ] && /usr/sbin/rtc -c > /dev/null 2>&1      (3)
30 3 * * * [ -x /usr/lib/gss/gsscred_clean ] && /usr/lib/gss/gsscred_clean      (4)
```

The following describes the output for each of these command lines:

- The first line runs the logadm command at 3:10 a.m. every day.
- The second line executes the nfsfind script every Sunday at 3:15 a.m.
- The third line runs a script that checks for daylight savings time (and make corrections, if necessary) at 2:10 a.m. daily.

  If there is no RTC time zone, nor an /etc/rtc_config file, this entry does nothing.

  x86 only – The /usr/sbin/rtc script can only be run on an x86 based system.

- The fourth line checks for (and removes) duplicate entries in the Generic Security Service table, /etc/gss/gsscred_db, at 3:30 a.m. daily.

For more information about the syntax of lines within a crontab file, see "Syntax of crontab File Entries" on page 53.

The crontab files are stored in the /var/spool/cron/crontabs directory. Several crontab files besides root are provided during SunOS software installation. See the following table.

TABLE 3–2 Default crontab Files

crontab File	Function
adm	Accounting

TABLE 3–2 Default crontab Files *(Continued)*

crontab File	Function
lp	Printing
root	General system functions and file system cleanup
sys	Performance data collection
uucp	General uucp cleanup

Besides the default crontab files, users can create crontab files to schedule their own system tasks. Other crontab files are named after the user accounts in which they are created, such as bob, mary, smith, or jones.

To access crontab files that belong to root or other users, superuser privileges are required.

Procedures explaining how to create, edit, display, and remove crontab files are described in subsequent sections.

How the cron Daemon Handles Scheduling

The cron daemon manages the automatic scheduling of crontab commands. The role of the cron daemon is to check the /var/spool/cron/crontab directory for the presence of crontab files.

The cron daemon performs the following tasks at startup:

- Checks for new crontab files.
- Reads the execution times that are listed within the files.
- Submits the commands for execution at the proper times.
- Listens for notifications from the crontab commands regarding updated crontab files.

In much the same way, the cron daemon controls the scheduling of at files. These files are stored in the /var/spool/cron/atjobs directory. The cron daemon also listens for notifications from the crontab commands regarding submitted at jobs.

Syntax of crontab File Entries

A crontab file consists of commands, one command per line, that execute automatically at the time specified by the first five fields of each command line. These five fields, described in the following table, are separated by spaces.

TABLE 3–3 Acceptable Values for crontab Time Fields

Time Field	Values
Minute	0-59
Hour	0-23
Day of month	1-31
Month	1-12
Day of week	0-6 (0 = Sunday)

Follow these guidelines for using special characters in crontab time fields:

- Use a space to separate each field.
- Use a comma to separate multiple values.
- Use a hyphen to designate a range of values.
- Use an asterisk as a wildcard to include all possible values.
- Use a comment mark (#) at the beginning of a line to indicate a comment or a blank line.

For example, the following crontab command entry displays a reminder in the user's console window at 4 p.m. on the first and fifteenth days of every month.

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

Each command within a crontab file must consist of one line, even if that line is very long. The crontab file does not recognize extra carriage returns. For more detailed information about crontab entries and command options, refer to the crontab(1) man page.

Creating and Editing crontab Files

The simplest way to create a crontab file is to use the crontab -e command. This command invokes the text editor that has been set for your system environment. The default editor for your system environment is defined in the EDITOR environment variable. If this variable has not been set, the crontab command uses the default editor, ed. Preferably, you should choose an editor that you know well.

The following example shows how to determine if an editor has been defined, and how to set up vi as the default.

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

When you create a crontab file, it is automatically placed in the /var/spool/cron/crontabs directory and is given your user name. You can create or edit a crontab file for another user, or root, if you have superuser privileges.

▼ How to Create or Edit a crontab File

Before You Begin If you are creating or editing a crontab file that belongs to root or another user you must become root.

You do not need to be root to edit your own crontab file.

1 Create a new crontab file, or edit an existing file.

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

where *username* specifies the name of the user's account for which you want to create or edit a crontab file. You can create your own crontab file without superuser privileges, but you must have superuser privileges to creating or edit a crontab file for root or another user.

Caution – If you accidentally type the crontab command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing crontab file would be overwritten with an empty file.

2 Add command lines to the crontab file.

Follow the syntax described in "Syntax of crontab File Entries" on page 53. The crontab file will be placed in the /var/spool/cron/crontabs directory.

3 Verify your crontab file changes.

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

Example 3–1 Creating a crontab File

The following example shows how to create a crontab file for another user.

```
# crontab -e jones
```

The following command entry added to a new crontab file automatically removes any log files from the user's home directory at 1:00 a.m. every Sunday morning. Because the command entry does not redirect output, redirect characters are added to the command line after *.log. Doing so ensures that the command executes properly.

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

▼ How to Verify That a crontab File Exists

- **To verify that a crontab file exists for a user, use the ls -l command in the /var/spool/cron/crontabs directory. For example, the following output shows that crontab files exist for users jones and smith.**

```
$ ls -l /var/spool/cron/crontabs
-rw-r--r--  1 root     sys          190 Feb 26 16:23 adm
-rw-------  1 root     staff        225 Mar  1  9:19 jones
-rw-r--r--  1 root     root        1063 Feb 26 16:23 lp
-rw-r--r--  1 root     sys          441 Feb 26 16:25 root
-rw-------  1 root     staff         60 Mar  1  9:15 smith
-rw-r--r--  1 root     sys          308 Feb 26 16:23 sys
```

Verify the contents of user's crontab file by using the crontab -l command as described in "How to Display a crontab File" on page 56.

Displaying crontab Files

The crontab -l command displays the contents of a crontab file much the same way that the cat command displays the contents of other types of files. You do not have to change the directory to /var/spool/cron/crontabs directory (where crontab files are located) to use this command.

By default, the crontab -l command displays your own crontab file. To display crontab files that belong to other users, you must be superuser.

▼ How to Display a crontab File

Before You Begin Become the root user to display a crontab file that belongs to root or another user.

You do not need to become root to display your own crontab file.

- **Display the crontab file.**

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

where *username* specifies the name of the user's account for which you want to display a crontab file. Displaying another user's crontab file requires superuser privileges.

Caution – If you accidentally type the crontab command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing crontab file would be overwritten with an empty file.

Example 3–2 Displaying a `crontab` File

This example shows how to use the `crontab -l` command to display the contents of the user's default `crontab` file.

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

Example 3–3 Displaying the Default `root` `crontab` file.

This example shows how to display the default `root` `crontab` file.

```
$ suPassword:

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

Example 3–4 Displaying the `crontab` File of Another User

This example shows how to display the `crontab` file that belongs to another user.

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

Removing `crontab` Files

By default, `crontab` file protections are set up so that you cannot inadvertently delete a `crontab` file by using the `rm` command. Instead, use the `crontab -r` command to remove `crontab` files.

By default, the `crontab -r` command removes your own `crontab` file.

You do not have to change the directory to `/var/spool/cron/crontabs` (where `crontab` files are located) to use this command.

▼ How to Remove a `crontab` File

Before You Begin Become the root user to remove a `crontab` file that belongs to `root` or another user. Roles contain authorizations and privileged commands.

You do not need to become root to remove your own crontab file.

1 **Remove the crontab file.**

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

where *username* specifies the name of the user's account for which you want to remove a crontab file. Removing crontab files for another user requires superuser privileges.

Caution – If you accidentally type the crontab command with no option, press the interrupt character for your editor. This character allows you to quit without saving changes. If you instead saved changes and exited the file, the existing crontab file would be overwritten with an empty file.

2 **Verify that the crontab file has been removed.**

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

Example 3–5 Removing a crontab File

The following example shows how user smith uses the crontab -r command to remove his own crontab file.

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

Controlling Access to the crontab Command

You can control access to the crontab command by using two files in the /etc/cron.d directory: cron.deny and cron.allow. These files permit only specified users to perform crontab command tasks such as creating, editing, displaying, or removing their own crontab files.

The cron.deny and cron.allow files consist of a list of user names, one user name per line.

These access control files work together as follows:

- If cron.allow exists, only the users who are listed in this file can create, edit, display, or remove crontab files.
- If cron.allow does not exist, all users can submit crontab files, except for users who are listed in cron.deny.
- If neither cron.allow nor cron.deny exists, superuser privileges are required to run the crontab command.

Superuser privileges are required to edit or create the cron.deny and cron.allow files.

The cron.deny file, which is created during SunOS software installation, contains the following user names:

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

None of the user names in the default cron.deny file can access the crontab command. You can edit this file to add other user names that will be denied access to the crontab command.

No default cron.allow file is supplied. So, after Oracle Solaris software installation, all users (except users who are listed in the default cron.deny file) can access the crontab command. If you create a cron.allow file, only these users can access the crontab command.

▼ How to Deny crontab Command Access

1 Become the root user.

```
$ su -
Password:
#
```

Note – This method works whether root is a user or a role.

---

2 Edit the /etc/cron.d/cron.deny file and add user names, one user per line. Include users who will be denied access to the crontab commands.

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

3 Verify that the /etc/cron.d/cron.deny file contains the new entries.

```
# cat /etc/cron.d/cron.deny
daemon
```

```
bin
nuucp
listen
nobody
noaccess
```

▼ How to Limit `crontab` Command Access to Specified Users

1. **Become the root user.**
2. **Create the `/etc/cron.d/cron.allow` file.**
3. **Add the `root` user name to the `cron.allow` file.**

   If you do not add `root` to the file, superuser access to `crontab` commands will be denied.
4. **Add the user names, one user name per line.**

   Include users that will be allowed to use the `crontab` command.

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

Example 3–6 Limiting `crontab` Command Access to Specified Users

The following example shows a `cron.deny` file that prevents user names `jones`, `temp`, and `visitor` from accessing the `crontab` command.

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

The following example shows a `cron.allow` file. The users `root`, `jones`, `lp`, and `smith` are the only users who can access the `crontab` command.

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

```
jones
lp
smith
```

How to Verify Limited crontab Command Access

To verify if a specific user can access the crontab command, use the crontab -l command while you are logged into the user account.

```
$ crontab -l
```

If the user can access the crontab command, and already has created a crontab file, the file is displayed. Otherwise, if the user can access the crontab command but no crontab file exists, a message similar to the following message is displayed:

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

Either this user either is listed in the cron.allow file (if the file exists), or the user is not listed in the cron.deny file.

If the user cannot access the crontab command, the following message is displayed whether or not a previous crontab file exists:

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

This message means that either the user is not listed in the cron.allow file (if the file exists), or the user is listed in the cron.deny file.

Using the at Command (Task Map)

Task	Description	For Instructions
Create an at job.	Use the atcommand to do the following: ■ Start the at utility from the command line. ■ Type the commands or scripts that you want to execute, one per line. ■ Exit the at utility and save the job.	"How to Create an at Job" on page 63
Display the at queue.	User the atq command to display the at queue.	"How to Display the at Queue" on page 64

Task	Description	For Instructions
Verify an at job.	Use the atq command to confirm that at jobs that belong to a specific user have been submitted to the queue.	"How to Verify an at Job" on page 64
Display at jobs.	Use the at -l[*job-id* to display at jobs. that have been submitted to the queue.	"How to Display at Jobs" on page 64
Remove at jobs.	Use the at -r [*job-id*] command to remove at jobs from the queue.	"How to Remove at Jobs" on page 65
Deny access to the at command.	To deny users access to the atcommand, edit the /etc/cron.d/at.deny file.	"How to Deny Access to the at Command" on page 66

Scheduling a Single System Task (at)

The following sections describe how to use the at command to perform the following tasks:

- Schedule jobs (command and scripts) for execution at a later time
- How to display and remove these jobs
- How to control access to the at command

By default, users can create, display, and remove their own at job files. To access at files that belong to root or other users, you must have superuser privileges.

When you submit an at job, it is assigned a job identification number along with the .a extension. This designation becomes the job's file name, as well as its queue number.

Description of the at Command

Submitting an at job file involves these steps:

1. Invoking the at utility and specifying a command execution time.
2. Typing a command or script to execute later.

> **Note –** If output from this command or script is important, be sure to direct the output to a file for later examination.

For example, the following at job removes core files from the user account smith near midnight on the last day of July.

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

Controlling Access to the at Command

You can set up a file to control access to the at command, permitting only specified users to create, remove, or display queue information about their at jobs. The file that controls access to the at command, /etc/cron.d/at.deny, consists of a list of user names, one user name per line. The users who are listed in this file cannot access at commands.

The at.deny file, which is created during SunOS software installation, contains the following user names:

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

With superuser privileges, you can edit the at.deny file to add other user names whose at command access you want to restrict.

▼ How to Create an at Job

1 Start the at utility, specifying the time you want your job executed.

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

-m Sends you email after the job is completed.

time Specifies the hour that you want to schedule the job. Add am or pm if you do not specify the hours according to the 24-hour clock. Acceptable keywords are midnight, noon, and now. Minutes are optional.

date Specifies the first three or more letters of a month, a day of the week, or the keywords today or tomorrow.

2 At the at prompt, type the commands or scripts that you want to execute, one per line.

You may type more than one command by pressing Return at the end of each line.

3 Exit the at utility and save the at job by pressing Control-D.

Your at job is assigned a queue number, which is also the job's file name. This number is displayed when you exit the at utility.

Example 3–7 Creating an at Job

The following example shows the at job that user jones created to remove her backup files at 7:30 p.m. She used the -m option so that she would receive an email message after her job completed.

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

She received a email message which confirmed the execution of her at job.

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

The following example shows how jones scheduled a large at job for 4:00 a.m. Saturday morning. The job output was directed to a file named big.file.

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

How to Display the at Queue

- **To check your jobs that are waiting in the at queue, use the atq command.**

  ```
  $ atq
  ```

  This command displays status information about the at jobs that you have created.

How to Verify an at Job

- **To verify that you have created an at job, use the atq command. In the following example, the atq command confirms that at jobs that belong to jones have been submitted to the queue.**

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

How to Display at Jobs

- **To display information about the execution times of your at jobs, use the at -l command.**

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

where the -l *job-id* option identifies the identification number of the job whose status you want to display.

Example 3–8 Displaying at Jobs

The following example shows output from the at -l command, which provides information about the status of all jobs submitted by a user.

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

The following example shows the output that is displayed when a single job is specified with the at -l command.

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

▼ How to Remove at Jobs

Before You Begin Become the root user to remove an at job that belongs to root or another user. Roles contain authorizations and privileged commands.

You do not need to become root to remove you own at job.

1 Remove the at job from the queue before the job is executed.

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

where the -r *job-id* option specifies the identification number of the job you want to remove.

2 Verify that the at job is removed by using the at -l (or the atq) command.

The at -l command displays the jobs remaining in the at queue. The job whose identification number you specified should not appear.

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

Example 3–9 Removing at Jobs

In the following example, a user wants to remove an at job that was scheduled to execute at 4 a.m. on July 17th. First, the user displays the at queue to locate the job identification number. Next, the user removes this job from the at queue. Finally, the user verifies that this job has been removed from the queue.

```
$ at -l
897543900.a    Sat Jul 14 23:45:00 2003
```

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

▼ How to Deny Access to the at Command

1. **Become the root user.**
2. **Edit the /etc/cron.d/at.deny file and add the names of users, one user name per line, that will be prevented from using the at commands.**

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

Example 3–10 Denying at Access

The following example shows an at.deny file that has been edited so that the users smith and jones cannot access the at command.

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

▼ How to Verify That at Command Access Is Denied

- **To verify that a username was added correctly to the /etc/cron.d/at.deny file, use the at -l command while logged in as the user. If the user smith cannot access the at command, the following message is displayed:**

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

Likewise, if the user tries to submit an at job, the following message is displayed:

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

This message confirms that the user is listed in the at.deny file.

If at command access is allowed, then the at -l command returns nothing.

◆ ◆ ◆ CHAPTER 4

Managing System Processes (Tasks)

This chapter describes the procedures for managing system processes.

For information about the procedures that are associated with managing system processes, see the following:

- "Managing System Processes (Task Map)" on page 69
- "Managing Process Class Information (Task Map)" on page 79

For overview information about managing system processes, see the following:

- "Commands for Managing System Processes" on page 70
- "Managing Process Class Information" on page 80

Managing System Processes (Task Map)

Task	Description	For Instructions
List processes.	Use the `ps` command to list all the processes on a system.	"How to List Processes" on page 73
Display information about processes.	Use the `pgrep` command to obtain the process IDs for processes that you want to display more information about.	"How to Display Information About Processes" on page 74
Control processes.	Locate processes by using the `pgrep` command. Then, use the appropriate `pcommand` (`/proc`) to control the process. See Table 4–3 for a description of the (`/proc`) commands.	"How to Control Processes" on page 76

Task	Description	For Instructions
Kill a process.	Locate a process, either by process name or process ID. You can use either the `pkill` or `kill` commands to terminate the process.	"How to Terminate a Process (`pkill`)" on page 76 "How to Terminate a Process (`kill`)" on page 77

Commands for Managing System Processes

The following table describes the commands for managing system processes.

TABLE 4–1 Commands for Managing Processes

Command	Description	Man Page
`ps`, `pgrep`, `prstat`, `pkill`	Checks the status of active processes on a system, as well as displays detailed information about the processes.	`ps`(1), `pgrep`(1), and`prstat`(1M)
`pkill`	Functions identically to `pgrep` but finds or signals processes by name or other attribute and terminates the process. Each matching process is signaled as if by the `kill` command, instead of having its process ID printed.	`pgrep`(1), and `pkill`(1) `kill`(1)
`pargs, preap`	Assists with processes debugging.	`pargs`(1), and `preap`(1)
`dispadmin`	Lists default process scheduling policies.	`dispadmin`(1M)
`priocntl`	Assigns processes to a priority class and manages process priorities.	`priocntl`(1)
`nice`	Changes the priority of a timesharing process.	`nice`(1)
`psrset`	Binds specific process groups to a group of processors rather than to just a single processor.	`psrset`(1M)

Using the ps Command

The ps command enables you to check the status of active processes on a system, as well as display technical information about the processes. This data is useful for administrative tasks, such as determining how to set process priorities.

Depending on which options you use, the ps command reports the following information:

- Current status of the process
- Process ID
- Parent process ID
- User ID
- Scheduling class
- Priority
- Address of the process
- Memory used
- CPU time used

The following table describes some fields that are reported by the ps command. Which fields are displayed depend on which option you choose. For a description of all available options, see the ps(1) man page.

TABLE 4–2 Summary of Fields in ps Reports

Field	Description
UID	The effective user ID of the process's owner.
PID	The process ID.
PPID	The parent process ID.
C	The processor xutilization for scheduling. This field is not displayed when the -c option is used.
CLS	The scheduling class to which the process belongs such as real-time, system, or timesharing. This field is included only with the -c option.
PRI	The kernel thread's scheduling priority. Higher numbers indicate a higher priority.
NI	The process's nice number, which contributes to its scheduling priority. Making a process “nicer” means lowering its priority.
ADDR	The address of the proc structure.
SZ	The virtual address size of the process.
WCHAN	The address of an event or lock for which the process is sleeping.
STIME	The starting time of the process in hours, minutes, and seconds.

TABLE 4–2 Summary of Fields in ps Reports *(Continued)*

Field	Description
TTY	The terminal from which the process, or its parent, was started. A question mark indicates that there is no controlling terminal.
TIME	The total amount of CPU time used by the process since it began.
CMD	The command that generated the process.

Using the /proc File System and Commands

You can display detailed information about the processes that are listed in the /proc directory by using process commands. The following table lists the /proc process commands. The /proc directory is also known as the process file system (PROCFS). Images of active processes are stored here by their process ID number.

TABLE 4–3 Process Commands (/proc)

Process Command	Description
pcred	Displays process credential information
pfiles	Reports fstat and fcntl information for open files in a process
pflags	Prints /proc tracing flags, pending signals and held signals, and other status information
pldd	Lists the dynamic libraries that are linked into a process
pmap	Prints the address space map of each process
psig	Lists the signal actions and handlers of each process
prun	Starts each process
pstack	Prints a hex+symbolic stack trace for each lwp in each process
pstop	Stops each process
ptime	Times a process by using microstate accounting
ptree	Displays the process trees that contain the process
pwait	Displays status information after a process terminates
pwdx	Displays the current working directory for a process

For more information, see proc(1).

The process tools are similar to some options of the ps command, except that the output that is provided by these commands is more detailed.

In general, the process commands do the following:

- Display more information about processes, such as `fstat` and `fcntl`, working directories, and trees of parent and child processes.
- Provide control over processes by allowing users to stop or resume them.

Managing Processes With Process Commands (`/proc`)

You can display detailed, technical information about processes or control active processes by using some of the process commands. Table 4–3 lists some of the `/proc` commands.

If a process becomes trapped in an endless loop, or if the process takes too long to execute, you might want to stop (kill) the process. For more information about stopping processes using the `kill` or the `pkill` command, see Chapter 4, "Managing System Processes (Tasks)."

The `/proc` file system is a directory hierarchy that contains additional subdirectories for state information and control functions.

The `/proc` file system also provides an xwatchpoint facility that is used to remap read-and-write permissions on the individual pages of a process's address space. This facility has no restrictions and is MT-safe.

Debugging tools have been modified to use `/proc`'s xwatchpoint facility, which means that the entire xwatchpoint process is faster.

The following restrictions have been removed when you set xwatchpoints by using the `dbx` debugging tool:

- Setting xwatchpoints on local variables on the stack due to SPARC based system register windows.
- Setting xwatchpoints on multithreaded processes.

For more information, see the `proc(4)`, and `mdb(1)` man pages.

▼ How to List Processes

- **Use the `ps` command to list all the processes on a system.**

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

`ps` Displays only the processes that are associated with your login session.

`-ef` Displays full information about all the processes that are being executed on the system.

`-c` Displays process scheduler information.

Example 4–1 Listing Processes

The following example shows output from the `ps` command when no options are used.

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

The following example shows output from the `ps -ef` command. This output shows that the first process that is executed when the system boots is `sched` (the swapper) followed by the `init` process, `pageout`, and so on.

```
$ ps -ef
     UID   PID  PPID  C    STIME TTY      TIME CMD
    root     0     0  0   Dec 20 ?        0:17 sched
    root     1     0  0   Dec 20 ?        0:00 /etc/init -
    root     2     0  0   Dec 20 ?        0:00 pageout
    root     3     0  0   Dec 20 ?        4:20 fsflush
    root   374   367  0   Dec 20 ?        0:00 /usr/lib/saf/ttymon
    root   367     1  0   Dec 20 ?        0:00 /usr/lib/saf/sac -t 300
    root   126     1  0   Dec 20 ?        0:00 /usr/sbin/rpcbind
    root    54     1  0   Dec 20 ?        0:00 /usr/lib/sysevent/syseventd
    root    59     1  0   Dec 20 ?        0:00 /usr/lib/picl/picld
    root   178     1  0   Dec 20 ?        0:03 /usr/lib/autofs/automountd
    root   129     1  0   Dec 20 ?        0:00 /usr/sbin/keyserv
    root   213     1  0   Dec 20 ?        0:00 /usr/lib/lpsched
    root   154     1  0   Dec 20 ?        0:00 /usr/sbin/inetd -s
    root   139     1  0   Dec 20 ?        0:00 /usr/lib/netsvc/yp/ypbind ...
    root   191     1  0   Dec 20 ?        0:00 /usr/sbin/syslogd
    root   208     1  0   Dec 20 ?        0:02 /usr/sbin/nscd
    root   193     1  0   Dec 20 ?        0:00 /usr/sbin/cron
    root   174     1  0   Dec 20 ?        0:00 /usr/lib/nfs/lockd
  daemon   175     1  0   Dec 20 ?        0:00 /usr/lib/nfs/statd
    root   376     1  0   Dec 20 ?        0:00 /usr/lib/ssh/sshd
    root   226     1  0   Dec 20 ?        0:00 /usr/lib/power/powerd
    root   315     1  0   Dec 20 ?        0:00 /usr/lib/nfs/mountd
    root   237     1  0   Dec 20 ?        0:00 /usr/lib/utmpd
    .
    .
    .
```

▼ How to Display Information About Processes

1 Obtain the process ID of the process that you want to display more information about.

```
# pgrep process
```

where *process* is the name of the process you want to display more information about.

The process ID is displayed in the first column of the output.

2 Display the process information that you need.

```
# /usr/bin/pcommand pid
```

pcommand Is the (`/proc`) command that you want to run. Table 4–3 lists and describes these commands.

pid Identifies the process ID.

Example 4–2 Displaying Information About Processes

The following example shows how to use process commands to display more information about a `cron` process.

```
# pgrep cron      1
4780
# pwdx 4780      2
4780:   /var/spool/cron/atjobs
# ptree 4780      3
4780  /usr/sbin/cron
# pfiles 4780      4
4780:   /usr/sbin/cron
  Current rlimit: 256 file descriptors
   0: S_IFCHR mode:0666 dev:290,0 ino:6815752 uid:0 gid:3 rdev:13,2
      O_RDONLY|O_LARGEFILE
      /devices/pseudo/mm@0:null
   1: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
      O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
      /var/cron/log
   2: S_IFREG mode:0600 dev:32,128 ino:42054 uid:0 gid:0 size:9771
      O_WRONLY|O_APPEND|O_CREAT|O_LARGEFILE
      /var/cron/log
   3: S_IFIFO mode:0600 dev:32,128 ino:42049 uid:0 gid:0 size:0
      O_RDWR|O_LARGEFILE
      /etc/cron.d/FIFO
   4: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
      O_RDWR|O_NONBLOCK
   5: S_IFIFO mode:0000 dev:293,0 ino:4630 uid:0 gid:0 size:0
      O_RDWR
```

1. Obtains the process ID for the `cron` process
2. Displays the current working directory for the `cron` process
3. Displays the process tree that contains the `cron` process
4. Displays `fstat` and `fcntl` information

▼ How to Control Processes

1 **Obtain the process ID of the process that you want to control.**

```
# pgrep process
```

where *process* is the name of the process you want to control.

The process ID displayed in the first column of the output.

2 **Use the appropriate process command to control the process.**

```
# /usr/bin/pcommand pid
```

pcommand Is the process (`/proc`) command that you want to run. Table 4–3 lists and describes these commands.

pid Identifies the process ID.

3 **Verify the process status.**

```
# ps -ef | grep pid
```

Terminating a Process (`pkill`, `kill`)

Sometimes, you might need to stop (kill) a process. The process might be in an endless loop. Or, you might have started a large job that you want to stop before it is completed. You can kill any process that you own. Superuser can kill any process in the system except for those processes with process IDs of `0`, `1`, `2`, `3`, and `4`. Killing these processes most likely will crash the system.

For more information, see the `pgrep(1)` and `pkill(1)` and `kill(1)` man pages.

▼ How to Terminate a Process (`pkill`)

1 **To terminate the process of another user, become root.**

2 **Obtain the process ID for the process that you want to terminate.**

```
$ pgrep process
```

where *process* is the name of the process that you want to terminate.

For example:

```
$ pgrep netscape
587
566
```

The process ID is displayed in the output.

Note – To obtain process information about a Sun Ray, use the following commands:

```
# ps -fu user
```

This command lists all user processes.

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

This command locates a specific process for a user.

3 Terminate the process.

```
$ pkill [signal] process
```

signal When no signal is included in the `pkill` command-line syntax, the default signal that is used is –15 (`SIGKILL`). Using the –9 signal (`SIGTERM`) with the `pkill` command ensures that the process terminates promptly. However, the –9 signal should not be used to kill certain processes, such as a database process, or an LDAP server process. The result is that data might be lost.

process Is the name of the process to stop.

Tip – When using the `pkill` command to terminate a process, first try using the command by itself, without including a signal option. Wait a few minutes to see if the process terminates before using the `pkill` command with the -9 signal.

4 Verify that the process has been terminated.

```
$ pgrep process
```

The process you terminated should no longer be listed in the output of the `pgrep` command.

▼ How to Terminate a Process (`kill`)

1 To terminate the process of another user, become root.

2 Obtain the process ID of the process that you want to terminate.

```
# ps -fu user
```

where *user* is the user that you want to display processes for.

The process ID is displayed in the first column of the output.

3 Terminate the process.

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

signal When no signal is included in the `kill` command-line syntax, the default signal that is used is –15 (`SIGKILL`). Using the –9 signal (`SIGTERM`) with the `kill` command ensures that the process terminates promptly. However, the –9 signal should not be used to kill certain processes, such as a database process, or an LDAP server process. The result is that data might be lost.

pid Is the process ID of the process that you want to terminate.

Tip – When using the `kill` command to stop a process, first try using the command by itself, without including a signal option. Wait a few minutes to see if the process terminates before using the `kill` command with the -9 signal.

---

4 Verify that the process has been terminated.

```
$ pgrep pid
```

The process you terminated should no longer be listed in the output of the `pgrep` command.

Debugging a Process (`pargs`, `preap`)

The `pargs` command and the `preap` command improve process debugging. The `pargs` command prints the arguments and environment variables that are associated with a live process or core file. The `preap` command removes defunct (zombie) processes. A zombie process has not yet had its exit status claimed by its parent. These processes are generally harmless but can consume system resources if they are numerous. You can use the `pargs` and `preap` commands to examine any process that you have the privileges to examine. As superuser, you can examine any process.

For information about using the `preap` command, see the `preap(1)` man page. For information about the using the `pargs` command, see the `pargs(1)` man page. See also, the `proc(1)` man page.

EXAMPLE 4–3 Debugging a Process (`pargs`)

The `pargs` command solves a long-standing problem of being unable to display with the `ps` command all the arguments that are passed to a process. The following example shows how to use the `pargs` command in combination with the `pgrep` command to display the arguments that are passed to a process.

```
# pargs 'pgrep ttymon'
579:    /usr/lib/saf/ttymon -g -h -p system-name console login:
-T sun -d /dev/console -l
argv[0]: /usr/lib/saf/ttymon
argv[1]: -g
argv[2]: -h
argv[3]: -p
```

EXAMPLE 4–3 Debugging a Process (`pargs`) *(Continued)*

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

The following example shows how to use the `pargs -e` command to display the environment variables that are associated with a process.

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

Managing Process Class Information (Task Map)

Task	Description	For Instructions
Display basic information about process classes.	Use the `priocntl -l` command. to Display process scheduling classes and priority ranges.	"How to Display Basic Information About Process Classes (`priocntl`)" on page 80
Display the global priority of a process.	Use the `ps -ecl` command to display the global priority of a process.	"How to Display the Global Priority of a Process" on page 81
Designate a process priority.	Start a process with a designated priority by using the `priocntl -e -c command.`	"How to Designate a Process Priority (`priocntl`)" on page 81
Change scheduling parameters of a timesharing process.	Use the `priocntl -s -m` command to change scheduling parameters in a timesharing process.	"How to Change Scheduling Parameters of a Timesharing Process (`priocntl`)" on page 82
Change the class of a process.	Use the `priocntl -s -c` command to change the class of a process.	"How to Change the Class of a Process (`priocntl`)" on page 83
Change the priority of a process.	Use the `/usr/bin/nice` command with the appropriate options to lower or raise the priority of a process.	"How to Change the Priority of a Process (`nice`)" on page 84

Managing Process Class Information

The following list identifies the process scheduling classes that can be configured on your system. Also included is the user priority range for the timesharing class.

The possible process scheduling classes are as follows:

- Fair share (`FSS`)
- Fixed (`FX`)
- System (`SYS`)
- Interactive (`IA`)
- Real-time (`RT`)
- Timesharing (`TS`)
  - The user-supplied priority ranges from -60 to +60.
  - The priority of a process is inherited from the parent process. This priority is referred to as the *user-mode priority*.
  - The system looks up the user-mode priority in the timesharing dispatch parameter table. Then, the system adds in any `nice` or `priocntl` (user-supplied) priority and ensures a 0–59 range to create a *global priority*.

Changing the Scheduling Priority of Processes (`priocntl`)

The scheduling priority of a process is the priority assigned by the process scheduler, according to scheduling policies. The `dispadmin` command lists the default scheduling policies. For more information, see the `dispadmin`(1M) man page.

You can use the `priocntl` command to assign processes to a priority class and to manage process priorities. For instructions on using the `priocntl` command to manage processes, see "How to Designate a Process Priority (`priocntl`)" on page 81.

▼ How to Display Basic Information About Process Classes (`priocntl`)

- **Display process scheduling classes and priority ranges with the `priocntl -l` command.**

```
$ priocntl -l
```

Example 4–4 Displaying Basic Information About Process Classes (`priocntl`)

The following example shows output from the `priocntl -l` command.

```
# priocntl -l
CONFIGURED CLASSES
==================

SYS (System Class)

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

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

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

▼ How to Display the Global Priority of a Process

- **Display the global priority of a process by using the `ps` command.**

  ```
  $ ps -ecl
  ```

  The global priority is listed under the `PRI` column.

Example 4–5 Displaying the Global Priority of a Process

The following example shows `ps -ecl` command output. The values in the `PRI` column show that the `pageout` process has the highest priority, while the `sh` process has the lowest priority.

```
$ ps -ecl
 F S UID PID  PPID CLS PRI  ADDR      SZ  WCHAN    TTY      TIME   COMD
19 T 0   0    0    SYS 96   f00d05a8   0           ?        0:03  sched
 8 S 0   1    0    TS  50   ff0f4678 185  ff0f4848 ?       36:51   init
19 S 0   2    0    SYS 98   ff0f4018   0  f00c645c ?        0:01 pageout
19 S 0   3    0    SYS 60   ff0f5998   0  f00d0c68 ?      241:01 fsflush
 8 S 0   269  1    TS  58   ff0f5338 303  ff49837e ?        0:07    sac
 8 S 0   204  1    TS  43   ff2f6008  50  ff2f606e console  0:02     sh
```

▼ How to Designate a Process Priority (`priocntl`)

1. **Become the root user.**

2. **Start a process with a designated priority.**

   ```
   # priocntl -e -c class -m user-limit -p pri command-name
   ```

`-e`	Executes the command.
`-c` *class*	Specifies the class within which to run the process. The valid classes are `TS` (timesharing), `RT` (real time), IA (interactive), FSS (fair share), and FX (fixed priority).

`-m` *user-limit*	When you use the `-p` option with this option, the maximum amount you can raise or lower your priority is also specified.
`-p` *pri command-name*	Enables you specify the relative priority in the RT class for a real-time thread. For a timesharing process, the `-p` option lets you specify the user-supplied priority, which ranges from -60 to +60.

3 Verify the process status.

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

Example 4–6 Designating a Process Priority (`priocntl`)

The following example shows how to start the `find` command with the highest possible user-supplied priority.

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

▼ How to Change Scheduling Parameters of a Timesharing Process (`priocntl`)

1 Become the root user.

2 Change the scheduling parameters of a running timesharing process.

```
# priocntl -s -m user-limit [-p user-priority] -i idtype idlist
```

`-s`	Lets you set the upper limit on the user priority range and change the current priority.
`-m` *user-limit*	When you use the `-p` option, specifies the maximum amount you can raise or lower the priority.
`-p` *user-priority*	Allows you to designate a priority.
`-i` *xidtype xidlist*	Uses a combination of *xidtype* and *xidlist* to identify the process or processes. The *xidtype* specifies the type of ID, such as the process ID or the user ID. Use *xidlist* to identify a list of process IDs or user IDs.

3 Verify the process status.

```
# ps -ecl | grep idlist
```

Example 4–7 Changing Scheduling Parameters of a Timesharing Process (`priocntl`)

The following example shows how to execute a command with a 500-millisecond time slice, a priority of 20 in the RT class, and a global priority of 120.

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

▼ How to Change the Class of a Process (`priocntl`)

1 (Optional) Become the root user.

2 Change the class of a process.

`# priocntl -s -c` *class* `-i` *idtype idlist*

`-s`	Lets you set the upper limit on the user priority range and change the current priority.
`-c` *class*	Specifies the class, `TS` for time-sharing or `RT` `for real-time`, to which you are changing the process.
`-i` *idtype idlist*	Uses a combination of *xidtype* and *xidlist* to identify the process or processes. The *xidtype* specifies the type of ID, such as the process ID or user ID. Use *xidlist* to identify a list of process IDs or user IDs.

Note – You must be the root user or working in a real-time shell to change a process from, or to, a real-time process. If, as superuser, you change a user process to the real-time class, the user cannot subsequently change the real-time scheduling parameters by using the `priocntl -s` command.

---

3 Verify the process status.

`# ps -ecl | grep` *idlist*

Example 4–8 Changing the Class of a Process (`priocntl`)

The following example shows how to change all the processes that belong to user `15249` to real-time processes.

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

Changing the Priority of a Timesharing Process (`nice`)

The `nice` command is only supported for backward compatibility to previous releases. The `priocntl` command provides more flexibility in managing processes.

The priority of a process is determined by the policies of its scheduling class and by its *nice number*. Each timesharing process has a global priority. The global priority is calculated by adding the user-supplied priority, which can be influenced by the `nice` or `priocntl` commands, and the system-calculated priority.

The execution priority number of a process is assigned by the operating system. The priority number is determined by several factors, including the process's scheduling class, how much CPU time it has used, and in the case of a timesharing process, its `nice` number.

Each timesharing process starts with a default `nice` number, which it inherits from its parent process. The `nice` number is shown in the `NI` column of the `ps` report.

A user can lower the priority of a process by increasing its user-supplied priority. However, only superuser can lower a `nice` number to increase the priority of a process. This restriction prevents users from increasing the priorities of their own processes, thereby monopolizing a greater share of the CPU.

The `nice` numbers range from 0 to +39, with 0 representing the highest priority. The default `nice` value for each timesharing process is 20. Two versions of the command are available: the standard version, `/usr/bin/nice`, and the C shell built-in command.

▼ How to Change the Priority of a Process (`nice`)

Using this procedure, a user can lower the priority of a process. However, superuser can raise or lower the priority of a process.

Note – This section describes the syntax of the `/usr/bin/nice` command and not the C-shell built-in `nice`command. For information about the C-shell `nice` command, see the `csh(1)` man page.

---

1 Determine whether you want to change the priority of a process, either as a user or as superuser. Then, select one of the following:

- **As a user, follow the examples in Step 2 to lower the priority of a command.**
- **As a superuser, follow the examples in Step 3 to raise or lower priorities of a command.**

2 As a user, lower the priority of a command by increasing the `nice` number.

The following `nice` command executes *command-name* with a lower priority by raising the nice number by 5 units.

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

In the preceding command, the minus sign designates that what follows is an option. This command could also be specified as follows:

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

The following `nice` command lowers the priority of *command-name* by raising the nice number by the default increment of 10 units, but not beyond the maximum value of 39.

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

3 As superuser, raise or lower the priority of a command by changing the `nice` number.

The following `nice` command raises the priority of *command-name* by lowering the nice number by 10 units, but not below the minimum value of 0.

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

In the preceding command, the first minus sign designates that what follows is an option. The second minus sign indicates a negative number.

The following `nice` command lowers the priority of *command-name* by raising the nice number by 5 units, but not beyond the maximum value of 39.

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

See Also For more information, see the `nice(1)` man page.

Troubleshooting Problems With System Processes

Here are some tips on obvious problems you might encounter:

- Look for several identical jobs that are owned by the same user. This problem might occur because of a running script that starts a lot of background jobs without waiting for any of the jobs to finish.
- Look for a process that has accumulated a large amount of CPU time. You can identify this problem by checking the `TIME` field in the `ps` output. Possibly, the process is in an endless loop.
- Look for a process that is running with a priority that is too high. Use the `ps -c` command to check the `CLS` field, which displays the scheduling class of each process. A process executing as a real-time (`RT`) process can monopolize the CPU. Or, look for a timesharing (`TS`) process

with a high `nice` number. A user with superuser privileges might have increased the priority of a process. The system administrator can lower the priority by using the `nice` command.

- Look for a runaway process. A runaway process progressively uses more and more CPU time. You can identify this problem by looking at the time when the process started (`STIME`) and by watching the cumulation of CPU time (`TIME`) for a while.

◆ ◆ ◆ CHAPTER 5

Monitoring System Performance (Tasks)

Achieving good performance from a computer or network is an important part of system administration. This chapter provides an overview of some factors that contribute to managing the performance of the computer systems in your care.

This is a list of the information that is in this chapter.

In addition, this chapter describes procedures for monitoring system performance by using the `vmstat`, `iostat`, `df`, and `sar` commands.

For information about the procedures that are associated with monitoring system performance, see the following:

What's New in Managing System Performance?

This section describes new or changed features in managing system performance in the Oracle Solaris release.

LatencyTOP Performance Tuning Utility

In this Oracle Solaris release, the LatencyTOP performance utility can be used to collect latency statistics on your Solaris system. Because the utility uses the DTrace feature, you must have DTrace privilege to run the LatencyTOP utility. Values for system-wide statistics are the

combined values of latencies that have the same cause as other processes that are running the system. You can also view data for one process or thread. There are also options available to help you focus on particular data. In addition, the LatencyTOP utility can assist you with system level tuning and locating an application bottleneck. For more information, see the Quick Start guide on the LatencyTOP project site at `http://hub.opensolaris.org/bin/view/Project+latencytop/`.

Where to Find System Performance Tasks

System Performance Task	For More Information
Manage processes	Chapter 4, "Managing System Processes (Tasks)"
Monitor system performance	Chapter 5, "Monitoring System Performance (Tasks)"
Change tunable parameters	*Oracle Solaris Tunable Parameters Reference Manual*
Manage system performance tasks	Chapter 2, "Projects and Tasks (Overview)," in *System Administration Guide: Oracle Solaris Zones, Oracle Solaris 10 Containers, and Resource Management*
Manage processes with FX and FS schedulers	Chapter 8, "Fair Share Scheduler (Overview)," in *System Administration Guide: Oracle Solaris Zones, Oracle Solaris 10 Containers, and Resource Management*

System Performance and System Resources

The performance of a computer system depends upon how the system uses and allocates its resources. Monitor your system's performance regularly so that you know how it behaves under normal conditions. You should have a good idea of what to expect, and be able to recognize a problem when it occurs.

System resources that affect performance are described in the following table.

System Resource	Description
Central processing unit (CPU)	The CPU processes instructions by fetching instructions from memory and executing them.
Input/output (I/O) devices	I/O devices transfer information into and out of the computer. Such a device could be a terminal and keyboard, a disk drive, or a printer.
Memory	Physical (or main) memory is the amount of random access memory (RAM) on the system.

Chapter 5, "Monitoring System Performance (Tasks)," describes the tools that display statistics about the system's activity and performance.

Processes and System Performance

The following table describes terms that are related to processes.

TABLE 5–1 Process Terminology

Term	Description
Process	Any system activity or job. Each time you boot a system, execute a command, or start an application, the system activates one or more processes.
Lightweight process (LWP)	A virtual CPU or execution resource. LWPs are scheduled by the kernel to use available CPU resources based on their scheduling class and priority. LWPs include a kernel thread and an LWP. A kernel thread contains information that has to be in memory all the time. An LWP contains information that is swappable.
Application thread	A series of instructions with a separate stack that can execute independently in a user's address space. Application threads can be multiplexed on top of LWPs.

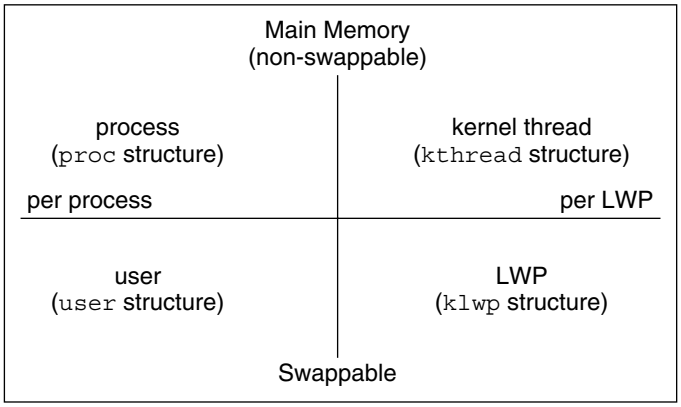
A process can consist of multiple LWPs and multiple application threads. The kernel schedules a kernel-thread structure, which is the scheduling entity in the SunOS environment. Various process structures are described in the following table.

TABLE 5–2 Process Structures

Structure	Description
`proc`	Contains information that pertains to the whole process and must be in main memory all the time
`kthread`	Contains information that pertains to one LWP and must be in main memory all the time
`user`	Contains the "per process" information that is swappable
`klwp`	Contains the "per LWP process" information that is swappable

The following figure illustrates the relationships among these process structures.

FIGURE 5–1 Relationships Among Process Structures



Most process resources are accessible to all the threads in the process. Almost all process virtual memory is shared. A change in shared data by one thread is available to the other threads in the process.

About Monitoring System Performance

While your computer is running, counters in the operating system are incremented to track various system activities.

System activities that are tracked are as follows:

- Central processing unit (CPU) utilization
- Buffer usage
- Disk and tape input/output (I/O) activity
- Terminal device activity
- System call activity
- Context switching
- File access
- Queue activity
- Kernel tables
- Interprocess communication
- Paging
- Free memory and swap space
- Kernel memory allocation (KMA)

Monitoring Tools

The Oracle Solaris software provides several tools to help you track how your system is performing. The following table describes these tools.

TABLE 5–3 Performance Monitoring Tools

Command	Description	For More Information
`cpustat` and `cputrack` commands	Monitors performance of a system or a process using CPU performance counters.	`cpustat`(1M) and `cputrack`(1)
`netstat` and `nfsstat` commands	Displays information about network performance.	`netstat`(1M) and `nfsstat`(1M)
`ps` and `prstat` commands	Displays information about active processes.	Chapter 4, "Managing System Processes (Tasks)"
`sar` and `sadc` commands	Collects and reports on system activity data.	Chapter 5, "Monitoring System Performance (Tasks)"
`swap` command	Displays information about available swap space on your system.	Chapter 21, "Configuring Additional Swap Space (Tasks)," in *System Administration Guide: Devices and File Systems*
`vmstat` and `iostat` commands	Summarizes system activity data, such as virtual memory statistics, disk usage, and CPU activity.	Chapter 5, "Monitoring System Performance (Tasks)"
`cputrack` and `cpustat` commands	Assists in accessing hardware performance counter facilities provided by microprocessors.	`cputrack`(1) and `cpustat`(1M) man pages
`kstat` and `mpstat` commands	Examines the available kernel statistics, or `kstats`, on the system and reports those statistics which match the criteria specified on the command line. The `mpstat` command reports processor statistics in tabular form.	`kstat`(1M) and `mpstat`(1M) man pages.

Displaying System Performance Information (Task Map)

Task	Description	For Instructions
Display virtual memory Statistics.	Collect virtual memory statistics by using the `vmstat` command.	"How to Display Virtual Memory Statistics (`vmstat`)" on page 93
Display system event information.	Display system event information by using the `vmstat` command with the `-s` option.	"How to Display System Event Information (`vmstat -s`)" on page 94

Task	Description	For Instructions
Display swapping statistics.	Use the vmstat command with the -S option to display swapping statistics.	"How to Display Swapping Statistics (vmstat -S)" on page 95
Display interrupts per device.	Use the vmstat command with the -i option to show the number of interrupts per device.	"How to Display Interrupts Per Device (vmstat -i)" on page 95
Display disk utilization.	Use the iostat command to report disk input and output statistics.	"How to Display Disk Utilization Information (iostat)" on page 96
Display extended disk statistics.	Use the iostat command with the -xtcoption to display extended disk statistics.	"How to Display Extended Disk Statistics (iostat -xtc)" on page 97
Display disk space information.	The df -k command displays disk space information in Kbytes.	"How to Display Disk Space Information (df -k)" on page 98

Displaying Virtual Memory Statistics (vmstat)

You can use the vmstat command to report virtual memory statistics and information about system events such as CPU load, paging, number of context switches, device interrupts, and system calls. The vmstat command can also display statistics on swapping, cache flushing, and interrupts.

The following table describes the fields in the vmstat command output.

TABLE 5–4 Output From the vmstat Command

Category	Field Name	Description
procs		Reports on the following:
	r	The number of kernel threads in the dispatch queue
	b	The number of blocked kernel threads that are waiting for resources
	w	The number of swapped out LWPs that are waiting for processing resources to finish
memory		Reports on usage of real memory and virtual memory:
	swap	Available swap space
	free	Size of the free list
page		Reports on page faults and paging activity, in units per second:

TABLE 5–4 Output From the vmstat Command *(Continued)*

Category	Field Name	Description
	re	Pages reclaimed
	mf	Minor faults and major faults
	pi	Kbytes paged in
	po	Kbytes paged out
	fr	Kbytes freed
	de	Anticipated memory that is needed by recently swapped-in processes
	sr	Pages scanned by the page daemon not currently in use. If sr does not equal zero, the page daemon has been running.
disk		Reports the number of disk operations per second, showing data on up to four disks
faults		Reports the trap/interrupt rates per second:
	in	Interrupts per second
	sy	System calls per second
	cs	CPU context switch rate
cpu		Reports on the use of CPU time:
	us	User time
	sy	System time
	id	Idle time

For a more detailed description of this command, see the vmstat(1M) man page.

▼ How to Display Virtual Memory Statistics (vmstat)

- **Collect virtual memory statistics by using the vmstat command with a time interval in seconds.**

```
$ vmstat n
```

where *n* is the interval in seconds between reports.

Example 5–1 Displaying Virtual Memory Statistics

The following example shows the vmstat display of statistics that were gathered at five-second intervals:

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

▼ How to Display System Event Information (`vmstat -s`)

- **Run the `vmstat -s` command to show how many system events have taken place since the last time the system was booted.**

```
$ vmstat -s
        0 swap ins
        0 swap outs
        0 pages swapped in
        0 pages swapped out
   522586 total address trans. faults taken
    17006 page ins
       25 page outs
    23361 pages paged in
       28 pages paged out
    45594 total reclaims
    45592 reclaims from free list
        0 micro (hat) faults
   522586 minor (as) faults
    16189 major faults
    98241 copy-on-write faults
   137280 zero fill page faults
    45052 pages examined by the clock daemon
        0 revolutions of the clock hand
       26 pages freed by the clock daemon
     2857 forks
       78 vforks
     1647 execs
 34673885 cpu context switches
 65943468 device interrupts
   711250 traps
 63957605 system calls
  3523925 total name lookups (cache hits 99%)
    92590 user   cpu
    65952 system cpu
 16085832 idle   cpu
     7450 wait   cpu
```

How to Display Swapping Statistics (vmstat -S)

- **Run `vmstat -S` to show swapping statistics.**

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

The swapping statistics fields are described in the following list. For a description of the other fields, see Table 5–4.

`si` Average number of LWPs that are swapped in per second

`so` Number of whole processes that are swapped out

Note – The `vmstat` command truncates the output of `si` and `so` fields. Use the `sar` command to display a more accurate accounting of swap statistics.

How to Display Interrupts Per Device (vmstat -i)

- **Run the `vmstat -i` command to show the number of interrupts per device.**

Example 5–2 Displaying Interrupts Per Device

The following example shows output from the `vmstat -i` command.

```
$ vmstat -i
interrupt         total     rate
--------------------------------
clock          52163269      100
esp0            2600077        4
zsc0              25341        0
zsc1              48917        0
cgsixc0             459        0
lec0             400882        0
fdc0                 14        0
bppc0                 0        0
audiocs0              0        0
--------------------------------
Total          55238959      105
```

Displaying Disk Utilization Information (iostat)

Use the iostat command to report statistics about disk input and output, and to produce measures of throughput, utilization, queue lengths, transaction rates, and service time. For a detailed description of this command, refer to the iostat(1M) man page.

▼ How to Display Disk Utilization Information (iostat)

- **You can display disk utilization information by using the iostat command with a time interval in seconds.**

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

The first line of output shows the statistics since the last time the system was booted. Each subsequent line shows the interval statistics. The default is to show statistics for the terminal (tty), disks (fd and sd), and CPU (cpu).

Example 5–3 Displaying Disk Utilization Information

The following example shows disk statistics that were gathered every five seconds.

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

The following table describes the fields in the output of the iostat *n* command.

Device Type	Field Name	Description
Terminal	Device Type	

Device Type	Field Name	Description
	tin	Number of characters in the terminal input queue
	tout	Number of characters in the terminal output queue
Disk	Device Type	
	bps	Blocks per second
	tps	Transactions per second
	serv	Average service time, in milliseconds
CPU	Device Type	
	us	In user mode
	sy	In system mode
	wt	Waiting for I/O
	id	Idle

How to Display Extended Disk Statistics (`iostat -xtc`)

- **Run the `iostat -xtc` command to display extended disk statistics.**

```
$ iostat -xtc
                 extended device statistics                      tty         cpu
device       r/s    w/s   kr/s   kw/s wait actv  svc_t  %w  %b  tin tout  us sy wt id
fd0          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0    0    0   0  0  0 100
sd0          0.0    0.0    0.4    0.4  0.0  0.0   49.5   0   0
sd6          0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs1         0.0    0.0    0.0    0.0  0.0  0.0    0.0   0   0
nfs49        0.0    0.0    0.0    0.0  0.0  0.0   15.1   0   0
nfs53        0.0    0.0    0.4    0.0  0.0  0.0   24.5   0   0
nfs54        0.0    0.0    0.0    0.0  0.0  0.0    6.3   0   0
nfs55        0.0    0.0    0.0    0.0  0.0  0.0    4.9   0   0
```

The `iostat -xtc` command displays a line of output for each disk. The output fields are described in the following list.

`r/s` Reads per second

`w/s` Writes per second

`kr/s` Kbytes read per second

`kw/s` Kbytes written per second

`wait` Average number of transactions that are waiting for service (queue length)

`actv`	Average number of transactions that are actively being serviced
`svc_t`	Average service time, in milliseconds
`%w`	Percentage of time that the queue is not empty
`%b`	Percentage of time that the disk is busy

Displaying Disk Space Statistics (df)

Use the `df` command to show the amount of free disk space on each mounted disk. The *usable* disk space that is reported by `df` reflects only 90 percent of full capacity, as the reporting statistics allows for 10 percent above the total available space. This *head room* normally stays empty for better performance.

The percentage of disk space actually reported by the `df` command is used space divided by usable space.

If the file system exceeds 90 percent capacity, you could transfer files to a disk that is not as full by using the `cp` command. Alternately, you could transfer files to a tape by using the `tar` or `cpio` commands. Or, you could remove the files.

For a detailed description of this command, see the `df(1M)` man page.

▼ How to Display Disk Space Information (df -k)

- **Use the `df -k` command to display disk space information in Kbytes.**

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

Example 5–4 Displaying File System Information

The following example shows the output from the `df -k` command.

```
$ df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t0d0s0     254966  204319   25151    90%    /
/devices                   0       0       0     0%    /devices
ctfs                       0       0       0     0%    /system/contract
proc                       0       0       0     0%    /proc
mnttab                     0       0       0     0%    /etc/mnttab
swap                  496808     376  496432     1%    /etc/svc/volatile
objfs                      0       0       0     0%    /system/object
/dev/dsk/c0t0d0s6    3325302 3073415  218634    94%    /usr
fd                         0       0       0     0%    /dev/fd
```

```
swap                   496472      40  496432     1%    /var/run
swap                   496472      40  496432     1%    /tmp
/dev/dsk/c0t0d0s5       13702    1745   10587    15%    /opt
/dev/dsk/c0t0d0s7        9450    1045    7460    13%    /export/home
```

The following table describes the output of the `df -k` command.

Field Name	Description
`kbytes`	Total size of usable space in the file system
`used`	Amount of space used
`avail`	Amount of space available for use
`capacity`	Amount of space used, as a percentage of the total capacity
`mounted on`	Mount point

Monitoring System Activities (Task Map)

Task	Description	For Instructions
Check file access.	Display file access operation status by using the `sar` command with the `-a` option.	"How to Check File Access (`sar -a`)" on page 101
Check buffer activity.	Display buffer activity statistics by using the `sar` command with the `-b` option.	"How to Check Buffer Activity (`sar -b`)" on page 102
Check system call statistics.	Display system call statistics by using the `sar` command with the `-c` option.	"How to Check System Call Statistics (`sar -c`)" on page 103
Check disk activity.	Check disk activity by using the `sar` command with the `-d` option.	"How to Check Disk Activity (`sar -d`)" on page 104
Check page-out and memory.	Use the `sar` command with the `-g` option to display page-out memory freeing activities.	"How to Check Page-Out and Memory (`sar -g`)" on page 106
Check kernel memory allocation.	The kernel memory allocation (KMA) allows a kernel subsystem to allocate and free memory, as needed. Use the `sar` command with the `-k` option to check KMA.	"How to Check Kernel Memory Allocation (`sar -k`)" on page 108
Check interprocess communication.	Use the `sar` command with the `-m` option to report interprocess communication activities.	"How to Check Interprocess Communication (`sar -m`)" on page 109
Check page-in activity.	Use the `sar` command with the `-p` option to report page-in activity.	"How to Check Page-In Activity (`sar -p`)" on page 110

Task	Description	For Instructions
Check queue activity.	Use the sar command with the -q option to check the following: ■ Average queue length while queue is occupied ■ Percentage of time that the queue is occupied	"How to Check Queue Activity (sar -q)" on page 111
Check unused memory.	Use the sar command with the -r option to report the number of memory pages and swap file disk blocks that are currently used.	"How to Check Unused Memory (sar -r)" on page 112
Check CPU utilization.	Use the sar command with the -u option to display CPU utilization statistics.	"How to Check CPU Utilization (sar -u)" on page 113
Check system table status.	Use the sar command with the -v option to report status on the following system tables: ■ Process ■ Inode ■ File ■ Shared memory record	"How to Check System Table Status (sar -v)" on page 114
Check swapping activity.	Use the sar command with the -w option to check swapping activity.	"How to Check Swapping Activity (sar -w)" on page 115
Check terminal activity.	Use the sar command with the -y option to monitor terminal device activity.	"How to Check Terminal Activity (sar -y)" on page 116
Check overall system performance.	The sar -A command displays statistics from all options to provide overall system performance information.	"How to Check Overall System Performance (sar -A)" on page 117
Set up automatic data collection.	To set up your system to collect data automatically and to run the sar commands, do the following: ■ Run the svcadm enable system/sar:default command ■ Edit the /var/spool/cron/crontabs/sys file	"How to Set Up Automatic Data Collection" on page 120

Monitoring System Activities (sar)

Use the sar command to perform the following tasks:

- Organize and view data about system activity.
- Access system activity data on a special request basis.
- Generate automatic reports to measure and monitor system performance, as well as special request reports to pinpoint specific performance problems. For information about how to set up the sar command to run on your system, as well as a description of these tools, see "Collecting System Activity Data Automatically (sar)" on page 117.

For a detailed description of this command, see the sar(1) man page.

How to Check File Access (sar -a)

- **Display file access operation statistics with the sar -a command.**

```
$ sar -a

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

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

Average        0       4       0
```

The following list describes the field names and description of operating system routines that are reported by the sar -a command.

`iget/s` The number of requests made for inodes that were not in the directory name look-up cache (DNLC).

`namei/s` The number of file system path searches per second. If `namei` does not find a directory name in the DNLC, it calls `iget` to get the inode for either a file or directory. Hence, most `igets` are the result of DNLC misses.

`dirbk/s` The number of directory block reads issued per second.

The larger the reported values for these operating system routines, the more time the kernel is spending to access user files. The amount of time reflects how heavily programs and applications are using the file systems. The `-a` option is helpful for viewing how disk-dependent an application is.

▼ How to Check Buffer Activity (sar -b)

- **Display buffer activity statistics with the sar -b command.**

  The buffer is used to cache metadata. Metadata includes inodes, cylinder group blocks, and indirect blocks.

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

Example 5–5 Checking Buffer Activity (sar -b)

The following example of sar -b command output shows that the %rcache and %wcache buffers are not causing any slowdowns. All the data is within acceptable limits.

```
$ sar -b

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00       0       0     100       0       0      94       0       0
02:00:01       0       0     100       0       0      94       0       0
03:00:00       0       0     100       0       0      92       0       0
04:00:00       0       1     100       0       1      94       0       0
05:00:00       0       0     100       0       0      93       0       0
06:00:00       0       0     100       0       0      93       0       0
07:00:00       0       0     100       0       0      93       0       0
08:00:00       0       0     100       0       0      93       0       0
08:20:00       0       1     100       0       1      94       0       0
08:40:01       0       1     100       0       1      93       0       0
09:00:00       0       1     100       0       1      93       0       0
09:20:00       0       1     100       0       1      93       0       0
09:40:00       0       2     100       0       1      89       0       0
10:00:00       0       9     100       0       5      92       0       0
10:20:00       0       0     100       0       0      68       0       0
10:40:00       0       1      98       0       1      70       0       0
11:00:00       0       1     100       0       1      75       0       0

Average        0       1     100       0       1      91       0       0
```

The following table describes the buffer activities that are displayed by the -b option.

Field Name	Description
bread/s	Average number of reads per second that are submitted to the buffer cache from the disk
lread/s	Average number of logical reads per second from the buffer cache
%rcache	Fraction of logical reads that are found in the buffer cache (100 % minus the ratio of bread/s to lread/s)

Field Name	Description
bwrit/s	Average number of physical blocks (512 bytes) that are written from the buffer cache to disk, per second
lwrit/s	Average number of logical writes to the buffer cache, per second
%wcache	Fraction of logical writes that are found in the buffer cache (100 % minus the ratio of bwrit/s to lwrit/s)
pread/s	Average number of physical reads, per second, that use character device interfaces
pwrit/s	Average number of physical write requests, per second, that use character device interfaces

The most important entries are the cache hit ratios %rcache and %wcache. These entries measure the effectiveness of system buffering. If %rcache falls below 90 percent, or if %wcache falls below 65 percent, it might be possible to improve performance by increasing the buffer space.

▼ How to Check System Call Statistics (sar -c)

- **Display system call statistics by using the sar -c command.**

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

Example 5–6 Checking System Call Statistics (sar -c)

The following example shows output from the sar -c command.

```
$ sar -c

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 scall/s sread/s swrit/s  fork/s  exec/s rchar/s wchar/s
01:00:00      89      14       9    0.01    0.00    2906    2394
02:00:01      89      14       9    0.01    0.00    2905    2393
03:00:00      89      14       9    0.01    0.00    2908    2393
04:00:00      90      14       9    0.01    0.00    2912    2393
05:00:00      89      14       9    0.01    0.00    2905    2393
06:00:00      89      14       9    0.01    0.00    2905    2393
07:00:00      89      14       9    0.01    0.00    2905    2393
08:00:00      89      14       9    0.01    0.00    2906    2393
08:20:00      90      14       9    0.01    0.01    2914    2395
08:40:01      90      14       9    0.01    0.00    2914    2396
09:00:00      90      14       9    0.01    0.01    2915    2396
09:20:00      90      14       9    0.01    0.01    2915    2396
09:40:00     880     207     156    0.08    0.08   26671    9290
```

```
10:00:00    2020     530     322    0.14    0.13   57675   36393
10:20:00     853     129      75    0.02    0.01   10500    8594
10:40:00    2061     524     450    0.08    0.08  579217  567072
11:00:00    1658     404     350    0.07    0.06 1152916 1144203

Average      302      66      49    0.02    0.01   57842   55544
```

The following table describes the system call categories that are reported by the `-c` option. Typically, reads and writes account for about half of the total system calls. However, the percentage varies greatly with the activities that are being performed by the system.

Field Name	Description
`scall/s`	The number of all types of system calls per second, which is generally about 30 per second on a system with 4 to 6 users.
`sread/s`	The number of `read` system calls per second.
`swrit/s`	The number of `write` system calls per second.
`fork/s`	The number of `fork` system calls per second, which is about 0.5 per second on a system with 4 to 6 users. This number increases if shell scripts are running.
`exec/s`	The number of `exec` system calls per second. If `exec/s` divided by `fork/s` is greater than 3, look for inefficient `PATH` variables.
`rchar/s`	The number of characters (bytes) transferred by `read` system calls per second.
`wchar/s`	The number of characters (bytes) transferred by `write` system calls per second.

▼ How to Check Disk Activity (`sar -d`)

- **Display disk activity statistics with the `sar -d` command.**

```
$ sar -d

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

Example 5–7 Checking Disk Activity

This abbreviated example illustrates the output from the `sar -d` command.

```
$ sar -d

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

12:36:32   device        %busy   avque   r+w/s  blks/s  avwait  avserv
```

```
12:40:01   dad1            15     0.7      26     399    18.1    10.0
           dad1,a          15     0.7      26     398    18.1    10.0
           dad1,b           0     0.0       0       1     1.0     3.0
           dad1,c           0     0.0       0       0     0.0     0.0
           dad1,h           0     0.0       0       0     0.0     6.0
           fd0              0     0.0       0       0     0.0     0.0
           nfs1             0     0.0       0       0     0.0     0.0
           nfs2             1     0.0       1      12     0.0    13.2
           nfs3             0     0.0       0       2     0.0     1.9
           nfs4             0     0.0       0       0     0.0     7.0
           nfs5             0     0.0       0       0     0.0    57.1
           nfs6             1     0.0       6     125     4.3     3.2
           nfs7             0     0.0       0       0     0.0     6.0
           sd1              0     0.0       0       0     0.0     5.4
           ohci0,bu         0     0.0       0       0     0.0     0.0
           ohci0,ct         0     0.0       0       0     0.0     0.0
           ohci0,in         0     0.0       7       0     0.0     0.0
           ohci0,is         0     0.0       0       0     0.0     0.0
           ohci0,to         0     0.0       7       0     0.0     0.0
```

The following table describes the disk device activities that are reported by the `-d` option.

Field Name	Description
`device`	Name of the disk device that is being monitored.
`%busy`	Portion of time the device was busy servicing a transfer request.
`avque`	Average number of requests during the time the device was busy servicing a transfer request.
`r+w/s`	Number of read-and-write transfers to the device, per second.
`blks/s`	Number of 512-byte blocks that are transferred to the device, per second.
`avwait`	Average time, in milliseconds, that transfer requests wait idly in the queue. This time is measured only when the queue is occupied.
`avserv`	Average time, in milliseconds, for a transfer request to be completed by the device. For disks, this value includes seek times, rotational latency times, and data transfer times.

Note that queue lengths and wait times are measured when something is in the queue. If `%busy` is small, large queues and service times probably represent the periodic efforts by the system to ensure that altered blocks are promptly written to the disk.

▼ How to Check Page-Out and Memory (sar -g)

- **Use the sar -g command to display page-out and memory freeing activities in averages.**

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

The output displayed by the sar -g command is a good indicator of whether more memory might be needed. Use the ps -elf command to show the number of cycles that are used by the page daemon. A high number of cycles, combined with high values for the pgfree/s and pgscan/s fields, indicates a memory shortage.

The sar -g command also shows whether inodes are being recycled too quickly and causing a loss of reusable pages.

Example 5–8 Checking Page-Out and Memory (sar -g)

The following example shows output from the sar -g command.

```
$ sar -g

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:00  pgout/s ppgout/s pgfree/s pgscan/s %ufs_ipf
01:00:00     0.00     0.00     0.00     0.00     0.00
02:00:00     0.01     0.01     0.01     0.00     0.00
03:00:00     0.00     0.00     0.00     0.00     0.00
04:00:00     0.00     0.00     0.00     0.00     0.00
05:00:00     0.00     0.00     0.00     0.00     0.00
06:00:00     0.00     0.00     0.00     0.00     0.00
07:00:00     0.00     0.00     0.00     0.00     0.00
08:00:00     0.00     0.00     0.00     0.00     0.00
08:20:01     0.00     0.00     0.00     0.00     0.00
08:40:00     0.00     0.00     0.00     0.00     0.00
09:00:00     0.00     0.00     0.00     0.00     0.00
09:20:01     0.05     0.52     1.62    10.16     0.00
09:40:01     0.03     0.44     1.47     4.77     0.00
10:00:02     0.13     2.00     4.38    12.28     0.00
10:20:03     0.37     4.68    12.26    33.80     0.00

Average      0.02     0.25     0.64     1.97     0.00
```

The following table describes the output from the -g option.

Field Name	Description
pgout/s	The number of page-out requests per second.
ppgout/s	The actual number of pages that are paged-out, per second. A single page-out request might involve paging-out multiple pages.
pgfree/s	The number of pages, per second, that are placed on the free list.

Field Name	Description
pgscan/s	The number of pages, per second, that are scanned by the page daemon. If this value is high, the page daemon is spending a lot of time checking for free memory. This situation implies that more memory might be needed.
%ufs_ipf	The percentage of ufs inodes taken off the free list by iget that had reusable pages that are associated with them. These pages are flushed and cannot be reclaimed by processes. Thus, this field represents the percentage of igets with page flushes. A high value indicates that the free list of inodes is page-bound, and that the number of ufs inodes might need to be increased.

Checking Kernel Memory Allocation

The KMA allows a kernel subsystem to allocate and free memory, as needed.

Rather than statically allocating the maximum amount of memory it is expected to require under peak load, the KMA divides requests for memory into three categories:

- Small (less than 256 bytes)
- Large (512 bytes to 4 Kbytes)
- Oversized (greater than 4 Kbytes)

The KMA keeps two pools of memory to satisfy small requests and large requests. The oversized requests are satisfied by allocating memory from the system page allocator.

If you are checking a system that is being used to write drivers or STREAMS that use KMA resources, then the sar -k command will likely prove useful. Otherwise, you will probably not need the information it provides. Any driver or module that uses KMA resources, but does not specifically return the resources before it exits, can create a memory leak. A memory leak causes the amount of memory that is allocated by KMA to increase over time. Thus, if the alloc fields of the sar -k command increase steadily over time, there might be a memory leak. Another indication of a memory leak is failed requests. If this problem occurs, a memory leak has probably caused KMA to be unable to reserve and allocate memory.

If it appears that a memory leak has occurred, you should check any drivers or STREAMS that might have requested memory from KMA and not returned it.

▼ How to Check Kernel Memory Allocation (sar -k)

- **Use the sar -k command to report on the following activities of the Kernel Memory Allocator (KMA).**

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

Example 5–9 Checking Kernel Memory Allocation (sar -k)

The following is an abbreviated example of sar -k output.

```
$ sar -k

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

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

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

The following table describes the output from the -k option.

Field Name	Description
sml_mem	The amount of memory, in bytes, that the KMA has available in the small memory request pool. In this pool, here a small request is less than 256 bytes.
alloc	The amount of memory, in bytes, that the KMA has allocated from its small memory request pool to small memory requests.
fail	The number of requests for small amounts of memory that failed.

Field Name	Description
lg_mem	The amount of memory, in bytes, that the KMA has available in the large memory request pool. In this pool, a large request is from 512 bytes to 4 Kbytes.
alloc	The amount of memory, in bytes, that the KMA has allocated from its large memory request pool to large memory requests.
fail	The number of failed requests for large amounts of memory.
ovsz_alloc	The amount of memory that is allocated for oversized requests, which are requests that are greater than 4 Kbytes. These requests are satisfied by the page allocator. Thus, there is no pool.
fail	The number of failed requests for oversized amounts of memory.

▼ How to Check Interprocess Communication (sar -m)

- **Use the sar -m command to report interprocess communication activities.**

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

These figures are usually zero (0.00), unless you are running applications that use messages or semaphores.

The following list describes the output from the -m option.

msg/s The number of message operations (sends and receives) per second

sema/s The number of semaphore operations per second

Example 5–10 Checking Interprocess Communication (sar -m)

The following abbreviated example shows output from the sar -m command.

```
$ sar -m

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:00   msg/s  sema/s
01:00:00    0.00    0.00
02:00:02    0.00    0.00
03:00:00    0.00    0.00
04:00:00    0.00    0.00
05:00:01    0.00    0.00
06:00:00    0.00    0.00

Average     0.00    0.00
```

▼ How to Check Page-In Activity (sar -p)

- **Use the sar -p command to report page-in activity, which includes protection and translation faults.**

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

Example 5–11 Checking Page-In Activity (sar -p)

The following example shows output from the sar -p command.

```
$ sar -p

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

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

Average     0.42    0.22    0.31    1.45    4.00    0.00
```

The following table describes the reported statistics from the -p option.

Field Name	Description
atch/s	The number of page faults, per second, that are satisfied by reclaiming a page currently in memory (attaches per second). Instances include reclaiming an invalid page from the free list and sharing a page of text that is currently being used by another process. An example is two or more processes that are accessing the same program text.
pgin/s	The number of times, per second, that file systems receive page-in requests.

Field Name	Description
ppgin/s	The number of pages paged in, per second. A single page-in request, such as a soft-lock request (see slock/s) or a large block size, might involve paging-in multiple pages.
pflt/s	The number of page faults from protection errors. Instances of protection faults indicate illegal access to a page and "copy-on-writes." Generally, this number consists primarily of "copy-on-writes."
vflt/s	The number of address translation page faults, per second. These faults are known as validity faults. Validity faults occur when a valid process table entry does not exist for a given virtual address.
slock/s	The number of faults, per second, caused by software lock requests that require physical I/O. An example of the occurrence of a soft-lock request is the transfer of data from a disk to memory. The system locks the page that is to receive the data so that the page cannot be claimed and used by another process.

▼ How to Check Queue Activity (sar -q)

- **Use the sar -q command to report the following information:**
  - The Average queue length while the queue is occupied.
  - The percentage of time that the queue is occupied.

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

The following list describes the output from the -q option.

runq-sz — The number of kernel threads in memory that are waiting for a CPU to run. Typically, this value should be less than 2. Consistently higher values mean that the system might be CPU-bound.

%runocc — The percentage of time that the dispatch queues are occupied.

swpq-sz — Swap queue of processes for the sar command.

%swpocc — Swap queue of processes for the sar command.

Example 5–12 Checking Queue Activity

The following example shows output from the sar -q command. If the %runocc value is high (greater than 90 percent) and the runq-sz value is greater than 2, the CPU is heavily loaded and response is degraded. In this case, additional CPU capacity might be required to obtain acceptable system response.

```
# sar -q
SunOS system2 5.10 Generic_142909-13 sun4u    06/28/2010

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

Average      1.0       7     0.0       0
```

▼ How to Check Unused Memory (sar -r)

- **Use the sar -r command to report the number of memory pages and swap-file disk blocks that are currently unused.**

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

The following list describes the output from the -r option:

freemem	The average number of memory pages that are available to user processes over the intervals sampled by the command. Page size is machine-dependent.
freeswap	The number of 512-byte disk blocks that are available for page swapping.

Example 5–13 Checking Unused Memory (sar -r)

The following example shows output from the sar -r command.

```
$ sar -r

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

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

```
10:40:00   33943  1598425
11:00:00   30500  1561959

Average    43312  1699242
```

▼ How to Check CPU Utilization (sar -u)

- **Use the sar -u command to display CPU utilization statistics.**

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

The sar command without any options is equivalent to the sar -u command. At any given moment, the processor is either busy or idle. When busy, the processor is in either user mode or system mode. When idle, the processor is either waiting for I/O completion or "sitting still" with no work to do.

The following list describes output from the -u option:

`%usr` Lists the percentage of time that the processor is in user mode.

`%sys` Lists the percentage of time that the processor is in system mode.

`%wio` Lists the percentage of time that the processor is idle and waiting for I/O completion.

`%idle` Lists the percentage of time that the processor is idle and not waiting for I/O.

A high `%wio` value generally means that a disk slowdown has occurred.

Example 5–14 Checking CPU Utilization (sar -u)

The following example shows output from the sar -u command.

```
$ sar -u

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04    %usr    %sys    %wio   %idle
01:00:00       0       0       0     100
02:00:01       0       0       0     100
03:00:00       0       0       0     100
04:00:00       0       0       0     100
05:00:00       0       0       0     100
06:00:00       0       0       0     100
07:00:00       0       0       0     100
08:00:00       0       0       0     100
08:20:00       0       0       0      99
08:40:01       0       0       0      99
09:00:00       0       0       0      99
09:20:00       0       0       0      99
09:40:00       4       1       0      95
```

```
10:00:00       4       2       0      94
10:20:00       1       1       0      98
10:40:00      18       3       0      79
11:00:00      25       3       0      72

Average        2       0       0      98
```

▼ How to Check System Table Status (sar -v)

- **Use the sar -v command to report the status of the process table, inode table, file table, and shared memory record table.**

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

Example 5–15 Checking System Table Status (sar -v)

The following abbreviated example shows output from the sar -v command. This example shows that all tables are large enough to have no overflows. These tables are all dynamically allocated based on the amount of physical memory.

```
$ sar -v

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04  proc-sz    ov  inod-sz    ov  file-sz    ov   lock-sz
01:00:00   69/8010    0 3476/34703    0    0/0       0    0/0
02:00:01   69/8010    0 3476/34703    0    0/0       0    0/0
03:00:00   69/8010    0 3476/34703    0    0/0       0    0/0
04:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
05:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
06:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
07:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:20:00   69/8010    0 3494/34703    0    0/0       0    0/0
08:40:01   69/8010    0 3494/34703    0    0/0       0    0/0
09:00:00   69/8010    0 3494/34703    0    0/0       0    0/0
09:20:00   69/8010    0 3494/34703    0    0/0       0    0/0
09:40:00   74/8010    0 3494/34703    0    0/0       0    0/0
10:00:00   75/8010    0 4918/34703    0    0/0       0    0/0
10:20:00   72/8010    0 4918/34703    0    0/0       0    0/0
10:40:00   71/8010    0 5018/34703    0    0/0       0    0/0
11:00:00   77/8010    0 5018/34703    0    0/0       0    0/0
```

Output from the -v option is described in the following table.

Field Name	Description
proc-sz	The number of process entries (proc structures) that are currently being used, or allocated, in the kernel.
inod-sz	The total number of inodes in memory compared to the maximum number of inodes that are allocated in the kernel. This number is not a strict high watermark. The number can overflow.
file-sz	The size of the open system file table. The sz is given as 0, because space is allocated dynamically for the file table.
ov	The overflows that occur between sampling points for each table.
lock-sz	The number of shared memory record table entries that are currently being used, or allocated, in the kernel. The sz is given as 0 because space is allocated dynamically for the shared memory record table.

▼ How to Check Swapping Activity (sar -w)

- **Use the sar -w command to report swapping and switching activity.**

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

The following list describes target values and observations related to the sar -w command output.

swpin/s — The number of LWP transfers into memory per second.

bswin/s — The number of blocks transferred for swap-ins per second. /* (float)PGTOBLK(xx->cvmi.pgswapin) / sec_diff */.

swpot/s — The average number of processes that are swapped out of memory per second. If the number is greater than 1, you might need to increase memory.

bswot/s — The number of blocks that are transferred for swap-outs per second.

pswch/s — The number of kernel thread switches, per second.

Note – All process swap-ins include process initialization.

Example 5–16 Checking Swap Activity (sar -w)

The following example shows output from the sar -w command.

```
$ sar -w

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 swpin/s bswin/s swpot/s bswot/s pswch/s
01:00:00    0.00     0.0    0.00     0.0     132
02:00:01    0.00     0.0    0.00     0.0     133
03:00:00    0.00     0.0    0.00     0.0     133
04:00:00    0.00     0.0    0.00     0.0     134
05:00:00    0.00     0.0    0.00     0.0     133
06:00:00    0.00     0.0    0.00     0.0     133
07:00:00    0.00     0.0    0.00     0.0     132
08:00:00    0.00     0.0    0.00     0.0     131
08:20:00    0.00     0.0    0.00     0.0     133
08:40:01    0.00     0.0    0.00     0.0     132
09:00:00    0.00     0.0    0.00     0.0     132
09:20:00    0.00     0.0    0.00     0.0     132
09:40:00    0.00     0.0    0.00     0.0     335
10:00:00    0.00     0.0    0.00     0.0     601
10:20:00    0.00     0.0    0.00     0.0     353
10:40:00    0.00     0.0    0.00     0.0     747
11:00:00    0.00     0.0    0.00     0.0     804

Average     0.00     0.0    0.00     0.0     198
```

▼ How to Check Terminal Activity (`sar -y`)

- **Use the `sar -y` command to monitor terminal device activities.**

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

If you have a lot of terminal I/O, you can use this report to determine if any bad lines exist. The activities recorded are defined in the following list.

`rawch/s` Input characters (raw queue) per second.

`canch/s` Input characters that are processed by canon (canonical queue) per second.

`outch/s` Output characters (output queue) per second.

`rcvin/s` Receiver hardware interrupts per second.

`xmtin/s` Transmitter hardware interrupts per second.

`mdmin/s` Modem interrupts per second.

The number of modem interrupts per second (`mdmin/s`) should be close to zero. The receive and transmit interrupts per second (`xmtin/s` and `rcvin/s`) should be less than or equal to the number of incoming or outgoing characters, respectively. If not, check for bad lines.

Example 5–17 Checking Terminal Activity (`sar -y`)

The following example shows output from the `sar -y` command.

```
$ sar -y

SunOS balmyday 5.10 s10_51 sun4u    03/18/2004

00:00:04 rawch/s canch/s outch/s rcvin/s xmtin/s mdmin/s
01:00:00       0       0       0       0       0       0
02:00:01       0       0       0       0       0       0
03:00:00       0       0       0       0       0       0
04:00:00       0       0       0       0       0       0
05:00:00       0       0       0       0       0       0
06:00:00       0       0       0       0       0       0
07:00:00       0       0       0       0       0       0
08:00:00       0       0       0       0       0       0
08:20:00       0       0       0       0       0       0
08:40:01       0       0       0       0       0       0
09:00:00       0       0       0       0       0       0
09:20:00       0       0       0       0       0       0
09:40:00       0       0       1       0       0       0
10:00:00       0       0      37       0       0       0
10:20:00       0       0       0       0       0       0
10:40:00       0       0       3       0       0       0
11:00:00       0       0       3       0       0       0

Average        0       0       1       0       0       0
```

▼ How to Check Overall System Performance (`sar -A`)

- **Use the `sar -A` command to display statistics from all options to provide a view of overall system performance.**

  This command provides a more global perspective. If data from more than a single time segment is shown, the report includes averages.

Collecting System Activity Data Automatically (`sar`)

Three commands are involved in the automatic collection of system activity data: `sadc`, `sa1`, and `sa2`.

The `sadc` data collection utility periodically collects data on system activity and saves the data in a file in binary format, one file for each 24-hour period. You can set up the `sadc` command to run periodically (usually once each hour), and whenever the system boots to multiuser mode. The data files are placed in the `/var/adm/sa` directory. Each file is named `sa`*dd*, where *dd* is the current date. The format of the command is as follows:

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

The command samples *n* times with an interval of *t* seconds, which should be greater than five seconds between samples. This command then writes to the binary *ofile* file, or to standard output.

Running the sadc Command When Booting

The sadc command should be run at system boot time to record the statistics from when the counters are reset to zero. To make sure that the sadc command is run at boot time, the svcadm enable system/sar:default command writes a record to the daily data file.

The command entry has the following format:

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

Running the sadc Command Periodically With the sa1 Script

To generate periodic records, you need to run the sadc command regularly. The simplest way to do so is to uncomment the following lines in the /var/spool/cron/crontabs/sys file:

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

The sys crontab entries do the following:

- The first two crontab entries cause a record to be written to the /var/adm/sa/sa*dd* file every 20 minutes from 8 a.m. to 5 p.m., Monday through Friday, and every hour on the hour otherwise.
- The third entry writes a record to the /var/adm/sa/sar*dd* file hourly, Monday through Friday, and includes all sar options.

You can change these defaults to meet your needs.

Producing Reports With the sa2 Shell Script

Another shell script, sa2, produces reports rather than binary data files. The sa2 command invokes the sar command and writes the ASCII output to a report file.

Setting Up Automatic Data Collection (sar)

The sar command can be used either to gather system activity data itself or to report what has been collected in the daily activity files that are created by the sadc command.

The sar command has the following formats:

```
sar [-aAbcdgkmpqruvwy] [-o file] t [n]

sar [-aAbcdgkmpqruvwy] [-s time] [-e time] [-i sec] [-f file]
```

The following sar command samples cumulative activity counters in the operating system every *t* seconds, *n* times. The *t* should be five seconds or greater. Otherwise, the command itself might affect the sample. You must specify a time interval in which to take the samples. Otherwise, the command operates according to the second format. The default value of *n* is 1. The following example takes two samples separated by 10 seconds. If the -o option were specified, samples are saved in binary format.

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

Other important information about the sar command includes the following:

- With no sampling interval or number of samples specified, the sar command extracts data from a previously recorded file. This file is either the file specified by the -f option or, by default, the standard daily activity file, /var/adm/sa/sa*dd*, for the most recent day.
- The -s and -e options define the starting time and the ending time for the report. Starting and ending times are of the form *hh*[:*mm*[:*ss*]], where *hh*, *mm*, and *ss* represent hours, minutes, and seconds.
- The -i option specifies, in seconds, the intervals between record selection. If the -i option is not included, all intervals that are found in the daily activity file are reported.

The following table lists the sar options and their actions.

TABLE 5–5 Options for the sar Command

Option	Actions
-a	Checks file access operations
-b	Checks buffer activity
-c	Checks system calls
-d	Checks activity for each block device
-g	Checks page-out and memory freeing
-k	Checks kernel memory allocation

TABLE 5–5 Options for the sar Command *(Continued)*

Option	Actions
-m	Checks interprocess communication
-nv	Checks system table status
-p	Checks swap and dispatch activity
-q	Checks queue activity
-r	Checks unused memory
-u	Checks CPU utilization
-w	Checks swapping and switching volume
-y	Checks terminal activity
-A	Reports overall system performance, which is the same as entering all options.

Using no option is equivalent to calling the sar command with the -u option.

▼ How to Set Up Automatic Data Collection

1. **Become the root user.**

2. **Run the `svcadm enable system/sar:default` command.**

   This version of the sadc command writes a special record that marks the time when the counters are reset to zero (boot time).

3. **Edit the `/var/spool/cron/crontabs/sys` `crontab` file.**

   Note – Do not edit a crontab file directly. Instead, use the crontab -e command to make changes to an existing crontab file.

   ```
   # crontab -e sys
   ```

4. **Uncomment the following lines:**

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

   For more information, see the crontab(1) man page.

◆ ◆ ◆ CHAPTER 6

Troubleshooting Software Problems (Tasks)

This chapter provides a general overview of troubleshooting software problems, including information abouttroubleshooting system crashes, managing crash dump information, and viewing and managing system messages.

This is a list of the information that is in this chapter.

Troubleshooting a System Crash

If a system running Oracle Solaris crashes, provide your service provider with as much information as possible, including crash dump files.

What to Do If the System Crashes

The most important information to remember is as follows:

1. Write down the system console messages.

   If a system crashes, making it run again might seem like your most pressing concern. However, before you reboot the system, examine the console screen for messages. These messages can provide some insight about what caused the crash. Even if the system reboots automatically and the console messages have disappeared from the screen, you might be able to check these messages by viewing the system error log, the `/var/adm/messages` file. For more information about viewing system error log files, see "How to View System Messages" on page 124.

If you have frequent crashes and can't determine their cause, gather all the information you can from the system console or the `/var/adm/messages` files, and have it ready for a customer service representative to examine. For a complete list of troubleshooting information to gather for your service provider, see "Troubleshooting a System Crash" on page 121.

If the system fails to reboot successfully after a system crash, see Chapter 9, "Troubleshooting Miscellaneous System Problems (Tasks)."

2. Synchronize the disks and reboot.

```
ok sync
```

If the system fails to reboot successfully after a system crash, see Chapter 9, "Troubleshooting Miscellaneous System Problems (Tasks)."

Check to see if a system crash dump was generated after the system crash. System crash dumps are saved by default. For information about crash dumps, see Chapter 8, "Managing System Crash Information (Tasks)."

Gathering Troubleshooting Data

Answer the following questions to help isolate the system problem. Use "Troubleshooting a System Crash Checklist" on page 123 for gathering troubleshooting data for a crashed system.

TABLE 6–1 Identifying System Crash Data

Question	Description
Can you reproduce the problem?	This is important because a reproducible test case is often essential for debugging really hard problems. By reproducing the problem, the service provider can build kernels with special instrumentation to trigger, diagnose, and fix the bug.
Are you using any third-party drivers?	Drivers run in the same address space as the kernel, with all the same privileges, so they can cause system crashes if they have bugs.
What was the system doing just before it crashed?	If the system was doing anything unusual like running a new stress test or experiencing higher-than-usual load, that might have led to the crash.
Were there any unusual console messages right before the crash?	Sometimes the system will show signs of distress before it actually crashes; this information is often useful.
Did you add any tuning parameters to the `/etc/system` *file*?	Sometimes tuning parameters, such as increasing shared memory segments so that the system tries to allocate more than it has, can cause the system to crash.

TABLE 6–1 Identifying System Crash Data *(Continued)*

Question	Description
Did the problem start recently?	If so, did the onset of problems coincide with any changes to the system, for example, new drivers, new software, different workload, CPU upgrade, or a memory upgrade.

Troubleshooting a System Crash Checklist

Use this checklist when gathering system data for a crashed system.

Item	Your Data
Is a system crash dump available?	
Identify the operating system release and appropriate software application release levels.	
Identify system hardware.	
Include `prtdiag` output for sun4u systems. Include Explorer output for other systems.	
Are patches installed? If so, include `showrev -p` output.	
Is the problem reproducible?	
Does the system have any third-party drivers?	
What was the system doing before it crashed?	
Were there any unusual console messages right before the system crashed?	
Did you add any parameters to the `/etc/system` file?	
Did the problem start recently?	

Managing System Messages

The following sections describe system messaging features Oracle Solaris .

Viewing System Messages

System messages display on the console device. The text of most system messages look like this:

[ID *msgid facility.priority*]

For example:

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

If the message originated in the kernel, the kernel module name is displayed. For example:

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

When a system crashes, it might display a message on the system console like this:

`panic:` *error message*

Less frequently, this message might be displayed instead of the panic message:

```
Watchdog reset !
```

The error logging daemon, `syslogd`, automatically records various system warnings and errors in message files. By default, many of these system messages are displayed on the system console and are stored in the `/var/adm` directory. You can direct where these messages are stored by setting up system message logging. For more information, see "Customizing System Message Logging" on page 126. These messages can alert you to system problems, such as a device that is about to fail.

The `/var/adm` directory contains several message files. The most recent messages are in `/var/adm/messages` file (and in `messages.*`), and the oldest are in the `messages.3` file. After a period of time (usually every ten days), a new `messages` file is created. The `messages.0` file is renamed `messages.1`, `messages.1` is renamed `messages.2`, and `messages.2` is renamed `messages.3`. The current `/var/adm/messages.3` file is deleted.

Because the `/var/adm` directory stores large files containing messages, crash dumps, and other data, this directory can consume lots of disk space. To keep the `/var/adm` directory from growing too large, and to ensure that future crash dumps can be saved, you should remove unneeded files periodically. You can automate this task by using the `crontab` file. For more information about automating this task, see "How to Delete Crash Dump Files" in *System Administration Guide: Devices and File Systems* and Chapter 3, "Scheduling System Tasks (Tasks)."

▼ How to View System Messages

- **Display recent messages generated by a system crash or reboot by using the `dmesg` command.**

```
$ dmesg
```

Or, use the `more` command to display one screen of messages at a time.

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

Example 6–1 Viewing System Messages

The following example shows output from the `dmesg` command on an Oracle Solaris 10 system.

```
% demsg
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]...
.
.
.
```

See Also For more information, see the `dmesg(1M)` man page.

System Log Rotation

System log files are rotated by the `logadm` command from an entry in the root `crontab` file. The `/usr/lib/newsyslog` script is no longer used.

The system log rotation is defined in the `/etc/logadm.conf` file. This file includes log rotation entries for processes such as `syslogd`. For example, one entry in the `/etc/logadm.conf` file specifies that the `/var/log/syslog` file is rotated weekly unless the file is empty. The most recent `syslog` file becomes `syslog.0`, the next most recent becomes `syslog.1`, and so on. Eight previous `syslog` log files are kept.

The `/etc/logadm.conf` file also contains time stamps of when the last log rotation occurred.

You can use the `logadm` command to customize system logging and to add additional logging in the `/etc/logadm.conf` file as needed.

For example, to rotate the Apache access and error logs, use the following commands:

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

In this example, the Apache `access_log` file is rotated when it reaches 100 MB in size, with a `.0`, `.1`, (and so on) suffix, keeping 10 copies of the old `access_log` file. The `error_log` is rotated when it reaches 10 MB in size with the same suffixes and number of copies as the `access_log` file.

The `/etc/logadm.conf` entries for the preceding Apache log rotation examples look similar to the following:

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

For more information, see `logadm`(1M).

You can use the `logadm` command as superuser or by assuming an equivalent role (with Log Management rights). With role-based access control (RBAC), you can grant non-root users the privilege of maintaining log files by providing access to the `logadm` command.

For example, add the following entry to the `/etc/user_attr` file to grant user `andy` the ability to use the `logadm` command:

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

Customizing System Message Logging

You can capture additional error messages that are generated by various system processes by modifying the `/etc/syslog.conf` file. By default, the `/etc/syslog.conf` file directs many system process messages to the `/var/adm/messages` files. Crash and boot messages are stored here as well. To view `/var/adm` messages, see "How to View System Messages" on page 124.

The `/etc/syslog.conf` file has two columns separated by tabs:

facility.level ... action

facility.level — A *facility* or system source of the message or condition. May be a comma-separated listed of facilities. Facility values are listed in Table 6–2. A *level*, indicates the severity or priority of the condition being logged. Priority levels are listed in Table 6–3.

Do not put two entries for the same facility on the same line, if the entries are for different priorities. Putting a priority in the syslog file indicates that all messages of that all messages of that priority or higher are logged, with the last message taking precedence. For a given facility and level, `syslogd` matches all messages for that level and all higher levels.

action — The action field indicates where the messages are forwarded.

The following example shows sample lines from a default `/etc/syslog.conf` file.

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

This means the following user messages are automatically logged:

- User errors are printed to the console and also are logged to the `/var/adm/messages` file.
- User messages requiring immediate action (`alert`) are sent to the root and operator users.
- User emergency messages are sent to individual users.

Note – Placing entries on separate lines might cause messages to be logged out of order if a log target is specified more than once in the `/etc/syslog.conf` file. Note that you can specify multiple selectors in a single line entry, each separated by a semi-colon.

The most common error condition sources are shown in the following table. The most common priorities are shown in Table 6–3 in order of severity.

TABLE 6–2 Source Facilities for `syslog.conf` Messages

Source	Description
`kern`	The kernel
`auth`	Authentication
`daemon`	All daemons
`mail`	Mail system
`lp`	Spooling system
`user`	User processes

Note – The number of `syslog` facilities that can be activated in the `/etc/syslog.conf` file is unlimited.

TABLE 6–3 Priority Levels for `syslog.conf` Messages

Priority	Description
`emerg`	System emergencies
`alert`	Errors requiring immediate correction
`crit`	Critical errors

TABLE 6–3 Priority Levels for `syslog.conf` Messages *(Continued)*

Priority	Description
`err`	Other errors
`info`	Informational messages
`debug`	Output used for debugging
`none`	This setting doesn't log output

▼ How to Customize System Message Logging

1. **Become the root user.**
2. **Edit the `/etc/syslog.conf` file, adding or changing message sources, priorities, and message locations according to the syntax described in `syslog.conf`(4).**
3. **Exit the file, saving the changes.**

Example 6–2 Customizing System Message Logging

This sample `/etc/syslog.conf` `user.emerg` facility sends user emergency messages to root *and* individual users.

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

Enabling Remote Console Messaging

The following new console features improve your ability to troubleshoot remote systems:

- The `consadm` command enables you to select a serial device as an *auxiliary* (or remote) console. Using the `consadm` command, a system administrator can configure one or more serial ports to display redirected console messages and to host `sulogin` sessions when the system transitions between run levels. This feature enables you to dial in to a serial port with a modem to monitor console messages and participate in `init` state transitions. (For more information, see `sulogin`(1M) and the step-by-step procedures that follow.)

  While you can log in to a system using a port configured as an auxiliary console, it is primarily an output device displaying information that is also displayed on the default console. If boot scripts or other applications read and write to and from the default console, the write output displays on all the auxiliary consoles, but the input is only read from the default console. (For more information abiyt using the `consadm` command during an interactive login session, see "Using the `consadm` Command During an Interactive Login Session" on page 130.)

- Console output now consists of kernel and `syslog` messages written to a new pseudo device, `/dev/sysmsg`. In addition, `rc` script startup messages are written to `/dev/msglog`. Previously, all of these messages were written to `/dev/console`.

  Scripts that direct console output to `/dev/console` need to be changed to `/dev/msglog` if you want to see script messages displayed on the auxiliary consoles. Programs referencing `/dev/console` should be explicitly modified to use `syslog()` or `strlog()` if you want messages to be redirected to an auxiliary device.
- The `consadm` command runs a daemon to monitor auxiliary console devices. Any display device designated as an auxiliary console that disconnects, hangs up or loses carrier, is removed from the auxiliary console device list and is no longer active. Enabling one or more auxiliary consoles does not disable message display on the default console; messages continue to display on `/dev/console`.

Using Auxiliary Console Messaging During Run Level Transitions

Keep the following in mind when using auxiliary console messaging during run level transitions:

- Input cannot come from an auxiliary console if user input is expected for an `rc` script that is run when a system is booting. The input must come from the default console.
- The `sulogin` program, invoked by `init` to prompt for the superuser password when transitioning between run levels, has been modified to send the superuser password prompt to each auxiliary device in addition to the default console device.
- When the system is in single-user mode and one or more auxiliary consoles are enabled using the `consadm` command, a console login session runs on the first device to supply the correct superuser password to the `sulogin` prompt. When the correct password is received from a console device, `sulogin` disables input from all other console devices.
- A message is displayed on the default console and the other auxiliary consoles when one of the consoles assumes single-user privileges. This message indicates which device has become the console by accepting a correct superuser password. If there is a loss of carrier on the auxiliary console running the single-user shell, one of two actions might occur:
  - If the auxiliary console represents a system at run level 1, the system proceeds to the default run level.
  - If the auxiliary console represents a system at run level S, the system displays the `ENTER RUN LEVEL (0-6, s or S):` message on the device where the `init s` or `shutdown` command had been entered from the shell. If there isn't any carrier on that device either, you will have to reestablish carrier and enter the correct run level. The `init` or `shutdown` command will not redisplay the run-level prompt.
- If you are logged in to a system using a serial port, and an `init` or `shutdown` command is issued to transition to another run level, the login session is lost whether this device is the auxiliary console or not. This situation is identical to releases without auxiliary console capabilities.

- Once a device is selected as an auxiliary console using the `consadm` command, it remains the auxiliary console until the system is rebooted or the auxiliary console is unselected. However, the `consadm` command includes an option to set a device as the auxiliary console across system reboots. (See the following procedure for step-by-step instructions.)

Using the `consadm` Command During an Interactive Login Session

If you want to run an interactive login session by logging in to a system using a terminal that is connected to a serial port, and then using the `consadm` command to see the console messages from the terminal, note the following behavior:

- If you use the terminal for an interactive login session while the auxiliary console is active, the console messages are sent to the `/dev/sysmsg` or `/dev/msglog` devices.
- While you issue commands on the terminal, input goes to your interactive session and not to the default console (`/dev/console`).
- If you run the `init` command to change run levels, the remote console software kills your interactive session and runs the `sulogin` program. At this point, input is accepted only from the terminal and is treated like it's coming from a console device. This allows you to enter your password to the `sulogin` program as described in "Using Auxiliary Console Messaging During Run Level Transitions" on page 129.

  Then, if you enter the correct password on the (auxiliary) terminal, the auxiliary console runs an interactive `sulogin` session, locks out the default console and any competing auxiliary console. This means the terminal essentially functions as the system console.
- From here you can change to run level 3 or go to another run level. If you change run levels, `sulogin` runs again on all console devices. If you exit or specify that the system should come up to run level 3, then all auxiliary consoles lose their ability to provide input. They revert to being display devices for console messages.

  As the system is coming up, you must provide information to `rc` scripts on the default console device. After the system comes back up, the `login` program runs on the serial ports and you can log back into another interactive session. If you've designated the device to be an auxiliary console, you will continue to get console messages on your terminal, but all input from the terminal goes to your interactive session.

▼ How to Enable an Auxiliary (Remote) Console

The `consadm` daemon does not start monitoring the port until after you add the auxiliary console with the `consadm` command. As a security feature, console messages are only redirected until carrier drops, or the auxiliary console device is unselected. This means carrier must be established on the port before you can successfully use the `consadm` command.

For more information about enabling an auxiliary console, see the `consadm(1m)` man page.

1 Log in to the system as the root user.

2. **Enable the auxiliary console.**

   ```
   # consadm -a devicename
   ```

3. **Verify that the current connection is the auxiliary console.**

   ```
   # consadm
   ```

Example 6–3 Enabling an Auxiliary (Remote) Console

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

How to Display a List of Auxiliary Consoles

1. **Log in to the system as superuser.**

2. **Select one of the following steps:**

   a. **Display the list of auxiliary consoles.**

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

   b. **Display the list of persistent auxiliary consoles.**

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

How to Enable an Auxiliary (Remote) Console Across System Reboots

1. **Log in to the system as superuser.**

2. **Enable the auxiliary console across system reboots.**

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

   This adds the device to the list of persistent auxiliary consoles.

3. **Verify that the device has been added to the list of persistent auxiliary consoles.**

   ```
   # consadm
   ```

Example 6–4 Enabling an Auxiliary (Remote) Console Across System Reboots

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

▼ How to Disable an Auxiliary (Remote) Console

1. **Log in to the system as superuser.**
2. **Select one of the following steps:**
   a. **Disable the auxiliary console.**

      ```
      # consadm -d devicename
      ```

      or

   b. **Disable the auxiliary console and remove it from the list of persistent auxiliary consoles.**

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

3. **Verify that the auxiliary console has been disabled.**

   ```
   # consadm
   ```

Example 6–5 Disabling an Auxiliary (Remote) Console

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

Troubleshooting File Access Problems

Users frequently experience problems, and call on a system administrator for help, because they cannot access a program, a file, or a directory that they could previously use.

Whenever you encounter such a problem, investigate one of three areas:

- The user's search path may have been changed, or the directories in the search path may not be in the proper order.
- The file or directory may not have the proper permissions or ownership.
- The configuration of a system accessed over the network may have changed.

This chapter briefly describes how to recognize problems in each of these three areas and suggests possible solutions.

Solving Problems With Search Paths (`Command not found`)

A message of `Command not found` indicates one of the following:

- The command is not available on the system.
- The command directory is not in the search path.

To fix a search path problem, you need to know the pathname of the directory where the command is stored.

If the wrong version of the command is found, a directory that has a command of the same name is in the search path. In this case, the proper directory may be later in the search path or may not be present at all.

You can display your current search path by using the `echo $PATH` command. For example:

```
$ echo $PATH
/home/kryten/bin:/sbin:/usr/sbin:/usr/bin:/usr/dt:/usr/dist/exe
```

Use the `type` command to determine whether you are running the wrong version of the command. For example:

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

▼ How to Diagnose and Correct Search Path Problems

1 Display the current search path to verify that the directory for the command is not in your path or that it isn't misspelled.

```
$ echo $PATH
```

2 Check the following:

- Is the search path correct?
- Is the search path listed before other search paths where another version of the command is found?
- Is the command in one of the search paths?

If the path needs correction, go to step 3. Otherwise, go to step 4.

3 Add the path to the appropriate file, as shown in this table.

Shell	File	Syntax	Notes
bash and ksh93	$HOME/.profile	$ PATH=*$HOME/bin:/sbin:/usr/local /bin ...* $ export PATH	A colon separates path names.

4 Activate the new path as follows:

Shell	Path Location	Command to Activate The Path
bash and ksh93	.profile	$ **. ./.profile**
	.login	*hostname*% **source .login**

5 Verify the new path.

$ **which** *command*

Example 6–6 Diagnosing and Correcting Search Path Problems

This example shows that the mytool executable is not in any of the directories in the search path using the type command.

```
$ 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
```

If you cannot find a command, look at the man page for its directory path. For example, if you cannot find the lpsched command (the lp printer daemon), the lpsched(1M) man page tells you the path is /usr/lib/lp/lpsched.

Changing File and Group Ownerships

Frequently, file and directory ownerships change because someone edited the files as superuser. When you create home directories for new users, be sure to make the user the owner of the dot (.) file in the home directory. When users do not own ".” they cannot create files in their own home directory.

Access problems can also arise when the group ownership changes or when a group of which a user is a member is deleted from the /etc/group database.

For information about how to change the permissions or ownership of a file that you are having problems accessing, see Chapter 7, "Controlling Access to Files (Tasks)," in *System Administration Guide: Security Services*.

Solving File Access Problems

When users cannot access files or directories that they previously could access, the permissions or ownership of the files or directories probably has changed.

Recognizing Problems With Network Access

If users have problems using the `rcp` remote copy command to copy files over the network, the directories and files on the remote system may have restricted access by setting permissions. Another possible source of trouble is that the remote system and the local system are not configured to allow access.

See "Strategies for NFS Troubleshooting" in *System Administration Guide: Network Services* for information about problems with network access and problems with accessing systems through AutoFS.

◆ ◆ ◆ CHAPTER 7

Managing Core Files (Tasks)

This chapter describes how to manage core files with the `coreadm` command.

For information about the procedures that are associated with managing core files, see "Managing Core Files (Task Map)" on page 137.

What's New in Managing Core Files

This section describes new or changed features for managing core files in the Oracle Solaris release.

`coreadm` Command Configuration Controlled by SMF

The `coreadm` configuration information is now stored in the Service Management Facility (SMF). This change does not impact administrative use of the `coreadm` command or any of its options.

Because the `coreadm` configuration has moved to SMF, use of certain `coreadm` options requires a particular set of authorizations. There is no obvious change to user logins that include the `Maintenance and Repair` rights profile. For more information, see the `coreadm(1M)` man page.

Managing Core Files (Task Map)

Task	Description	For Instructions
1. Display the current core dump configuration.	Display the current core dump configuration by using the `coreadm` command.	"How to Display the Current Core Dump Configuration" on page 140

Task	Description	For Instructions
2. Modify the core dump configuration.	Modify the core dump configuration to do one of the following: Set a core file name pattern. Enable a per-process core file path. Enable a global core file path.	"How to Set a Core File Name Pattern" on page 141 "How to Enable a Per-Process Core File Path" on page 141 "How to Enable a Global Core File Path" on page 141
3. Examine a core dump file.	Use the `proc` tools to view a core dump file.	"Examining Core Files" on page 142

Managing Core Files Overview

Core files are generated when a process or application terminates abnormally. Core files are managed with the `coreadm` command.

For example, you can use the `coreadm` command to configure a system so that all process core files are placed in a single system directory. This means it is easier to track problems by examining the core files in a specific directory whenever a process or daemon terminates abnormally.

Configurable Core File Paths

Two new configurable `core` file paths that can be enabled or disabled independently of each other are:

- A per-process core file path, which defaults to `core` and is enabled by default. If enabled, the per-process core file path causes a `core` file to be produced when the process terminates abnormally. The per-process path is inherited by a new process from its parent process.

  When generated, a per-process core file is owned by the owner of the process with read/write permissions for the owner. Only the owning user can view this file.
- A global core file path, which defaults to `core` and is disabled by default. If enabled, an *additional* core file with the same content as the per-process core file is produced by using the global core file path.

  When generated, a global core file is owned by superuser with read/write permissions for superuser only. Non-privileged users cannot view this file.

When a process terminates abnormally, it produces a core file in the current directory by default. If the global core file path is enabled, each abnormally terminating process might produce two files, one in the current working directory, and one in the global core file location.

By default, a `setuid` process does not produce core files using either the global or per-process path.

Expanded Core File Names

If a global core file directory is enabled, `core` files can be distinguished from one another by using the variables described in the following table.

Variable Name	Variable Definition
`%d`	Executable file directory name, up to a maximum of `MAXPATHLEN` characters
`%f`	Executable file name, up to a maximum of `MAXCOMLEN` characters
`%g`	Effective group ID
`%m`	Machine name (`uname -m`)
`%n`	System node name (`uname -n`)
`%p`	Process ID
`%t`	Decimal value of time(2)
`%u`	Effective user ID
`%z`	Name of the zone in which process is executed (`zonename`)
`%%`	Literal %

For example, if the global core file path is set to:

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

and a `sendmail` process with PID `12345` terminates abnormally, it produces the following `core` file:

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

Setting the Core File Name Pattern

You can set a core file name pattern on a global, zone, or per-process basis. In addition, you can set the per-process defaults that persist across a system reboot.

For example, the following `coreadm` command sets the default per-process core file pattern. This setting applies to all processes that have not explicitly overridden the default core file pattern. This setting persists across system reboots.

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

The following `coreadm` command sets the per-process core file name pattern for any processes:

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

The `$$` symbols represent a placeholder for the process ID of the currently running shell. The per-process core file name pattern is inherited by all child processes.

Once a global or per-process core file name pattern is set, it must be enabled with the `coreadm -e` command. See the following procedures for more information.

You can set the core file name pattern for all processes run during a user's login session by putting the command in a user's `$HOME/.profile` or `.login` file.

Enabling `setuid` Programs to Produce Core Files

You can use the `coreadm` command to enable or disable `setuid` programs to produce core files for all system processes or on a per-process basis by setting the following paths:

- If the global `setuid` option is enabled, a global core file path allows all `setuid` programs on a system to produce `core` files.
- If the per-process `setuid` option is enable, a per-process core file path allows specific `setuid` processes to produce `core` files.

By default, both flags are disabled. For security reasons, the global core file path must be a full pathname, starting with a leading `/`. If superuser disables per-process core files, individual users cannot obtain core files.

The `setuid` core files are owned by superuser with read/write permissions for superuser only. Regular users cannot access them even if the process that produced the `setuid` core file was owned by an ordinary user.

For more information, see the `coreadm(1M)` man page.

How to Display the Current Core Dump Configuration

Use the `coreadm` command without any options to display the current core dump configuration.

```
$ 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
```

How to Set a Core File Name Pattern

- **Determine whether you want to set a per-process or global core file and select one of the following:**

  a. **Set a per-process file name pattern.**

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

  b. **Become the root user.**

  c. **Set a global file name pattern.**

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

How to Enable a Per-Process Core File Path

1. **Become the root user.**
2. **Enable a per-process core file path.**

   ```
   # coreadm -e process
   ```

3. **Display the current process core file path to verify the configuration.**

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

How to Enable a Global Core File Path

1. **Become the root user.**
2. **Enable a global core file path.**

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

3. **Display the current process core file path to verify the configuration.**

   ```
   # 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
```

Troubleshooting Core File Problems

Error Message

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

Cause

You have an obsolete parameter that allows `setuid` core files in your `/etc/system` file.

Solution

Remove `allow_setid_core=1` from the `/etc/system` file. Then use the `coreadm` command to enable global `setuid` core file paths.

Examining Core Files

Some of the `proc` tools have been enhanced to examine process core files, as well as live processes. The `proc` tools are utilities that can manipulate features of the `/proc` file system.

The `/usr/proc/bin/pstack`, `pmap`, `pldd`, `pflags`, and `pcred` tools can now be applied to core files by specifying the name of the core file on the command line, similar to the way you specify a process ID to these commands.

For more information about using `proc` tools to examine core files, see `proc(1)`.

EXAMPLE 7–1 Examining Core Files With `proc` Tools

```
$ ./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
```

◆ ◆ ◆ CHAPTER 8

Managing System Crash Information (Tasks)

This chapter describes how to manage system crash information in the Oracle Solaris OS.

For information abiyt the procedures that are associated with managing system crash information, see "Managing System Crash Information (Task Map)" on page 144.

What's New in Managing System Crash Information

This section describes new or changed features for managing system resources in this Oracle Solaris release.

Fast Crash Dump Facility

This feature enhancement enables the system to save crash dumps in less time, using less space. The time that is required for a crash dump to complete is now 2 to 10 times faster, depending on the platform. The amount of disk space that is required to save crash dumps in the `savecore` directory is reduced by the same factors. To accelerate the creation and compression of the crash dump file, the fast crash dump facility utilizes lightly used CPUs on large systems. A new crash dump file, `vmdump.`*n*, is a compressed version of the `vmcore.`*n* and `unix.`*n* files. Compressed crash dumps can be moved over the network more quickly and then analyzed off-site. Note that the dump file must first be uncompressed to use it with tools like the `mdb` utility. You can uncompress a dump file by using the `savecore` command, either locally or remotely.

To support the new crash dump facility, the `-z` option has been added to the `dumpadm` command. Use this option to specify whether to save dumps in a compressed or an uncompressed format. The default format is compressed.

For more detailed information, see the `dumpadm(1M)` and the `savecore(1M)` man pages.

Managing System Crash Information (Task Map)

The following task map identifies the procedures needed to manage system crash information.

Task	Description	For Instructions
1. Display the current crash dump configuration.	Display the current crash dump configuration by using the `dumpadm` command.	"How to Display the Current Crash Dump Configuration" on page 148
2. Modify the crash dump configuration.	Use the `dumpadm` command to specify the type of data to dump, whether or not the system will use a dedicated dump device, the directory for saving crash dump files, and the amount of space that must remain available after crash dump files are written.	"How to Modify a Crash Dump Configuration" on page 148
3. Examine a crash dump file.	Use the `mdb` command to view crash dump files.	"How to Examine a Crash Dump" on page 150
4. (Optional) Recover from a full crash dump directory.	The system crashes, but no room is available in the `savecore` directory, and you want to save some critical system crash dump information.	"How to Recover From a Full Crash Dump Directory (Optional)" on page 151
5. (Optional) Disable or enable the saving of crash dump files.	Use the `dumpadm` command to disable or enable the saving the crash dump files. Saving of crash dump files is enabled by default.	"How to Disable or Enable the Saving of Crash Dumps" on page 151

System Crashes (Overview)

System crashes can occur due to hardware malfunctions, I/O problems, and software errors. If the system crashes, it will display an error message on the console, and then write a copy of its physical memory to the dump device. The system will then reboot automatically. When the system reboots, the `savecore` command is executed to retrieve the data from the dump device and write the saved crash dump to your `savecore` directory. The saved crash dump files provide invaluable information to your support provider to aid in diagnosing the problem.

The crash dump information is written in a compressed format to the `vmdump.`*n* file, where `n` is an integer that identifies the crash dump. Afterwards, the `savecore` command can be invoked on the same system or another system to expand the compressed crash dump to a pair of files that are named `unix.`*n* and `vmcore.`*n*. The directory in which the crash dump is saved upon reboot can also be configured by using the `dumpadm` command.

For systems that have an Oracle Solaris ZFS root file system, dedicated ZFS volumes are used for swap and dump areas. See "Oracle Solaris ZFS Support for Swap Area and Dump Devices" on page 145 for more information.

Oracle Solaris ZFS Support for Swap Area and Dump Devices

If you select an Oracle Solaris ZFS root file system during an initial software installation to migrate from a UFS root file system to a ZFS root file system, a swap area is created on a ZFS volume in the ZFS root pool. Swap volume size is calculated as half the size of physical memory, but no more than 2 Gbytes and no less than 512 Mbytes. Dump volume size is calculated by the kernel based on `dumpadm` information and the size of physical memory. You can adjust the sizes of your swap and dump volumes in a JumpStart profile or during an initial installation to sizes of your choosing as long as the new sizes support system operation. For more information, see "Managing Your ZFS Swap and Dump Devices" in *Oracle Solaris ZFS Administration Guide*.

If you need to modify your ZFS swap area or dump area after installation, use the `swap` or `dumpadm` commands, as in previous releases.

For information about managing dump devices in this document, see "Managing System Crash Dump Information" on page 147.

x86: System Crashes in the GRUB Boot Environment

If a system crash occurs on an x86 based system in the GRUB boot environment, it is possible that the SMF service that manages the GRUB boot archive, `svc:/system/boot-archive:default`, might fail on the next system reboot. For more information aobut GRUB based booting, see "Booting an x86 Based System (Task Map)" in *System Administration Guide: Basic Administration*.

System Crash Dump Files

The `savecore` command runs automatically after a system crash to retrieve the crash dump information from the dump device and writes a pair of files called `unix.`*X* and `vmcore.`*X*, where *X* identifies the dump sequence number. Together, these files represent the saved system crash dump information.

Crash dump files are sometimes confused with *core* files, which are images of user applications that are written when the application terminates abnormally.

Crash dump files are saved in a predetermined directory, which by default, is `/var/crash/`*hostname*. In previous releases, crash dump files were overwritten when a system

rebooted, unless you manually enabled the system to save the images of physical memory in a crash dump file. Now, the saving of crash dump files is enabled by default.

System crash information is managed with the `dumpadm` command. For more information, see "The `dumpadm` Command" on page 146.

Saving Crash Dumps

You can examine the control structures, active tables, memory images of a live or crashed system kernel, and other information about the operation of the kernel by using the `mdb` utility. Using `mdb` to its full potential requires a detailed knowledge of the kernel, and is beyond the scope of this manual. For information about using this utility, see the `mdb(1)` man page.

Additionally, crash dumps saved by `savecore` can be useful to send to a customer service representative for analysis of why the system is crashing.

The `dumpadm` Command

Use the `dumpadm` command to manage system crash dump information in the Oracle Solaris OS.

- The `dumpadm` command enables you to configure crash dumps of the operating system. The `dumpadm` configuration parameters include the dump content, dump device, and the directory in which crash dump files are saved.
- Dump data is stored in compressed format on the dump device. Kernel crash dump images can be as big as 4 Gbytes or more. Compressing the data means faster dumping and less disk space needed for the dump device.
- Saving crash dump files is run in the background when a dedicated dump device, not the swap area, is part of the dump configuration. This means a booting system does not wait for the `savecore` command to complete before going to the next step. On large memory systems, the system can be available before `savecore` completes.
- System crash dump files, generated by the `savecore` command, are saved by default.
- The `savecore -L` command is a new feature which enables you to get a crash dump of the live running the Oracle Solaris OS. This command is intended for troubleshooting a running system by taking a snapshot of memory during some bad state, such as a transient performance problem or service outage. If the system is up and you can still run some commands, you can execute the `savecore -L` command to save a snapshot of the system to the dump device, and then immediately write out the crash dump files to your `savecore` directory. Because the system is still running, you can only use the `savecore -L` command if you have configured a dedicated dump device.

The following table describes `dumpadm`'s configuration parameters.

Dump Parameter	Description
dump device	The device that stores dump data temporarily as the system crashes. When the dump device is not the swap area, `savecore` runs in the background, which speeds up the boot process.
`savecore` directory	The directory that stores system crash dump files.
dump content	Type of memory data to dump.
minimum free space	Minimum amount of free space required in the `savecore` directory after saving crash dump files. If no minimum free space has been configured, the default is one Mbyte.

For more information, see `dumpadm(1M)`.

Dump configuration parameters are managed by the `dumpadm` command.

How the `dumpadm` Command Works

During system startup, the `dumpadm` command is invoked by the `svc:/system/dumpadm:default` service to configure crash dumps parameters.

Specifically, `dumpadm` initializes the dump device and the dump content through the `/dev/dump` interface.

After the dump configuration is complete, the `savecore` script looks for the location of the crash dump file directory. Then, `savecore` is invoked to check for crash dumps and check the content of the `minfree` file in the crash dump directory.

Managing System Crash Dump Information

Keep the following key points in mind when you are working with system crash information:

- You must be the root user to access and manage system crash information.
- Do not disable the option of saving system crash dumps. System crash dump files provide an invaluable way to determine what is causing the system to crash.
- Do not remove important system crash information until it has been sent to your customer service representative.

How to Display the Current Crash Dump Configuration

1. **Become the root user.**

2. **Display the current crash dump configuration.**

```
# dumpadm
Dump content: kernel pages
Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/venus
 Savecore enabled: yes
  Saved compressed: on
```

The preceding example output means:

- The dump content is kernel memory pages.
- Kernel memory will be dumped on a swap device, `/dev/dsk/c0t3d0s1`. You can identify all your swap areas with the `swap -l` command.
- System crash dump files will be written in the `/var/crash/venus` directory.
- Saving crash dump files is enabled.
- Save crash dumps in compressed format.

How to Modify a Crash Dump Configuration

1. **Become the root user.**

2. **Identify the current crash dump configuration.**

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
Savecore directory: /var/crash/pluto
 Savecore enabled: yes
     Save commpressed: on
```

This output identifies the default dump configuration for a system running the Oracle Solaris 10 release.

3. **Modify the crash dump configuration.**

```
# /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* — Specifies the type of data to dump. Use `kernel` to dump of all kernel memory, `all` to dump all of memory, or `curproc`, to dump kernel memory and the memory pages of the process whose thread was executing when the crash occurred. The default dump content is kernel memory.

`-d` *dump-device*	Specifies the device that stores dump data temporarily as the system crashes. The primary swap device is the default dump device.
`-m` *nnnk* | *nnnm* | *nnn%*	Specifies the minimum free disk space for saving crash dump files by creating a `minfree` file in the current `savecore` directory. This parameter can be specified in Kbytes (`nnnk`), Mbytes (`nnnm`) or file system size percentage (`nnn%`). The `savecore` command consults this file prior to writing the crash dump files. If writing the crash dump files, based on their size, would decrease the amount of free space below the `minfree` threshold, the dump files are not written and an error message is logged. For information about recovering from this scenario, see "How to Recover From a Full Crash Dump Directory (Optional)" on page 151.
`-n`	Specifies that `savecore` should not be run when the system reboots. This dump configuration is not recommended. If system crash information is written to the swap device, and `savecore` is not enabled, the crash dump information is overwritten when the system begins to swap.
`-s`	Specifies an alternate directory for storing crash dump files. The default directory is `/var/crash/hostname` where `hostname` is the output of the `uname -n` command.
`-u`	Forcibly updates the kernel dump configuration based on the contents of the `/etc/dumpadm.conf` file.
`-y`	Modifies the dump configuration to automatically execute the `savecore` command upon reboot, which is the default for this dump setting.
`-z on | off`	Modifies the dump configuration to control the operation of the `savecore` command upon reboot. The `on` setting enables the saving of core file in a compressed format. The `off` setting automatically uncompresses the crash dump file. Because crash dump files can be extremely large and therefore require less file system space if they are saved in a compressed forma, the default is `on`.

Example 8–1 Modifying a Crash Dump Configuration

In this example, all of memory is dumped to the dedicated dump device, `/dev/dsk/c0t1d0s1`, and the minimum free space that must be available after the crash dump files are saved is 10% of the file system space.

```
# dumpadm
      Dump content: kernel pages
       Dump device: /dev/dsk/c0t3d0s1 (swap)
```

```
Savecore directory: /var/crash/pluto
  Savecore enabled: yes
   Save compressed: on
 # dumpadm -c all -d /dev/dsk/c0t1d0s1 -m 10%
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: yes
    Save compressed: on
```

▼ How to Examine a Crash Dump

1 Become the root user.

2 Examine a crash dump by using the `mdb` utility.

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

`-k` Specifies kernel debugging mode by assuming the file is an operating system crash dump file.

crashdump-file Specifies the operating system crash dump file.

3 Display crash status information.

```
# /usr/bin/mdb file-name
> ::status
   .
   .
   .
> ::system
   .
   .
   .
```

Example 8–2 Examining a Crash Dump

The following example shows sample output from the `mdb` utility, which includes system information and identifies the tunables that are set in this system's `/etc/system` file.

```
# /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 (sun4u)
> ::system
set ufs_ninode=0x9c40 [0t40000]
set ncsize=0x4e20 [0t20000]
set pt_cnt=0x400 [0t1024]
```

▼ How to Recover From a Full Crash Dump Directory (Optional)

In this scenario, the system crashes but no room is left in the `savecore` directory, and you want to save some critical system crash dump information.

1. **After the system reboots, log in as superuser.**
2. **Clear out the `savecore` directory, typically, `/var/crash/`*hostname*, by removing existing crash dump files that have already been sent to your service provider.**

   - **Alternatively, you can manually run the `savecore` command to specify an alternate directory that has sufficient disk space.**

     ```
     # savecore [ directory ]
     ```

▼ How to Disable or Enable the Saving of Crash Dumps

1. **Become the root user.**
2. **Disable or enable the saving of crash dumps on your system.**

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

Example 8–3 Disabling the Saving of Crash Dumps

This example illustrates how to disable the saving of crash dumps on your system.

```
# dumpadm -n
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: no
   Save Compressed: on
```

Example 8–4 Enabling the Saving of Crash Dumps

This example illustrates how to enable the saving of crash dump on your system.

```
# dumpadm -y
      Dump content: all pages
       Dump device: /dev/dsk/c0t1d0s1 (dedicated)
Savecore directory: /var/crash/pluto (minfree = 77071KB)
  Savecore enabled: yes
   Save compressed: on
```

◆ ◆ ◆ CHAPTER 9

Troubleshooting Miscellaneous System Problems (Tasks)

This chapter describes miscellaneous software problems that might occur occasionally and are relatively easy to fix. Troubleshooting miscellaneous software problems includes solving problems that aren't related to a specific software application or topic, such as unsuccessful reboots and full file systems. Resolving these problems are described in the following sections.

This is a list of the information that is in this chapter.

What to Do If Rebooting Fails

Note – Some of the information in this section pertains to systems that are running the Oracle Solaris 10 release *only*.

If the system does not reboot completely, or if it reboots and then crashes again, there might be a software or hardware problem that is preventing the system from booting successfully.

Cause of System Not Booting	How to Fix the Problem
The system can't find `/platform/‘uname -m‘/kernel/unix`.	You may need to change the `boot-device` setting in the PROM on a SPARC based system. For information about changing the default boot device, see "How to Change the Default Boot Device by Using the Boot PROM" in *System Administration Guide: Basic Administration*.

Cause of System Not Booting	How to Fix the Problem
Oracle Solaris 11 Express: The GRUB boot archive has become corrupted or the GRUB menu is lost. Or, the SMF boot archive service has failed. An error message is displayed if you run the `svcs -x` command.	Create a second boot environment (BE) that is a backup of the primary BE. In the event the primary BE is not bootable, boot the backup BE. Alternatively, you can boot from the live CD or USB media. For more information about creating and managing BEs, see *Managing Boot Environments With Oracle Solaris 11 Express*.
There's an invalid entry in the `/etc/passwd` file.	For information about recovering from an invalid `passwd` file, see Chapter 7, "Booting an Oracle Solaris System (Tasks)," in *System Administration Guide: Basic Administration*.
There's a hardware problem with a disk or another device.	Check the hardware connections: ■ Make sure the equipment is plugged in. ■ Make sure all the switches are set properly. ■ Look at all the connectors and cables, including the Ethernet cables. ■ If all this fails, turn off the power to the system, wait 10 to 20 seconds, and then turn on the power again.

If none of the above suggestions solve the problem, contact your local service provider.

What to Do If You Forgot the Root Password

If you forget the root password and you cannot log into the system, you will have to do the following:

- Stop the system by using the keyboard stop sequence.
- Boot the system the installation media.
- Mount the root (`/`) file system.
- Remove the root password from the `/etc/shadow` file.
- Reboot the system.
- Log in and set root's password.

These procedures are fully described in Chapter 7, "Booting an Oracle Solaris System (Tasks)," in *System Administration Guide: Basic Administration*.

The following examples describe how to recover from a forgotten root password on a SPARC based systems or an x86 based system that does not implement GRUB.

EXAMPLE 9–1 SPARC: What to Do If You Forgot the Root Password

The following example shows how to recover when you forget the root password by booting from the network. This example assumes that the boot server is already available. Be sure to apply a new root password after the system has rebooted.

```
(Use keyboard abort sequence--Press Stop A keys to stop the system)
ok boot net -s
# mount /dev/dsk/c0t3d0s0 /a
# cd /a/etc
# TERM=vt100
# export TERM
# vi shadow
(Remove root's encrypted password string)
# cd /
# umount /a
# init 6
```

What to Do If a System Hangs

A system can freeze or hang rather than crash completely if some software process is stuck. Follow these steps to recover from a hung system.

1. Determine whether the system is running a window environment and follow these suggestions. If these suggestions do not solve the problem, go to step 2.
   - Make sure the pointer is in the window where you are typing the commands.
   - Press Control-q in case the user accidentally pressed Control-s, which freezes the screen. Control-s freezes only the window, not the entire screen. If a window is frozen, try using another window.
   - If possible, log in remotely from another system on the network. Use the `pgrep` command to look for the hung process. If it looks like the window system is hung, identify the process and kill it.
2. Press Control-\ to force a "quit" in the running program and (probably) write out a `core` file.
3. Press Control-c to interrupt the program that might be running.
4. Log in remotely and attempt to identify and kill the process that is hanging the system.
5. Log in remotely, become root and then reboot the system.
6. If the system still does not respond, force a crash dump and reboot. For information about forcing a crash dump and booting, see "Forcing a Crash Dump and Reboot of the System" in *System Administration Guide: Basic Administration*.
7. If the system still does not respond, turn the power off, wait a minute or so, then turn the power back on.
8. If you cannot get the system to respond at all, contact your local service provider for help.

What to Do If a File System Fills Up

When the root (/) file system or any other file system fills up, you will see the following message in the console window:

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

There are several reasons why a file system fills up. The following sections describe several scenarios for recovering from a full file system. For information about how to routinely clean out old and unused files to prevent file systems from becoming too full, see Chapter 16, "Managing Disk Use (Tasks)," in *System Administration Guide: Devices and File Systems*.

File System Fills Up Because a Large File or Directory Was Created

Reason Error Occurred	How to Fix the Problem
Someone accidentally copied a file or directory to the wrong location. This also happens when an application crashes and writes a large `core` file into the file system.	Log in as superuser and use the `ls -tl` command in the specific file system to identify which large file is newly created and remove it. For information about removing `core` files, see "How to Find and Delete core Files" in *System Administration Guide: Devices and File Systems*.

A `TMPFS` File System Is Full Because the System Ran Out of Memory

Reason Error Occurred	How to Fix the Problem
This can occur if `TMPFS` is trying to write more than it is allowed or some current processes are using a lot of memory.	For information about recovering from `tmpfs`-related error messages, see the `tmpfs`(7FS) man page.

What to Do If File ACLs Are Lost After Copy or Restore

Reason Error Occurred	How to Fix the Problem
If files or directories with ACLs are copied or restored into the `/tmp` directory, the ACL attributes are lost. The `/tmp` directory is usually mounted as a temporary file system, which doesn't support UFS file system attributes such as ACLs.	Copy or restore files into the `/var/tmp` directory instead.

Index

D

E

F

N

P

Q

R

S

T

U

V

W

Z