

Managing System Information, Processes, and Performance in Oracle® Solaris 11.1

Copyright © 1998, 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

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

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf disposition de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, breveter, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est concédé sous licence au Gouvernement des Etats-Unis, ou à toute entité qui délivre la licence de ce logiciel ou l'utilise pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique:

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

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer des dommages corporels. Si vous utilisez ce logiciel ou matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour ce type d'applications.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée d'The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation.

Contents

Preface	7
1 Managing System Information (Tasks)	11
What's New in Displaying and Changing System Information	11
Enhancements to <code>hostname</code> and <code>domainname</code> Commands	11
Displaying System Information	12
Displaying System Information (Task Map)	12
Commands That Are Used to Display System Information	13
▼ How to Display a System's Release Information	13
▼ How to Display the Date and Time	14
▼ How to Display a System's Host ID Number	14
Displaying a System's Architecture and Processor Type	14
▼ How to Display a System's Product Name	15
▼ How to Display a System's Installed Memory	16
▼ How to Display Default and Customized Property Values for a Device	16
▼ How to Display System Diagnostic Information	20
Identifying Information About Chip Multithreading Features	22
▼ How to Display a System's Physical Processor Type	22
▼ How to Display a System's Virtual Processor Type	23
Changing System Information	24
Changing System Information (Task Map)	24
▼ How to Manually Set a System's Date and Time	24
▼ How to Set Up a Message-Of-The-Day	25
▼ How to Change a System's Identity	26
2 Managing System Processes (Tasks)	27
What's New in Managing System Processes	27

Pseudo System Processes	27
Managing System Processes	28
Managing System Processes (Task Map)	28
Commands for Managing System Processes	28
Managing Process Class Information	38
Managing Process Class Information (Task Map)	38
Changing the Scheduling Priority of Processes (prioctl)	39
▼ How to Display Basic Information About Process Classes (prioctl)	39
▼ How to Display the Global Priority of a Process	40
▼ How to Designate a Process Priority (prioctl)	41
▼ How to Change Scheduling Parameters of a Timesharing Process (prioctl)	42
▼ How to Change the Class of a Process (prioctl)	43
Changing the Priority of a Timesharing Process (nice)	43
▼ How to Change the Priority of a Process (nice)	44
Troubleshooting Problems With System Processes	45
3 Monitoring System Performance (Tasks)	47
Where to Find System Performance Tasks	47
System Performance and System Resources	48
Processes and System Performance	48
About Monitoring System Performance	50
Monitoring Tools	50
Displaying System Performance Information	51
Displaying System Performance Information (Task Map)	51
Displaying Virtual Memory Statistics (vmstat)	52
Displaying Disk Utilization Information (iostat)	55
Displaying Disk Space Statistics (df)	58
Monitoring System Activities	59
Monitoring System Activities (Task Map)	60
Monitoring System Activities (sar)	61
Collecting System Activity Data Automatically (sar)	77
4 Scheduling System Tasks (Tasks)	81
Ways to Automatically Executing System Tasks	81
For Scheduling Repetitive Jobs: crontab	82

For Scheduling a Single Job: at	82
Scheduling System Tasks	83
Creating and Editing crontab Files (Task Map)	83
Scheduling a Repetitive System Task (cron)	84
Creating and Editing crontab Files	86
Displaying crontab Files	88
Removing crontab Files	89
Controlling Access to the crontab Command	90
Scheduling Tasks by Using the at Command	93
Using the at Command (Task Map)	93
Scheduling a Single System Task (at)	94
5 Managing the System Console, Terminal Devices, and Power Services (Tasks)	101
What's New in Managing the System Console, Terminal Devices, and Power Services	101
Changes to How System Power Services Are Managed	101
Managing System Console and Locally Connected Terminal Devices	102
SMF Services That Manage the System Console and Locally Connected Terminal Devices	102
Managing System Power Services	105
Troubleshooting System Power Problems	108
Index	109

Preface

Managing System Information, Processes and Performance in Oracle Solaris 11.1 is part of a documentation set that provides a significant portion of the Oracle Solaris system administration information. This guide contains information for both SPARC based and x86 based systems.

This book assumes you have completed the following tasks:

- Installed the Oracle Solaris software
- Set up all the networking software that you plan to use

For Oracle Solaris, new features that might be 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*. 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 running the Oracle Solaris 11 release. To use this book, you should have 1–2 years of UNIX system administration experience. Attending UNIX system administration training courses might be helpful.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Typographic Conventions

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

TABLE P-1 Typographic Conventions

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

TABLE P-2 Shell Prompts (Continued)

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

Managing System Information (Tasks)

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

This chapter does 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 *Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management*.

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

- “What's New in Displaying and Changing System Information” on page 11
- “Displaying System Information” on page 12
- “Changing System Information” on page 24

What's New in Displaying and Changing System Information

Enhancements to `hostname` and `domainname` Commands

In this release, the `hostname` and `domainname` commands have been enhanced to enable you to permanently set `hostname` and `domainname` more easily. When you use these commands, the corresponding SMF properties and associated SMF service, are also automatically updated.

For more information, see [“How to Change a System's Identity”](#) on page 26 and the `hostname(1)`, `domainname(1M)`, and `nodename(4)` man pages.

Displaying System Information

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

Displaying System Information (Task Map)

Task	Description	For Instructions
Display a system's release information.	Display the contents of the <code>/etc/release</code> file to identify the Oracle Solaris release version.	“How to Display a System's Release Information” on page 13
Display a system's date and time.	Use the <code>date</code> command to display your system's date and time.	“How to Display the Date and Time” on page 14
Display a system's host ID number.	Use the <code>hostid</code> command to display your system's host id.	“How to Display a System's Host ID Number” on page 14
Display a system's architecture or processor type.	Use the <code>isainfo</code> command to display a system's architecture type. Use the <code>isalist</code> command to display a system's processor type.	“Displaying a System's Architecture and Processor Type” on page 14
Display a system's product name.	You can use the <code>prtconf -b</code> command to display the product name of a system.	“How to Display a System's Product Name” on page 15
Display a system's installed memory.	Use the <code>prtconf</code> command to display information about your system's installed memory.	“How to Display a System's Installed Memory” on page 16
Display the original and default values for a device.	Use the <code>prtconf</code> command with the <code>-u</code> option to display both the default and updated property values for a device.	“How to Display Default and Customized Property Values for a Device” on page 16
Display a system's configuration and diagnostic information.	Use the <code>prtdiag</code> command with the appropriate option to display a system's configuration and diagnostic information.	“How to Display System Diagnostic Information” on page 20

Task	Description	For Instructions
Display a system's physical and virtual processor information.	<p>Use the <code>psrinfo -p</code> command to list the total number of physical processors on a system.</p> <p>Use the <code>psrinfo -pv</code> command to list all physical processors on a system and the virtual processors that are associated with each physical processor.</p>	“How to Display a System's Physical Processor Type” on page 22

Commands That Are Used to Display System Information

TABLE 1-1 Commands for Displaying System Information

Command	System Information Displayed	Man Page
<code>date</code>	Date and time	date(1)
<code>hostid</code>	Host ID number	hostid(1)
<code>isainfo</code>	The number of bits supported by <i>native</i> applications on the running system, which can be passed as a token to scripts	isainfo(1)
<code>isalist</code>	Processor type	isalist(1)
<code>prtconf</code>	System configuration information, installed memory, device properties, and product name	prtconf(1M)
<code>prtdiag</code>	System configuration and diagnostic information, including any failed field replacement units (FRUs)	prtdiag(1M)
<code>psrinfo</code>	Processor information	psrinfo(1M)
<code>uname</code>	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
```

▼ How to Display the Date and Time

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

Example 1-1 Displaying the Date and Time

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

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

▼ How to Display a System's Host ID Number

- To display the host ID number in a numeric (hexadecimal) format, use the `hostid` command.

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

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

```
$ hostid
80a5d34c
```

Displaying a System's Architecture and Processor Type

The following examples show the output of the `isainfo` and `isalist` commands when run on x86 and SPARC based systems.

EXAMPLE 1-3 Displaying a System's Architecture Type

The following examples show how to display the architecture type and names of the native instruction sets for applications that are supported by the current operating system.

The following output is from an x86 based system:

```
$ isainfo
amd64 i386
```

The following output is from a SPARC based system:

```
$ isainfo
sparcv9 sparc
```

The `isainfo -v` command displays 32-bit and 64-bit application support. For example, the following output is from a SPARC based system:

EXAMPLE 1-3 Displaying a System's Architecture Type *(Continued)*

```
$ isainfo -v
64-bit sparcv9 applications
    asi_blk_init
32-bit sparc applications
    asi_blk_init v8plus div32 mul32
#
```

This example shows the output of the `isainfo -v` command from an x86 based system:

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

See the [isainfo\(1\)](#) man page.

EXAMPLE 1-4 Displaying a System's Processor Type

The following 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
```

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

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

See the [isalist\(1\)](#) man page.

▼ How to Display a System's Product Name

The `-b` option to the `prtconf` command enables you to display the product name of a system. For more information, 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 1-5 SPARC: Displaying a System's Product Name

This example shows sample output from the `prtconf -b` command on a SPARC based system.

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

This example shows sample output from the `prtconf -vb` command on a SPARC based system.

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

▼ 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 1-6 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: 523776 Megabytes
```

▼ How to Display Default and Customized Property Values for a Device

To display both the default and customized property values for devices, use the `prtconf` command with the `-u` option. For more information about this option, see the [prtconf\(1M\)](#) man page.

- Display the default and customized properties of a `driver.conf` file.

```
$ prtconf -u
```

The output of the `prtconf -u` command displays the default and customized properties for all of the drivers that are on the system.

Example 1-7 SPARC: Displaying Default and Custom Device Properties

This example shows the default and custom properties for the `bge.conf` file. Note that vendor-provided configuration files are located in the `/kernel` and `/platform` directories, while the corresponding modified driver configuration files are located in the `/etc/driver/drv` directory.


```

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

ORCL,SPARC-T3-4
  scsi_vhci, instance #0
    disk, instance #4
    disk, instance #5
    disk, instance #6
    disk, instance #8
    disk, instance #9
    disk, instance #10
    disk, instance #11
    disk, instance #12
  packages (driver not attached)
    SUNW,builtin-drivers (driver not attached)
    deblocker (driver not attached)
    disk-label (driver not attached)
    terminal-emulator (driver not attached)
    dropins (driver not attached)
    SUNW,asr (driver not attached)
    kbd-translator (driver not attached)
    obp-tftp (driver not attached)
    zfs-file-system (driver not attached)
    hsfs-file-system (driver not attached)
  chosen (driver not attached)
  openprom (driver not attached)
    client-services (driver not attached)
  options, instance #0
  aliases (driver not attached)
  memory (driver not attached)
  virtual-memory (driver not attached)
  iscsi-hba (driver not attached)
    disk, instance #0 (driver not attached)
  virtual-devices, instance #0
    flashprom (driver not attached)
    tpm, instance #0 (driver not attached)
    n2cp, instance #0
    ncp, instance #0
    random-number-generator, instance #0
    console, instance #0
    channel-devices, instance #0
      virtual-channel, instance #0
      virtual-channel, instance #1
      virtual-channel-client, instance #2
      virtual-channel-client, instance #3
      virtual-domain-service, instance #0
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)
  cpu (driver not attached)

```

Example 1-8 x86: Displaying Default and Custom Device Properties

This example shows the default and custom properties for the `bge.conf` file. Note that vendor-provided configuration files are located in the `/kernel` and `/platform` directories, while the corresponding modified driver configuration files are located in the `/etc/driver/drv` directory.

```
$ prtconf -u
System Configuration: Oracle Corporation i86pc
Memory size: 8192 Megabytes
System Peripherals (Software Nodes):

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

```

        pci8086,1011, instance #1
        pci1000,3060, instance #0
            sd, instance #1
            sd, instance #2
        pci1022,7459 (driver not attached)
ioapics (driver not attached)
    ioapic, instance #0 (driver not attached)
    ioapic, instance #1 (driver not attached)
fw, instance #0
    cpu (driver not attached)
    cpu (driver not attached)
    cpu (driver not attached)
    cpu (driver not attached)
    sb, instance #1
used-resources (driver not attached)
iscsi, instance #0
fcoe, instance #0
pseudo, instance #0
options, instance #0
xsvc, instance #0
vga_arbiter, instance #0

```

Example 1–9 x86: Displaying System Configuration Information

The following example shows how to use the `prtconf` command with the `-v` option on an x86 based system to identify which disk, tape, and DVD devices are connected to a system. The output of this command displays the driver not attached messages next to the device instances. This message usually indicates that no device exists at that device instance.

```

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

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

```

See Also For more information, see the `driver(4)`, `driver.conf(4)`, and `prtconf(1M)` man pages.

For instructions on how to create administratively provided configuration files, see [Chapter 3, “Managing Devices \(Overview/Tasks\)”](#) in *Oracle Solaris 11.1 Administration: Devices and File Systems*.

▼ How to Display System Diagnostic Information

- Display configuration and diagnostic information for a system.

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

-v Verbose mode.

-l Log output. If failures or errors exist in the system, output this information to `syslogd(1M)` only.

Example 1–10 SPARC: Displaying System Diagnostic Information

The following example shows the output for the `prtdiag -v` command on a SPARC based system. For the sake of brevity, the example has been truncated.

```
$ prtdiag -v | more
System Configuration: Oracle Corporation sun4v Sun Fire T200
Memory size: 16256 Megabytes

===== Virtual CPUs =====

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

===== Physical Memory Configuration =====
Segment Table:
-----
Base      Segment  Interleave  Bank  Contains
Address   Size     Factor     Size  Modules
-----
0x0      16 GB    4          2 GB  MB/CMP0/CH0/R0/D0
          2 GB    4          2 GB  MB/CMP0/CH0/R0/D1
          2 GB    4          2 GB  MB/CMP0/CH0/R1/D0
          2 GB    4          2 GB  MB/CMP0/CH0/R1/D1
          2 GB    4          2 GB  MB/CMP0/CH1/R0/D0
```

```

                2 GB      MB/CMP0/CH1/R0/D1
                MB/CMP0/CH1/R1/D0
.
.
System PROM revisions:
-----
OBP 4.30.4.d 2011/07/06 14:29

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

```

Example 1-11 x86: Displaying System Diagnostic Information

The following example shows the output for the `prtdiag -l` command on an x86 based system.

```

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

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

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

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

==== Upgradeable Slots =====
ID  Status  Type          Description
-----

```

```

1  available PCI Express      PCIExp SLOT0
2  available PCI Express      PCIExp SLOT1
3  available PCI-X           PCIX SLOT2
4  available PCI Express      PCIExp SLOT3
5  available PCI Express      PCIExp SLOT4
$

```

Identifying Information About 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 22.

▼ 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. For example:

```

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

```

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

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

▼ How to Display a System's Virtual Processor Type

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

```
$ psrinfo -v
```

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

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

Example 1–12 SPARC: Displaying a System's Virtual Processor Type

This example shows how to display information about a SPARC based system's virtual 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 1–13 SPARC: Displaying the Virtual Processor That Is Associated With Each Physical Processor on a System

The following example shows the output of the `psrinfo` command, when run with the `-pv` options on an Oracle SPARC T4-4 server. The output displays both the chip (physical processor) and the core information about the thread location. This information can be helpful in determining which physical CPU a thread is on, and how it is mapped at the core level.

```
$ psrinfo -pv
The physical processor has 8 cores and 64 virtual processors (0-63)
```

```

The core has 8 virtual processors (0-7)
The core has 8 virtual processors (8-15)
The core has 8 virtual processors (16-23)
The core has 8 virtual processors (24-31)
The core has 8 virtual processors (32-39)
The core has 8 virtual processors (40-47)
The core has 8 virtual processors (48-55)
The core has 8 virtual processors (56-63)
    SPARC-T4 (chipid 0, clock 2998 MHz)
The physical processor has 8 cores and 64 virtual processors (64-127)
The core has 8 virtual processors (64-71)
The core has 8 virtual processors (72-79)
The core has 8 virtual processors (80-87)
The core has 8 virtual processors (88-95)
The core has 8 virtual processors (96-103)
The core has 8 virtual processors (104-111)
The core has 8 virtual processors (112-119)
The core has 8 virtual processors (120-127)
    SPARC-T4 (chipid 1, clock 2998 MHz)
    
```

Changing System Information

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

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 <code>date mmddHHMM[[cc]yy]</code> command-line syntax.	“How to Manually Set a System's Date and Time” on page 24
Set up a message-of-the-day.	Set up a message-of-the-day on your system by editing the <code>/etc/motd</code> file.	“How to Set Up a Message-Of-The-Day” on page 25
Change a system's identity.	Change a system's identity by using the <code>hostname</code> command.	“How to Change a System's Identity” on page 26

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

1 Become an administrator.

See [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: 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 1–14 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

You can 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 Assume the root role or a role that has the Administrator Message Edit profile assigned to it.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

2 Use the `pfedit` command to edit the `/etc/motd` file and add a message of your choice.

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

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.
```

▼ **How to Change a System's Identity**

1 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

2 Set the name of the host for the system.

```
# hostname mynodename
```

Managing System Processes (Tasks)

This chapter describes the procedures for managing system processes.

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

- “What's New in Managing System Processes” on page 27
- “Managing System Processes” on page 28
- “Managing Process Class Information” on page 38
- “Troubleshooting Problems With System Processes” on page 45

What's New in Managing System Processes

The following features for managing system processes are new or changed in this release.

Pseudo System Processes

Oracle Solaris 10 and Oracle Solaris 11 releases include system processes that perform a specific task, but do not require any administration.

Process	Description
<code>fsflush</code>	System daemon that flushes pages to disk
<code>init</code>	Initial system process that starts and restarts other processes and SMF components
<code>intrd</code>	System process that monitors and balances system load due to interrupts
<code>kmem_task</code>	System process that monitors memory cache sizes
<code>pageout</code>	System process that controls memory paging to disk

Process	Description
sched	System process that is responsible for OS scheduling and process swapping
vm_tasks	System process with one thread per processor that balances and distributes virtual memory related workloads across CPUs for better performance.
zpool - pool-name	System process for each ZFS storage pool containing the I/O taskq threads for the associated pool

Managing System Processes

This section describes the various tasks for managing system processes.

Managing System Processes (Task Map)

Task	Description	For Instructions
List processes.	Use the <code>ps</code> command to list all the processes on a system.	“How to List Processes” on page 32
Display information about processes.	Use the <code>pgrep</code> command to obtain the process IDs for processes that you want to display more information about.	“How to Display Information About Processes” on page 33
Control processes.	Locate processes by using the <code>pgrep</code> command. Then, use the appropriate <code>pcommand (/proc)</code> to control the process. See Table 2–3 for a description of the <code>(/proc)</code> commands.	“How to Control Processes” on page 34
Kill a process.	Locate a process, either by process name or process ID. You can use either the <code>pkill</code> or <code>kill</code> commands to terminate the process.	“How to Terminate a Process (pkill)” on page 35 “How to Terminate a Process (kill)” on page 36

Commands for Managing System Processes

The following table describes the commands for managing system processes.

TABLE 2-1 Commands for Managing Processes

Command	Description	Man Page
<code>ps</code> , <code>pgrep</code> , <code>prstat</code> , <code>pkill</code>	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)
<code>pkill</code>	Functions identically to <code>pgrep</code> but finds or signals processes by name or other attribute and terminates the process. Each matching process is signaled as if by the <code>kill</code> command, instead of having its process ID printed.	pgrep(1) , and pkill(1) kill(1)
<code>pargs</code> , <code>preap</code>	Assists with processes debugging.	pargs(1) , and preap(1)
<code>dispadmin</code>	Lists default process scheduling policies.	dispadmin(1M)
<code>priocntl</code>	Assigns processes to a priority class and manages process priorities.	priocntl(1)
<code>nice</code>	Changes the priority of a timesharing process.	nice(1)
<code>psrset</code>	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 2-2 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 utilization for scheduling. This field is not displayed when the <code>-c</code> 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 <code>-c</code> 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 <code>proc</code> 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.
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 2-3 Process Commands (`/proc`)

Process Command	Description
<code>pcrred</code>	Displays process credential information

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

You can display detailed, technical information about processes or control active processes by using some of the process commands. [Table 2-3](#) 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 2, “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 2-1 Listing Processes

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

```
$ ps
  PID TTY          TIME CMD
 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   18:04:04 ?        0:15 sched
root  5    0     0   18:04:03 ?        0:05 zpool-rpool
root  1    0     0   18:04:05 ?        0:00 /sbin/init
root  2    0     0   18:04:05 ?        0:00 pageout
root  3    0     0   18:04:05 ?        2:52 fsflush
root  6    0     0   18:04:05 ?        0:02 vmtasks
daemon 739  1     0   19:03:58 ?        0:00 /usr/lib/nfs/nfs4cbd
```



```

root      9      1      0 18:04:06 ?           0:14 /lib/svc/bin/svc.startd
root     11      1      0 18:04:06 ?           0:45 /lib/svc/bin/svc.configd
daemon   559     1      0 18:04:49 ?           0:00 /usr/sbin/rpcbind
netcfg   47      1      0 18:04:19 ?           0:01 /lib/inet/netcfgd
dladm    44      1      0 18:04:17 ?           0:00 /sbin/dlmgmt
netadm   51      1      0 18:04:22 ?           0:01 /lib/inet/ipmgmt
root    372     338     0 18:04:43 ?           0:00 /usr/lib/hal/hald-addon-cpufreq
root     67      1      0 18:04:30 ?           0:02 /lib/inet/in.mpathd
root    141     1      0 18:04:38 ?           0:00 /usr/lib/pfexecd
netadm   89      1      0 18:04:31 ?           0:03 /lib/inet/nwamd
root    602     1      0 18:04:50 ?           0:02 /usr/lib/inet/inetd start
root    131     1      0 18:04:35 ?           0:01 /sbin/dhcpagent
daemon  119     1      0 18:04:33 ?           0:00 /lib/crypto/kcfd
root    333     1      0 18:04:41 ?           0:07 /usr/lib/hal/hald --daemon=yes
root    370     338     0 18:04:43 ?           0:00 /usr/lib/hal/hald-addon-network-discovery
root    159     1      0 18:04:39 ?           0:00 /usr/lib/sysevent/syseventd
root    236     1      0 18:04:40 ?           0:00 /usr/lib/ldoms/drd
root    535     1      0 18:04:46 ?           0:09 /usr/sbin/nscd
root    305     1      0 18:04:40 ?           0:00 /usr/lib/zones/zonestatd
root    326     1      0 18:04:41 ?           0:03 /usr/lib/devfsadm/devfsadm
root    314     1      0 18:04:40 ?           0:00 /usr/lib/dbus-daemon --system

```

▼ 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 2–3](#) lists and describes these commands.

pid Identifies the process ID.

Example 2–2 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 2–3](#) lists and describes these commands.

pid Identifies the process ID.

3 Verify the process status.

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

Terminating a Process (`kill`, `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 `kill(1)` and `kill(1)` man pages.

▼ How to Terminate a Process (`kill`)

- 1 To terminate the process of another user, assume the root role.
- 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.

```
$ kill [signal] process
```

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.

process Is the name of the process to stop.

Tip – When using the `kill` 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 `kill` 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, assume the root role.

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 2-3 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
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

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*.

Managing Process Class Information (Task Map)

Task	Description	For Instructions
Display basic information about process classes.	Use the <code>priocntl -l</code> command to display process scheduling classes and priority ranges.	“How to Display Basic Information About Process Classes (priocntl)” on page 39
Display the global priority of a process.	Use the <code>ps -ecl</code> command to display the global priority of a process.	“How to Display the Global Priority of a Process” on page 40
Designate a process priority.	Start a process with a designated priority by using the <code>priocntl -e -c</code> command.	“How to Designate a Process Priority (priocntl)” on page 41
Change scheduling parameters of a timesharing process.	Use the <code>priocntl -s -m</code> command to change scheduling parameters in a timesharing process.	“How to Change Scheduling Parameters of a Timesharing Process (priocntl)” on page 42
Change the class of a process.	Use the <code>priocntl -s -c</code> command to change the class of a process.	“How to Change the Class of a Process (priocntl)” on page 43

Task	Description	For Instructions
Change the priority of a process.	Use the <code>/usr/bin/nice</code> command with the appropriate options to lower or raise the priority of a process.	“How to Change the Priority of a Process (<code>nice</code>)” on page 44

Changing the Scheduling Priority of Processes (`prionctl`)

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 `prionctl` command to assign processes to a priority class and to manage process priorities. For instructions on using the `prionctl` command to manage processes, see [“How to Designate a Process Priority \(`prionctl`\)” on page 41](#).

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

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

```
$ prionctl -l
```

Example 2-4 Displaying Basic Information About Process Classes (`prionctl`)

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

```
# prionctl -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 2-5 Displaying the Global Priority of a Process

The following example shows `ps -ecl` command output. The values in the PRI column show the priority for each process.

```
$ ps -ecl
 F S      UID    PID  PPID  CLS  PRI      ADDR      SZ      WCHAN  TTY      TIME  CMD
 1 T      0      0      0  SYS  96      ?      0      ?      ?      0:11  sched
 1 S      0      5      0  SDC  99      ?      0      ? ?    ? ?    0:01  zpool-rp
 0 S      0      1      0  TS   59      ?     688      ? ?    ? ?    0:00  init
 1 S      0      2      0  SYS  98      ?      0      ? ?    ? ?    0:00  pageout
 1 S      0      3      0  SYS  60      ?      0      ? ?    ? ?    2:31  fsflush
 1 S      0      6      0  SDC  99      ?      0      ? ?    ? ?    0:00  vmtasks
 0 S     16     56      1  TS   59      ?    1026      ? ?    ? ?    0:01  ipmgmt
 0 S      0      9      1  TS   59      ?    3480      ? ?    ? ?    0:04  svc.star
 0 S      0     11      1  TS   59      ?    3480      ? ?    ? ?    0:13  svc.conf
 0 S      0    162      1  TS   59      ?     533      ? ?    ? ?    0:00  pfexecd
 0 S      0   1738   1730  TS   59      ?     817      ? pts/ 1  0:00  bash
 0 S      1     852      1  TS   59      ?     851      ? ?    ? ?    0:17  rpcbind
 0 S     17     43      1  TS   59      ?    1096      ? ?    ? ?    0:01  netcfgd
 0 S     15     47      1  TS   59      ?     765      ? ?    ? ?    0:00  dlmgmt
 0 S      0     68      1  TS   59      ?     694      ? ?    ? ?    0:01  in.mpath
 0 S      1   1220      1  FX   60      ?     682      ? ?    ? ?    0:00  nfs4cbd
 0 S     16     89      1  TS   59      ?    1673      ? ?    ? ?    0:02  nwamd
 0 S      0    146      1  TS   59      ?     629      ? ?    ? ?    0:01  dhcpgen
 0 S      1    129      1  TS   59      ?    1843      ? ?    ? ?    0:00  kcf
 0 S      1   1215      1  FX   60      ?     738      ? ?    ? ?    0:00  lockd
 0 S      0    829    828  TS   59      ?     968      ? ?    ? ?    0:00  hald-run
 0 S      0    361      1  TS   59      ?    1081      ? ?    ? ?    0:01  devfsadm
 0 S      0    879      1  TS   59      ?    1166      ? ?    ? ?    0:01  inetd
 0 0 119764  1773   880  TS   59      ?     557      ? cons ole 0:00  ps
 0 S      0    844    829  TS   59      ?     996      ? ?    ? ?    0:00  hald-add
 0 S      0    895   866  TS   59      ?     590      ? ?    ? ?    0:00  ttymon
 0 S      0    840      1  TS   59      ?     495      ? ?    ? ?    0:00  cron
 0 S      0    874      1  TS   59      ?     425      ? ?    ? ?    0:00  utmpd
 0 S      0   1724   956  TS   59      ?    2215      ? ?    ? ?    0:00  sshd
 0 S 119764  880      9  TS   59      ?     565      ? cons ole 0:00  csh
 0 S      0    210      1  TS   59      ?    1622      ? ?    ? ?    0:00  sysevent
 0 S      0    279      1  TS   59      ?     472      ? ?    ? ?    0:00  iscsid
 0 S      1   1221      1  TS   59      ?    1349      ? ?    ? ?    0:00  nfsmapid
 1 S      0    374      0  SDC  99      ?      0      ? ?    ? ?    0:00  zpool-us
 0 S      0   1207      1  TS   59      ?    1063      ? ?    ? ?    0:00  rmvolmgr
 0 S      0    828      1  TS   59      ?    1776      ? ?    ? ?    0:03  hald
 0 S      0    853    829  TS   59      ?     896      ? ?    ? ?    0:02  hald-add
 0 S      0    373      1  TS   59      ?     985      ? ?    ? ?    0:00  picld
 0 S      0    299      1  TS   59      ?     836      ? ?    ? ?    0:00  dbus-dae
 0 S   12524  1730  1725  TS   59      ?     452      ? pts/ 1  0:00  csh
```



```

0 S      0  370      1 TS  59      ?  574      ? ?      0:00 powerd
0 S      0  264      1 FX  60      ?  637      ? ?      0:00 zonestat
0 S      0  866      9 TS  59      ?  555      ? ?      0:00 sac
0 S      0  851     829 TS  59      ?  998      ? ?      0:00 halD-add
0 S 12524 1725 1724 TS  59      ? 2732      ? ?      0:00 sshd
0 S      1 1211      1 TS  59      ?  783      ? ?      0:00 statd
0 S      0 1046      1 TS  59      ? 1770      ? ?      0:13 intrd
0 S      0  889      1 TS  59      ? 1063      ? ?      0:00 syslogd
0 S      0 1209      1 TS  59      ?  792      ? ?      0:00 in.ndpd
0 S      0 1188 1186 TS  59      ?  951      ? ?      0:15 automoun
0 S      0 1172  829 TS  59      ?  725      ? ?      0:00 halD-add
0 S      0 1186      1 TS  59      ?  692      ? ?      0:00 automoun
0 S     101 1739 1738 TS  59      ?  817      ? pts/ 1      0:00 bash
0 S      0 1199      1 TS  59      ? 1495      ? ?      0:02 sendmail
0 S      0  956      1 TS  59      ? 1729      ? ?      0:00 sshd
0 S      25 1192      1 TS  59      ? 1528      ? ?      0:00 sendmail
0 S      0  934      1 TS  59      ? 6897      ? ?      0:14 fmd
0 S      0 1131      1 TS  59      ? 1691      ? ?      0:07 nscd
0 S      1 1181      1 TS  59      ?  699      ? ?      0:00 ypbind

```

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

1 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

2 Start a process with a designated priority.

```
# prctl -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 2-6 Designating a Process Priority (`prionctl`)

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

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

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

1 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

2 Change the scheduling parameters of a running timesharing process.

```
# prionctl -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 2-7 Changing Scheduling Parameters of a Timesharing Process (`prionctl`)

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.

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

▼ How to Change the Class of a Process (priosctl)

1 (Optional) Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

2 Change the class of a process.

```
# priosctl -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 <i>xidtype</i> and <i>xidlist</i> to identify the process or processes. The <i>xidtype</i> specifies the type of ID, such as the process ID or user ID. Use <i>xidlist</i> to identify a list of process IDs or user IDs.

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

3 Verify the process status.

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

Example 2-8 Changing the Class of a Process (priosctl)

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

```
# priosctl -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 `priosctl` 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 `prionctl` 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, the root role can raise or lower the priority of a process.

- 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.

Monitoring System Performance (Tasks)

Achieving good performance from a computer or network is an important part of system administration. This chapter provides overview of some factors that contribute to managing the performance of the computer systems in your care. In addition, this chapter describes procedures for monitoring system performance by using the `vmstat`, `iostat`, `df`, and `sar` commands.

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

- “Where to Find System Performance Tasks” on page 47
- “System Performance and System Resources” on page 48
- “Processes and System Performance” on page 48
- “About Monitoring System Performance” on page 50
- “Displaying System Performance Information” on page 51
- “Monitoring System Activities” on page 59

Where to Find System Performance Tasks

System Performance Task	For More Information
Manage processes	Chapter 2, “Managing System Processes (Tasks)”
Monitor system performance	Chapter 3, “Monitoring System Performance (Tasks)”
Change tunable parameters	<i>Oracle Solaris 11.1 Tunable Parameters Reference Manual</i>
Manage system performance tasks	Chapter 2, “Projects and Tasks (Overview),” in <i>Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management</i>

System Performance Task	For More Information
Manage processes with FX and FS schedulers	Chapter 8, “Fair Share Scheduler (Overview),” in <i>Oracle Solaris Administration: Oracle Solaris Zones, Oracle Solaris 10 Zones, and Resource Management</i>

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 3, “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 3-1 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.

TABLE 3-1 Process Terminology (Continued)

Term	Description
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.

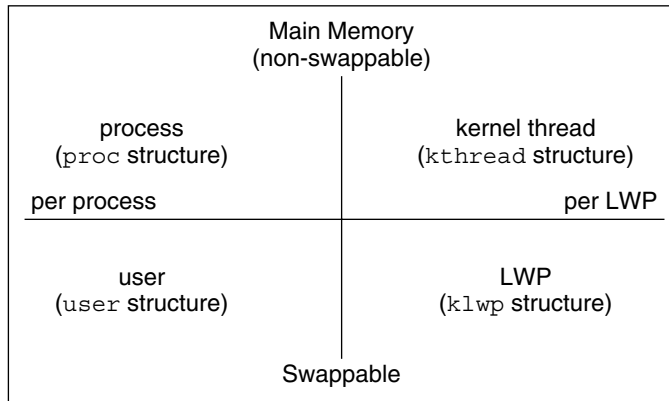
A process can consist of multiple LWPs and multiple application threads. The kernel schedules a kernel-thread structure, which is the scheduling entity in the SunOS environment. Various process structures are described in the following table.

TABLE 3-2 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 3-1 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.

TABLE 3-3 Performance Monitoring Tools

Command	Description	For More Information
<code>cpustat</code> and <code>cputrack</code> commands	Monitors performance of a system or a process using CPU performance counters.	cpustat(1M) and cputrack(1)
<code>netstat</code> and <code>nfsstat</code> commands	Displays information about network performance.	netstat(1M) and nfsstat(1M)
<code>ps</code> and <code>prstat</code> commands	Displays information about active processes.	Chapter 2, “Managing System Processes (Tasks)”
<code>sar</code> and <code>sadc</code> commands	Collects and reports on system activity data.	Chapter 3, “Monitoring System Performance (Tasks)”
<code>swap</code> command	Displays information about available swap space on your system.	Chapter 16, “Configuring Additional Swap Space (Tasks),” in <i>Oracle Solaris 11.1 Administration: Devices and File Systems</i>
<code>vmstat</code> and <code>iostat</code> commands	Summarizes system activity data, such as virtual memory statistics, disk usage, and CPU activity.	Chapter 3, “Monitoring System Performance (Tasks)”
<code>kstat</code> and <code>mpstat</code> commands	Examines the available kernel statistics, or <code>kstats</code> , on the system and reports those statistics which match the criteria specified on the command line. The <code>mpstat</code> command reports processor statistics in tabular form.	kstat(1M) and mpstat(1M) man pages.

Displaying System Performance Information

This section describes the tasks for monitoring displaying system performance information.

Displaying System Performance Information (Task Map)

Task	Description	For Instructions
Display virtual memory Statistics.	Collect virtual memory statistics by using the <code>vmstat</code> command.	“How to Display Virtual Memory Statistics (<code>vmstat</code>)” on page 53

Task	Description	For Instructions
Display system event information.	Display system event information by using the <code>vmstat</code> command with the <code>-s</code> option.	“How to Display System Event Information (<code>vmstat -s</code>)” on page 54
Display swapping statistics.	Use the <code>vmstat</code> command with the <code>-S</code> option to display swapping statistics.	“How to Display Swapping Statistics (<code>vmstat -S</code>)” on page 54
Display interrupts per device.	Use the <code>vmstat</code> command with the <code>-i</code> option to show the number of interrupts per device.	“How to Display Interrupts Per Device (<code>vmstat -i</code>)” on page 55
Display disk utilization.	Use the <code>iostat</code> command to report disk input and output statistics.	“How to Display Disk Utilization Information (<code>iostat</code>)” on page 56
Display extended disk statistics.	Use the <code>iostat</code> command with the <code>-xtc</code> option to display extended disk statistics.	“How to Display Extended Disk Statistics (<code>iostat -xtc</code>)” on page 57
Display disk space information.	The <code>df -k</code> command displays disk space information in Kbytes.	“How to Display Disk Space Information (<code>df -k</code>)” on page 58

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.

TABLE 3-4 Output From the `vmstat` Command

Category	Field Name	Description
procs		Reports on the following:
	<code>r</code>	The number of kernel threads in the dispatch queue
	<code>b</code>	The number of blocked kernel threads that are waiting for resources
	<code>w</code>	The number of swapped out LWPs that are waiting for processing resources to finish
memory		Reports on usage of real memory and virtual memory:
	<code>swap</code>	Available swap space
	<code>free</code>	Size of the free list

TABLE 3-4 Output From the `vmstat` Command (Continued)

Category	Field Name	Description
page		Reports on page faults and paging activity, in units per second:
	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 <code>sr</code> 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 3-1 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 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 3-4](#).

- 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 3-2 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 3-3 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	
	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

Device Type	Field Name	Description
	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
device      r/s    w/s    kr/s    kw/s  wait  actv  svc_t  %w  %b  tty  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  0  100
sd0         0.0    0.0    0.4    0.4  0.0  0.0  49.5   0  0   0   0   0   0  0  0  0
sd6         0.0    0.0    0.0    0.0  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   0   0   0   0  0  0  0
nfs49       0.0    0.0    0.0    0.0  0.0  0.0  15.1   0  0   0   0   0   0  0  0  0
nfs53       0.0    0.0    0.4    0.0  0.0  0.0  24.5   0  0   0   0   0   0  0  0  0
nfs54       0.0    0.0    0.0    0.0  0.0  0.0   6.3   0  0   0   0   0   0  0  0  0
nfs55       0.0    0.0    0.0    0.0  0.0  0.0   4.9   0  0   0   0   0   0  0  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 3-4 Displaying File System Information

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

```
$ df -k
Filesystem          1024-blocks    Used  Available Capacity  Mounted on
rpool/ROOT/solaris-161 191987712    6004395  140577816    5%      /
/devices              0            0          0    0%      /devices
/dev                  0            0          0    0%      /dev
ctfs                  0            0          0    0%      /system/contract
proc                  0            0          0    0%      /proc
mnttab                0            0          0    0%      /etc/mnttab
swap                  4184236      496      4183740    1%      /system/volatile
objfs                 0            0          0    0%      /system/object
sharefs               0            0          0    0%      /etc/dfs/sharetab
/usr/lib/libc/libc_hwcapi.so.1 146582211    6004395  140577816    5%      /lib/libc.so.1
fd                    0            0          0    0%      /dev/fd
swap                  4183784      60        4183724    1%      /tmp
rpool/export          191987712     35      140577816    1%      /export
rpool/export/home     191987712     32      140577816    1%      /export/home
rpool/export/home/123 191987712    13108813 140577816    9%      /export/home/123
rpool/export/repo     191987712    11187204 140577816    8%      /export/repo
rpool/export/repo2010_11 191987712     31      140577816    1%      /export/repo2010_11
rpool                 191987712    5238974 140577816    4%      /rpool
/export/home/123     153686630    13108813 140577816    9%      /home/123
```

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

Example 3-5 Displaying File System Information by Using the `df` Command Without Any Options

When the `df` command is used without operands or options, it reports on all mounted file systems, as shown in the following example:

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

Monitoring System Activities

This section describes tasks for monitoring system activities.

Monitoring System Activities (Task Map)

Task	Description	For Instructions
Check file access.	Display file access operation status by using the <code>sar</code> command with the <code>-a</code> option.	“How to Check File Access (sar -a)” on page 61
Check buffer activity.	Display buffer activity statistics by using the <code>sar</code> command with the <code>-b</code> option.	“How to Check Buffer Activity (sar -b)” on page 62
Check system call statistics.	Display system call statistics by using the <code>sar</code> command with the <code>-c</code> option.	“How to Check System Call Statistics (sar -c)” on page 63
Check disk activity.	Check disk activity by using the <code>sar</code> command with the <code>-d</code> option.	“How to Check Disk Activity (sar -d)” on page 65
Check page-out and memory.	Use the <code>sar</code> command with the <code>-g</code> option to display page-out memory freeing activities.	“How to Check Page-Out and Memory (sar -g)” on page 66
Check kernel memory allocation.	The kernel memory allocation (KMA) allows a kernel subsystem to allocate and free memory, as needed. Use the <code>sar</code> command with the <code>-k</code> option to check KMA.	“How to Check Kernel Memory Allocation (sar -k)” on page 68
Check interprocess communication.	Use the <code>sar</code> command with the <code>-m</code> option to report interprocess communication activities.	“How to Check Interprocess Communication (sar -m)” on page 69
Check page-in activity.	Use the <code>sar</code> command with the <code>-p</code> option to report page-in activity.	“How to Check Page-In Activity (sar -p)” on page 70
Check queue activity.	Use the <code>sar</code> command with the <code>-q</code> option to check the following: <ul style="list-style-type: none"> ■ Average queue length while queue is occupied ■ Percentage of time that the queue is occupied 	“How to Check Queue Activity (sar -q)” on page 71
Check unused memory.	Use the <code>sar</code> command with the <code>-r</code> 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 72
Check CPU utilization.	Use the <code>sar</code> command with the <code>-u</code> option to display CPU utilization statistics.	“How to Check CPU Utilization (sar -u)” on page 73
Check system table status.	Use the <code>sar</code> command with the <code>-v</code> option to report status on the following system tables: <ul style="list-style-type: none"> ■ Process ■ Inode ■ File ■ Shared memory record 	“How to Check System Table Status (sar -v)” on page 74

Task	Description	For Instructions
Check swapping activity.	Use the <code>sar</code> command with the <code>-w</code> option to check swapping activity.	“How to Check Swapping Activity (sar -w)” on page 75
Check terminal activity.	Use the <code>sar</code> command with the <code>-y</code> option to monitor terminal device activity.	“How to Check Terminal Activity (sar -y)” on page 76
Check overall system performance.	The <code>sar -A</code> command displays statistics from all options to provide overall system performance information.	“How to Check Overall System Performance (sar -A)” on page 77
Set up automatic data collection.	To set up your system to collect data automatically and to run the <code>sar</code> commands, do the following: <ul style="list-style-type: none"> ▪ Run the <code>svcadm enable system/sar:default</code> command ▪ Edit the <code>/var/spool/cron/crontabs/sys</code> file 	“How to Set Up Automatic Data Collection” on page 80

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 77](#).

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

```
00:00:00 iget/s namei/s dirbk/s
01:00:00      0        3        0
02:00:00      0        3        0
03:00:00      0        3        0
04:00:00      0        3        0
05:00:00      0        3        0
06:00:00      0        3        0
07:00:00      0        3        0
08:00:00      0        3        0
```

```

08:20:01      0      3      0
08:40:00      0      3      0
09:00:00      0      3      0
09:20:01      0     10      0
09:40:01      0      1      0
10:00:02      0      5      0

Average      0      4      0

```

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 `iget/s` 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 3-6 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 t2k-brm-24 5.10 Generic_144500-10 sun4v  ...

00:00:04 bread/s lread/s %rcache bwrit/s lwrit/s %wcache pread/s pwrit/s
01:00:00      0      0     100      0      0      94      0      0
02:00:01      0      0     100      0      0      94      0      0
03:00:00      0      0     100      0      0      92      0      0
04:00:00      0      1     100      0      1      94      0      0
05:00:00      0      0     100      0      0      93      0      0

```

06:00:00	0	0	100	0	0	93	0	0
07:00:00	0	0	100	0	0	93	0	0
08:00:00	0	0	100	0	0	93	0	0
08:20:00	0	1	100	0	1	94	0	0
08:40:01	0	1	100	0	1	93	0	0
09:00:00	0	1	100	0	1	93	0	0
09:20:00	0	1	100	0	1	93	0	0
09:40:00	0	2	100	0	1	89	0	0
10:00:00	0	9	100	0	5	92	0	0
10:20:00	0	0	100	0	0	68	0	0
10:40:00	0	1	98	0	1	70	0	0
11:00:00	0	1	100	0	1	75	0	0
Average	0	1	100	0	1	91	0	0

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

Field Name	Description
<code>bread/s</code>	Average number of reads per second that are submitted to the buffer cache from the disk
<code>lread/s</code>	Average number of logical reads per second from the buffer cache
<code>%rcache</code>	Fraction of logical reads that are found in the buffer cache (100 % minus the ratio of <code>bread/s</code> to <code>lread/s</code>)
<code>bwrit/s</code>	Average number of physical blocks (512 bytes) that are written from the buffer cache to disk, per second
<code>lwrit/s</code>	Average number of logical writes to the buffer cache, per second
<code>%wcache</code>	Fraction of logical writes that are found in the buffer cache (100 % minus the ratio of <code>bwrit/s</code> to <code>lwrit/s</code>)
<code>pread/s</code>	Average number of physical reads, per second, that use character device interfaces
<code>pwrit/s</code>	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 3-7 Checking System Call Statistics (sar -c)

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

```
$ sar -c
```

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

Average      302      66      49      0.02      0.01      57842      55544
```

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
<code>scall/s</code>	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.
<code>sread/s</code>	The number of read system calls per second.
<code>swrit/s</code>	The number of write system calls per second.
<code>fork/s</code>	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.
<code>exec/s</code>	The number of exec system calls per second. If <code>exec/s</code> divided by <code>fork/s</code> is greater than 3, look for inefficient PATH variables.
<code>rchar/s</code>	The number of characters (bytes) transferred by read system calls per second.

Field Name	Description
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  await  avserv
```

Example 3-8 Checking Disk Activity

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

```
$ sar -d
```

```
SunOS balmy 5.10 Generic_144500-10 sun4v  ...
```

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

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.

Field Name	Description
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.
await	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 3–9 Checking Page-Out and Memory (sar -g)

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

```
$ sar -g
SunOS balmy 5.10 Generic_144500-10 sun4v ...
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
<code>pgout/s</code>	The number of page-out requests per second.
<code>ppgout/s</code>	The actual number of pages that are paged-out, per second. A single page-out request might involve paging-out multiple pages.
<code>pgfree/s</code>	The number of pages, per second, that are placed on the free list.
<code>pgscan/s</code>	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.
<code>%ufs_ipf</code>	The percentage of <code>ufs</code> inodes taken off the free list by <code>iget</code> 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 <code>igets</code> with page flushes. A high value indicates that the free list of inodes is page-bound, and that the number of <code>ufs</code> 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 3–10 Checking Kernel Memory Allocation (`sar -k`)

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

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

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

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.
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 3-11 Checking Interprocess Communication (sar -m)

The following abbreviated example shows output from the `sar -m` command.

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

```

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

Average     0.00    0.00

```

▼ 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 3-12 Checking Page-In Activity (sar -p)

The following example shows output from the `sar -p` command.

```

$ sar -p

SunOS balmy 5.10 Generic_144500-10 sun4v    ...

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

Average   0.42    0.22    0.31    1.45    4.00    0.00

```

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.
ppgin/s	The number of pages paged in, per second. A single page-in request, such as a soft-lock request (see <code>slock/s</code>) 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	The average number of swapped out processes.
%swpocc	The percentage of time in which the processes are swapped out.

Example 3–13 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 balmy 5.10 Generic_144500-10 sun4v ...

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

Average      1.0      7      0.0      0
```

▼ 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 3–14 Checking Unused Memory (sar -r)

The following example shows output from the `sar -r` command.

```
$ sar -r

SunOS balmy 5.10 Generic_144500-10 sun4v ...

00:00:04 freemem freeswap
01:00:00      44717      1715062
02:00:01      44733      1715496
03:00:00      44715      1714746
04:00:00      44751      1715403
05:00:00      44784      1714743
06:00:00      44794      1715186
07:00:00      44793      1715159
```



```

08:00:00  44786  1714914
08:20:00  44805  1715576
08:40:01  44797  1715347
09:00:00  44761  1713948
09:20:00  44802  1715478
09:40:00  41770  1682239
10:00:00  35401  1610833
10:20:00  34295  1599141
10:40:00  33943  1598425
11:00:00  30500  1561959

Average   43312  1699242

```

▼ 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 3–15 Checking CPU Utilization (sar -u)

The following example shows output from the `sar -u` command.

```

$ sar -u
00:00:04  %usr  %sys  %wio  %idle
01:00:00      0     0     0    100
02:00:01      0     0     0    100
03:00:00      0     0     0    100
04:00:00      0     0     0    100
05:00:00      0     0     0    100
06:00:00      0     0     0    100
07:00:00      0     0     0    100
08:00:00      0     0     0    100
08:20:00      0     0     0     99

```

08:40:01	0	0	0	99
09:00:00	0	0	0	99
09:20:00	0	0	0	99
09:40:00	4	1	0	95
10:00:00	4	2	0	94
10:20:00	1	1	0	98
10:40:00	18	3	0	79
11:00:00	25	3	0	72
Average	2	0	0	98

▼ 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 3-16 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
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.pgswpin) / 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 3–17 Checking Swap Activity (sar -w)

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

```

$ sar -w

00:00:04 swpin/s bswin/s swpot/s bswoT/s pswch/s
01:00:00 0.00 0.0 0.00 0.0 132
02:00:01 0.00 0.0 0.00 0.0 133
03:00:00 0.00 0.0 0.00 0.0 133
04:00:00 0.00 0.0 0.00 0.0 134
05:00:00 0.00 0.0 0.00 0.0 133
06:00:00 0.00 0.0 0.00 0.0 133
07:00:00 0.00 0.0 0.00 0.0 132
08:00:00 0.00 0.0 0.00 0.0 131
08:20:00 0.00 0.0 0.00 0.0 133
08:40:01 0.00 0.0 0.00 0.0 132
09:00:00 0.00 0.0 0.00 0.0 132
09:20:00 0.00 0.0 0.00 0.0 132
09:40:00 0.00 0.0 0.00 0.0 335
10:00:00 0.00 0.0 0.00 0.0 601
10:20:00 0.00 0.0 0.00 0.0 353
10:40:00 0.00 0.0 0.00 0.0 747
11:00:00 0.00 0.0 0.00 0.0 804

Average 0.00 0.0 0.00 0.0 198

```

▼ 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 3–18 Checking Terminal Activity (sar -y)

The following example shows output from the `sar -y` command.

```

$ sar -y

00:00:04 rawch/s  canch/s  outch/s  rcvin/s  xmtin/s  mdmin/s
01:00:00      0       0       0       0       0       0
02:00:01      0       0       0       0       0       0
03:00:00      0       0       0       0       0       0
04:00:00      0       0       0       0       0       0
05:00:00      0       0       0       0       0       0
06:00:00      0       0       0       0       0       0
07:00:00      0       0       0       0       0       0
08:00:00      0       0       0       0       0       0
08:20:00      0       0       0       0       0       0
08:40:01      0       0       0       0       0       0
09:00:00      0       0       0       0       0       0
09:20:00      0       0       0       0       0       0
09:40:00      0       0       1       0       0       0
10:00:00      0       0      37       0       0       0
10:20:00      0       0       0       0       0       0
10:40:00      0       0       3       0       0       0
11:00:00      0       0       3       0       0       0

Average      0       0       1       0       0       0

```

▼ 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 `sadd`, 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/sadd` 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/sardd` 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 [-aAbcdgkmpqruvw] [-o file] t [n]
```

```
sar [-aAbcdgkmpqruvw] [-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/sadd`, 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 3-5 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
-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

TABLE 3-5 Options for the sar Command (Continued)

Option	Actions
-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 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

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.

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`

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

- [“Ways to Automatically Executing System Tasks” on page 81](#)
- [“Scheduling System Tasks” on page 83](#)
- [“Scheduling Tasks by Using the `at` Command” on page 93](#)

Ways to Automatically Executing 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 4-1 Command Summary: Scheduling System Tasks

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

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 87](#).

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 94](#).

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 95](#).

Scheduling System Tasks

This section includes tasks for scheduling system tasks by using `crontab` files.

Creating and Editing `crontab` Files (Task Map)

Task	Description	For Instructions
Create or edit a <code>crontab</code> file.	Use the <code>crontab -e</code> command to create or edit a <code>crontab</code> file.	“How to Create or Edit a <code>crontab</code> File” on page 87
Verify that a <code>crontab</code> file exists.	Use the <code>ls -l</code> command to verify the contents of the <code>/var/spool/cron/crontabs</code> file.	“How to Verify That a <code>crontab</code> File Exists” on page 88
Display a <code>crontab</code> file.	Use the <code>ls -l</code> command to display the <code>crontab</code> file.	“How to Display a <code>crontab</code> File” on page 88
Remove a <code>crontab</code> file.	The <code>crontab</code> file is set up with restrictive permissions. Use the <code>crontab -r</code> command, rather than the <code>rm</code> command to remove a <code>crontab</code> file.	“How to Remove a <code>crontab</code> File” on page 89
Deny <code>crontab</code> access.	To deny users access to <code>crontab</code> commands, add user names to the <code>/etc/cron.d/cron.deny</code> file by editing this file.	“How to Deny <code>crontab</code> Command Access” on page 91

Task	Description	For Instructions
Limit crontab access to specified users.	To allow users access to the crontab command, add user names to the <code>/etc/cron.d/cron.allow</code> file.	“How to Limit crontab Command Access to Specified Users” on page 92

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 85](#).

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 4-2 Default crontab Files

crontab File	Function
adm	Accounting
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 4-3 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 root 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 assume the root role. See [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

You do not need to assume the root role 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 85](#). The crontab file will be placed in the `/var/spool/cron/crontabs` directory.

3 Verify your crontab file changes.

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

Example 4-1 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
```

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 88.

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 Assume the root role to display a crontab file that belongs to root or another user. See [“How to Use Your Assigned Administrative Rights”](#) in *Oracle Solaris 11.1 Administration: Security Services*.

You do not need to assume the root role 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 4-2 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 4-3 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 4-4 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 Assume the root role to remove a crontab file that belongs to root or another user. Roles contain authorizations and privileged commands. See [“How to Use Your Assigned Administrative Rights”](#) in *Oracle Solaris 11.1 Administration: Security Services*.

You do not need to assume the root role 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 4–5 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   root    smith   sys     uucp
$ crontab -r
$ ls /var/spool/cron/crontabs
adm   jones   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 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

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

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 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

2 Create the `/etc/cron.d/cron.allow` file.

3 Add the root role 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 4-6 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`, and `smith` are the only users who can access the crontab command.

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

Scheduling Tasks by Using the at Command

This section includes tasks for scheduling system tasks by using the at command.

Using the at Command (Task Map)

Task	Description	For Instructions
Create an at job.	Use the at command to do the following: <ul style="list-style-type: none"> ▪ 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 95
Display the at queue.	User the atq command to display the at queue.	“How to Display the at Queue” on page 96

Task	Description	For Instructions
Verify an at job.	Use the <code>atq</code> 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 96
Display at jobs.	Use the <code>at -l [job-id]</code> command to display at jobs that have been submitted to the queue.	“How to Display at Jobs” on page 96
Remove at jobs.	Use the <code>at -r [job-id]</code> command to remove at jobs from the queue.	“How to Remove at Jobs” on page 97
Deny access to the at command.	To deny users access to the <code>at</code> command, edit the <code>/etc/cron.d/at.deny</code> file.	“How to Deny Access to the at Command” on page 98

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 4-7 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 4-8 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 Assume the root role to remove an at job that belongs to root or another user. Roles contain authorizations and privileged commands. See [“How to Use Your Assigned Administrative Rights” in Oracle Solaris 11.1 Administration: Security Services](#).

You do not need to assume the root role 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 4-9 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 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

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 4–10 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.

Managing the System Console, Terminal Devices, and Power Services (Tasks)

This chapter describes how to manage the system console and locally connected terminal devices through the `ttymon` program and system power services.

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

- [“What's New in Managing the System Console, Terminal Devices, and Power Services” on page 101](#)
- [“Managing System Console and Locally Connected Terminal Devices” on page 102](#)
- [“Managing System Power Services” on page 105](#)

What's New in Managing the System Console, Terminal Devices, and Power Services

The following features are new or changed in Oracle Solaris 11.

Changes to How System Power Services Are Managed

Power management configuration has moved into an SMF configuration repository. The `poweradm` command is used to manage system power management properties directly rather than using a combination of power-related command, daemon, and configuration file. These changes are part of a wider set of changes to modernize the power management framework in Oracle Solaris 11. For more information, see [“Managing System Power Services” on page 105](#).

Managing System Console and Locally Connected Terminal Devices

The system console is a terminal that has special attributes and is used for certain purposes. For example, kernel messages that are meant for an administrator are sent to the Console and not other terminals.

A terminal is a means of interacting with Oracle Solaris. 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.

A terminal could also be associated with the physical monitor and keyboard layout of a computer. What sets the graphical terminal apart is that it must be associated with the graphics card and monitor of a computer. So, instead of transmitting characters out of a serial port, the characters are drawn onto the memory of the graphics card that is in the computer.

SMF Services That Manage the System Console and Locally Connected Terminal Devices

The system console and locally connected terminal devices are represented as instances of the SMF service, `svc:/system/console`. This service defines most of the behavior, with each instance having specific overrides to the settings that are inherited from the service. The `ttymon` program is used to offer login services for these terminals. Each terminal uses a separate instance of the `ttymon` program. Command-line arguments that are passed by the service to the `ttymon` program govern its behavior.

The service instances that are supplied with the system are as follows:

- `svc:/system/console-login:default`
The default instance always represents that the `ttymon` program offer a login to the system hardware console. For an example, see [“How to Modify Settings for the System Console” on page 103](#).
- `svc:/system/console-login:{vt2, vt3, vt4, vt5, vt6}`
Additional service instances are provided for the system's virtual consoles. If virtual consoles are not available, these services are automatically disabled. For more information, see the `vtdaemon(1M)` man page.
- `svc:/system/console-login:{terma, termb}`
The `svc:/system/console-login:terma` and `svc:/system/console-login:termb` services are provided as a convenience. These services can assist you in setting up login services for additional `/dev/term/a` and `/dev/term/b` ports. These services are *disabled* by default.

You can define additional service instances as part of the `svc:system/console-login` service. For example, if you had a `/dev/term/f` device which you needed to support, you could instantiate `svc:/system/console-login:termf` and configure it appropriately.

▼ How to Modify Settings for the System Console

Administration of the system console is managed by SMF. Use the `svccfg` command to set the system console properties.

The following procedure shows how to change the console terminal type by using the `svccfg` command.

1 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

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

2 Use the `svccfg` command to set the property for the service instance that you want to change.

For example, to change the terminal type for the system console, which is represented by the `:default` service, you would type the following command:

```
# svccfg -s svc:/system/console-login:default "setprop ttymon/terminal_type = xterm"
```



Caution – It is not advisable to set the terminal type of the `svc:/system/console-login` service because the change will affect *all* instances.

▼ How to Set Up Login Services on Auxiliary Terminals

For terminals that are connected to `/dev/term/a` or `/dev/term/b` serial ports on a system, predefined services are provided.

To enable login services for `/dev/term/a`, use the following procedure.

1 Assume the root role.

See “How to Use Your Assigned Administrative Rights” in *Oracle Solaris 11.1 Administration: Security Services*.

2 Enable the service instance as follows:

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

3 Check that the service is online.

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

The output should show that the service is online. If the service is in maintenance mode, consult the service's log file for further details.

▼ How to Set the Baud Rate Speed on the System Terminal

This procedure shows how to set the baud rate speed on the console. 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.

See “[How to Use Your Assigned Administrative Rights](#)” in *Oracle Solaris 11.1 Administration: 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 version of the `options.conf` file that is in the `/etc/driver/drv` directory.

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.

Managing System Power Services

In the Oracle Solaris 11 release, power management configuration has moved into an SMF configuration repository. The new `poweradm` command is used to manage system power management properties directly rather than using a combination of power-related command, daemon, and configuration file. These changes are part of a wider set of changes to modernize the power management framework in Oracle Solaris 11.

The following power management features are no longer available:

- `/etc/power.conf`
- `pmconfig` and `powerd`
- Device power management

The following properties describe power management components:

- `administrative-authority` – Defines the source of administrative control for Oracle Solaris power management. This property can be set to `none`, `platform` (default value), or `smf`. When set to `platform`, the values of `time-to-full-capacity` and `time-to-minimum-responsiveness` are taken from the platform's power management commands. When set to `smf`, the values of `time-to-full-capacity` and `time-to-minimum-responsiveness` are taken from SMF. If you attempt to set `time-to-full-capacity` or `time-to-minimum-responsiveness` from either a platform command or an SMF service property when in the opposite venue, the value is ignored.

When `administrative-authority` is set to `none`, power management within the Solaris instance is turned off,

- `time-to-full-capacity` – Defines the maximum time (in microseconds) the system is allowed to reach its full capacity, from any lower-capacity or less-responsive state, while the system is in active state. The maximum time includes when the system is allowed to reach (re-provision and make available) its full capacity, returning from any lower-capacity/less-responsive state, while it has been using any or all of the PM features falling within this boundary.

By default, this value is taken from the platform, `i86pc` for example, because the default setting for `administrative-authority` is set to `platform`.

Alternatively, if `administrative-authority` is set to `smf`, this value is taken from the definition provided by the SMF power service. At installation time, this value is undefined. If you choose to modify this property, a value appropriate to the needs of the system's workload or applications should be considered.

- `time-to-minimum-responsiveness` – Defines how long the system is allowed to return to its active state in milliseconds. This parameter provides the minimum capacity required to meet the `time-to-full-capacity` constraint. By default, this parameter value is taken from the platform, `i86pc` for example, because the default setting for `administrative-authority` is set to `platform`.

Alternatively, if `administrative-authority` is set to `smf`, this value is taken from the definition provided by the SMF power service. At installation time, this value is undefined. If you choose to modify this property, a value appropriate to the needs of the system's workload or applications should be considered.

Moderate values, seconds for example, allow hardware components or subsystems on the platform to be placed in slower-response inactive states. Larger values, 30 seconds to minutes for example, allow for whole system suspension, using techniques such as `suspend-to-RAM`.

- `suspend-enable` – By default, no system running Solaris is permitted to attempt a suspend operation. Setting this property to true permits a suspend operation to be attempted. The value of the `administrative-authority` has no effect upon this property.
- `platform-disabled` – When `platform-disabled` is set to true, the platform has disabled power management. When set to false, the default value, power management is controlled by the value of the above properties.

A brief summary of power management status can be displayed by using the following command:

```
$ /usr/sbin/poweradm show
Power management is enabled with the hardware platform as the authority:
time-to-full-capacity set to 250 microseconds
time-to-minimum-responsiveness set to 0 milliseconds
```

All power management properties can be displayed by using the following command:

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

In the above output, the `active_control/administrative-authority` indicates the source of the configuration with two settings:

- `platform` – Configuration for power management comes from the platform. This is the default value.
- `smf` – Allows the other power management properties to be set using the `poweradm` command.

The `platform-disabled` property in the above output indicates that the platform power management is enabled:

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

For more information, see [poweradm\(1M\)](#).

EXAMPLE 5-1 Enabling and Disabling Power Management

If you previously enabled S3-support in the `/etc/power.conf` file to suspend and resume your system, similar `poweradm` syntax is:

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

The `suspend-enable` property is set to `false` by default.

Use the following syntax to disable power management:

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

Disabling the following SMF power management service does not disable power management:

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

Use the following syntax to disable suspend and resume.

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

EXAMPLE 5-2 Setting and Displaying Power Management Parameters

The following example shows how to set `time-to-full-capacity` to 300 microseconds and `time-to-minimum-responsiveness` to 500 milliseconds. Lastly, the Oracle Solaris instance is informed of the new values.

```
# poweradm set time-to-full-capacity=300
# poweradm set time-to-minimum-responsiveness=500
# poweradm set administrative-authority=smf
```

The following command shows the current `time-to-full-capacity` value.

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

The following command retrieves the `time-to-full-capacity` value set by the platform.

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

Note that this value will only be the same as the current value if `administrative-authority` is set to `platform`. For more information, see the above `administrative-authority` property description.

Troubleshooting System Power Problems

▼ How to Recover from Power Service in Maintenance Mode

If `administrative-authority` is set to `smf` before both `time-to-full-capacity` and `time-to-minimum-responsiveness` have been set, the service will go into maintenance mode. See the steps below to recover from this scenario.

1 Become an administrator.

See “[How to Use Your Assigned Administrative Rights](#)” in *Oracle Solaris 11.1 Administration: Security Services*.

2 Set `administrative-authority` to `none`.

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

3 Set both `time-to-full-capacity` and `time-to-minimum-responsiveness` to their desired values.

```
# poweradm set time-to-full-capacity=value  
# poweradm set time-to-minimum-responsiveness=value
```

4 Clear the service.

```
# svcadm clear power
```

5 Set `administrative-authority` to `smf`.

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

Index

A

- address space map, 31
- application threads, 48, 50
- at command, 94, 95, 98
 - l option (list), 97
 - m option (mail), 95, 96
 - automatic scheduling of, 85
 - controlling access to, 95, 98
 - overview, 81
 - denying access, 98
 - error messages, 98
 - overview, 81, 82, 94
- at.deny file, 95, 98
 - description, 81
- at.job files, 94, 97
 - creating, 95, 96
 - deleting, 97
 - description, 82
 - displaying, 97
 - location of, 83
 - submitting, 94
- at.jobs directory, 85
 - description, 81
- automatic system activity data collection, 77, 78
- automatic system activity reporting, 77, 78
- automatic system task execution
 - repetitive tasks, 91, 92
 - single tasks, 94, 95, 98
- automating system task execution, 81

B

- baud rate
 - how to set on `ttymon` terminal, 104
 - how to set with the `eeprom` command, 104
- booting, running `sadc` command when, 78

C

- changing
 - crontab files, 86
 - date, 25
 - priority, 42, 44
 - timesharing processes, 43, 44
 - scheduling classes, 43
- console terminal, how to set the baud rate on, 104
- console terminal baud rate, setting with the `eeprom` command, 104
- controlling
 - access to `at` command, 81, 95, 98
 - access to `crontab` command, 91, 92
 - overview, 81
 - processes, 34
- core files, automatically deleting, 94
- CPU (central processing unit)
 - displaying information on
 - time usage, 30, 45
 - high-usage processes, 45
- creating
 - at jobs, 96
 - at jobs, 95
 - crontab files, 86, 87

- cron.allow file, 90, 91, 92
- cron daemon, 83, 85
- cron.deny file, 90, 91
 - defaults, 90
- crontab command, 91
 - controlling access to, 90, 91, 92
 - denying access, 90, 91
 - limiting access to specific users, 90, 91, 92
 - overview, 81, 90, 91
 - cron daemon and, 85
 - e option (edit), 86, 87
 - l option (list), 88
 - r option (remove), 89, 90
 - daily tasks, 82
 - error messages, 93
 - files used by, 85
 - overview, 81, 82
 - quitting without saving changes, 87
 - scheduling of, 85
- crontab files
 - creating, 86, 87
 - creating and editing, 83–84
 - defaults, 84
 - deleting, 89, 90
 - denying access, 91
 - description, 85
 - displaying, 88
 - editing, 86, 87
 - location of, 84
 - removing, 89–90
 - syntax, 85, 86

D

- daily tasks (scheduling with crontab), 82
- defaults, nice number, 44
- deleting
 - at jobs, 97
 - crontab files, 89, 90
 - log files, 87
 - old/inactive files, 82
- df command, 58
 - k option (kilobytes), 58
 - examples, 58

- df command (*Continued*)
 - overview, 58
- directories
 - current working directory for processes, 31
- disk drives
 - displaying information about
 - free disk space, 58
 - finding and deleting old/inactive files, 87
- disk space
 - displaying information about
 - df command, 58
 - mount point, 59
- dispadm command, overview, 39
- display
 - date and time, 14
 - host ID, 14
 - system's installed memory, 16
- displaying
 - at jobs, 97
 - crontab files, 88
 - linked libraries, 31
 - LWP information, 31
 - priority information, 30, 40
 - process information (how to), 33–34
 - scheduling class information, 30, 39, 40
 - system activity information, 61, 78
 - system information
 - commands for, 14
- displaying a system's physical processor type, psrinfo
 - p, 22–23
- displaying product name information, prtconf
 - command, 15–16

E

- editing
 - crontab files, 86, 87
- eeprom command, using to set the baud rate on the ttymon terminal, 104
- error messages
 - at command, 98
 - crontab command, 93
 - /etc/cron.d/at.deny file, 95, 98
 - /etc/cron.d/cron.allow file, 90, 91, 92

/etc/cron.d/cron.deny file, 90, 91
 executing routine tasks automatically (overview), 81

F

fcntl information, 31, 33
 file systems
 disk space usage, 58
 mount point, 59
 files
 checking access operations, 61, 62
 fstat and fcntl information display, 31, 33
 fsck command, 82
 fstat information, 31, 33

G

global priorities
 defined, 38
 displaying, 40

I

iostat command
 basic information display, 56
 overview, 55

K

kernel thread
 scheduling and, 30
 structures, 30, 49
 killing processes, 31, 35
 klwp structure, 49
 kthread structure, 49

L

listing
 processes, 32

listing (*Continued*)

 processes being executed, 32
 log files, deleting automatically, 87
 LWPs (lightweight processes)
 defined, 48
 displaying information about, 31
 processes and, 48, 49
 structures for, 49

M

maximums, nice number, 44
 memory
 example of displaying information about, 16
 process structures and, 49
 shared
 process virtual memory, 50
 virtual
 process, 50
 message of the day (MOTD) facility, 25–26
 minimums, nice number, 44
 monthly tasks (scheduling with crontab), 82
 MOTD (message of the day) facility, 25–26
 motd file, 25–26

N

new features, svcadm enable system/sar:default
 command, 78
 nice command, 43, 44, 45
 nice number, 30, 44

P

perf file, 78
 performance
 activities that are tracked, 50
 automatic collection of activity data, 77, 78
 file access, 61, 62
 manual collection of activity data, 61, 78
 process management, 31, 44, 48
 reports on, 61

performance (*Continued*)

- system activity monitoring, 50, 61, 77
- tools for monitoring, 50
- pf files command, 31, 33
- pf lags command, 31
- pkill command, 31, 35
- pldd command, 31
- pmap command, 31
- prionctl command
 - overview, 39
 - c option (scheduling class designation), 43
 - i option (ID type), 42, 43
 - l option (scheduling class display), 39
 - m option (max/min priority), 42
 - p option (priority designation), 42
 - s option (priority upper limit/change priority), 42, 43
- priority (process)
 - changing, 42, 44
 - timesharing processes, 42, 43, 44
 - designating, 42
 - displaying information about, 30, 40
 - global
 - defined, 38
 - displaying, 40
 - overview, 38, 44
 - scheduling classes and, 42
 - user-mode priority, 38
- /proc directory, 30
- proc structure, 30, 49
- process file system (PROCFS), 30
- processes

- address space map, 31
- application threads and, 48, 50
- controlling, 34
- current working directory for, 31, 33
- defined, 48
- displaying information (how to), 33–34
- displaying information about
 - prionctl command, 39
- displaying information on, 29
 - listing processes, 32
 - listing processes being executed, 32
 - LWPs, 31

processes, displaying information on (*Continued*)

- prionctl command, 39
- ps command, 29, 32, 40
- displaying information with proc tool
 - commands, 31
- displaying information with proc tools, 30
- fstat and fcntl information for open files, 31, 33
- killing, 31, 35
- libraries linked into, 31
- nice number of, 30, 43, 44, 45
- priority, 44
 - changing, 42, 44
 - changing timesharing process priority, 42, 43, 44
 - designating, 42
 - displaying information about, 30, 40
 - global priorities, 38, 40
 - overview, 38, 44
 - scheduling classes and, 38, 42
 - user-mode priority, 38
- proc tool commands, 30
- restarting, 31
- runaway, 45
- scheduling classes, 38
 - changing, 43
 - changing priority of, 42, 44
 - designating, 42
 - displaying information about, 30, 39
 - displaying information on, 40
 - priority levels and, 38, 42
- signal actions, 31
- stack trace, 31
- stopping temporarily, 31
- structures for, 30, 49
- terminology, 48, 50
- tool commands, 31
- tracing flags, 31
- trees, 31, 33
- troubleshooting, 45
- PROCFS (process file system), 30
- product name for a system, displaying with prtconf command, 15–16
- programs, disk-dependency of, 62
- prtconf command, 16
 - displaying a system's product name, 15–16

ps command, 29, 32
 fields reported, 30
 overview, 29
 -c option (scheduling class), 30, 45
 -ecl option (global priority), 40
 -ef option (full information), 32
 psig command, 31
 psrinfo command option to identify chip
 multithreading features, psrinfo -p, 22
 pstack command, 31
 ptime command, 31
 ptree command, 31, 33
 pwait command, 31
 pwdx command, 31, 33

R

real-time processes, changing class of, 43
 removing, crontab files, 89–90
 repetitive system tasks, 91
 restarting, processes, 31
 runaway processes, 45

S

sa1 command, 77
 sa2 command, 77, 78
 sadc command, 77, 78
 sadd file, 78
 sar command, 61, 78
 description of all options, 79
 options listed, 79
 overview, 61, 78
 -a option (file access), 61, 62
 -A option (overall performance), 77, 79
 -b option (buffers), 62
 -c option (system calls), 64
 -e option (ending time), 79
 -f option (file to extract data from), 79
 -i option (interval), 79
 -m option (interprocess communication), 69
 -p option (page-in/page faults), 70
 -q option (queue), 71, 72

sar command (*Continued*)
 -r option (unused memory), 72
 -s option (starting time), 79
 -u option (CPU usage), 73
 -v option (system tables), 74
 -y option (terminal devices), 76
 scheduling
See also crontab command, atcommand
 one-time system tasks, 82, 94
 repetitive system tasks, 82, 84
 scheduling classes, 38
 changing, 43
 changing priority of, 42, 44
 designating, 42
 displaying information about, 30, 39, 40
 priority levels and, 38, 42
 security
 at command, 95
 crontab command, 91
 setting the baud rate on the ttymon console terminal,
 how to, 104
 shared memory, process virtual memory, 50
 stopping, processes temporarily, 31
 svcadm enable system/sar:default command, 78
 sys crontab, 78
 system activities
 automatic collection of data on, 77, 78
 list of activities tracked, 50
 manual collection of data on, 78
 system resources
 monitoring, 95
 automatic, 95
 overview, 48
 system tasks
See also crontab command, at command
 scheduling
 one-time tasks, 82, 94
 repetitive tasks, 82, 84
 scheduling automatically, 81

T

terminals, process controlling, 30

time

- CPU usage, 30, 45
- processes accumulating large amounts of CPU time, 45

timesharing processes

- changing scheduling parameters, 42
- priority of
 - changing, 42, 43, 44
 - overview, 38
 - range of, 38

tools

- for displaying process information, 30
- process, 31
- system performance monitoring, 50

tracing flags, 31

troubleshooting

- processes, 45

U

user-mode priority, 38

user processes

- changing priority, 43, 44
- priority of, 38

user structure, 49

/usr/proc/bin directory, 30, 31

V

/var/adm/sa/sadd file, 78

/var/spool/cron/atjobs directory, 81, 83, 85

/var/spool/cron/crontabs directory, 84, 85

/var/spool/cron/crontabs/root file, 84

/var/spool/cron/crontabs/sys crontab, 78

vmstat command, overview, 52

W

weekly tasks (scheduling with crontab), 82