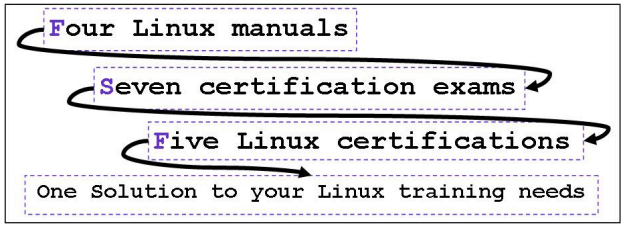
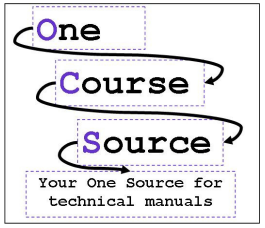
UNIX Administration I

Student Manual



www.onecoursesource.com



Copyright One Course Source, 2008, 2014 ALL RIGHTS RESERVED

This publication contains proprietary and confidential information, which is the property of One Course Source, 2340 Tampa Ave, Suite J, El Cajon, CA 92020. No part of this publication is be reproduced, copied, disclosed, transmitted, stored in a retrieval system or translated into any human or computer language, in any form, by any means, in whole or in part, without the prior express written consent of One Course Source.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

REFERENCES TO CORPORATIONS, THEIR SERVICES AND PRODUCTS, ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. IN NO EVENT SHALL ONE COURSE SOURCE BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, SPECIAL, EXEMPLARY OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS INFORMATION.

Descriptions of, or references to, products or publications within this publication do not imply endorsement of that product or publication. One Course Source makes no warranty of any kind with respect to the subject matter included herein, the products listed herein, or the completeness or accuracy of this publication. One Course Source specifically disclaims all warranties, express, implied or otherwise, including without limitation, all warranties of merchantability and fitness for a particular purpose.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. ONE COURSE SOURCE MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

This notice may not be removed or altered.

ver 3.3 - 4/16/2014

Module Table of Contents

Introduction

Unit One Administration Concepts **Page**

1.1 ..... Brief history of UNIX 19
1.2 ..... Three parts of the OS 21
1.3 ..... Default Shells 22
1.4 ..... Add-on Shells 23
1.5 ..... Terms 24
1.6 ..... Man Pages and Other Documentation 26
1.7 ..... Additional Resources 32
1.8 ..... Summary of Commands and Files 33
1.9 ..... Lab Exercises 34

Unit Two Installation **Page**

2.1 ..... Hardware Requirements for Solaris 36
2.2 ..... Installation Methods 37
2.3 ..... Bringing the System Down 38
2.4 ..... Starting the Installation 40
2.5 ..... Step-by-step Installation Options 41
2.6 ..... Additional Resources 86
2.7 ..... Summary of Commands and Files 87
2.8 ..... Lab Exercises 88

Unit Three Post Installation **Page**

3.1 ..... Configuring the Host Table 90
3.2 ..... Post Install Files 92
3.3 ..... Displaying Disk Usage by Directory 96
3.4 ..... Displaying Disk Usage by Filesystem 97
3.5 ..... Displaying Disk Usage by User 99
3.6 ..... Introduction the Solaris Management Console 100
3.7 ..... Additional Resources 102
3.8 ..... Summary of Commands and Files 103
3.9 ..... Lab Exercises 104

Unit Four The Open Boot PROM **Page**

4.1 ..... What is the Open Boot PROM? 107
4.2 ..... Displaying Basic System Information 111
4.3 ..... Getting Help 112
4.4 ..... Displaying SCSI or IDE Devices 113
4.5 ..... Displaying Device Alias Names 117
4.6 ..... Creating Custom Device Alias Names 120
4.7 ..... Displaying PROM Variables 121
4.8 ..... Setting PROM Variables 123
4.9 ..... Resetting PROM Variables to Factory Preset 126
4.10 ..... Booting from PROM 127
4.11 ..... Recovering from a stop-a 128
4.12 ..... Additional Resources 130
4.13 ..... Summary of Commands and Files 131
4.14 ..... Lab Exercises 132

Unit Five Software administration **Page**

5.1 ..... Displaying Package Information 135
5.2 ..... Removing Packages 138
5.3 ..... Adding Packages via Scripts or the Command Line 140
5.4 ..... Spooling Packages 143
5.5 ..... Installing from Spooled Package 145
5.6 ..... Removing a Spooled Package 146
5.7 ..... The pkgchk Command 147
5.8 ..... Review of the /var/sadm Files 151
5.9 ..... Using admintool to Administer Packages **Error! Bookmark not defined.**
5.10 ..... Additional Resources 152
5.11 ..... Summary of Commands and Files 153
5.12 ..... Lab Exercises 154

Unit Six Patches **Page**

6.1 ..... What are Patches? 156
6.2 ..... Where to Get Patches 157
6.3 ..... Patch Reports 158
6.4 ..... Patch Documentation 159
6.5 ..... Preparing Patches 160
6.6 ..... Installing Patches 161
6.7 ..... The /var/sadm/patch Directory 164
6.8 ..... Displaying installed Patches 165
6.9 ..... Removing Patches 166
6.10 ..... Additional Resources 167
6.11 ..... Summary of Commands and Files 168

6.12 ..... Lab Exercises 169

Unit Seven User Accounts **Page**
7.1 ..... The /etc/passwd File 171
7.2 ..... The /etc/shadow File 173
7.3 ..... The /etc/group File 176
7.4 ..... Using Admintool to Administer Group and User Accounts **Error! Bookmark not defined.**
7.5 ..... Using smuser & smgroup to Administer Group and User Accounts 178
7.6 ..... Using Solaris Management Console to Administer Group & User accounts 182
7.7 ..... Adding Group Accounts via the Command Line 187
7.8 ..... Deleting Group Accounts via the Command Line 188
7.9 ..... Adding User Accounts via the Command Line 189
7.10 ..... Deleting User Accounts via the Command Line 192
7.11 ..... Additional Resources 193
7.12 ..... Summary of Commands and Files 194
7.13 ..... Lab Exercises 195

Unit Eight System Security **Page**

8.1 Password Security 198
8.2 Recording Failed Login Attempts 200
8.3 Shifting to Another Account 201
8.4 User Information 203
8.5 Changing the User Ownership of a File 207
8.6 Changing the Group Ownership of a File 209
8.7 Who is Logged In Now? 210
8.8 Who has Logged In the Past? 211
8.9 Controlling Logging In 212
8.10 Controlling su Attempts 214
8.11 Preventing stop-a 216
8.12 Preventing FTP Access 217
8.13 Additional Resources 218
8.14 Summary of Commands and Files 219
8.15 Lab Exercises 220

Unit Nine Administering Initialization Files **Page**

9.1 What are Initialization Files? 224
9.2 A Review of Common Customizations 225
9.3 Bourne Shell Initialization Files 226
9.4 Korn Shell Initialization Files 229
9.5 C Shell Initialization Files 231
9.6 Testing Initialization Files 233
9.7 Additional Resources 234
9.8 Summary of Commands and Files 235
9.9 Lab Exercises 236

Unit Ten Permissions **Page**

10.1 Basic UNIX Permissions 238
10.2 Setting Default Permissions 244
10.3 Special Permission: setuid 246
10.4 Special Permission: setgid 249
10.5 Special Permission: sticky bit 254
10.6 Additional Resources 256
10.7 Summary of Commands and Files 257
10.8 Lab Exercises 258

Unit Eleven Boot Process **Page**

11.1 Run Levels 262
11.2 Overview of the Boot Phases 263
11.3 The init Phase 266
11.4 Modifying a Run Level 272
11.5 Changing to Different Run Levels 273
11.6 Displaying Boot Messages 276
11.7 Additional Resources 277
11.8 Summary of Commands and Files 278
11.9 Lab Exercises 280

Unit Twelve Directory Hierarchy **Page**

12.1 Physical Device Names 283
12.2 Logical Device Names 285
12.3 The /etc/path_to_inst File 287
12.4 Displaying Devices 288
12.5 Virtual Filesystems vs. Physical Filesystems 289
12.6 Why Have Partitions? 291
12.7 The format Utility 292
12.8 Partitioning with SMC 301
12.9 Filesystem Types 303
12.10 The UNIX Filesystem 305
12.11 Creating a New Filesystem 315
12.12 Adding New Devices 317
12.13 Additional Resources 319
12.14 Summary of Commands and Files 320
12.15 Lab Exercises 321

Unit Thirteen Local Disk Management **Page**

13.1 What is Mounting? 323
13.2 The mount Command 324
13.3 Mount Rules 327
13.4 The umount Command 328
13.5 umount Rules 329
13.6 Mounting Automatically at Boot 332
13.7 The mountall Command 336
13.8 The umountall Command 337
13.9 Review: the df Command 338
13.10 Mounting PCFS and HSFS 339
13.11 Mounting from the CD-ROM 341
13.12 Additional Resources 343
13.13 Summary of Commands and Files 344
13.14 Lab Exercises 345

Unit Fourteen Configuring Filesystems **Page**

14.1 Why Filesystems Break 347
14.2 Fixing Filesystems with fsck 348
14.3 fsck Examples 350
14.4 Additional Resources 354
14.5 Summary of Commands and Files 355
14.6 Lab Exercises 356

Unit Fifteen Administering Processes **Page**

15.1 Listing and Stopping Processes 358
15.2 CDE Process Manager (Solaris 8 and Higher Only) 359
15.3 The prstat Command (Solaris 8 and Higher Only) 361
15.4 The Solaris Management Console (Solaris 9 and Higher Only) 362
15.5 The ps and kill Commands 364
15.6 The pgrep and pkill Commands (Solaris 8 and Higher Only) 368
15.7 The xkill Command 371
15.8 Controlling Job Priority 372
15.9 Using crontab 373
15.10 Using the at Command 379
15.11 Additional Resources 384
15.12 Summary of Commands and Files 385
15.13 Lab Exercises 386

Unit Sixteen Backup and Recovery

		Page
16.1	Why Perform Backups?	390
16.2	Full and Incremental Backups	391
16.3	Logical Tape Device Names	392
16.4	The ufsdump Command	394
16.5	Backing up the / and /usr Filesystems	396
16.6	The ufsrestore Command	397
16.7	The restoresymtable File	399
16.8	Restoring the / Filesystem	400
16.9	The mt Command	401
16.10	The tar Command	402
16.11	Compressing Large Files: compress	407
16.12	Compressing Large Files: gzip	412
16.13	Compressing Large Files: zip	415
16.14	Encoding Binary Files	419
16.15	Emailing Files	421
16.16	UFS Snapshots	422
16.17	Additional Resources	426
16.18	Summary of Commands and Files	428
16.19	Lab Exercises	429

Unit Seventeen Networking

		Page
17.1	Setting up Remote Access Authentication	431
17.2	Remote Login	435
17.3	Remote Shell	436
17.4	Remote Copy	437
17.5	rlogin vs. telnet	438
17.6	Restricting telnet	439
17.7	Review: /etc/default/login	441
17.8	rcp vs. FTP	442
17.9	Restricting FTP	443
17.10	Disabling rlogin, rsh and rcp	446
17.11	Displaying Remote Users	448
17.12	Checking the Status of Machines	449
17.13	Additional Resources	452
17.14	Summary of Commands and Files	453
17.15	Lab Exercises	454

Unit Eighteen Printing

		Page
18.1	Using admintool to Set Up Local Printers	456
18.2	Using admintool to Set Up Network Printers	458
18.3	Printing Files	460
18.4	Monitoring the Print Queue	461
18.5	Canceling Print Jobs	462
18.6	Specifying a Default Printer	463
18.7	Printer Classes	464
18.8	Checking Printer Status	465
18.9	Managing the Print Queue	466
18.10	Printing with Different Priorities	467
18.11	Rejecting Print Jobs	468
18.12	Moving Print Jobs	469
18.13	Recovering from Common Printer Problems	471
18.14	Removing a Printer	472
18.15	Additional Resources	473
18.16	Summary of Commands and Files	475
18.17	Lab Exercises	476

Notes:

Introduction

About this course

This manual was designed with the goal of assisting instructors in their efforts of teaching students to be able to administer a UNIX system.

Typographical syntax

Examples in this text of commands will appear in **bold** text and the output of the commands will appear in *italic* text. The commands and the output of the commands will be placed in a box to separate them from other text. Example:

```
# pwd
/home/student
```

Note: "#" is a **prompt**, a method the shell uses to say “I’m ready for a new command”.

Bold text within a sentence will indicate an important term or a command. Files and directories are highlighted by being placed in `courier` font.

Notes:

Using this manual while in class

In many ways, class manuals are different from textbooks. Textbooks are often filled with lengthy paragraphs that explain a topic in detail. Unfortunately, this style doesn't work well in a classroom environment.

Class manuals often are much more concise than textbooks. Its difficult to follow the instructor's example and read lengthy paragraphs in a book at the same time. For this purpose, class manuals are often more terse.

Lab Exercises

The lab exercises provided in this class are intended to provide practical, hands on experience with a UNIX Operating System. Students are strongly encouraged to perform the labs provided at the end of each Unit to reinforce the knowledge provided in class.

Lab Solutions

The unix_admin.zip file that accompanies this course contains the following:

- All of the answers to the labs provided in the manual

Unit One
Administration Concepts

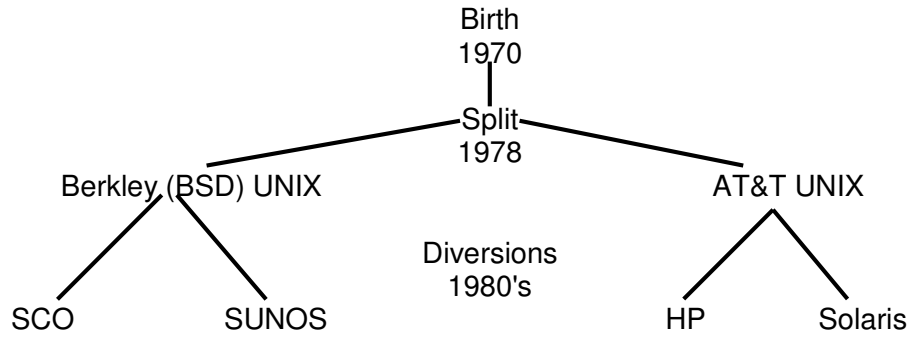
Unit topics

		Page #
1.1	Brief history of UNIX	19
1.2	Three parts of the OS	21
1.3	Default Shells	22
1.4	Add-on Shells	23
1.5	Terms	24
1.6	Man Pages and Other Documentation	26
1.7	Additional Resources	32
1.8	Summary of Commands and Files	33
1.9	Lab Exercises	34

1.1 Brief History of UNIX

Birth of UNIX: UNIX was created in the late 1960's when AT&T (Bell Labs) pulled its research team off another project (MULTICS). Its official birth date is typically placed as January 1, 1970.

Development of UNIX:



The chart above gives a brief overview of the development of UNIX since its birth in 1970.

1970-1978: During these years, AT&T was the primary developer of the UNIX operating system. While other companies developed software or tools that ran on AT&T's UNIX operating system, no other organization developed UNIX itself.

Notes:

1978: In 1978, the University of California at Berkley revised AT&T's latest version of UNIX and released that version (called BSD) to the "public".

1979-present: Since the "big split" several companies have taken either the AT&T version of UNIX or the BSD version of UNIX and modified it to suit their own needs. Many of these companies, including Sun Microsystems, HP and SCO, have released a commercial version of their UNIX operating systems. At present, there are approximately 2 dozen different versions (or **flavors**) of UNIX in the marketplace.

Notes:

1.2 Three Parts of the OS

Kernel: The kernel is a process (program) that controls the operating system. It determines how the user's commands are sent to the machine's hardware, handles input and output devices on the system, controls the boot process and controls system daemons.

Shell: The shell is the primary method for users to tell the kernel what actions to take.

Filesystem: The filesystem is consists on files and directories. It is also sometimes called the directory structure.

Notes:

1.3 Default Shells

The method that users interface with the UNIX operating system is through a **shell**. There are three primary shells available in most UNIX systems: the Bourne shell, the Korn shell and the C shell.

The Bourne shell: The original shell, not often used by the average user anymore due to its limited capabilities. The default shell for the root user is the Bourne Shell.

The Korn shell: A superset of the Bourne shell; All of the commands and syntax (how you type a command) that work in the Bourne shell also work in the Korn shell, but additional features were added.

The C shell: Based upon the C programming language, the C shell has many of the same capabilities as the Korn shell, although the syntax is different.

Notes:

1.4 Add-on Shells

While there are only three shells that come with all versions of Solaris by default, there are other shells that you can install on your system. Here are a couple of the common ones:

tcsh: A superset of the C shell, the tcsh provides additional functionality such as bringing up previous commands for editing via the up arrow key.

bash: The bash shell's syntax and capabilities are very similar to the Korn shell's syntax.

To see a list of available shells for the system, first look at the man page for shells and then look at the `/etc/shells` file.

Notes:

Notes:

1.5 Terms

Host: A host is simply a Sparc station (or any computer, really).

IP Address: An IP (Internet Protocol) address is a "number" given to a machine that is unique to the network. This address consists of four 8-bit (decimal numbers from 0-255) numbers. IP Addresses are broken into "classes":

Class	Network IP	# of Networks	# of Hosts
A	**1-127**.H.H.H	127	16 Million
B	**128-191.N**.H.H	16,384	65,000
C	**192-223.N.N**.H	2,097,152	254

Host name: A name given to a machine that is unique to the network. While machines use IP addresses (or hardware addresses) to communicate with each other, users use host names (which are translated to IP addresses by the system).

Network: A collection of 2 or more Sparc Stations (or other computers) which are physically connected and are able to communicate with each other.

Ethernet address: A host's hardware address.

Swap space: Swap space is space on the hard drive the system can use to store RAM data during times of heavy system use.

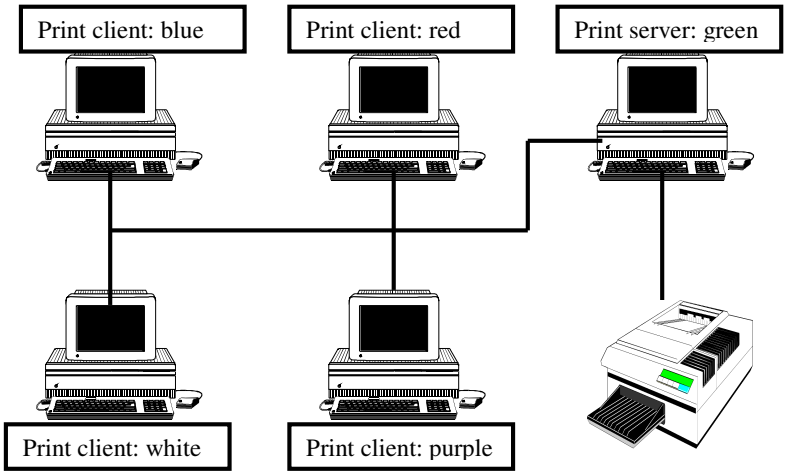
Notes:

Daemons: Daemons are processes that run in the background "waiting" for an activity to take place. Once this activity occurs, the daemon "awakes" and performs a function (ex: the **cron** daemon handles cron jobs).

Servers: These hosts provide other hosts with a service of some sort. This service might be to allow access to the Internet or to provide access to a network printer.

Client: The hosts make use of the services of a server.

An example of a UNIX network environment:



1.6 Man pages and Other Documentation

A critical part of being a System Administrator is being able to find a solution to a problem using documentation. In addition to volumes of textbooks and manuals, there are two excellent (and free!) sources of documentation: man pages and on line documentation.

Man pages

Most UNIX users know about man pages. Initially man pages appear to be clunky and unusable. While the man pages may be difficult to use to begin with, they are an excellent source of information.

Notes:

Notes:

Viewing man pages

To view the man page of a UNIX command, do the following:

```
# man ls
Reformatting page.  Please Wait... done

User Commands                                   ls(1)

NAME
    ls - list contents of directory

SYNOPSIS
    /usr/bin/ls [ -aAbcCdfFgilLmnopqrRstux1 ]  [ file ... ]

    /usr/xpg4/bin/ls [ -aAbcCdfFgilLmnopqrRstux1 ]  [ file ... ]

DESCRIPTION
    For each file that is a directory, ls lists the contents  of
    the  directory;  for  each file that is an ordinary file, ls
    repeats its name and any other  information  requested.  The
    output is sorted alphabetically by default. When no argument
{remaining output omitted}
```

Notes:

Viewing man pages by keywords

Sometimes you want to view a man page but you don't remember the exact name of the command. You can list man pages that match a keyword.

In order to set up this feature, you need to run the **catman** on the system:

```
# catman &
```

The **catman** will generate a database base on keywords. It will also reformat man pages (which will make the viewing of man pages quicker on your system).

After the **catman** command has completed, you can use the **-k** option to view man pages based on keywords:

```
# man -k permissions
check-permissions     check-permissions (1m)  - check permissions on mail rerouting files
chmod                 chmod (1)       - change the permissions mode of a file
fbtab                 logindevperm (4)    - login-based device permissions
logindevperm          logindevperm (4)    - login-based device permissions
uucheck               uucheck (1m)    - check the uucp directories and permissions file
```

Notes:

Man page sections

Man pages are broken into sections. The sections are listed in the following table:

Section	Description
1	Basic user commands
1M	System Admin commands
2	OS Calls
3	Library functions
4	Configuration files
5	Miscellaneous topics
6	Games and demos
7	Hardware and device driver files
9	DDI (Device Driver Interface) and DKI (Driver/Kernel Interface)

By default, if you ask to see a man page, the first match that is found is displayed. Therefore, the command “man passwd” will display the man page for the **passwd** command which exists in section 1.

Notes:

To display man pages from a specific section, use the **-s** option:

```
# man -s 4 passwd

File Formats                    passwd(4)

NAME
    passwd - password file

SYNOPSIS
    /etc/passwd
{remaining output omitted}
```

To *list* all man pages that match, use the **-l** option:

```
# man -l passwd
passwd (1)      -M /usr/share/man
passwd (4)      -M /usr/share/man
```

To *display* all man pages that match, use the **-a** option:

```
# man -a passwd
{output omitted}
```

Notes:

Specifying alternative paths

If you install additional software, you will need to specify to the **man** command where the man pages are for this new software. For example, if you install Perl in `/usr/local`, the man pages will be in `/usr/local/man`. The **man** command needs to be told about this by using one of two methods:

#1. Add the path to the MANPATH variable.

By adding the path to the MANPATH variable, the man command will know to look in this directory as well as the default system directories.

#2. Use the **-M** option.

When you run the **man** command, use the **-M** option:

```
# man -M /usr/local/man perltoc
```

Documentation Web Sites

Site	Contents
docs.sun.com	Excellent documentation on all of Sun's products
sunfreeware.com	Freeware software for Solaris



1.7 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #1

Web sites

None

Man pages

sh	csh	ksh	bash
tcsh	man	catman	

1.8 Summary of Commands and Files

Command	Description
catman	Generates a database to query man pages using a keyword
man	Displays a command's manual

File	Description
/etc/shells	A list of legal shells on the system

1.9 Lab Exercises

There are no lab exercises for Unit 1.

Unit Two
Installation

Unit topics

		Page #
2.1	Hardware Requirements for Solaris	36
2.2	Installation Methods	37
2.3	Bringing the System Down	38
2.4	Starting the Installation	40
2.5	Step-by-step Installation Options	41
2.6	Additional Resources	86
2.7	Summary of Commands and Files	87
2.8	Lab Exercises	88

2.1 Hardware Requirements for Solaris

Device	Requirement
CPU	Sparc or Intel
Hard Drive	2.3 GB or larger
RAM	1 GB or more

Note: Individual versions of Solaris have different hardware requirements. The requirements listed above are for Solaris 10.

2.2 Installation Methods

There are several methods of installing Solaris:

Interactive Installation (Desktop Session) – A GUI-based installation that only installs the OS.

Interactive Installation (Console Session) – A command line installation that only installs the OS.

Jumpstart – An automated installation method.
.

2.3 Bringing the System Down

In order to start the "interactive" installation, you will need to boot off the DVD-ROM. There are two primary methods (and several commands) that you can use to bring the system down to a state in which you can boot off the DVD-ROM: the "clean" method and the "quick and dirty".

The "clean" method

The clean method can be accomplished by several commands (which we will learn in a later Unit). The "clean" method brings down the system gently and prevents disk corruption. The **halt** command is one command that will bring the system down cleanly:

```
# halt
{output omitted}
```

The “quick and dirty” method

The “quick and dirty” method will bring down the system immediately. Because the system is not brought down gently this method will most likely cause disk corruption. This method is necessary in some case (for example, when a system "freezes"). To perform a “quick and dirty” shutdown, hold down the stop button and press the “a” key.

Note: When performing an upgrade, always use the clean method to shut the system down. When performing an initial install (blowing out the entire OS) the “stop-a” method is fine.

2.4 Starting the Installation

For Solaris 10

To start the installation, take the following steps:

#1. Bring the system down to PROM

#2. Place the installation DVD-ROM into the DVD-ROM drive for the “interactive” installation

#3. Type the following command at the boot prompt:

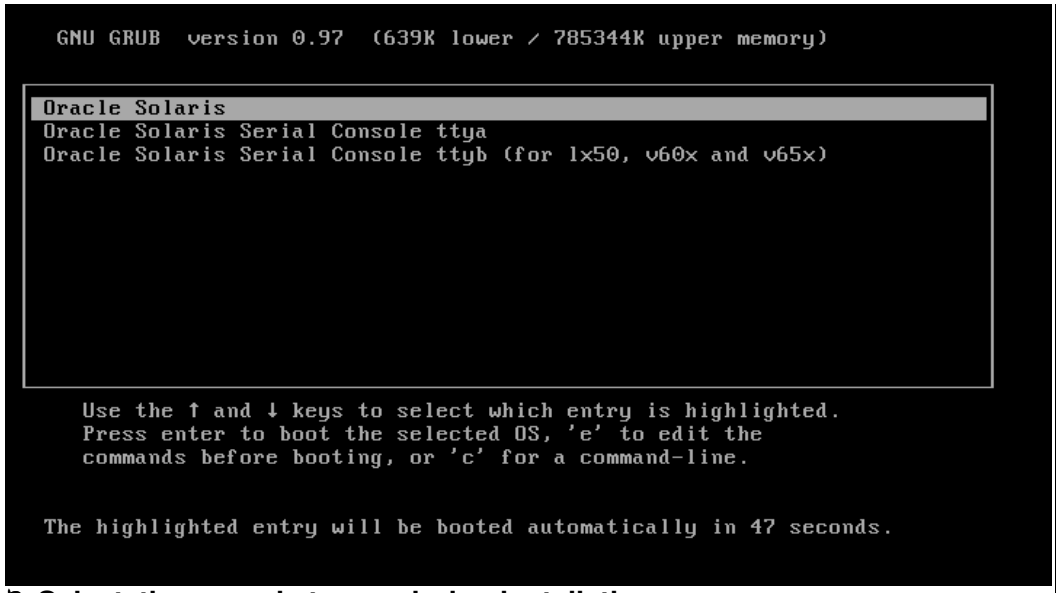
```
ok boot cdrom
```

2.5 Step-by-step Installation Options

Solaris 10 (x86) Installation Steps

Note: The steps to installing Solaris 10 **(x86)** are very similar to installation of other versions of Solaris. While the examples in this guide are specific to the Solaris 10 installations, they can also be applied to previous or later versions of Solaris.

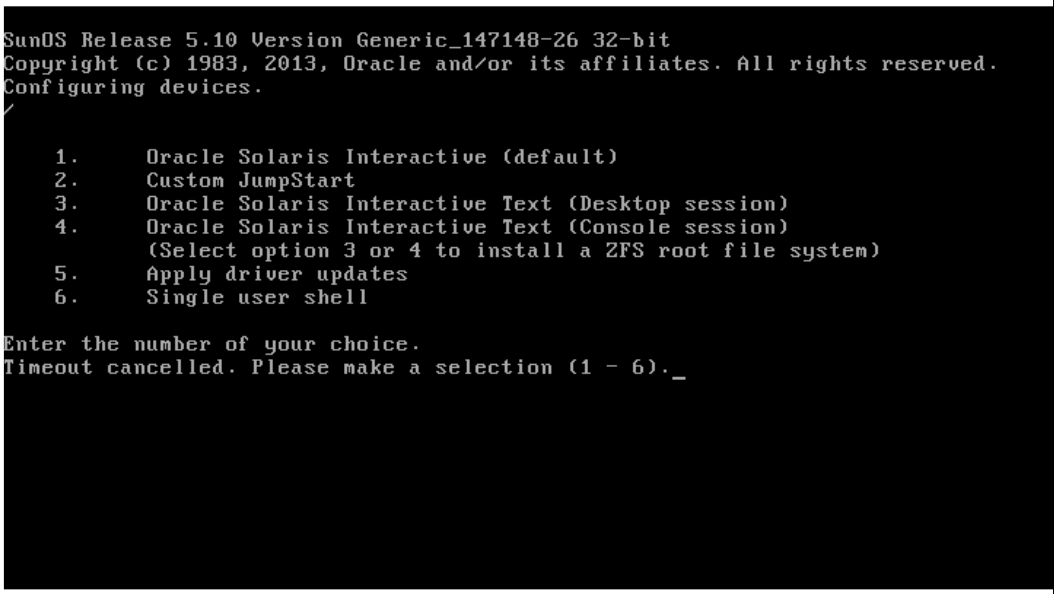
When the installation DVD-ROM boots, the GRUB bootloader appears to allow you to choose whether to boot a normal or serial port console:



Select the console to use during installation:

- It is recommended to use the first item on the menu, the normal console. Press Enter to continue.

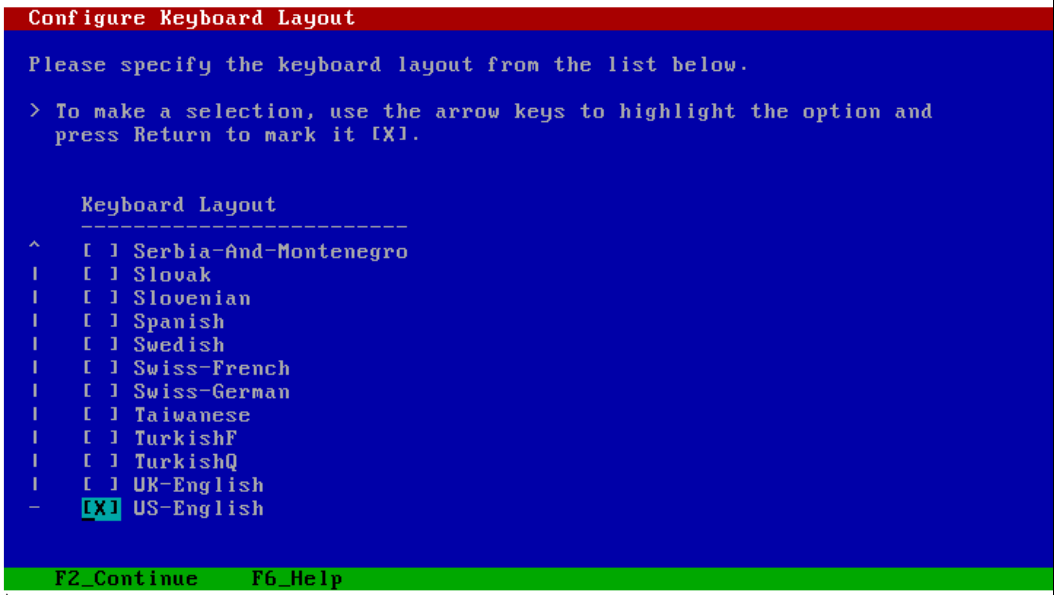
Once you've booted into the installation console, you need to choose the installation method:



Select the method to use to perform the installation:

- It is recommended to use the first item on the menu, the Oracle Solaris Interactive method, which is the same as #3, a GUI Desktop session. Press Enter to continue.

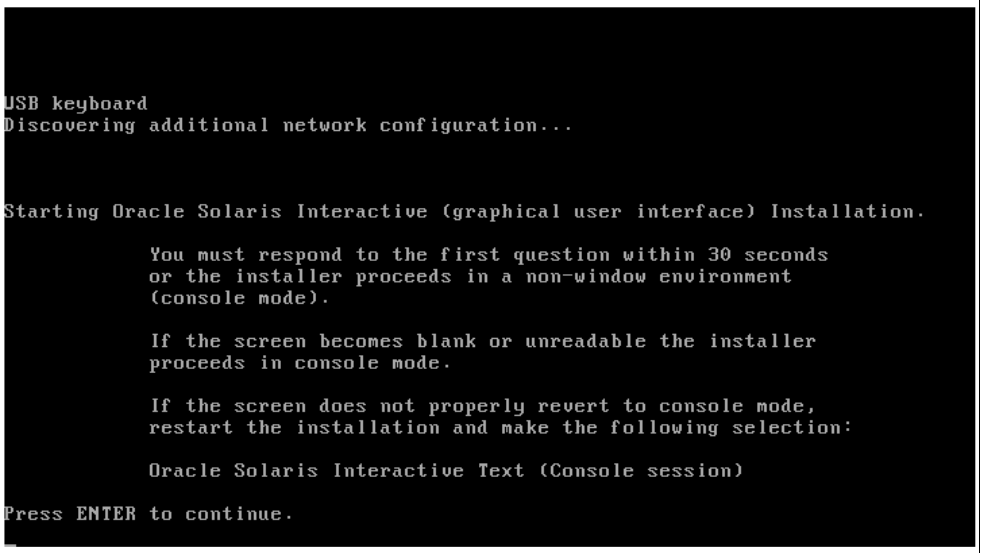
After taking a few moments to load the installer, you will be asked to configure the keyboard:



Configure the Keyboard Layout:

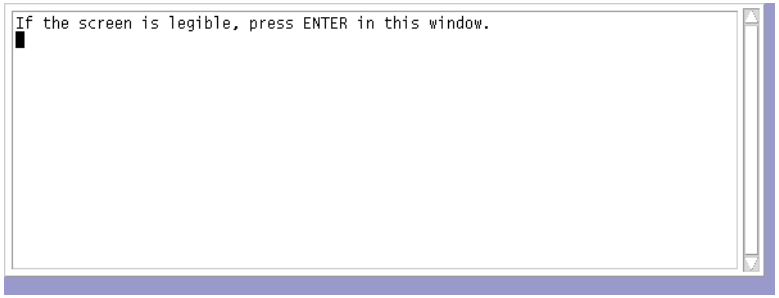
- For users living in the United States, it is recommended to use the default selection: US-English. Press the F2 key to continue.

Next, the installer attempts to discover the hardware of the system, and waits for the user to press Enter to start the Graphical User Interface (GUI):



Once you press Enter, be prepared to answer the next question within 30 seconds or else the installer will revert to a command line only installation. The allows the installer to be able to continue, even if it is unable to start the GUI.

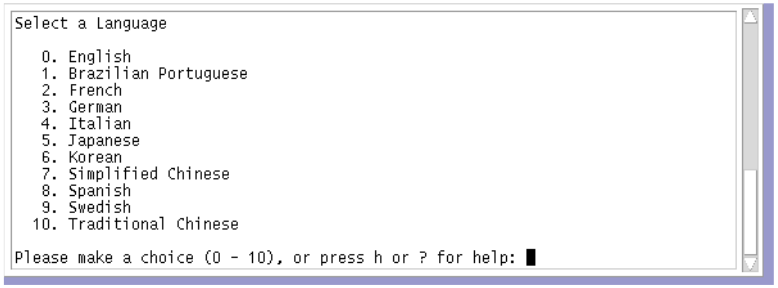
If the GUI is able to start, you will see dialog like this in the upper corner of the screen:



If the screen is readable, confirm that fact:

- You must have the mouse pointer over the window and press ENTER

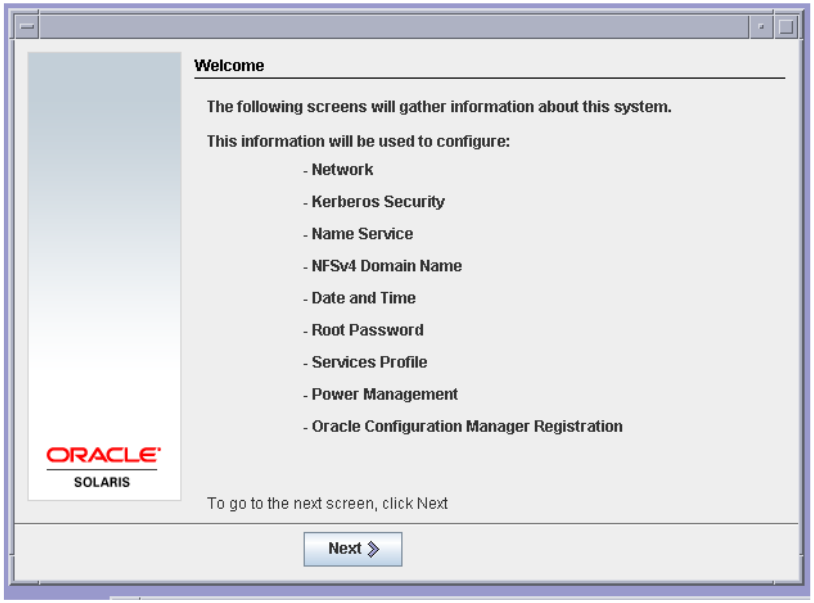
In the next dialog, you choose the language to use during the installation process and for the resulting installation:



Select a language to use:

- You must have the mouse pointer over the window, type "0" for English and press ENTER

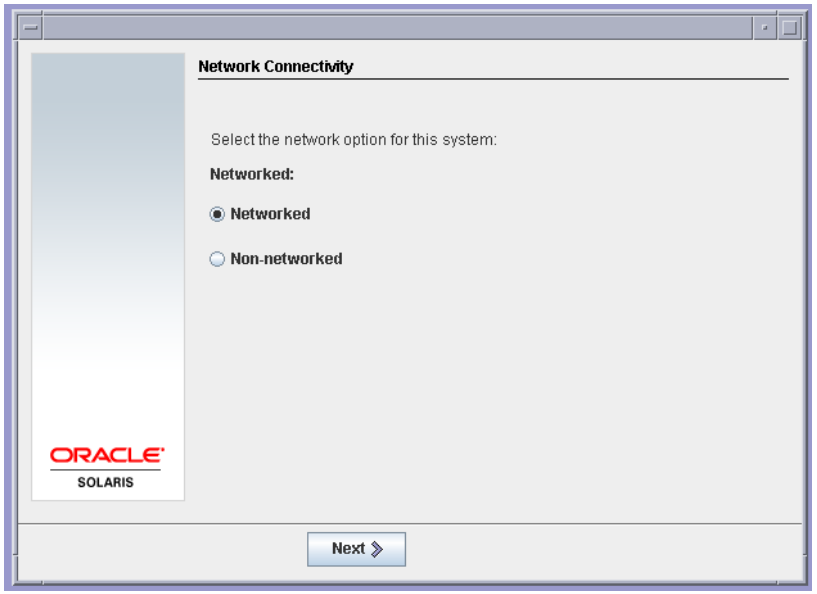
Now, the GUI dialog boxes start appearing to take you through the remaining steps of the installation process:



☞Welcome to the configuration dialogs:

- Click the Next button to continue.

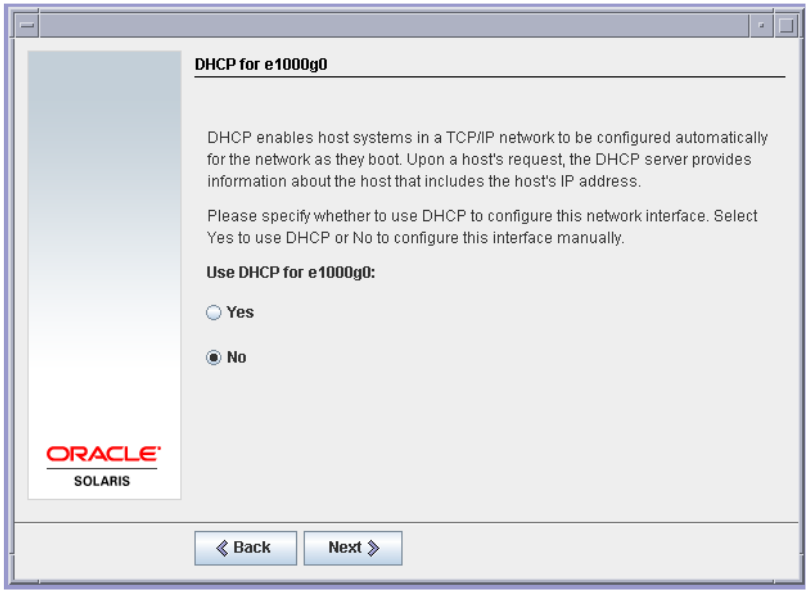
The first GUI configuration dialog allows you to enable or disable networking:



Select Network Connectivity:

- Click the radio button to choose between Network or Non-networked and click Next to continue.

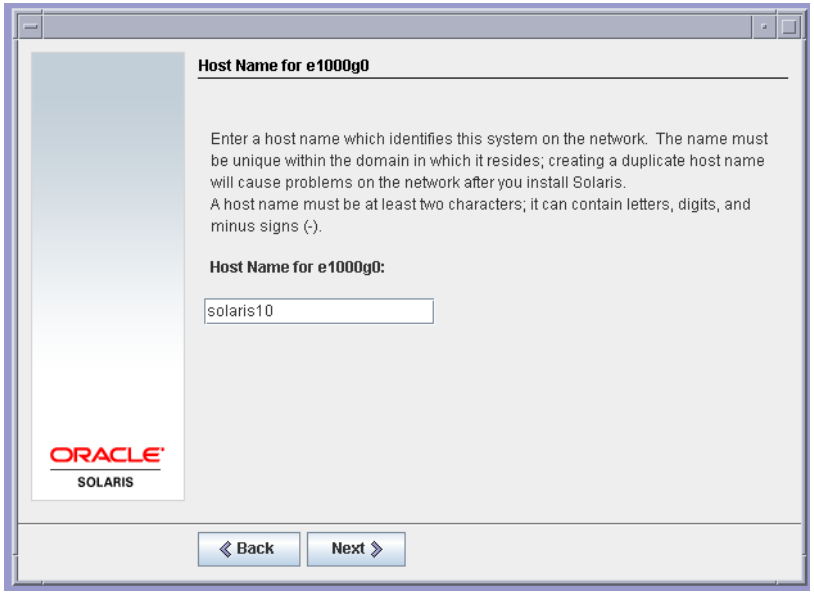
If you selected to have the system networked (generally recommended), then the next dialog allows you to choose between having the network interface automatically configured with DHCP or to manually configure the interface:



Select DHCP for interface:

- Click the radio button to choose between using DHCP or not and click Next to continue.

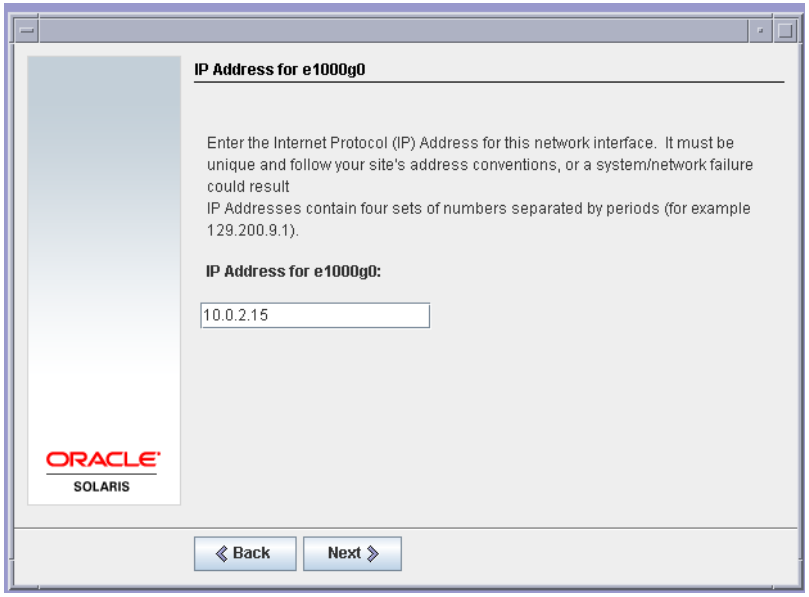
The host name of the system is configured next, if you are not using DHCP. Read the dialog carefully to be aware of the naming rules:



Enter a host name:

- Type a unique name to identify the host and click Next to continue.

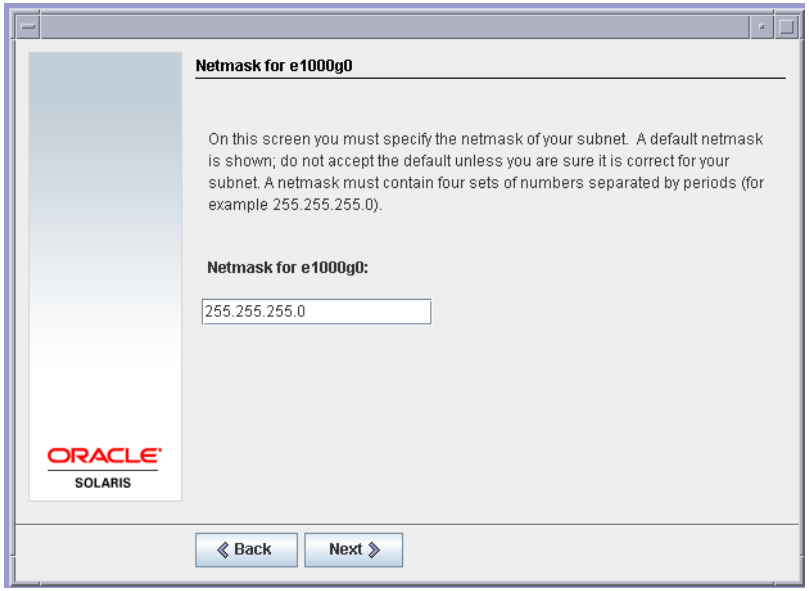
If you are not using DHCP, you will also have to set a unique IP address for the host:



Enter the IP address:

- Type the IP address for the interface and click Next to continue.

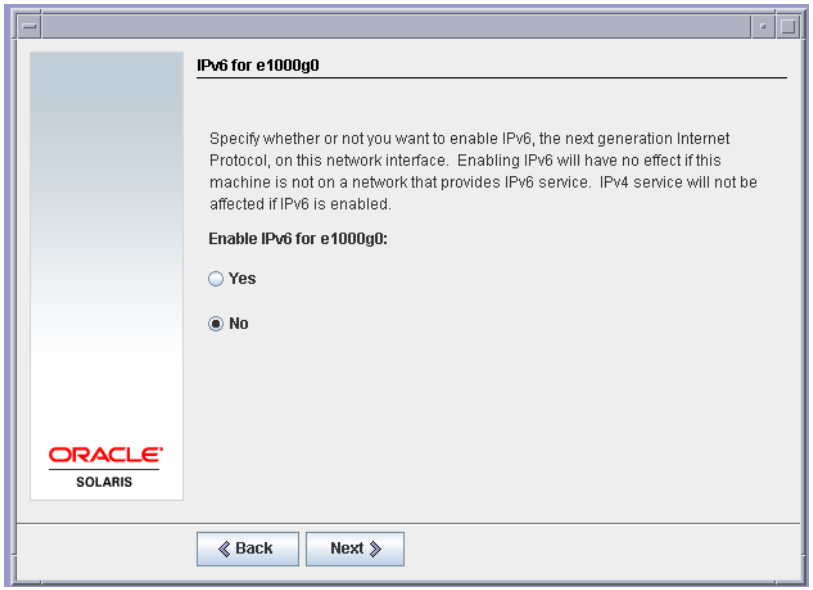
Without DHCP you will also need to configure the subnet mask for the interface:



Set the interface netmask:

- Accept the default subnet mask, or type a new one and click Next to continue.

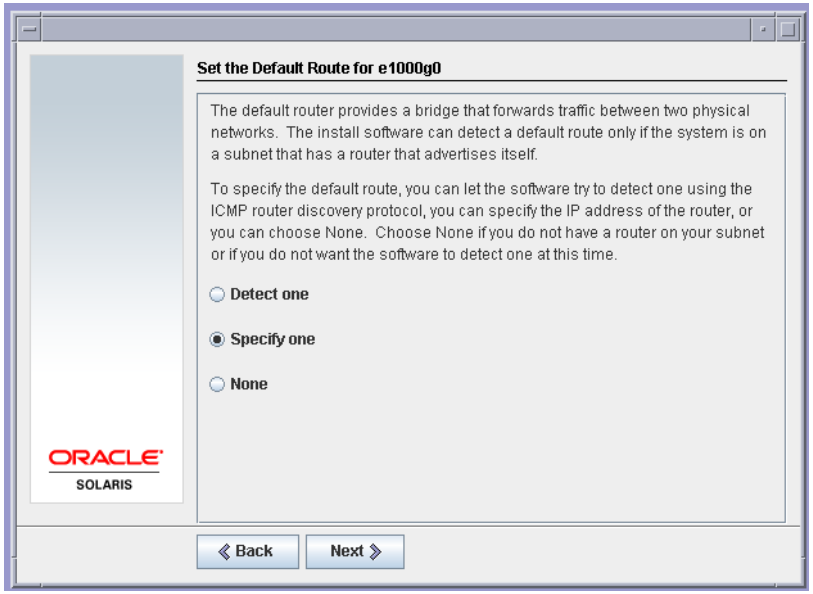
Configuration of IPv6 is available in the next dialog:



Select IPv6 configuration:

- Click the radio button to choose between enabling IPv6 or not for the interface and click Next to continue.

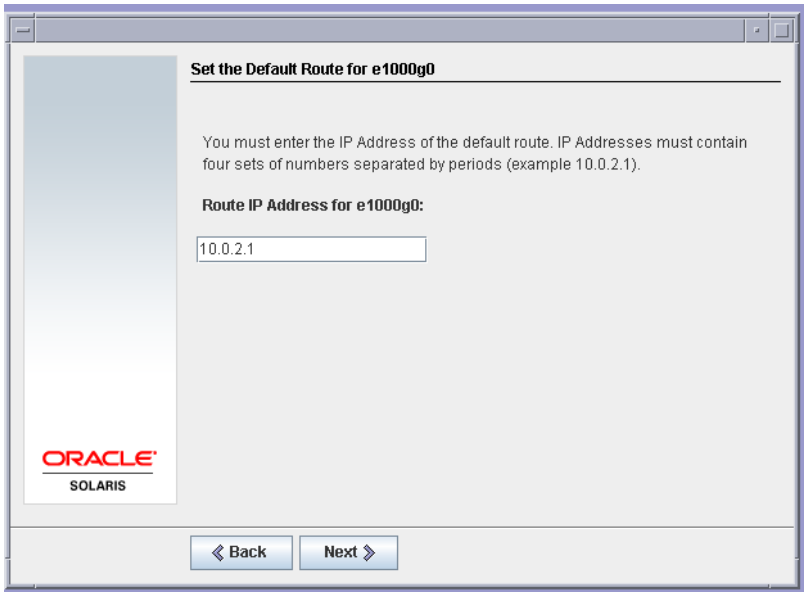
If you want to make a router available to the system, you can choose how to do that in the next dialog:



Set the Default Route:

- Click the radio button to choose between Detecting one, Specifying one, or None and click Next to continue.

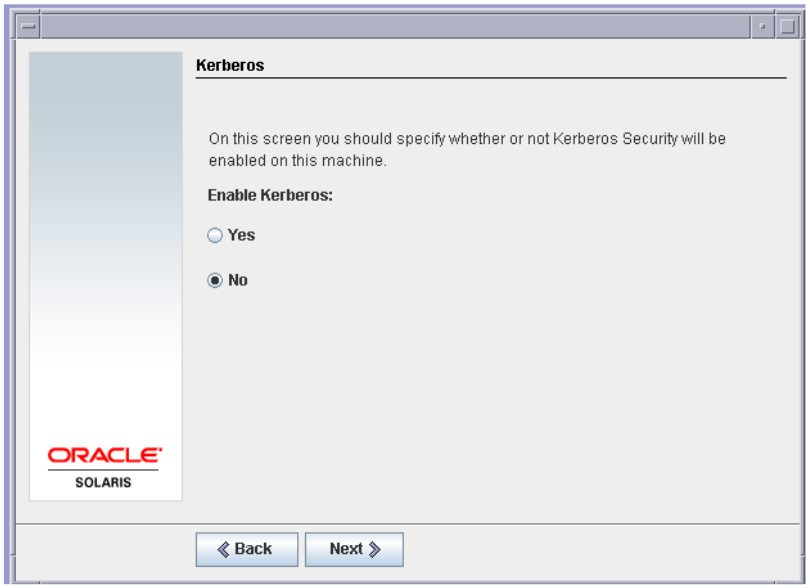
If you chose to specify a Default Route in the previous dialog, the next dialog that appears will allow you to specify that route:



Specify the Default Route:

- Type the IP address of the default route and click Next to continue.

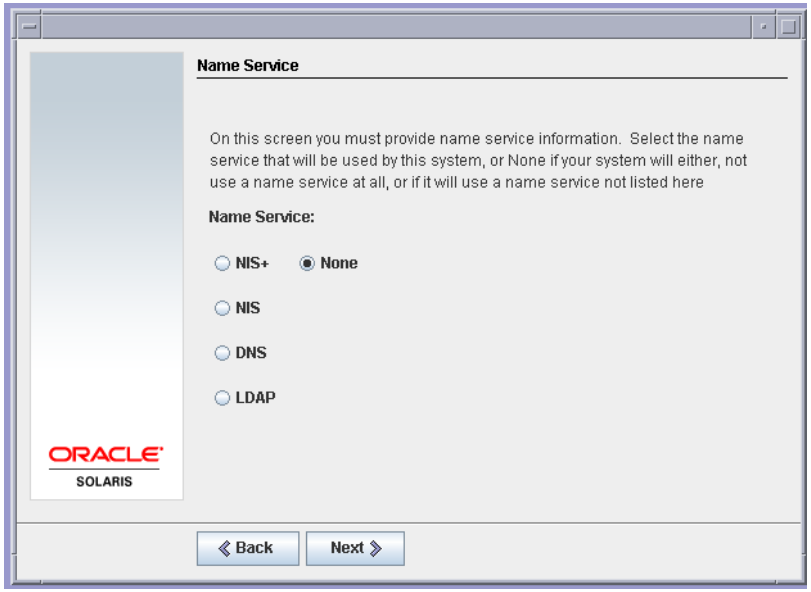
If you want to use Kerberos for authentication, you can enable it in the following dialog:



Select Kerberos Authentication:

- Click the radio button to choose between enabling Kerberos authentication or not and click Next to continue.

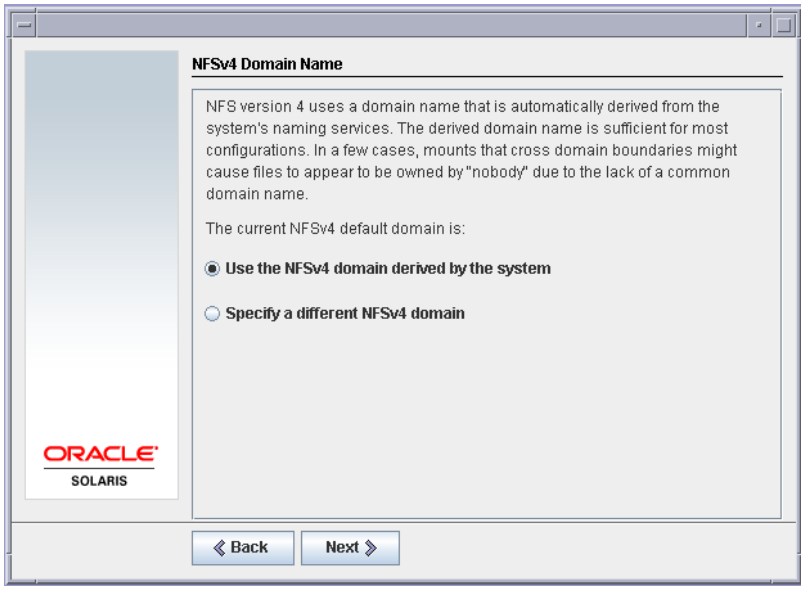
If you want to use any name services, then you can enable them in this dialog:



Enable Name Service:

- Click the radio button to choose to enable a name service and click Next to continue.

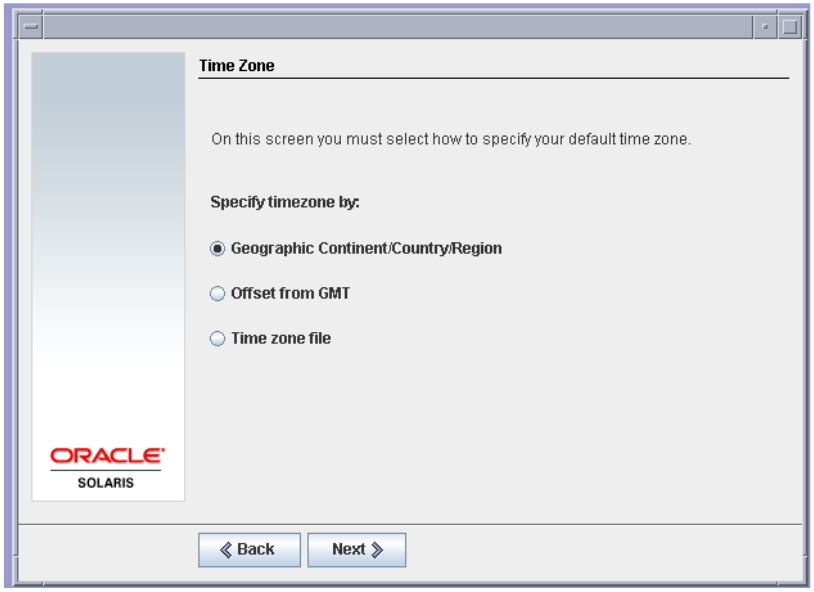
If no name services are selected, then the next dialog will allow you to specify how the NFSv4 domain name should be set:



Select NFSv4 Domain Name:

- Click the radio button to choose between deriving the domain versus specifying the domain name and click Next to continue.

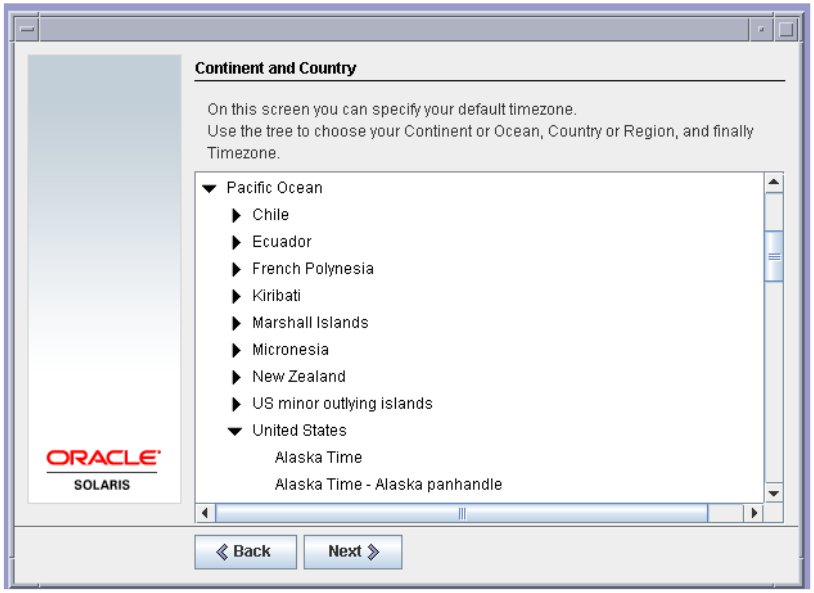
In the next dialog, you can choose how you want to specify the timezone for the system:



Select Network Connectivity:

- Click the radio button to choose how to specify the timezone and click Next to continue.

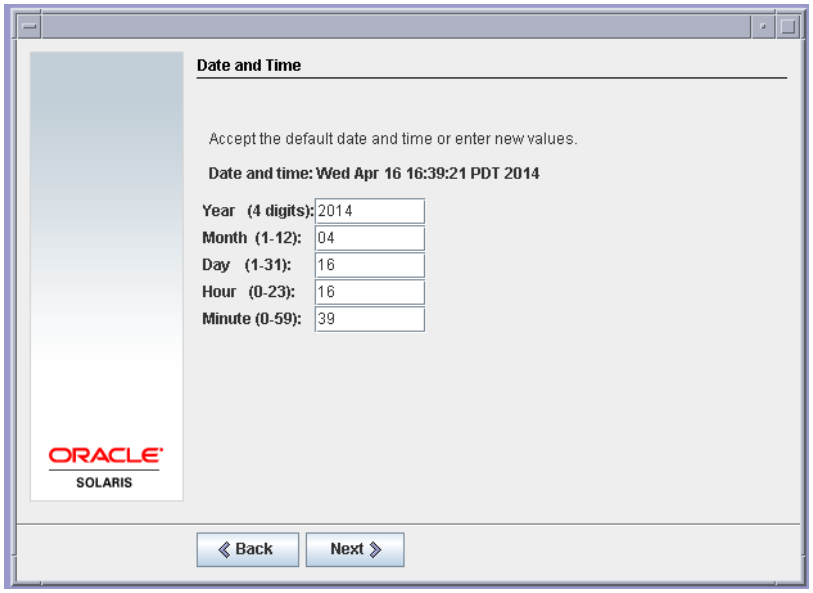
Depending on the method chosen in the previous dialog, various dialogs similar to this will allow you to specify the timezone:



Select Your Timezone:

- Click the timezone and click Next to continue.

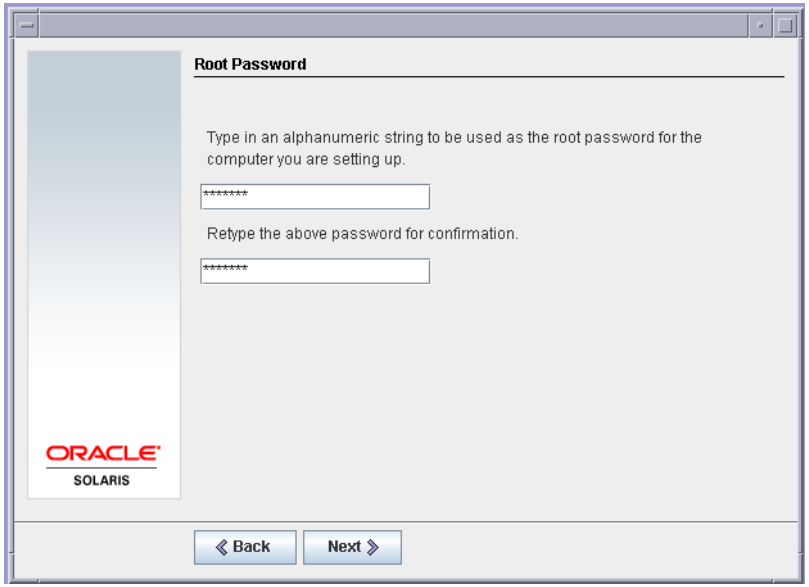
Verify or set the time for the local time in the following dialog:



Set the Date and Time:

- Verify or enter the correct date and time and click Next to continue.

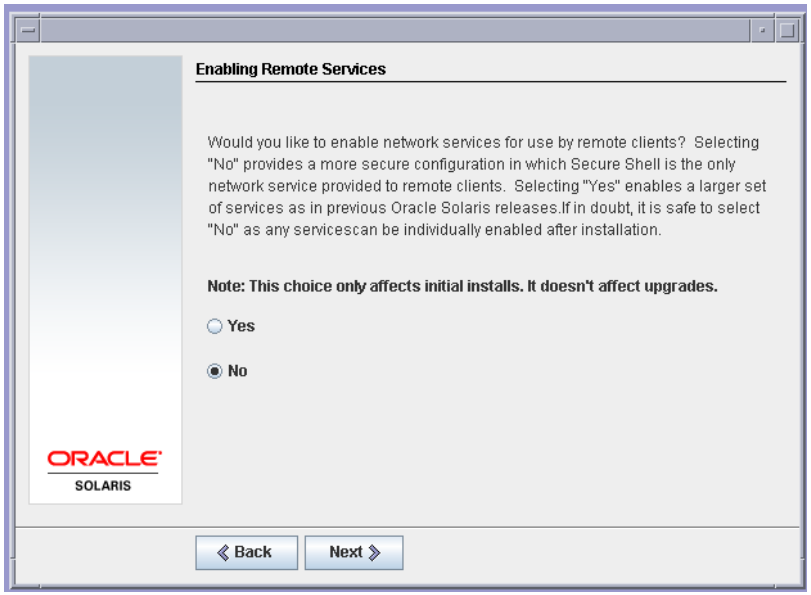
The root password must be set in the following dialog:



Set the Root Password:

- Type and retype the root user password and click Next to continue.

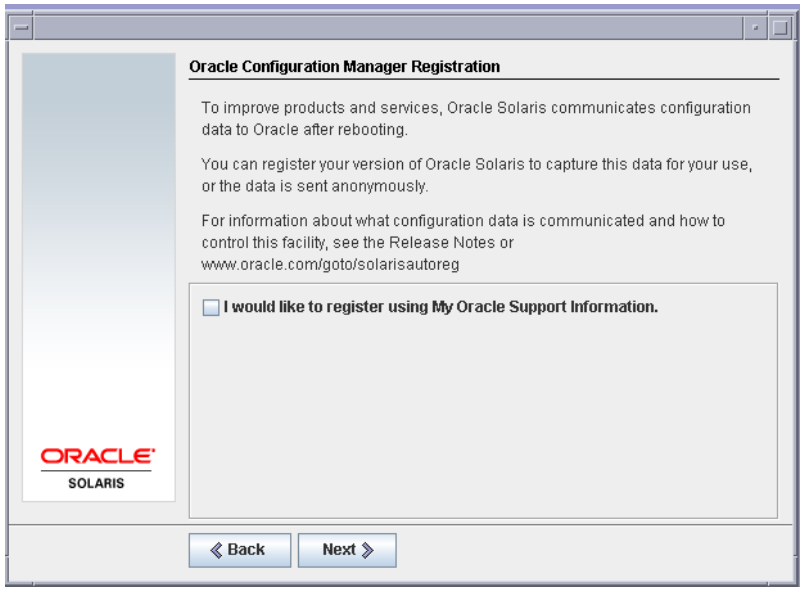
For a more secure system, you should not enable remote services for clients in the following dialog:



Enable Remote Services:

- Click the radio button to choose between enabling or not remote services for clients in addition to ssh and click Next to continue.

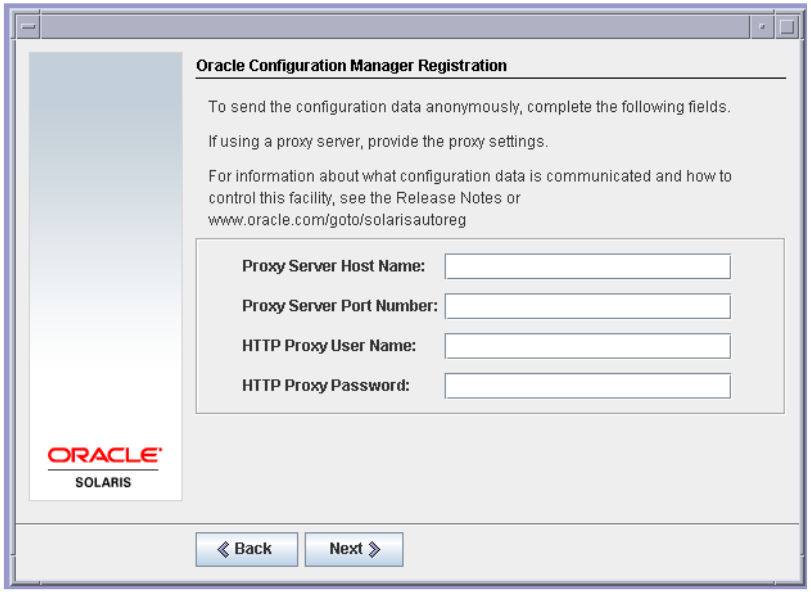
If you want to register your system, then you can enable that in the following dialog:



Register System:

- Click the check box to choose if you want to register your system with Oracle and click Next to continue.

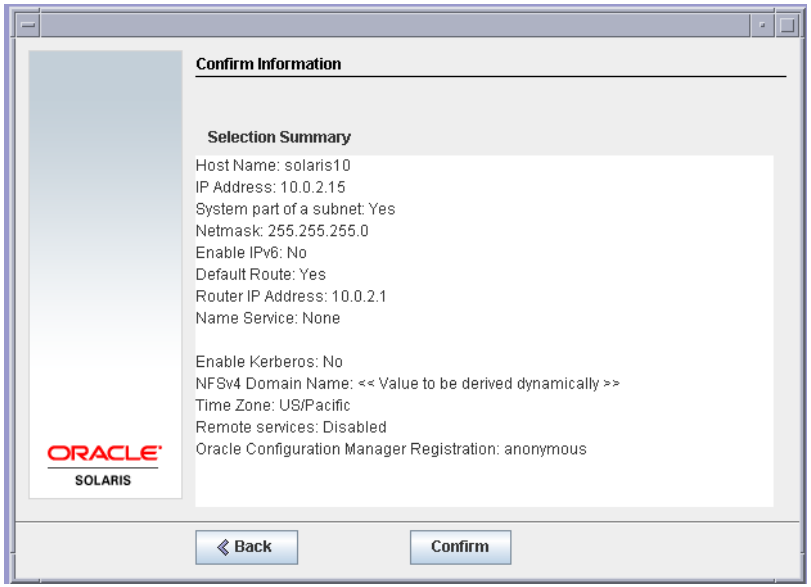
If necessary, you can configure proxies to be used when sending data to Oracle in this dialog:



Configure Proxy Servers:

- Enter any necessary configuration for proxy servers and click Next to continue.

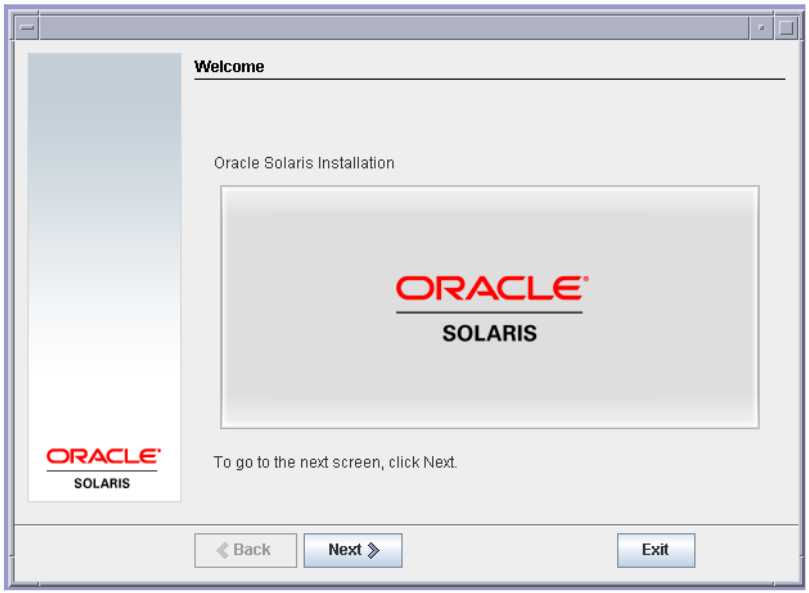
Finally, you can confirm your choices before the installation begins:



Confirm Information:

- Click the Confirm button to continue or click Back to go back to change information.

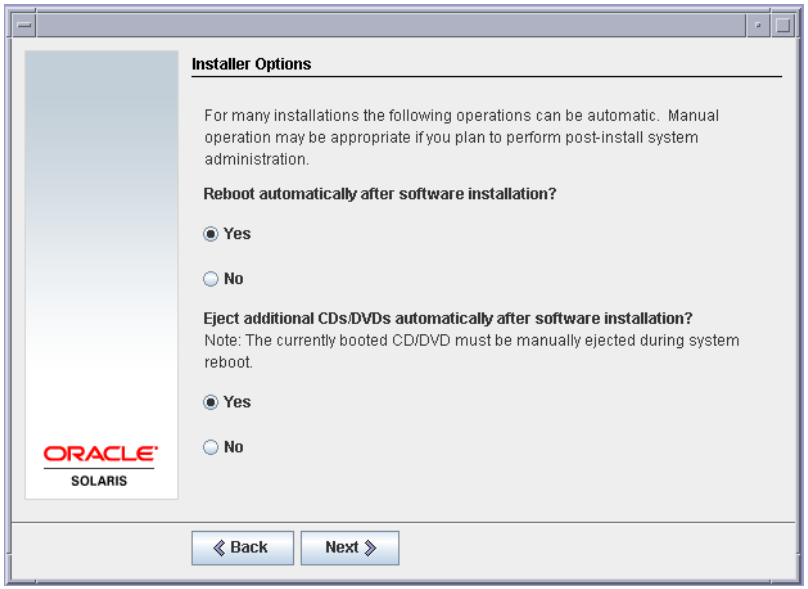
Now that you've provided most of the information, you are ready to start the actual installation:



Welcome to the Installation:

- Click the Next button to continue.

In the next dialog, you set the the installer options for rebooting and ejecting the DVD:



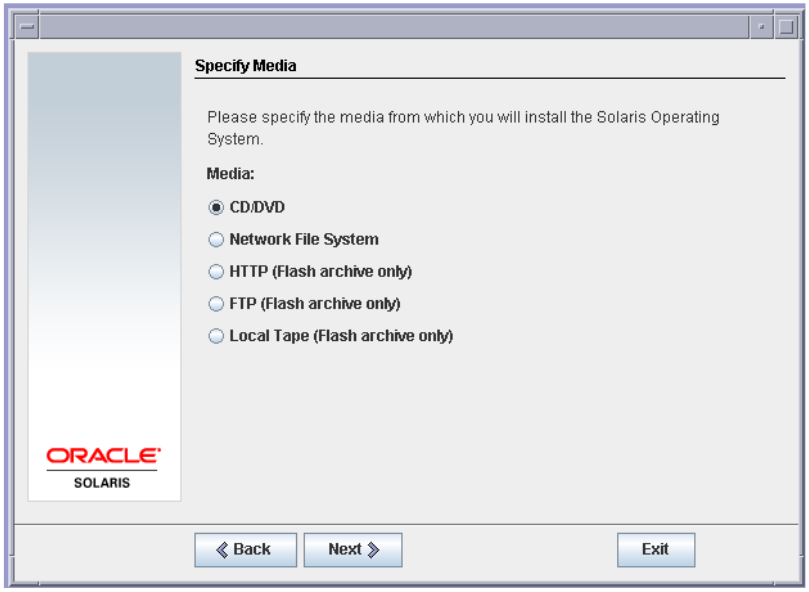
Select Installer Options:

- Click the radio button to choose the installer options and click Next to continue.

If you are using the DVD to perform an installation, don't forget to eject the DVD at the end of the install or else the installation process will restart:



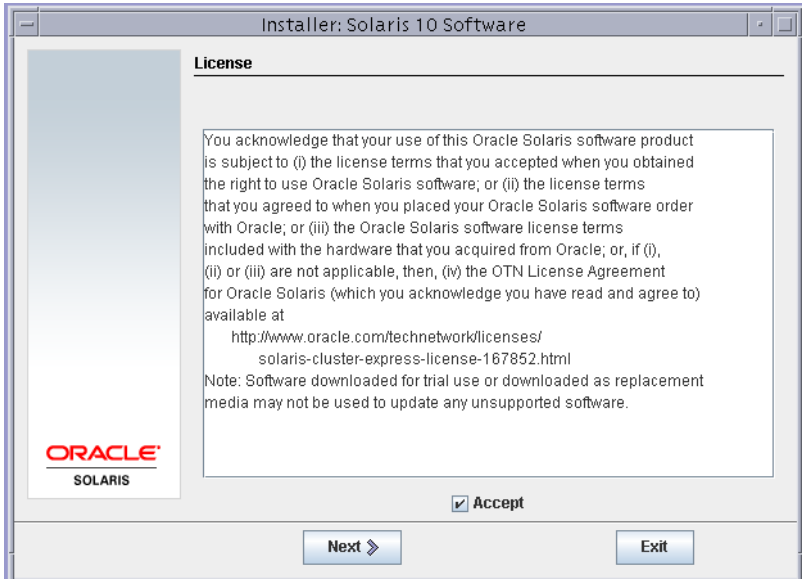
Before the installation can begin, you must choose the location of the installation media:



Specify Media:

- Click the radio button to choose the media source and click Next to continue.

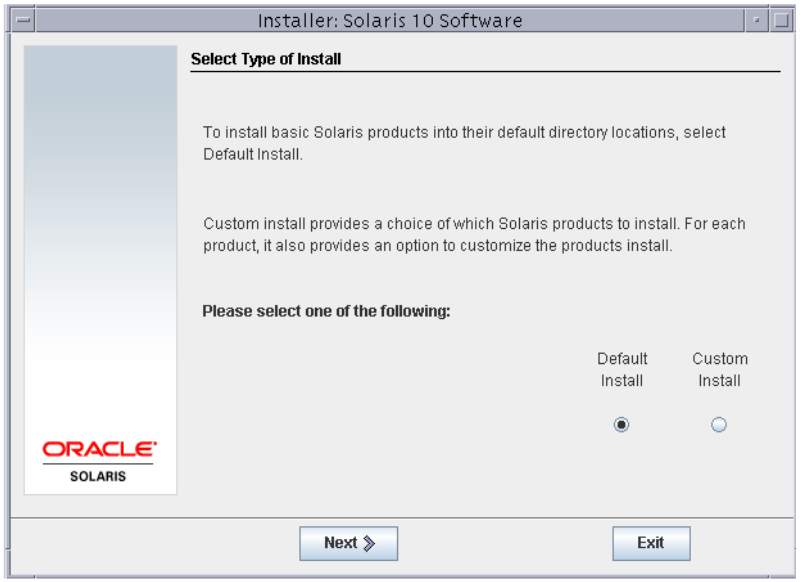
Of course, you must accept the license before you can begin the installation:



Accept License:

- Click the check box to Accept the license and click Next to continue.

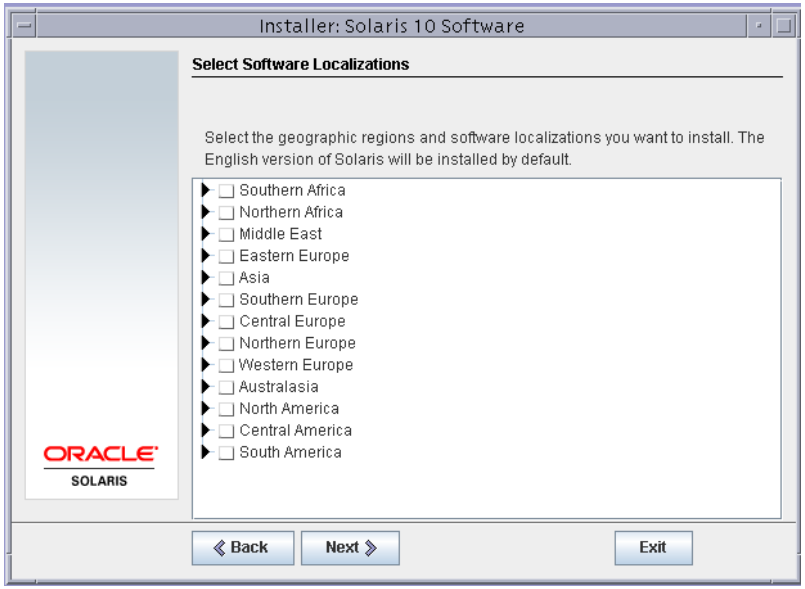
If you want to customize your software packages, you can choose to do a Custom Install:



Specify Media:

- Click the radio button to choose the Default or Custom install and click Next to continue.

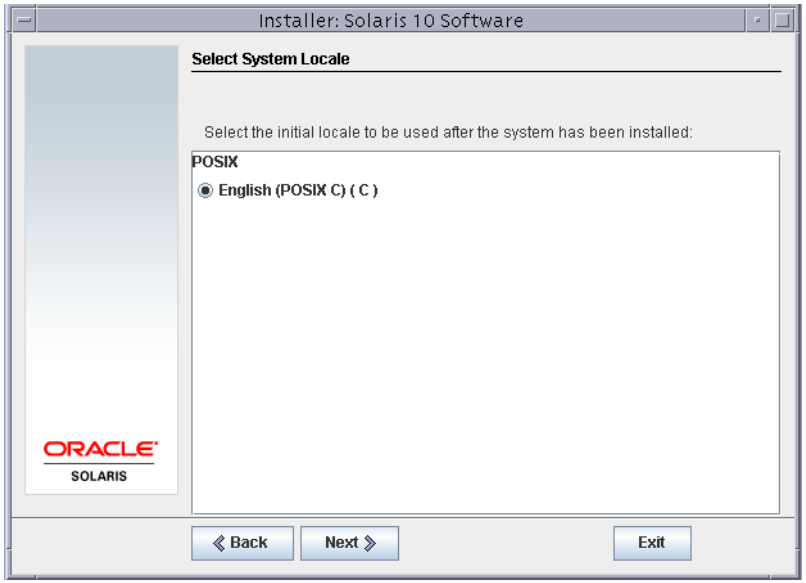
If you use other languages besides English, then you should select those locales:



Select Software Localization:

- Select any software localizations besides English and click Next to continue.

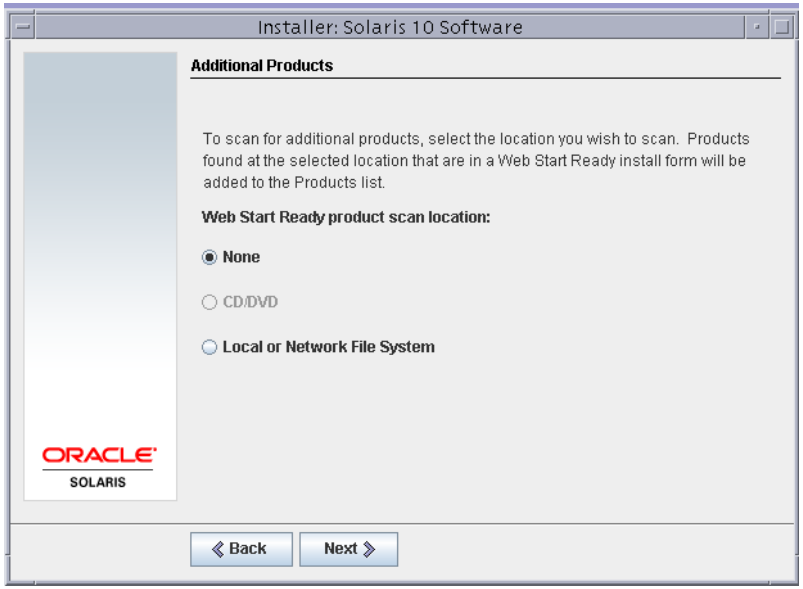
Some dialogs are purely informational, such as the system locale. Using Posix C means that system applications interpret text as classic US English ASCII text:



Select System Locale:

- Click Next to continue.

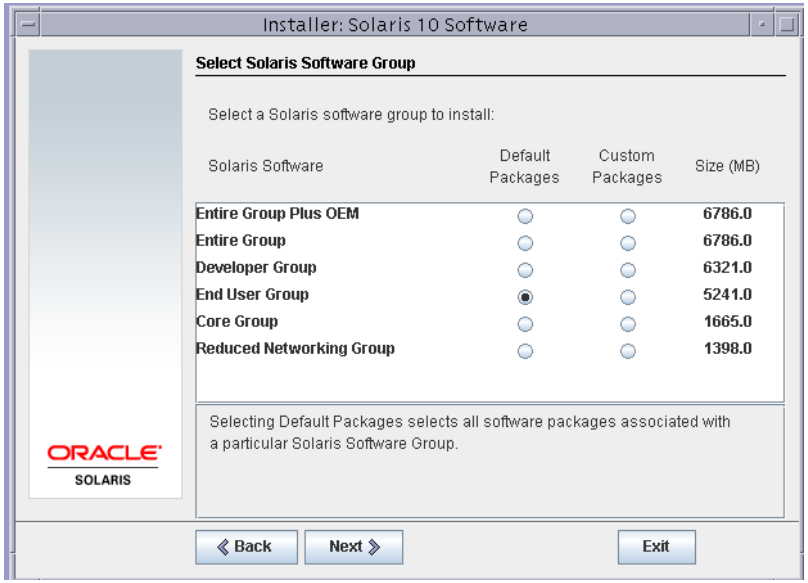
If you have any additional products prepared to install, then you can choose to scan for them:



Additional Products:

- Click the radio button to choose the location to scan and click Next to continue.

The following dialog may be one of the most important. By choosing a package group, you select a metacluster of packages that determines what software and services will be available on the installed system.

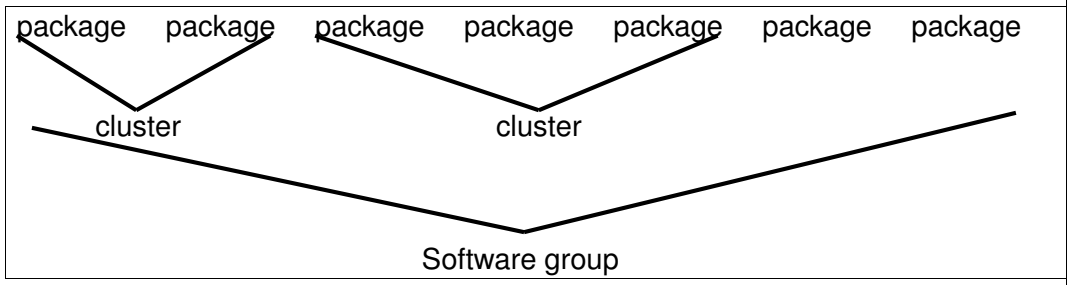


Although you can add software packages later, it is most convenient to select the correct group of packages at installation time.

Software is bundled in packages for easier installation. Common packages are then bundled into software clusters; for example, all of the packages that make up OpenWindows are bundled into the OpenWindows cluster.

Not all software packages are grouped into clusters.

All of the software packages and clusters on the installation DVD are grouped into five logical software groups. The following diagram gives a visual representation of this concept:



The following table describes the groups and what they provide:

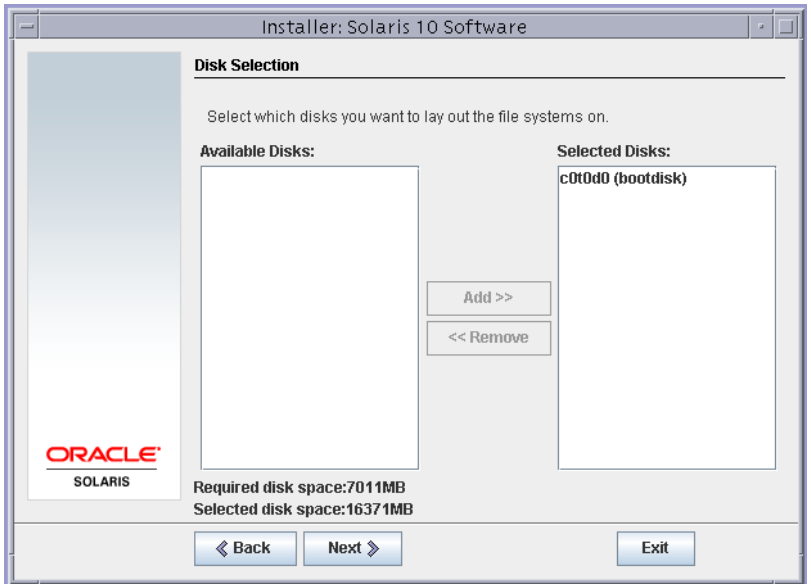
Group / Cluster	Description
Reduced Networking SUNWCrnet	This is the bare minimum of packages to boot the system and provide a multiuser text-based console. Network interfaces are detected, but no network services are activated.

Core SUNWCreq	Includes the Reduced Networking packages plus the packages need to run network services.
End User SUNWCuser	Includes the Core packages plus packages for the CDE and GNOME graphical desktop environments.
Developer SUNWCprog	Includes the End User packages plus development software libraries, include files, man pages, and programming tools not including compilers.
Entire SUNWCall	Includes the Developer packages plus additional packages for server deployments.
Entire + OEM SUNWCXall	Includes the Entire packages plus additional hardware drivers..

Select Software Group:

- Click the radio button to choose the desired software group and click Next to continue.

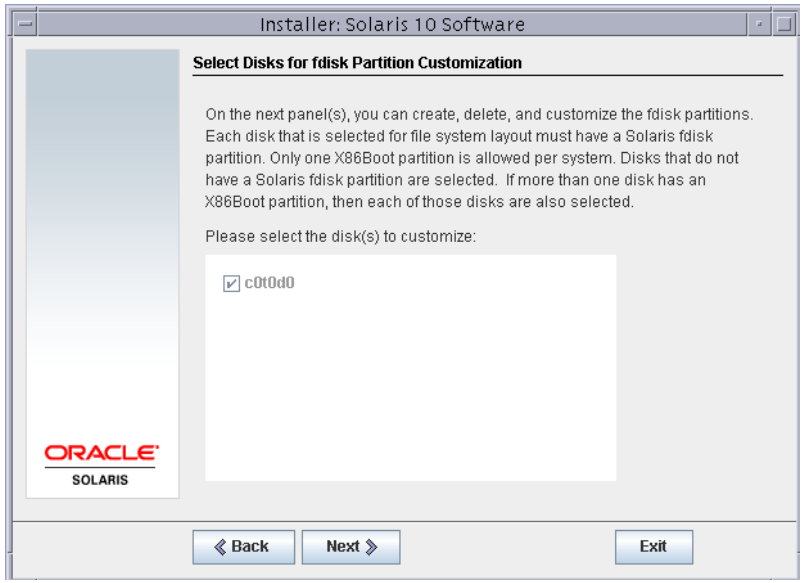
If you have more than one disk, then select the disk to prepare the filesystem on:



Select Disk:

- Use the Add and Remove buttons to choose the disk to make a filesystem on and click Next to continue.

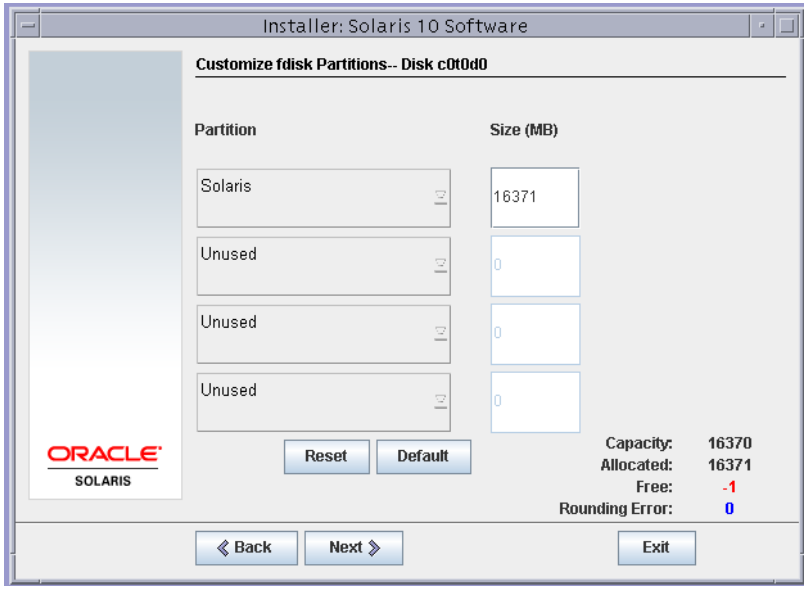
Choose the partition that you want to customize:



Select Partition:

- Click the check box to choose the partition and click Next to continue.

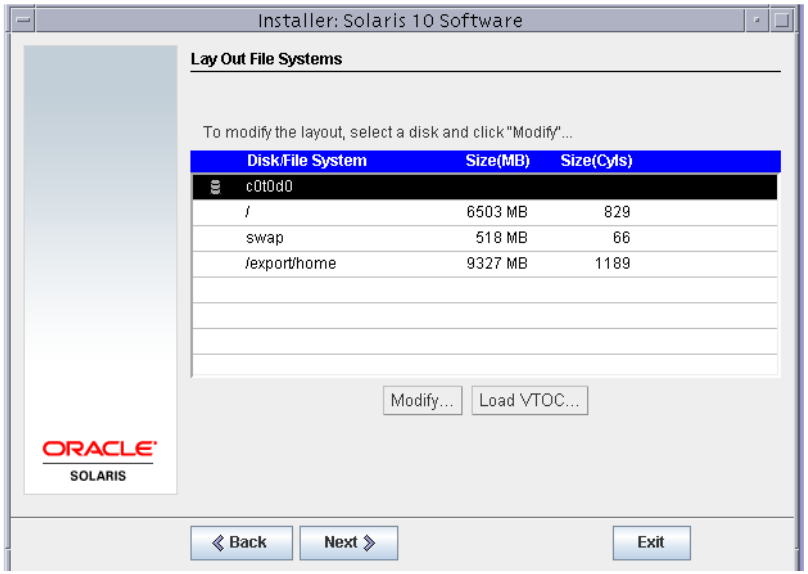
Use the following dialog to create multiple partitions if desired:



Customize fdisk Partitions:

- Type values for the partitions or use the default settings and click Next to continue.

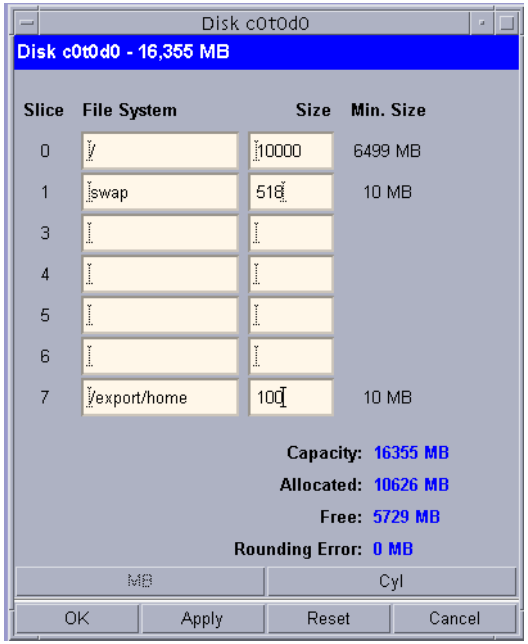
Advanced users may want to customize the layout of the filesystem in the following dialog, but new users should use the defaults:



Lay Out File Systems:

- Use the Modify button to make adjustments or click Next to continue.

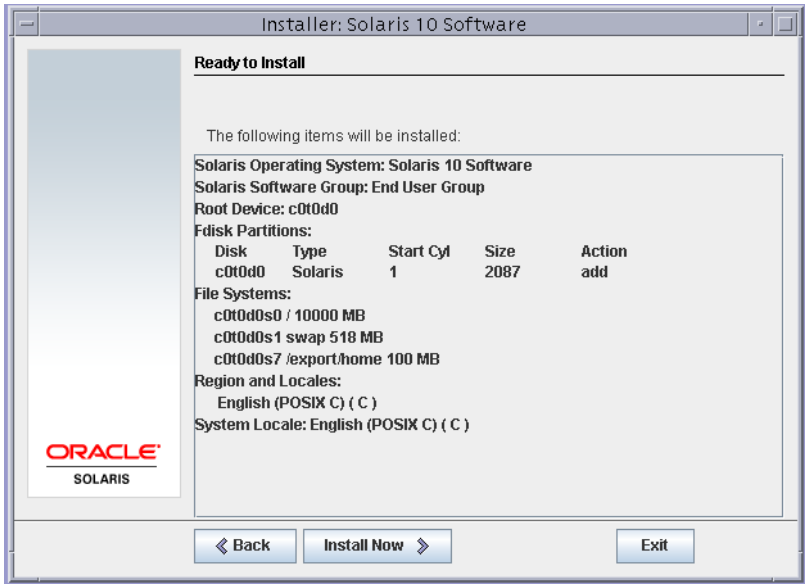
If you want to customize the partitions or slices, the dialog will look like the following:



Adjust Slice Sizes:

- Type the values for the mount point and size of each slice and click OK to continue.

At last, the installer is ready to install the Solaris Operating System:



Ready to Install:

- Click the Install Now button to begin the installation or use Back to go back to make changes.

2.6 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #3

Web sites

Solaris 10 Installation Guide: Basic Installations: http://docs.oracle.com/cd/E19253-01/817-0544/

Man pages

halt

2.7 Summary of Commands and Files

Command	Description
halt	Brings the system down cleanly

2.8 Lab Exercises

Based in the data provided to you by your instructor, perform an installation of Solaris.

Unit Three
Post Installation

Unit topics

		Page #
3.1	Configuring the Host Table	90
3.2	Post Install Files	92
3.3	Displaying Disk Usage by Directory	96
3.4	Displaying Disk Usage by Filesystem	97
3.5	Displaying Disk Usage by User	99
3.6	Introduction the Solaris Management Console	100
3.7	Additional Resources	102
3.8	Summary of Commands and Files	103
3.9	Lab Exercises	104

3.1 Configuring the Host Table

Sparc stations “talk” to each other via a network. In order to distinguish between different machines, each machine is assigned an IP address. Because IP addresses are difficult to remember and type, machines are given nicknames, host names, for users and System Administrators to use.

After installation, the only host name a machine knows about is its own. To tell your machine about other host names, place these names (and their corresponding IP addresses) in the `/etc/inet/hosts` table:

```
# more /etc/inet/hosts table
#
# Internet host table
#
127.0.0.1          localhost
129.46.24.50       sun       loghost
```

The keyword **loghost** should only appear on the line of the machine that will hold the log files for this host. Typically, this is the local machine.

Notes:

Notes:

Note: The file `/etc/hosts` is just a symbolic link to the `/etc/inet/hosts` file. The system never uses the `/etc/hosts` file. It was only placed there for convenience.

Note: The IP address 127.0.0.1 is a special IP that represents the machine itself. Don't delete this IP address!

To test your `/etc/inet/hosts` file, use the **rup** command:

```
# rup
        glory           up 30 days, 20:31,    load average: 0.03, 0.01, 0.02
        sun             up 10 days, 15:34,    load average: 0.04, 0.01, 0.02
        129.8.23.10     up 30 days, 21:01,    load average: 0.03, 0.01, 0.02
```

rup sends a broadcast to all machines on the network asking them to report back to the local machine. If the machine's IP address is in the `/etc/inet/hosts` table, **rup** lists the machine's name. If not, **rup** lists the machine's IP address.

Commented [KW1]: The rstatd daemon must be enabled for a host to respond. It can be enabled by executing:

inetadm -e svc:/network/rpc/rstat:default

3.2 Post Install Files

Notes:

The Installation Log

As a result of the installation process, log files were created in the `/var` directory. These files contain information regarding what was installed on the system.

To display the log of what occurred during the installation process, look at the `/var/sadm/install_data/intall_log` (note: much of the text in the following example was {omitted}):



```
# more  /var/sadm/install_data/install_log
        Expanding Locale en_US
        Expanded en_US to include locale: en_US.ISO8859-1

Processing profile
      - Selecting cluster (SUNWCXall)

Installing 64-bit Solaris packages
      - Selecting locale (en_US.ISO8859-1)
      - Selecting locale (en_US)
{Remaining output omitted}
```

All of the information displayed by this command will be explained as the class progresses.

Notes:

The Package Directory

In the `/var/sadm/pkg` directory, each package that has been installed has its own subdirectory:

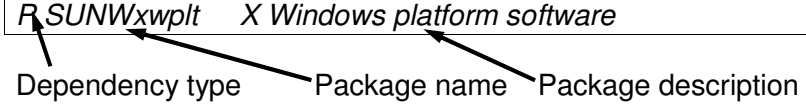
```
# ls /var/sadm/pkg
AXILvplr      SUNWadmc  SUNWdfb   SUNWdtmad  SUNWhea   SUNWjvrt
SUNWmfrun     SUNWpcelx SUNWpsf   SUNWsutl   SUNWxcu4  SUNWxwfs
AXILvplu      SUNWadmfw SUNWdfbh  SUNWdtrme  SUNWhiu8  SUNWkcspf
{note: remaining text omitted}
```

Under each of these subdirectories, there are files that contain information regarding that package. One such file is the **pkginfo** file that contains general information about the package:

```
# cd /var/sadm/pkg/SUNWaudio
# ls
install  pkginfo  save
# more pkginfo
CLASSES=none
BASEDIR=/
TZ=US/Pacific
PATH=/sbin:/usr/sbin:/usr/bin:/usr/sadm/install/bin
{note: remaining text omitted}
```

Another important file is located under the **install** directory. This file, called the **depend** file, describes dependencies between packages:

```
# cd install
# more depend
{note: text omitted here}
P SUNWcar       Core Architecture,  (Root)
P SUNWkvm       Core Architecture,  (Kvm)
P SUNWcsr       Core Solaris,  (Root)
P SUNWcsu       Core Solaris,  (Usr)
P SUNWcsd       Core Solaris Devices
R SUNWxwplt     X Windows platform software
```



Dependency types

P - The current package depends on the listed package
R - The listed package depends on the current package
I - The packages are incompatible

Note: if there is no `depend` file, then this file doesn't depend on another package (besides the basic OS packages). However, there may be packages that depend on this package.

Notes:

Notes:

The System Contents File

During the installation of packages, the original attributes of each file that was added to the system is stored in the contents file:

```
#  more /var/sadm/install/contents
{note: text omitted here}
/etc/cron.d/logchecker f none 0555 bin bin 1659 598 869026971 SUNWcsr
/etc/cron.d/queuedefs f none 0644 root sys 17 1164 28800 SUNWcsr
/etc/datemsk f none 0444 root sys 446 24974 869026985 SUNWaccr
/etc/dcopy=../usr/sbin/dcopy s none SUNWcsr
{note: remaining text omitted}
```

This file is useful for two reasons:

1. To determine if a file has changed ownership, permissions, or size since it was installed. This is one means of determining if a hacker has compromised the system.
2. To search quickly for system files using **grep** (instead of **find**):

```
# grep shadow /var/sadm/install/contents
/etc/shadow e shadow 0400 root sys 252 17245 28800 SUNWcsr
/usr/include/shadow.h f none 0644 bin bin 2474 4583 869024258 SUNWhea
/usr/share/man/man4/shadow.4 f none 0444 bin bin 3316 20991 867354858
SUNWman
```

Note: If you add or remove packages after the installation process, the **contents** file is automatically updated.

3.3 Displaying Disk Usage by Directory

To determine how much space is being used by a directory (and its subdirectories), use the **du** command:

```
# du -sk /usr/openwin
202758  /usr/openwin
```

Common options:
-s Summary
-k Show in kilobytes (default is to show in blocks)

This command is useful to determine how much space a particular software is using.

Notes:

Notes:

3.4 Displaying Disk Usage by Filesystem

To determine how much space is being used by a filesystem (partition), use the **df** command:

```
# df -k
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/dsk/c0t3d0s0     72407     28861  43474   40%     /
/dev/dsk/c0t3d0s6     583807    406870 176354  70%     /usr
/proc                 0         0      0       0%      /proc
fd                    0         0      0       0%      /dev/fd
/dev/dsk/c0t3d0s1     25919     13914  11980   54%     /var
/dev/dsk/c0t3d0s7     57143     347    56739   1%      /export/home
swap                  103972    196    103776  1%      /tmp
/dev/dsk/c0t3d0s4     102871    62056  40713   61%     /opt2
/dev/dsk/c0t3d0s5     925974    883640 40791   96%     /opt
```



Notes:

Human Readable output

The **-h** option displays the output in a more "readable" format by showing sizes in kilobytes, megabytes or gigabytes:

```
# df -h
Filesystem             size   used  avail capacity  Mounted on
/dev/dsk/c0t0d0s0      939M    85M   798M    10%    /
/dev/dsk/c0t0d0s4      8.2G   1.1G   7.1G    14%    /usr
/proc                  0K     0K     0K      0%     /proc
mnttab                 0K     0K     0K      0%     /etc/mnttab
fd                     0K     0K     0K      0%     /dev/fd
/dev/dsk/c0t0d0s3      1.9G    76M   1.8G     5%    /var
swap                   530M   168K   530M    1%     /var/run
swap                   530M   328K   530M    1%     /tmp
/dev/dsk/c0t0d0s5      1.9G   416M   1.4G    23%    /opt
/dev/dsk/c0t0d0s7      469M   151K   422M    1%     /export/home
```

Notes:

3.5 Displaying Disk Usage by User

The **quot** command will display how much space each user on the system is using:

```
# quot –a
/dev/rdsk/c0t3d0s0 (/):
 8799   smtp
 2840   bin
   32   lp
   31   uucp
    1   adm
    1   sys
/dev/rdsk/c0t3d0s6 (/usr):
114034  smtp
89140   bin
 2212   lp
  801   uucp
    1   adm
{Remaining output omitted}
```

Common options:

-a Generate a report for all mounted filesystems.

-f Display count of number of files as well as space owned by each user.

-v In addition to the default output, display three columns containing the number of blocks not accessed in the last 30, 60, and 90 days.

3.6 Introduction the Solaris Management Console

The Solaris Management Console (SMC) is designed to simplify the process of administering the Solaris OE. This tool can be executed either in GUI-mode or via command line tools.

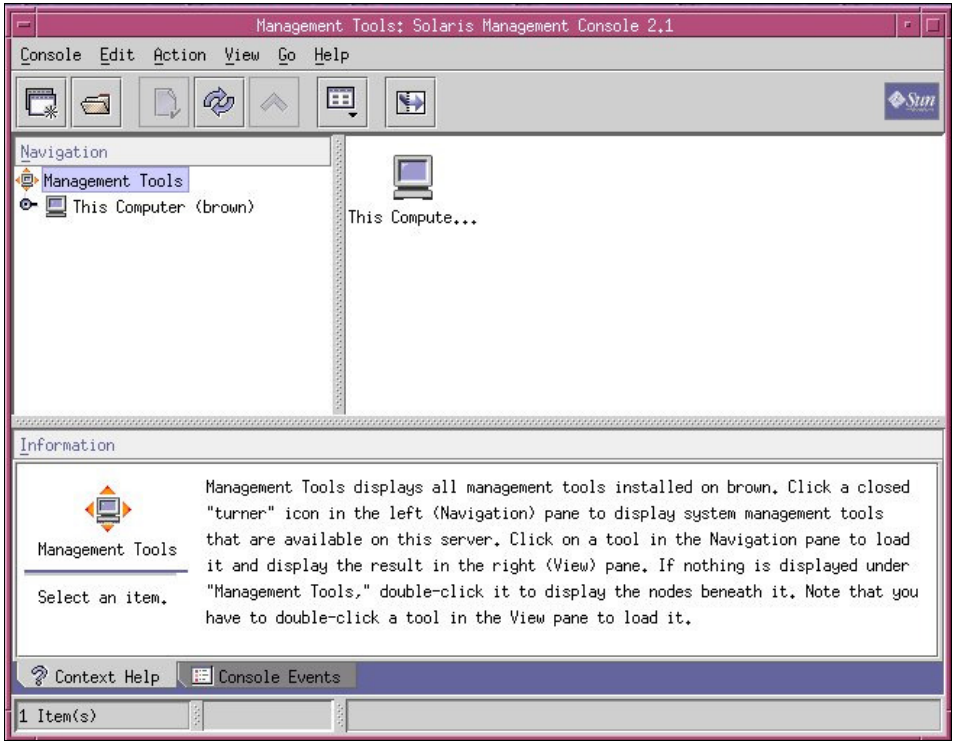
While the SMC can't be used to administer all of the features of the Solaris OE, it can provide the following features:

- Display system information
- Administer processes
- Display system logs
- Display system performance
- Administer users and groups
- Display network information
- Administer patches
- Administer cron jobs
- Display disk space usage
- Administer disk partitions including RAID
- Administer serial ports

Notes:

To start the SMC in GUI mode, run the **smc** command:

```
# smc
```



Notes:

Notes:

3.7 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #2

Web sites

None

Man pages

df
du
quot
rup
smc

Notes:

3.8 Summary of Commands and Files

Command	Description
df	Displays disk usage by filesystem (partition)
du	Displays disk usage by directory
quot	Displays disk usage in a filesystem (partition) on a user basis
rup	Sends a broadcast request on the network to determine what machines are in the network.
smc	Starts the Solaris Management Console

File	Description
/etc/hosts	A symbolic link to /etc/inet/hosts
/etc/inet/hosts	Contains a list of host name to IP address entries
/var/sadm/install/contents	A files that contains a list of each file (and its original attributes) that is currently installed on the system.
/var/sadm/install_data/install_log	A log of what took place during the installation process.
/var/sadm/pkg	A directory that contains detailed information for each package that is currently installed.

Notes:

3.9 Lab Exercises

1. What are the original permissions of the file `/etc/init.d/nfs.server`?

   Command: _____

2. Determine if the package SUNWauda was successfully installed during the installation of the operating system
   Command: _____

3. Are there any dependencies for this package?

   Command: _____

4. Which filesystems have at least 100MB of space available?

   Command: _____

5. Which user is using the most space in the `/` partition?

   Command: _____

6. How large is the `/usr/openwin/demo` directory?

   Command: _____

Notes:

Scenario #1: You wish to install a 50 MB program on your system. Determine if there is enough room on the hard drive to do this (FYI: the program must completely fix within one partition).



Notes:

Unit Four
The Open Boot PROM

Unit topics

		Page #
4.1	What is the Open Boot PROM?	107
4.2	Displaying Basic System Information	111
4.3	Getting Help	112
4.4	Displaying SCSI or IDE Devices	113
4.5	Displaying Device Alias Names	117
4.6	Creating Custom Device Alias Names	120
4.7	Displaying PROM Variables	121
4.8	Setting PROM Variables	123
4.9	Resetting PROM Variables to Factory Preset	126
4.10	Booting from PROM	127
4.11	Recovering from a stop-a	128
4.12	Additional Resources	130
4.13	Summary of Commands and Files	131
4.14	Lab Exercises	132

Notes:

4.1 What is the Open Boot PROM?

The Open Boot PROM is commonly referred to as just the PROM. PROM stands for Programmable Read-Only Memory.

PROM is not a UNIX "thing". PROM is the hardware level of the Sparc Station.

As a System Administrator, you can customize and modify how the PROM functions. This Unit will focus on the most common customizations you can make to PROM.

Please note: all of these command will only work at the PROM prompt. None of the commands in this Unit (except the **halt** command) are UNIX commands.

To bring the system down to the PROM level, use the **halt** command. We will discuss this command in detail in a future Unit:

```
# halt
{output omitted}
```

Notes:

PROM versions

There are five versions of PROM: 1.x, 2.x, 3.x, 4.x, 5.x. Most likely you will not come across any machines with a 1.x or 2.x PROM as these systems are old and not used very often.

You may encounter some machines with 3.x PROM. These include systems like the Sparc 4, Sparc 5 and Sparc 10.

Most likely the machines you will encounter will have 4.x or 5.x PROM. The 4.x PROM was introduced with the 64-bit UltraSPARC III models while the 5.x PROM is available on the Sun Enterprise models (3500 and higher).

This Unit focuses on the 4.x and 5.x versions of PROM.

To see the version of PROM that your system is using, either use the **banner** command at the PROM prompt (see this command in a few pages), or use the **prtdiag** command at a shell prompt:

```
# prtdiag -v | grep OBP
OBP 4.0.45 2001/02/08 14:33 Sun Blade 100
```

Notes:

The PROM chip

The PROM is a chip that stores the programs used to boot the system and test the hardware. On older systems (1.x and 2.x PROM) this was a removable chip. On newer systems (3.x and higher) the chip is soldered on to the system board.

On some systems you can update the PROM software. These systems use a different type of PROM chip called a FPROM (Flash PROM). In order to update the PROM on these systems, you will need change a jumper (it is set to be write-protected by default). Consult the documentation of you system for complete details.

The NVRAM chip

The Nonvolatile Random Access Memory (NVRAM) chip is used by PROM to store data that can be modified. This data is often referred to as the Electronically Erasable Programmable Read-Only Memory (EEPROM).

This chip contains information that can either be modified from the PROM command level or from the shell. Examples of data that can be modified include the device name to boot from, boot parameters (options) and the system's time.

The chip also contains some information that can not be changed. For example, the system's Ethernet address is stored on the NVRAM chip, but it can not be changed.

Notes:

POST

The Power On Self Test (POST) is software that is stored on the PROM chip that is used to test the system's hardware. This software is used to perform actions during the boot process such as testing memory and the CPU. It also creates a device tree (a directory-like structure of the devices on the system) by probing the system bus.

Stop operations

The stop key can be used to control the system during the boot process. The following stop operations are allowed:

Operation	Description
stop-d	Used to have the POST software switch to diagnostic mode. In diagnostic mode the POST performs more extensive tests on the hardware. This feature can also be enabled by setting the **diag-switch?** parameter to true (see details in a future section).
stop-n	This stop operation will reset all PROM parameter back to their factory preset values. This can be very useful when a system won't boot because of incorrect PROM parameters
stop-a	This stop operation is used to halt the boot process and presents a PROM prompt.

Notes:

4.2 Displaying Basic System Information

The banner command

The **banner** command will display basic system information, including:

1. What ROM revision you are using (version of the PROM software).
2. How much memory is installed on the system.
3. The system's Ethernet address

An example of the **banner** command:

```
ok banner
SPARCstation 4 (1 X 387Z34), Keyboard Present
ROM Rev. 2.2, 64 MB memory installed, Serial #3243878.
Ethernet address 8:0:20:e5:1f:3b, Host ID: 79873645
```

The .enet-addr command

To display only the Ethernet address of the system, use the **.enet-addr** command:

```
ok .enet-addr
8:0:20:e5:1f:3b
```


© 2014 One Course Source

4.3 Getting Help

Help at the PROM level is sparse, but it does exist. Using the command **help** by itself will list the topics available:

```
ok help
{output omitted}
```

Using a topic name or command after the **help** command will give you details about that topic:

```
ok help probe-scsi
{output omitted}
```

Notes:

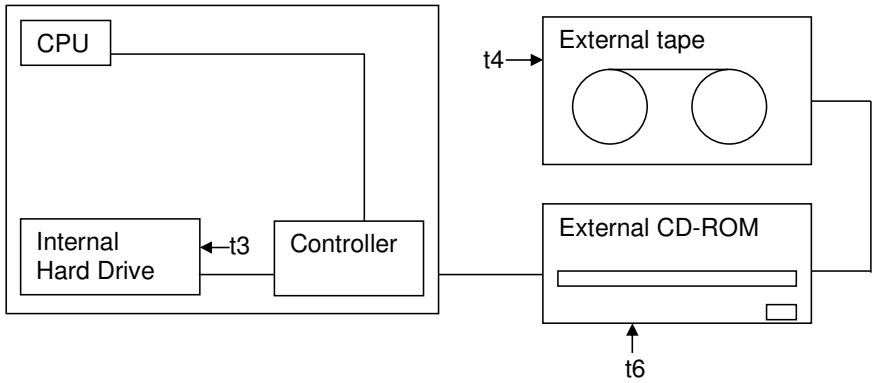
Notes:

4.4 Displaying SCSI or IDE Devices

SPARC stations use either SCSI or IDE devices. The following sections show how you can display these devices from within PROM.

SCSI devices

SCSI devices are typically either hard drives, CD-ROMS, and tape devices. The chart below illustrates a typical system:



Since all of the devices are connected to one SCSI controller, there needs to be a method of distinguish between each device. Therefore, unique **target numbers** are assigned to each. In the example above, the internal hard drive has been assigned target 3 (t3), the external tape device has been assigned target 4 and the CD-ROM has been assigned to target 6.

These targets are not assigned at random. There are typical locations for hard drives, CD-ROMS and tape devices.

For ROM Rev. 1.x and 2.x (typically up to Ultra Sparcs):

Target	Device
0	**Hard drive**
1	Hard drive
2	Hard drive
3	**Boot hard drive**
4	Tape device
5	Tape device
6	CD-ROM

For ROM Rev. 3.x, 4.x and 5.x (typically Ultra Sparcs and beyond):

Target	Device
0	**Boot hard drive**
1	Hard drive
2	Hard drive
3	**Hard drive**
4	Tape device
5	Tape device
6	CD-ROM

Notes:

To display the SCSI devices that are currently attached (and powered on) to the system, use the **probe-scsi** command:

```
ok probe-scsi
Target 3
  Unit 0    Disk  SEAGATE ST2310
Target 6
  Unit 0    Removable Read Only device Sony       CD-ROM
```

Notes:

- At each target, you can place a number of devices. In order to distinguish between two (or more devices) on a single target, the system will assign them a LUN (Logical Unit Number). The “Unit 0” in the example above refers to which LUN at that target the device is located.

- If there is more than one SCSI controller in your system, use the command **probe-scsi-all**; **probe-scsi** by itself only shows the first SCS Is controller's devices.

Notes:

Notes:

IDE Devices

The **probe-ide** command displays which IDE devices (typically hard drives and CD-ROMs) are attached to the system:

```
ok probe-ide
  Device 0    ( Primary Master )
              ATA Model: ST315320A

  Device 1    ( Primary Slave )
              Removable ATAPI Model: LTN486S

  Device 2    ( Secondary Master )
              Not Present

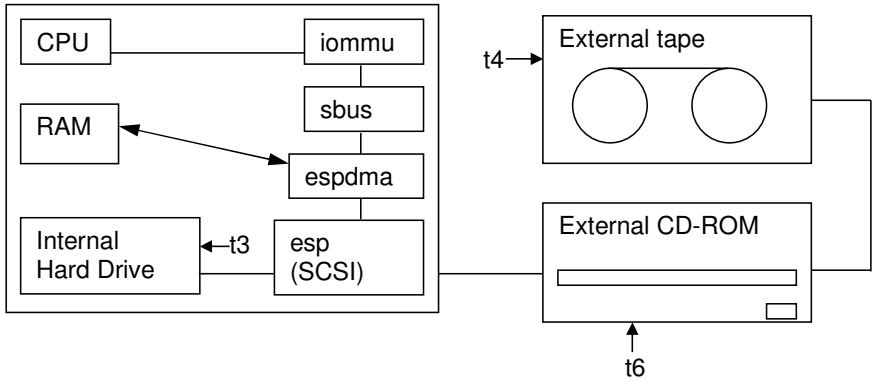
  Device 3    ( Secondary Slave )
              Not Present
```

If you use the **probe-scsi** or **probe-ide** command and don't see a device that you think should be displayed, perform the following checks:

- Check to see if power is on (for external device)
- Check all cables for good connections
- Check last device for SCSI termination
- Check all device targets for conflicting numbers

4.5 Displaying Device Alias Names

When PROM accesses devices, it does so by using the path from the CPU to the device itself. This path is quite long and difficult to type. Here is a look at the path from the CPU to the devices for SCSI disks:



Notes:

Notes:

To get from the CPU to the SCSI controller, the system goes through the following:

iommu: Input/Output Memory Management Unit

sbus: System bus

espdma: SCSI Direct Memory Access

esp: The SCSI device itself

To see the complete path for devices (and their nicknames), use the **devalias** command:

```
ok devalias
{output omitted}
disk            /iommu/sbus/espdma@f,400000/esp@f,80000/sd@3,0
cdrom           /iommu/sbus/espdma@f,400000/esp@f,80000/sd@6,0
{output omitted}
```

Notes:

Displaying device tree

While the **devalias** command shows the device aliases, the **show-devs** command will show what devices are attached to the system:

```
ok show-devs
/os-io
/SUNW,UltraSPARC-IIe@0,0
/pci@1f,0
/virtual-memory
/memory@0,0
/aliases
/options
/openprom
/chosen
/packages
/pci@1f,0/pci@5
{remaining output omitted}
```

Note: The example above is taken from a system that has PCI devices while the example on the previous page is taken from a system that has SCSI devices.

Notes:

4.6 Creating Custom Device Alias Names

To create your own custom device alias names, use the **nvalias** command:

```
ok nvalias hd2 /iommu/sbus/espdma@f,400000/esp@f,80000/sd@1,0
```

You can remove this device alias name at a later date with the **nvunalias** command:

```
ok nvunalias hd2
```

An easy method to do this is to use the **show-disks** command. This command will display devices that you might want to have device aliases for:

```
ok show-disks
a) /pci@1f,0/ide@d/cdrom
b) /pci@1f,0/ide@d/disk
q) NO SELECTION
Enter Selection, q to quit:: b
/pci@1f,0/ide@d/disk has been selected.
Type ^Y ( Control-Y ) to insert it in the command line.
e.g. ok nvalias mydev ^Y
             for creating devlias mydev for
/pci@1f,0/ide@d/disk
ok
```

4.7 Displaying PROM Variables

PROM variables can be used to control how the system boots. To view the current settings of the PROM variables, use the **printenv** command:

```
ok printenv
Parameter Name      Value       Default

{output omitted}
input-device        ttya        keyboard
diag-file
diag-device         net         net
boot-file
boot-device         disk        disk net
auto-boot?          true        true
{output omitted}
```

Each line describes one variable. The first column tells the variable name, the second is what the variable is currently set to, and the third is the factory preset for that variable.

Notes:

Notes:

You can also display PROM variables while the system is booted up by using the **eeprom** command:

```
# eeprom | more
tpe-link-test?=true
output-device=screen
input-device=keyboard
keyboard-click?=false
keymap: data not available.
ttyb-rts-dtr-off=false
ttyb-ignore-cd=true
ttya-rts-dtr-off=false
ttya-ignore-cd=true
ttyb-mode=9600,8,n,1,-
ttya-mode=9600,8,n,1,-
fcode-debug?=false
local-mac-address?=false
screen-#columns=80
screen-#rows=34
selftest-#megs=1
scsi-initiator-id=7
silent-mode?=false
{remaining output omitted}
```

4.8 Setting PROM Variables

To modify PROM variables, use the **setenv** command. The following will change the **boot-device** variable to cdrom. When the system boots, it will boot off the cdrom instead of the internal disk:

```
ok setenv boot-device cdrom
boot-device = cdrom
```

You can set PROM variables while logged in as root with the **eeprom** command:

```
# eeprom boot-device=cdrom
```

Notes:

Notes:

Security Risk

There is a security risk currently on the system. Anyone can perform the following actions to gain superuser access on the system:

1. stop-a
2. boot cdrom -s
3. mount the hard drive's boot partition
4. edit the hard drive's `/etc/shadow` file and remove the root password
5. reboot and log in as root

While this method is good because it will allow you to recover your root password (and fix system files), it also allows hackers to gain access to your system.

One method of making this more difficult is to provide a password at the PROM level. If you set this password, anyone who performs a "stop-a" will need to know the password in order to boot the system from the cdrom.

To set this password, perform the following commands:

```
ok setenv security-mode full
security-mode = full
ok setenv security-password *****
security-password =
```

Note: please do NOT perform these commands on classroom machines.

Recovering PROM password

If you forget the PROM password, there are only a few methods of recovering your system:

1. Physically remove the NVRAM chip from the computer and leave it out overnight. It will lose all of its power and the variables revert back to the factory presets.
2. Physically remove the motherboard's battery and leave it out overnight. This has the same result as option #1.
3. "Short out" the NVRAM chip (NOT RECOMMENDED).
4. Return the motherboard to the manufacturer.

Notes:

4.9 Resetting PROM Variables to Factory Preset

To set "all" of the variables back to the factory presets, use the **set-defaults** command:

```
ok printenv
{output omitted}
boot-device             disk                disk net
auto-boot?              true                true
{output omitted}
ok set-defaults
ok printenv
{output omitted}
boot-device             disk net            disk net
auto-boot?              true                true
{output omitted}
```

Notes:

- This command doesn't reset ALL variables. To reset the security variables (security-mode and security-password), you will have to use the **setenv** command.
- An alternative method is to use stop-n.

Notes:

Notes:

4.10 Booting from PROM

The **boot** command with no arguments will boot the system using the device specified by the boot-device variable:

```
ok printenv boot-device
boot-device = disk
ok boot
{output omitted}
```

To boot off a different device, use the alias name for the device:

```
ok devalias
{output omitted}
disk            /iommu/sbus/espdma@f,400000/esp@f,80000/sd@3,0
cdrom           /iommu/sbus/espdma@f,400000/esp@f,80000/sd@6,0
{output omitted}
ok boot cdrom
{output omitted}
```

Common options:

-s Boot to single-user run level (more on run levels in Unit 11).
-v Verbose boot.
-r Reconfigure boot. Required whenever adding devices to the system.
-a "Ask me" boot. When the system boots, it reads system files. This option prompts you as to where to read the files.

4.11 Recovering from a stop-a

Performing a "stop-a" can cause problems on the hard drive (which we will see in Unit 15). There are two methods of recovering from a stop-a; the first method will take you back to your UNIX session and the second method will attempt to fix the problems and reboot the system:

To return to your UNIX session after a stop-a, use the **go** command:

```
ok go
{output omitted}
```

Note: this method may not always work.

To attempt to fix the problems that stop-a creates and reboot the system, use the **sync** command:

```
ok sync
{output omitted}
```

Notes:

If the system hangs, perform the following steps, in order, to try to bring the system back (once a method works, don't try any more methods):

- Try the control character: ^c, ^d, ^z to stop the program that froze.
- Try opening another terminal window and use the **ps** & **kill** (shown in a later Unit) commands to stop the frozen program.
- Try using the **telnet** or **rlogin** commands to log into the system remotely. Then use the **ps** and **kill** commands.
- Perform a stop-a, followed by a **sync**.
- If all else fails, manually power off the system. When the system boots, you may have to fix the problems on the hard disk (shown in a later Unit).

Notes:

4.12 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #4

Web sites

System Administration Guide: Basic Administration, Chapter 10. Using the Boot PROM:
http://docs.sun.com/app/docs/doc/817-1985/6mhm8o5nu?a=view

Man pages

none

4.13 Summary of Commands and Files

Command	Description
.enet-addr	Displays Ethernet address information
banner	Displays basic system information
boot	Boots the system from PROM
devalias	Displays PROM device aliases
eeprom	Solaris command that displays and sets PROM variables
go	Resumes a suspended system
help	Displays basic help information for PROM
nvalias	Creates a device alias
nvunalias	Removes a device alias
printenv	Displays PROM variables
probe-ide	Displays the IDE devices attached to system
probe-scsi	Displays the SCSI devices attached to system
prtdiag	Displays the PROM version information
set-defaults	Sets PROM variables back to factory preset values
setenv	Sets PROM variables
show-devs	Displays the devices that are attached to the system
show-disks	Displays the disks that are attached to the system
sync	Flushes memory to hard drive and boots the system

4.14 Lab Exercises

Scenario #1: You want the boot variable **keyboard-click?** set to true. Bring the system down to the PROM level, set this parameter to true, and then boot the system.

Scenario #2: Imagine your system has hung. Use stop-a and fix the problem.

Scenario #3: You added a CDROM to your system. Bring down the system to PROM and perform a reconfigure boot.

Scenario #4: Determine how many internal hard drives you have.

Scenario #5: Return all boot parameters back to their original factory preset settings.

Unit Five
Software administration

Unit topics

		Page #
5.1	Displaying Package Information	135
5.2	Removing Packages	138
5.3	Adding Packages via Scripts or the Command Line	140
5.4	Spooling Packages	143
5.5	Installing from Spooled Package	145
5.6	Removing a Spooled Package	146
5.7	The pkgchk Command	147
5.8	Review of the /var/sadm Files	151
5.9	Additional Resources	152
5.10	Summary of Commands and Files	153
5.11	Lab Exercises	154

5.1 Displaying Package Information

The **pkginfo** command will provide information about the packages that are either on the local hard drive or on a CD-ROM.

To list all packages that are installed on the system, use the **pkginfo** command with no options or arguments:

```
# pkginfo | more
system      FJSVcpc       Fujitsu CPU Performance Counter package
system      FJSVcpcx      Fujitsu CPU Performance Counter package (64-bit)
system      FJSVhea       Fujitsu SunOS Header Files
system      FJSVmdb       Fujitsu Platform Modular Debugger
system      FJSVmdbx      Fujitsu Platform Modular Debugger (64-bit)
system      FJSVvplr      Fujitsu platform links
system      FJSVvplu      Fujitsu usr/platform links
system      IPLTadcon     Administration Server Console
system      IPLTadman     Administration Server Documentation
system      IPLTadmin     Administration Server
system      IPLTcons      Console Client Base
system      IPLTdscon     Directory Server Console
{remaining output omitted}
```

To list the details of a particular package, use the **-l** option and give the package name as the argument:

```
# pkginfo -l SUNWnfscr
   PKGINST:  SUNWnfscr
      NAME:  Network Filesystem (NFS) client support (Root)
  CATEGORY:  system
      ARCH:  sparc
   VERSION:  11.9.0,REV=2002.04.06.15.27
   BASEDIR:  /
    VENDOR:  Sun Microsystems, Inc.
      DESC:  Network Filesystem (NFS) client support (Root)
    PSTAMP:  crash20020406153651
  INSTDATE:  Jun 13 2003 16:10
   HOTLINE:  Please contact your local service provider
    STATUS:  completely installed
     FILES:       21 installed pathnames
                   9 shared pathnames
                   3 linked files
                  11 directories
                   6 executables
                2652 blocks used (approx)
```

To list package information from the CD-ROM, use the **-d** option and specify the directory that contains the packages:

```
# pkginfo -d /cdrom/cdrom0/s0/Solaris_9/Product | more
system      FJSVhea           Fujitsu SunOS Header Files
system      FJSVvplr.u        Fujitsu platform links
system      FJSVvplr.us       Fujitsu platform links
system      FJSVvplu.u        Fujitsu usr/platform links
system      FJSVvplu.us       Fujitsu usr/platform links
{remaining output omitted}
```

Note: the directory location will vary from CD to CD. Most of the time, documentation that comes with the software will indicate where the packages are stored. If not, you may just have to go exploring.

5.2 Removing Packages

To remove a package from the system, use the **pkgrm** command:

```
# pkgrm SUNWatfsu

The following package is currently installed:
  SUNWatfsu      AutoFS, (Usr)
                 (sparc) 11.9.0,REV=2002.04.06.15.27

Do you want to remove this package? [y,n,?,q] y

## Removing installed package instance <SUNWatfsu>
## Verifying package dependencies.
## Processing package information.
## Removing pathnames in class <none>
/usr/sbin/automount
/usr/sbin <shared pathname not removed>
/usr/lib/fs/autofs/unshare
/usr/lib/fs/autofs/share
/usr/lib/fs/autofs/mount
/usr/lib/fs/autofs/fn.so.2
/usr/lib/fs/autofs/dfshares
/usr/lib/fs/autofs/automount
/usr/lib/fs/autofs
/usr/lib/fs <shared pathname not removed>
```

```
/usr/lib/autofs/automountd
/usr/lib/autofs
/usr/lib/adb/fnnode
/usr/lib/adb <shared pathname not removed>
/usr/lib <shared pathname not removed>
/usr <shared pathname not removed>
## Updating system information.

Removal of <SUNWatfsu> was successful.
```

Notes:

- If there are other packages that depend on this package (dependencies), then you will be warned of these dependencies.
- The `/var/sadm/install/contents` file is updated when the **pkgrm** command is used.

5.3 Adding Packages via Scripts or the Command Line

We will discuss three methods of installing and removing software:

1. Installation scripts
2. Command line method

If the software comes with an installation script, use that to install the software. Typically the installation script will be called “install****”.

To add packages via the command line, use the **pkgadd** command:

```
# pkgadd -d /cdrom/cdrom0/s0/Solaris_9/Product SUNWatfsu

Processing package instance <SUNWatfsu> from
</cdrom/sol_9_sparc/s0/Solaris_9/Product>

AutoFS, (Usr)
(sparc) 11.9.0,REV=2002.04.06.15.27
{output omitted}

Do you want to install these conflicting files [y,n,?,q] y
## Checking for setuid/setgid programs.
```

This package contains scripts which will be executed with super-user permission during the process of installing this package.

Do you want to continue with the installation of <SUNWatfsu> [y,n,?] **y**

Installing AutoFS, (Usr) as <SUNWatfsu>

Installing part 1 of 1.
483 blocks

Installation of <SUNWatfsu> was successful.

Notes:

- The `/var/sadm/install/contents` file is updated when the **pkgadd** command is used.
- If you don't specify a package name when using the **pkgadd** command, the system will display the packages available and allow you to choose one.

If you get errors when trying to install software, check the following:

- Make sure you are logged in as root (most software can only be installed by root)
- Make sure you have enough room in the partition the software is being installed in (check your documentation or run **pkginfo -l** on package and look at basedir).
- Look at the error message that is displayed on screen and compare the error to your documentation.

5.4 Spooling Packages

Before you learn how to spool, it's helpful to understand why you would want to spool.

Spooling is the process of taking a package **in its original format** from the CD-ROM and placing it (copying) onto the hard drive. After spooling a package onto the hard drive, the package can be installed, either on the local machine or on another host on the network.

Use the **-s** option to the **pkgadd** command to spool a package onto the hard drive:

```
# pkgadd -s spool -d /cdrom/cdrom0/s0/Solaris_9/Product SUNWatfsu
Transferring <SUNWatfsu> package instance
```

The keyword **spool** is a nickname for the directory `/var/spool/pkg`. If you want the spooled package go to another directory, specify the directory name instead of **spool**:

```
# pkgadd –s /export/spool –d /cdrom/cdrom0/s0/Solaris_9/Product
SUNWatfsu
Transferring <SUNWatfsu> package instance
```

Note: make sure that `/export/spool` exists. If `/export` exists and `spool` does not, you will end up with SUNWatfsu being spooled under `/export` that is

called "spool"!

5.5 Installing from Spooled Package

To install from a spooled package use the **-d** option and specify where the spooled package resides:

```
# pkgadd -d /var/spool/pkg SUNWatfsu

Processing package instance <SUNWatfsu> from </var/spool/pkg>
{remaining output omitted}
```

Note: the keyword “spool” can also be used with the **-d** option of the **pkgadd** command.

5.6 Removing a Spooled Package

Many Administrators remove spooled packages directly by using the **rm -r** command. While not a bad method, it is better to use the following method to remove spooled packages:

```
# rm -r /var/spool/pkg/ SUNWatfsu
{no output}
```

Remember, whenever software is installed via the **pkgadd** command, the file `/var/sadm/install/contents` if updated. This includes when you spool packages onto the hard drive. The **rm** command does not update this "contents" file.

To properly remove a package, use the **-s** option to the **pkgrm** command:

```
# pkgrm -s /var/spool/pkg SUNWatfsu
{output omitted}
```

5.7 The pkgchk Command

The **pkgchk** command compares current attributes of files against the attributes for these files in the `/var/sadm/install/contents` file. If there are discrepancies, they are reported.

To check all of the files that were installed with a package, use **pkgchk** followed by the package name:

```
# pkgchk SUNWatfsr
ERROR: /etc/auto_home
   modtime <04/06/02 03:06:48 PM> expected <06/13/03 04:16:46 PM> actual
ERROR: /etc/auto_master
   modtime <04/06/02 03:06:47 PM> expected <06/13/03 04:16:47 PM> actual
```

Note: No output indicated that there are no discrepancies.

To check an individual file, use the **-p** option:

```
# pkgchk -p /etc/inet/hosts
ERROR: /etc/inet/hosts
   file size <46> expected <292> actual
   file cksum <3463> expected <23035> actual
```

Even if there are discrepancies, it doesn't mean there are problems on the system. The `/etc/inet/hosts` file reported two errors: file size and cksum had changed size. The file size changed because you edited the file. The cksum is a number that represents a combination of attributes (including the file size).

Errors such as file size changes in binary files and the change of ownership of system files may indicate problems (like a hacker) on the system.

Displaying file details

To display detail information regarding the original attributes of a file, use the **-l** option:

```
# pkgchk -l -p /etc/inet/hosts
Pathname: /etc/inet/hosts
Type: editted file
Expected mode: 0444
Expected owner: root
Expected group: sys
Referenced by the following packages:
      SUNWcsr
Current status: installed
```

Displaying files associated with a package

If you want to see a list of all of the files that were installed with the package, use the **-v** option:

```
# pkgchk -v SUNWatfsr
/etc
ERROR: /etc/auto_home
    modtime <04/06/02 03:06:48 PM> expected <06/13/03 04:16:46 PM> actual
/etc/auto_home
ERROR: /etc/auto_master
    modtime <04/06/02 03:06:47 PM> expected <06/13/03 04:16:47 PM> actual
/etc/auto_master
/etc/init.d
/etc/init.d/autofs
/etc/rc0.d
/etc/rc0.d/K41autofs
/etc/rc1.d
/etc/rc1.d/K41autofs
/etc/rc2.d
/etc/rc2.d/S74autofs
/etc/rcS.d
/etc/rcS.d/K41autofs
/kernel
/kernel/fs
/kernel/fs/autofs
```

5.8 Review of the /var/sadm Files

Because they play such an important role in software administration, the files and directories of `/var/pkg` have been included here for a quick review:

File	Purpose
/var/sadm/install/contents	Contains original attributes of installed files and directories
/var/sadm/pkg	Contains subdirectories of each package on the system
/var/sadm/pkg/SUNW*/pkginfo	Contains information regarding the package
/var/sadm/pkg/SUNW*/install/depend	Contains dependencies for this package
/var/sadm/install_data/install_log	Contains a log of what occurred during the installation process (including what packages were successfully installed)

5.9 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #5

Web sites

System Administration Guide: Basic Administration, Chapter 20 Managing Software(Overview):
http://docs.oracle.com/cd/E26505_01/html/E29492/swmgrpkgsconcepts-52619.html

Man pages

pkgadd
pkgchk
pkginfo
pkgrm

5.11 Summary of Commands and Files

Command	Description
pkgadd	Installs packages
pkgchk	Verifies files that were installed
pkginfo	Displays package information
pkgrm	Uninstalls packages

5.12 Lab Exercises

1. Using the command line, perform the following:

   A. Delete a package called **SUNWntpu** (aka, NTP, (Usr))

2. What is the description of the **SUNWxwdem** package?

Command: _____

3. After determining the dependencies for **SUNWxwdem**, remove this package using command line syntax.

Command: _____

4. Spool the **SUNWxwdem** package onto your hard drive.

Command: _____

5. Install the **SUNWxwdem** package using your spooled package.

Command: _____

6. Remove the spooled **SUNWxwdem** package from your hard drive.

Command: _____

Unit Six
Patches

Unit topics

		Page #
6.1	What are Patches?	156
6.2	Where to Get Patches	157
6.3	Patch Reports	158
6.4	Patch Documentation	159
6.5	Preparing Patches	160
6.6	Installing Patches	161
6.7	The /var/sadm/patch Directory	164
6.8	Displaying installed Patches	165
6.9	Removing Patches	166
6.10	Additional Resources	167
6.11	Summary of Commands and Files	168
6.12	Lab Exercises	169

Notes:

6.1 What are Patches?

Patches are fixes to bugs in software packages. Instead of creating an entirely new revision of a software package, Sun makes scripts called patches that fix problems with software.

A few warnings about patches:

1. If it isn't broken, don't try to fix it. In other words, don't install a patch just because it exists; install a patch to fix a problem you are experiencing.

2. A good exception to rule #1 is in regard to security patches and "Recommended Patches". Whenever you are aware of security patches, you should review them and install them unless they might cause problems on your system. In most cases, you should install Sun's Recommended Patches.

3. When installing many patches, try installing one at a time. If the system breaks because of a patch, it will be easier to determine which patch it was if you install only one at a time (example: install one per day).

4. Patches have versions to them. An example of a patch name would be "102837-01" where "102837" is the name of the patch while "01" is the version number to the patch. Make sure when you install a patch you are installing the most current version.

6.2 Where to Get Patches

There are several locations for getting patches:

1. From the Installation CD-ROM
2. From Oracles's web site (www.oracle.com)
3. From the Patch Update CD-ROM (distributed every 6-8 weeks)

Notes:

6.3 Patch Reports

A patch report is a summary of all patches that Oracle recommends you install on your system. This report is available either on the web sites or on the Patch Update CD-ROM.

Read the patch reports carefully and install each patch that Oracle recommends as long you have determined the patch will not cause problems on your system.

Notes:

Notes:

6.4 Patch Documentation

Each patch should come with its own documentation. This documentation will be in the form of a text file.

In Solaris 2.6 and higher the name of the file that contains the documentation is “Install.info”:

```
# more Install.info
{output omitted}
```

In Solaris 2.5 and previous the name of the file that contains the documentation is “README******-**”:

```
# more README102837-01
{output omitted}
```

Note: for some Solaris 2.6 patches there is still a README file.

6.5 Preparing Patches

When you download patches from Oracle they will come in a compressed format. If you are downloading patches for Solaris 7, 8 or 9 the patches will come in zip format. You can uncompress these files with the **unzip** command:

```
# unzip 9_Recommended.zip
{no output}
```

Prior to Solaris 7 the patches were first joined in a tar file and then compressed with the **compress** command. To extract these files, use the following commands:

```
# uncompress 2.6_Recommended.Z
# tar -xf 2.6_Recommeded.tar
```

Notes:

Notes:

6.6 Installing Patches

There are two commands you want to be aware of when installing patches: **patchadd** is the Solaris 2.6 command while **installpatch** is used for Solaris 2.5 and some Solaris 2.6 patches.

Solaris 2.6 and higher - patchadd

```
# patchadd 102181-01
{output omitted}
```

Solaris 2.5 & 2.6- installpatch

```
# cd 106828-01
# ./installpatch .
{output omitted}
New versions of /usr/sbin/patchadd and /usr/sbin/patchrm exist
that provide essential bug fixes and a speed enhancement.
Please install patch 106125 for significant patch
installation improvements.
{output omitted}
Patch number 106828-01 has been successfully installed.
See /var/sadm/patch/106828-01/log for details

Patch packages installed:
  SUNWcsu
```

Notes:

Installing multiple patches

To install more than one patch at a time, use the **-M** option to the **patchadd** command. The first argument after the **-M** option must be the location of the patches:

```
# patchadd -M /var/spool/patch 102181-01 102675-04
```

Saving backout data

By default the original files that the patch modifies are saved in the `/var` filesystem so that the changes can be "undone". The process of "undoing" a patch is called "backing out". Saving the original files can create space issue problems in the `/var` filesystem. To avoid saving the original files, use the **-d** option.

If you want to save the original files but you don't have enough room in the `/var` filesystem, use the **-B** option to place the original files in a different directory:

```
# patchadd -B backout_dir patch_name
```

Notes:

The install_cluster script

If you download the recommended patches (or other patch groups), you can have the system install all of the patches by running the **install_cluster** script. This script will install all of the patches based on the **patch_order** file. The **patch_order** file contains a list of patches.

By default the **install_cluster** script will save the original files. To avoid this, run the **install_cluster** script with the **-nosave** option.

Error messages

Several problems may occur during the installation of patches. The man page for the **patchadd** command has an extensive list of error messages and their meaning.

6.7 The /var/sadm/patch Directory

When a patch is installed, a subdirectory is created for it in the `/var/sadm/patch` directory. That directory contains the patch's documentation and a command to back out (uninstall) the patch:

```
# ls /var/sadm/patch/106828-01
README.106828-01  backoutpatch      log
```

Notes:

Notes:

6.8 Displaying Installed Patches

Solaris 2.6 command that displays current patches is **patchadd -p** while the Solaris 2.5 command is **showrev -p**.

Solaris 2.6 and higher - patchadd -p

```
# patchadd -p
Patch: 106125-06 Obsoletes: Requires: Incompatibles: Packages: SUNWswmt
```

Solaris 2.5 and previous - showrev -p

```
# showrev -p
Patch: 106125-06 Obsoletes:  Requires:  Incompatibles:  Packages: SUNWswmt
```

Note: the **showrev -p** still functions in Solaris 2.6.

Displaying patches for a package

Use the **pkgparam** command to display what patches have been installed on a specific package:

```
# pkgparam SUNWdtdte PATCHLIST
{Output omitted}
```

6.9 Removing Patches

There are two commands you want to be aware of when removing patches: **patchrm** is the Solaris 2.6 command while **backoutpatch** is used for Solaris 2.5 and some Solaris 2.6 patches.

Notes:

Solaris 2.6 and higher - patchrm

```
# patchrm 106828-01
{output omitted}
```

Solaris 2.5 or previous - backoutpatch

```
# cd /var/sadm/patch/106828-01
# ./backoutpatch 106828-01
{output omitted}
New versions of /usr/sbin/patchadd and /usr/sbin/patchrm exist
that provide essential bug fixes and a speed enhancement.
Please install patch 106125 for significant patch
installation improvements.

Checking installed packages and patches...

Patch 106828-01 has been backed out.
```

6.10 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #5

Web sites

System Administration Guide: Basic Administration, Chapter 23 Managing Solaris Patches by Using the patchadd Command (Tasks): http://docs.oracle.com/cd/E26505_01/html/E29492/swmgrpatchtasks-1.html

Man pages

patchadd
pkgparam
pkgrm
showrev

6.11 Summary of Commands and Files

Command	Description
patchadd	Installs a patch on Solaris 2.6 and higher systems
pkgparam	Displays patches applied to a specific package
pkgrm	Removes a patch on Solaris 2.6 and higher systems
showrev	Displays currently installed patches

File	Description
/var/sadm/patch	Patch documentation directory

6.12 Lab Exercises

1. What patches are currently installed on your system?

Command: _____

Scenario: Patches vary from OS release to OS release. It's important to know which patches are available so you can determine which patches should be installed on your particular system. Take a little time to explore the patches on the CD-ROM or those provided by your instructor. Try installing a few and backing out of (removing) a couple.

Unit Seven
Maintaining User Accounts

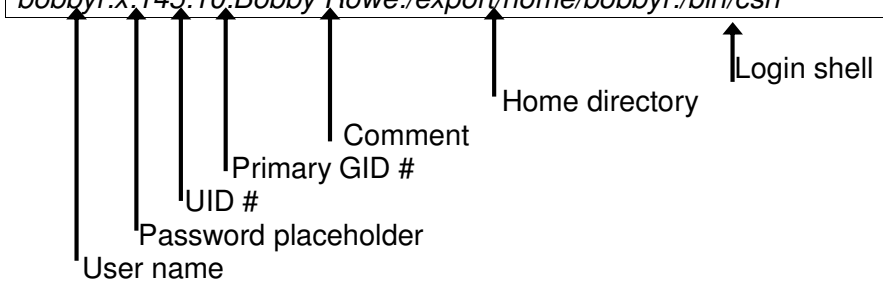
Unit topics

		Page #
7.1	The /etc/passwd File	171
7.2	The /etc/shadow File	173
7.3	The /etc/group File	176
7.4	Using Admintool to Administer Group and User Accounts	**Error! Bookmark not defined.**
7.5	Using smuser & smgroup to Administer Group and User Accounts	178
7.6	Using Solaris Management Console to Administer Group & User accounts	182
7.7	Adding Group Accounts via the Command Line	187
7.8	Deleting Group Accounts via the Command Line	188
7.9	Adding User Accounts via the Command Line	189
7.10	Deleting User Accounts via the Command Line	192
7.11	Additional Resources	193
7.12	Summary of Commands and Files	194
7.13	Lab Exercises	195

7.1 The /etc/passwd File

Some of the user account information is stored in the `/etc/passwd` file:

```
# more /etc/passwd
root:x:0:1:Super-User:/:/sbin/sh
daemon:x:1:1::/:
bin:x:2:2::/usr/bin:
sys:x:3:3::/:
lp:x:71:8:Line Printer Admin:/usr/spool/lp:
smtp:x:0:0:Mail Daemon User:/:
uucp:x:5:5:uucp Admin:/usr/lib/uucp:
listen:x:37:4:Network Admin:/usr/net/nls:
nobody:x:60001:60001:Nobody:/:
noaccess:x:60002:60002:No Access User:/:
bobbyr:x:145:10:Bobby Rowe:/export/home/bobbyr:/bin/csh
```



Notes:

Notes:

Description of the fields

User Name: The user name is what users on the system use to log in and access resources. This name should be in all lower case and no longer than 8 characters long.

Password Placeholder: The password itself is not stored in the `/etc/passwd` file. The password placeholder field (2nd field) was where the password used to be stored (in Solaris 1.x). It is now placed in the `/etc/shadow` file.

UID #: A unique number that the system uses to identify the user. The numbers 1-100 are reserved for system use.

Primary GID #: The user's primary group ID number. All users must belong to at least one group. When a user creates a new file or directory, the group ownership of that new object is the user's primary GID #.

Comment: Can be anything the System Administrator wants to place here.

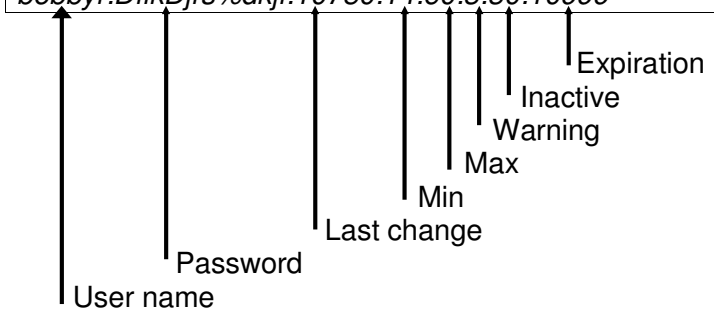
Home Directory: Where the user will be placed when the user logs in.

Login Shell: The default shell for the user.

7.2 The /etc/shadow File

The `/etc/shadow` file contains the encrypted password and information regarding the password (password aging information):

```
# more /etc/shadow
root:4eir.cYfMaG5U:6445::::::
daemon:NP:6445::::::
bin:NP:6445::::::
sys:NP:6445::::::
lp:NP:6445::::::
smtp:NP:6445::::::
uucp:NP:6445::::::
listen:*LK*:::::::
nobody:NP:6445::::::
noaccess:NP:6445::::::
bobbyr:DflkDjrs%dkjf:10780:14:60:3:30:10999
```



Notes:

Notes:

Description of the fields

Password: The password is the encrypted password and can not be "broken" by any user (including the superuser). The superuser can, however, change and delete the user's password. NP means "no password will work" and is a special entry for daemon accounts. *LK* means the account is locked. The System Administrator can unlock it by giving the account a normal password.

Last change: This field is used only by the system. It contains a number that represents the number of days since January 1, 1970 and the day the user last changed their password.

Min: The minimum number of days that must pass before the user can change his/her password.

Max: The maximum number of days that can go by before the user has to change his/her password (or be locked out of the account).

Warning: The number of days prior to being locked out of that the system will issue a warning to the user at login.

Inactive: The number of days that can go by without the user logging in before the account will be locked out due to inactivity.

Expiration: The **useradd** and **admintool** commands allow you to specify a "date" for the account to expire. This is converted to a number that is the number of days from January 1, 1970 and the day the account is supposed to expire.



Notes:

Recovering the /etc/passwd file

If your `/etc/passwd` file has become corrupted, you may be able to recover it partially by using the `/etc/opasswd` file. When you use **useradd**, **usermod**, or **userdel** to modify user accounts, a backup of the old `/etc/passwd` file is created. This file is called `/etc/opasswd`. While it won't contain the latest changes to the `/etc/passwd` file, it does provide a starting point for generating a new file.

Recovering the /etc/shadow file

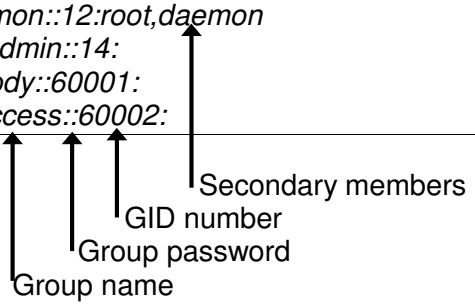
If your `/etc/shadow` file has become corrupted, you may be able to recover it partially by using the `/etc/shadow` file. When you use **useradd**, **usermod**, **userdel**, or **passwd** to modify user accounts, a backup of the old `/etc/shadow` file is created. This file is called `/etc/oshadow`. While it won't contain the latest changes to the `/etc/shadow` file, it does provide a starting point for generating a new file.

If the `/etc/shadow` file has become corrupted, you can recreate one with the **pwconv** command. This command will use your `/etc/passwd` file to recreate the `/etc/shadow` file.

7.3 The /etc/group File

The file that stores all group account information is the `/etc/group` file. Each line contains four fields separated by colons and defines one group:

```
# more /etc/group
root::0:root
other::1:
bin::2:root,bin,daemon
sys::3:root,bin,sys,adm
adm::4:root,adm,daemon
uucp::5:root,uucp
mail::6:root
tty::7:root,tty,adm
lp::8:root,lp,adm
nuucp::9:root,nuucp
staff::10:
daemon::12:root,daemon
sysadmin::14:
nobody::60001:
noaccess::60002:
```



Notes:

Notes:

Description of the fields

Group name: The group name is used by users to refer to groups. The system itself uses the GID number. The group name should be in all lower case letters and be no more than 8 characters long.

Group password: The command **newgrp** will change the user's primary group by opening a new shell with a new primary group number for the user. This group password field can be used to restrict or limit users from using this command.

GID number: A unique number that the system uses to distinguish different groups. GID numbers 1-100 are reserved for system use.

Secondary Member list: This is a list of the people who are secondary members of the group.

Notes:

7.4 Using smuser & smgroup to Administer Group and User Accounts

The **smuser** and **smgroup** are command line equivalents of the Solaris Management Console (SMC). Note: These commands are located in the `/usr/sadm/bin` directory which is not typically part of the root's PATH (hence the complete pathname in the following examples).

To add a group

```
# /usr/sadm/bin/smgroup add -- -g 900 -m guest -n sales
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::

Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from spare
Login to spare as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from spare was successful.
```

Notes:

To modify a group



```
# /usr/sadm/bin/smgroup modify -- -n sales -m root
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from spare
Login to spare as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from spare was successful.
```

To delete a group

```
# /usr/sadm/bin/smgroup delete -- -n sales
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.usermgr.cli.group.UserMgrGroupCli from spare
Login to spare as user root was successful.
Download of com.sun.admin.usermgr.cli.group.UserMgrGroupCli from spare was successful.
```



Notes:

To add a user

```
# /usr/sadm/bin/smuser add -- -n joe -g staff -d /export/home/joe -s /bin/tcsh -u 985 -c "Joe Smith"
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from spare
Login to spare as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from spare was successful.
```

Common options to the **smuser add** command:

-u Specify UID #
-g Specify primary GID #
-G Specify secondary GID #
-d Specify user's home directory
-s Specify user's shell (example above is the TC shell)
-c Specify a comment

Notes:

To modify a user

```
# /usr/sadm/bin/smuser modify -- -n joe -c "Joe Smyth"
Authenticating as user: root

Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from spare
Login to spare as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from spare was successful.
```

To delete a user

```
# /usr/sadm/bin/smuser delete -- -n joe
Authenticating as user: root

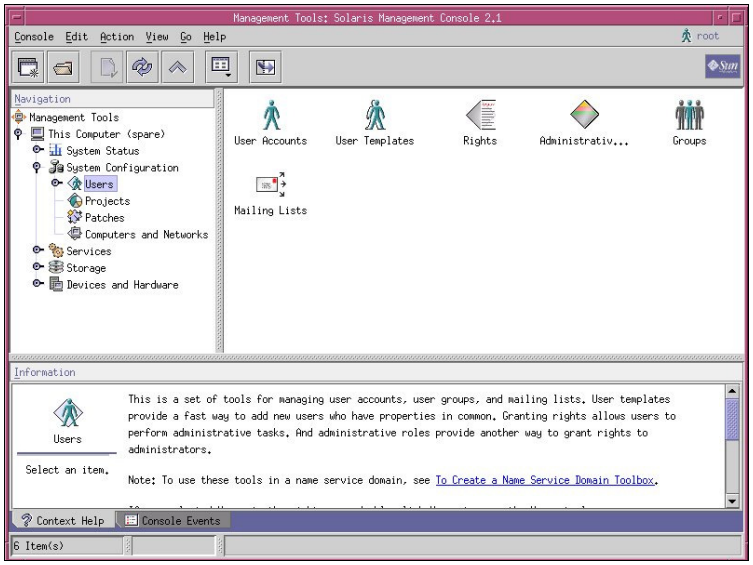
Type /? for help, pressing <enter> accepts the default denoted by [ ]
Please enter a string value for: password ::
Loading Tool: com.sun.admin.usermgr.cli.user.UserMgrCli from spare
Login to spare as user root was successful.
Download of com.sun.admin.usermgr.cli.user.UserMgrCli from spare was successful.
```

7.6 Using Solaris Management Console to Administer Group & User Accounts

The Solaris Management Console (SMC) is the preferred GUI tool in Solaris 9 to administer users and groups. To start the SMC, run the **smc** command:

```
# smc
```

To administer users and groups, first expand “This computer”, then “System Configuration” and then “Users”:



Notes:

Notes:

Add a new user using the wizard

To add a new user using the wizard, click on the “User Accounts” icon and then click “Action” + “Add User” + “with wizard…”:



You can also create multiple users with the wizard.

Add a new template

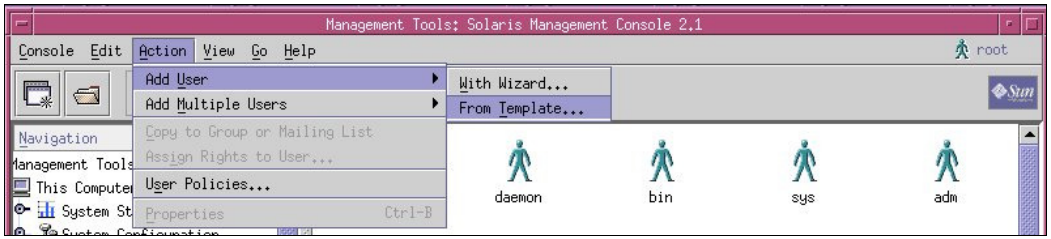
If you find yourself creating similar user accounts on a regular basis, you can create a template to make this process easier. Click on the “User Templates” icon and then “Action” + “Add User Template…”



Notes:

Add a new user using a template

To add a new user account using an existing template, click on the “User Accounts” icon and then click “Action” + “Add User” + “with template…”:



Modify a user account

To modify a user account, either double-click on the account icon or click on the account icon and then click “Action” + “Properties”.

Delete a user account

To delete a user account, either right-click on the account icon and click “delete” or click on the account icon and then click “Edit” + “Delete”.

Rights and Administrative Roles

Rights grant additional access to users. There are predefined rights that you can make use of immediately, or you can create your own rights.

Administrative Roles are special accounts that make use of Rights.

A detailed discussion of Rights and Administrative Roles is beyond the scope of this class. To learn more about Rights and Administrative Roles, attend the "UNIX Administration II" class.

Notes:

Notes:

Add a group account

To add a new group account using an existing template, click on the “Groups” icon and then click “Action” + “Add Group…”



Modify a group account

To modify a group account, either double-click on the account icon or click on the account icon and then click “Action” + “Properties”.

Delete a group account

To delete a group account, either right-click on the account icon and click “delete” or click on the account icon and then click “Edit” + “Delete”.

7.7 Adding Group Accounts via the Command Line

To add group accounts on the command line, user the **groupadd** command:

```
# groupadd -g 120 class
{No output}
```

The **-g** option allows you to specify group number.

Note: You can modify group accounts with the **groupmod** command (often it is easier to modify the `/etc/group` file directly).

Notes:

Notes:

7.8 Deleting Group Accounts via the Command Line

To delete a group account from the command line, use the **groupdel** command:

```
# groupdel class
{No output}
```

Note: if you delete a group and a file is “owned” by that group, the group number will be listed when you use the **ls -l** command:

```
# groupadd -g 111 test
# cd /tmp
# ls
{No output}
# touch sample_file
# chgrp test sample_file
# ls -l
-rw-r--r--   1 root     test          0 Feb 18 08:11 sample_file
# groupdel test
# ls -l
-rw-r--r--   1 root     111           0 Feb 18 08:11 sample_file
```

7.9 Adding User Accounts via the Command Line

To add user accounts from the command line, user the **useradd** command:

```
# useradd -u 125 -g 10 -G 14 -d /export/home/bob -m -s /bin/csh bob
6 blocks
```

Common options to the **useradd** command:

-u Specify UID #
-g Specify primary GID #
-G Specify secondary GID #
-d Specify user's home directory
-m Makes user's home directory (required if home directory does not exist)
-s Specify user's shell (example above is the C shell)
-c Specify a comment
-e Specify expiration date for password (see man page for format examples)
-f Specify the Min and Max inactive field
-k Specify a directory whose contents will be copied into the user's home directory

Notes:

Notes:

The "quick and dirty" method to create an account

```
# useradd steve
{No output}
```

When you use this method, the system "fills in the blanks" with default information. To list the default information, use the **-D** option to the **useradd** command:

```
# useradd -D
group=other,1  basedir=/home  skel=/etc/skel
shell=/sbin/sh  inactive=0  expire=
```

Note: Other options can be used in conjunction with the **-D** option to override its defaults.



Note: You can modify user accounts with the **usermod** command.

Notes:

After you create a user account, be sure to assign a password to the account with the **passwd** command:

```
# passwd bob
# passwd bob
New password: djh7rDi
Re-enter new password: djh7rDi
```

More details regarding passwords will be discussed in a later Unit.

Notes:

7.10 Deleting User Accounts via the Command Line

To delete user accounts from the command line, use the **userdel** command:

```
# userdel bob
{No output}
```

The method that is shown above WILL NOT delete the user's home directory. To remove the user's home directory, use the **-r** option to the **userdel** command:

```
# userdel -r steve
{No output}
```

7.11 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #12

Web sites

System Administration Guide: Basic Administration, Chapter 4. Managing User Accounts and Groups (Overview): http://docs.oracle.com/cd/E26505_01/html/E29492/userconcept-97366.html

System Administration Guide: Basic Administration, Chapter 5. Managing User Accounts and Groups (Tasks): http://docs.oracle.com/cd/E26505_01/html/E29492/usersetup-92366.html

Man pages

admintool	groupmod	smc	useradd
groupadd	passwd	smgroup	userdel
groupdel	pwconv	smuser	usermod

7.12 Summary of Commands and Files

Command	Description
admintool	GUI-based tool to administer users and groups
groupadd	Adds group accounts
groupdel	Deletes group accounts
groupmod	Modifies group accounts
passwd	Sets a user account password
pwconv	Creates a `/etc/shadow` file based on entries in the `/etc/passwd` file
smc	GUI-based tool to administer users and groups
smgroup	Command-based utility to administer groups
smuser	Command-based utility to administer users
useradd	Adds user accounts
userdel	Deletes user accounts
usermod	Modifies user accounts

File	Description
/etc/group	Contains group account information
/etc/passwd	Contains user account information
/etc/shadow	Contains user password information

7.13 Lab Exercises

1. Add the following group accounts using the method indicated

Method	Groupname	GID	Members
SMC	webdev	2000	root
SMC	payroll	2001	
smgroup or groupadd	acct	2002	bin
smgroup or groupadd	prog	2003	

2. Add the following user accounts using the method indicated:

Method	Username	UID	Primary GID	Secondary GID	Shell	Password
SMC	bob	1020	10		C	test
SMC	sue	1021	10	2000	Korn	test
smuser or useradd	ted	1022	10	2002 & 14	Korn	test
smuser or useradd	nick	1023	2003		Bourne	test

3. Lock out nick's account.

4. Change sue's login shell to the C shell by modifying the `/etc/passwd` file.

5. Using SMC, change ted's account so his password will expire in 90 days from today.

6. Add sue to the acct group.

7. Delete nick's account (but not his home directory) via the command line.

Command: _____

8. Delete your bob's account (and his home directory) via SMC.

Unit Eight
System Security

Unit topics

		Page #
8.1	Password Security	198
8.2	Recording Failed Login Attempts	200
8.3	Shifting to Another Account	201
8.4	User Information	203
8.5	Changing the User Ownership of a File	207
8.6	Changing the Group Ownership of a File	209
8.7	Who is Logged In Now?	210
8.8	Who has Logged In the Past?	211
8.9	Controlling Logging In	212
8.10	Controlling su Attempts	214
8.11	Preventing stop-a	216
8.12	Preventing FTP Access	217
8.13	Additional Resources	218
8.14	Summary of Commands and Files	219
8.15	Lab Exercises	220

8.1 Password Security

Many UNIX flavors have rules for passwords. The flavor this manual is written for (Solaris) has the following password rules:

1. Must have at least 6 characters
2. Any character after the 8th is ignored
3. Must have 2 alphabetic characters
4. Must have 1 numeric or special character
5. Must be different than the user's login name, the reverse of the user's login name or a circular shift of the user's login name.

6 Must be different than old password by at least 3 characters.

Note: The root (superuser) on the system it not affected by any rule above except rule #2. The encrypting method only allows up to 8 characters.

Users change their password with the **passwd** command. This command first prompts for the current (old) password then prompts twice for the new password.

Notes:



Commented [KW2]: Rules for passwords can be set in /etc/default/passwd. The encryption method that is used is stored in /etc/security/policy.conf.

Good passwords are hard to create. The password #FW&31z is a good password in the sense that it is fairly complex, but it is not an easy one to remember. Here are a few techniques you can use to come up with a complex, yet easy-to-remember password:

The "sentence" password: Pick a sentence from a book (or make one up) that is easy to remember. For example, "The old, lazy dog ate his food." Now take the first letter of each word of the sentence (you might want to include punctuation also) and there is your password (To,ldah. in this case).

The "flip-flop" password: Take a word that is six or seven characters. Split it in half and flip the second half in front of the first half. Lastly, add a number or special character in the middle before you rejoin the two halves. For example, the word "windows" could become "dows4win".

The "Translation" password: This one takes a bit of practice. The basic concept is to take a word and substitute the letters for similar characters on the keyboard. For example, take the word "password". The "o" is similar to "0" (zero), so substitute "o" for "0" and you get "passw0rd". The "s" looks similar to a "5", so you get "pa55w0rd". The "w" looks similar to the "~" (or so some people think), so you get "pa55~0rd". You can probably make this password even more complex by making the first and last letters capitals, but even the small changes made so far makes it fairly complex.

Notes:

8.2 Recording Failed Login Attempts

By default, Solaris does not record failed login attempts. You can have Solaris record these failures by creating the `/var/adm/loginlog` file:

```
# touch /var/adm/loginlog
```

This file won't record every failed attempt. If a user attempts to log in five times and fails, an entry will be made in the `/var/adm/loginlog` file:

```
# more /var/adm/loginlog
bob:/dev/pts/2:Tue Nov 13 15:50:28 2001
bob:/dev/pts/2:Tue Nov 13 15:50:36 2001
bob:/dev/pts/2:Tue Nov 13 15:50:42 2001
bob:/dev/pts/2:Tue Nov 13 15:50:49 2001
bob:/dev/pts/2:Tue Nov 13 15:50:55 2001
```

Notes:

8.3 Shifting to Another Account

Sometimes you will want to be “another user” in order to test a user’s interface. To switch to another user without logging out, use the **su** command:

```
# su bob
sun%
```

Note: the prompt changed to “sun%” because bob’s default shell is the C shell (# is the Bourne Shell's default prompt for the root user).

There are two advantages of the **su** command over logging off and logging into another account:

1. It is quicker
2. As the root user, you don’t have to know the user’s password to su to that account

Notes:

Notes:

The **su** command itself allows you to become another user, but you retain your interface (your aliases, your history, and your variables). To shift to another user and implement that user's interface, use the **-** (hyphen) option to the **su** command:

```
# su - bob
sun%
```

To return back to your original shell, use the **exit** command:

```
sun% exit
#
```

8.4 User Information

There are several commands that give you information regarding who you are logged in as: **id**, **groups**, and **who am i.**

Who you are now

The **id** command will tell you who you are right now:



```
# id
uid=0(root) gid=1(other)
# su - bob
# id
uid=127(bob) gid=1(other)
```

You can also use the BSD command **whoami**:

```
# /usr/ucb/whoami
root
# su - bob
# /usr/ucb/whoami
bob
```

Notes:

Notes:

What groups you belong to

The **id** command by itself will tell you who you are and what your primary group is. To see secondary groups also, either use the **-a** option to the **id** command or use the **groups** command:

```
# id –a
uid=0(root) gid=1(other) groups=1(other),0(root),2(bin),3(sys),
4(adm),5(uucp),6(mail),7(tty),8(lp),9(nuucp),12(daemon)
# groups
other root bin sys adm uucp mail tty lp nuucp daemon
```

The **groups** command accepts a username as an argument, so you can see the groups of another user:

```
# groups bob
other staff
```

Notes:

Who you were originally

When you use the **su** command a lot, it gets confusing as to who you were originally. The command **who am i** tells you who your were originally:

```
# id
uid=0(root) gid=1(other)
# su - bob
# id
uid=127(bob) gid=1(other)
# who am I
root      pts/3       Feb 18 11:39
```

This is important because some commands (such as **passwd** and **mail**), run as the "original" user, not the current user:

```
# id
uid=127(bob) gid=1(other)
# who am I
root      pts/3       Feb 18 11:39
# passwd
passwd:  Changing password for root
passwd: Permission denied
passwd: Can't change local passwd file

Permission denied
```

Notes:

The finger command

The **finger** command will provide you useful user information:

```
# finger bob
Login name: bob
Directory: /export/home/bob          Shell: /usr/local/bin/tcsh
On since Nov 20 11:55:11 on pts/4 from blade
No unread mail
No Plan.
```

The whodo command

The **whodo** command is useful to determine what a specific user is currently doing on the system:

```
# whodo bob
Tue Nov 20 12:35:07 PST 2001
tan

pts/4        bob   11:55
   pts/4        8253    0:00 tcsh
   pts/4        8258    0:00 tcsh
   pts/4        8299    0:00 grep
```


© 2014 One Course Source

8.5 Changing the User Ownership of a File

To change the ownership of a file, use the **chown** command:

```
# ls -l /etc/skel
total 6
-rw-r--r--   1 root     sys          124 Dec  2 10:06 local.cshrc
-rw-r--r--   1 root     sys          575 Oct 25  1995 local.login
-rw-r--r--   1 root     sys          560 Oct 25  1995 local.profile
# chown bob /etc/skel/local.login
# ls -l /etc/skel
total 6
-rw-r--r--   1 root     sys          124 Dec  2 10:06 local.cshrc
-rw-r--r--   1 bob      sys          575 Oct 25  1995 local.login
-rw-r--r--   1 root     sys          560 Oct 25  1995 local.profile
```

The only user that can change the ownership of a file is the root user. Even if you own the file, you can not give its ownership to another user. You can modify the system so that regular users can change the ownership of a file by putting the line “set rstchown=0” in the `/etc/system` file and rebooting the machine. This also affects the **chgrp** command (see the next section).

Notes:

Notes:

To perform a recursive (an entire directory structure) ownership change, use the **-R** option to the **chown** command:

```
# ls -l /etc/skel
total 6
-rw-r--r--   1 root     sys          124 Dec  2 10:06 local.cshrc
-rw-r--r--   1 bob      sys          575 Oct 25  1995 local.login
-rw-r--r--   1 root     sys          560 Oct 25  1995 local.profile
# chown -R bob /etc/skel
# ls -l /etc/skel
total 6
-rw-r--r--   1 bob      sys          124 Dec  2 10:06 local.cshrc
-rw-r--r--   1 bob      sys          575 Oct 25  1995 local.login
-rw-r--r--   1 bob      sys          560 Oct 25  1995 local.profile
```

This option is very useful when a user leaves a company. You can delete the user's account and keep the home directory so when a new user "takes over" the job responsibilities of the old user, you can just give ownership of the home directory to the new user.

8.6 Changing the Group Ownership of a File

To change the group ownership of a file, you must own the file and be a member of the group you are changing ownership to. You can modify the system so that all users can change the group ownership of a file by putting the line "set rstchown=0" in the `/etc/system` file and rebooting the machine.

The root user can also use the **chgrp** command:

```
# ls -l /etc/skel
total 6
-rw-r--r--   1 bob      sys        124 Dec  2 10:06 local.cshrc
-rw-r--r--   1 bob      sys        575 Oct 25  1995 local.login
-rw-r--r--   1 bob      sys        560 Oct 25  1995 local.profile
# chgrp other /etc/skel/local.login
# ls -l /etc/skel
total 6
-rw-r--r--   1 bob      sys        124 Dec  2 10:06 local.cshrc
-rw-r--r--   1 bob      other      575 Oct 25  1995 local.login
-rw-r--r--   1 bob      sys        560 Oct 25  1995 local.profile
```

Notes:

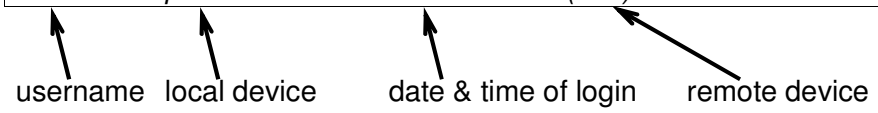
- The **chgrp** command also supports the **-R** option for recursive changes.
- You can use the **chown** command to modify both use and group ownership by using the following syntax: **chown user:group filename**

Notes:

8.7 Who is Logged In Now?

To see which users are currently logged into the system, use the **who** command:

```
# who
bob      console   Feb  1 09:26     (:0)
bob      pts/2     Feb  1 09:26     (:0.0)
root     pts/3     Feb 18 11:39     (sun)
```



Notes:

8.8 Who has Logged In the Past?

The **last** command will display a log of who has logged into the system and where they logged in from:

```
# last
bob      console     :0      Thu Dec  3 10:50 - down  (59+22:33)
reboot   system boot         Thu Dec  3 10:49
root     console     :0       Thu Dec  3 10:43 - 10:48  (00:04)
reboot   system boot         Thu Dec  3 10:42
root     console              Wed Dec  2 15:15 - down   (19:27)
reboot   system boot          Wed Dec  2 10:32

wtmp begins Wed Dec  2 10:32
```

To limit the listing, type the user's name you are interested in seeing:

```
# last root
root     console     :0          Mon Feb  1 09:25 - 09:26  (00:00)
root     console     :0           Thu Dec  3 10:43 - down   (00:05)
root     console                 Wed Dec  2 15:15 - down   (19:27)
```

The file that stored this information is `/var/adm/wtmpx`. If this file is deleted, all information regarding user logins is lost.

Notes:

Notes:

8.9 Controlling Logging In

There are a couple of files on the system that control who (or when) users can log in.

The `/etc/default/passwd` file

This file contains three variables:

Variable	Meaning
MAXWEEKS	**MAXWEEKS=6** Maximum number of weeks ALL USERS passwords are good before they have to change their password.
MINWEEKS	**MINWEEKS=1** Minimum number of weeks ALL USERS must wait to change their password.
PASSLENGTH	**PASSLENGTH=6** Can be set to the values 6, 7 or 8. Will set the minimum number of characters a password must be for typical users on the system.

Note: if the min or max fields are set for a user in the `/etc/shadow` file, those setting will take precedence over the MAXWEEKS and MINWEEKS variables in this file.

Notes:

The `/etc/default/login` file

The important variables of this file are:

Variable	Meaning
CONSOLE	**CONSOLE=/dev/console** If this is set to `/dev/console`, then root can only log in from the console (no telnet). **#CONSOLE=/dev/console** If not set (or commented out), root can log in from anywhere. **CONSOLE=** If set to nothing, root CAN NOT log in at all (but you can **su** to the root account).
PASSREQ	**PASSREQ=YES** If set to YES, then the system will force users to have a password (if they have no password, the system will require them to make a password). **PASSREQ=NO** If set to NO, users can have no password.
PATH	**PATH=/usr/bin** Sets the initial shell PATH variable.
UMASK	**UMASK=022** Sets the initial umask for users.



Notes:

8.10 Controlling su Attempts

The `/etc/default/su` file contains variables that will allow the System Administrator to control **su** attempts on the system:

Variable	Meaning
SULOG	**SULOG=/var/adm/sulog** su attempts will be logged in this file (`/var/adm/sulog`).
CONSOLE	**CONSOLE=/dev/console** If set, any attempt to su to the root account will be reported to the console.

If you wish to not allow anyone to use the **su** command, one method is to change the permissions of the command to 500:

```
# chmod 500 /bin/su
{no output}
```

Notes:

Viewing the sulog

If you are concerned that someone is trying to access an account that they should not be accessing (like the root account), look at the `/var/adm/sulog` file:

```
# more /var/adm/sulog
SU 02/18 11:39 + pts/3 root-bob
SU 02/18 11:40 + pts/3 root-bob
SU 02/18 11:43 + pts/3 root-bob
```

8.11 Preventing stop-a

As mentioned previously, there is a potential security concern on the system. Anyone can perform the following actions to gain superuser access on the system:

1. stop-a
2. boot cdrom -s
3. mount the boot partition
4. edit the `/etc/shadow` file and remove the root password
5. reboot and log in as root

One method of preventing this is to place a password on the PROM. Another method is to place the following line in the `/etc/default/kbd` file:

KEYBOARD_ABORT=disable

This will prevent anyone from performing a stop-a while the system is up.

Note 1: You must reboot the system for this file to take effect.

Note 2: This DOES NOT prevent someone from manually turning off your system and performing a stop-a as the system is coming up.

Notes:

8.12 Preventing FTP Access

You can control which users can connect to the system via the File Transfer Protocol by creating a file called `/etc/ftpd/ftpusers`. This file will contain a list of users who are not allowed to connect to the system via FTP.

The default `/etc/ftpd/ftpusers` file:

```
# more /etc/ftpd/ftpusers
root
daemon
bin
sys
adm
lp
uucp
nuucp
smmsp
listen
nobody
noaccess
nobody4
```

Notes:

8.13 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #9

Web sites

None

Man pages

chgrp	finger	last	who
chown	groups	passwd	who am i
exit	id	su	whodo

8.14 Summary of Commands and Files

Command	Description
chgrp	Changes the group ownership of a file
chown	Changes the user ownership of a file
exit	Closes current shell
finger	Provides user account information
groups	Provides list of user's groups
id	Provides user account information
last	Displays login information of current and past logins
passwd	Changes a user's password
su	Allows you to shift to another user account
who	Displays users who are currently logged into the system
who am i	Displays your original login information
whodo	Provides information as to a user's current activity

File	Description
/etc/default/kbd	Can contain a setting to prevent stop-a attempts
/etc/default/login	Controls login settings
/etc/default/passwd	Controls default password aging and length
/etc/default/su	Sets parameters for logging of su attempts
/etc/ftpd/ftpusers	Can contain a list of users who can not FTP to system
/var/adm/loginlog	Existence of this file enables logging of failed login attempts
/var/adm/sulog	Default sulog file

8.15 Lab Exercises

1. On your computer, find out who is currently logged in.

   Command: _____

2. Find out how many times you have logged into the root account since Monday.

   Command: _____

3. What is UID of the bin user? Of the nobody user?

   Command: _____

4. What is the home directory of the root user?

   Command: _____

5. What is the encrypted password of the root user and when does this password expire?

   Command: _____

6. What are the usernames assigned to the sys group.

   Command: _____

7. Log out of the system and then log in using another user account. Now become the superuser (don't log out!).

   Command: _____

8. Go back to the user account by closing your superuser status.

   Command: _____

9. What is the current's users group?

   Command: _____

Scenario Lab:

You want to make your system more secure. In order to accomplish this, take the following steps:

#1. Don't allow root login from anywhere
#2. Don't allow stop-a attempts
#3. Log all su attempts to the console window
#4. Set up the system to log failed login attempts

Scenario lab:

The user sue is leaving the company. A new user (julia) will be taking sue's place. Delete sue's account, but don't delete the home directory. Add a new account for julia, using sue's old home directory as julia's new directory.

Unit Nine
Administering Initialization Files

Unit topics

		Page #
9.1	What are Initialization Files?	224
9.2	A Review of Common Customizations	225
9.3	Bourne Shell Initialization Files	226
9.4	Korn Shell Initialization Files	229
9.5	C Shell Initialization Files	231
9.6	Testing Initialization Files	233
9.7	Additional Resources	234
9.8	Summary of Commands and Files	235
9.9	Lab Exercises	236

9.1 What are Initialization Files?

Initialization files allow the user (or the System Administrator) to modify the working environment. In other words, initialization files modify how commands work in UNIX.

Each of the primary shells (C, Bourne and Korn) has different initialization files. This is due to the syntax differences between the shells. As a System Administrator, you should know how to modify (or assist a user in modifying) the initialization files of all three shells.

There are also two types of initialization files:

> **Global** initialization files affect all users on the system and are located in the `/etc` directory.
>
> **Local** initialization files affect only a particular user and are located in the user's home directory.

Typically when a user logs in, the global initialization files is read first and all of the customization placed in that file are enforced on the user's environment. After the global initialization file has been read, the user's local initialization file is read and enacted on.

Notes:

9.2 A Review of Common Customizations

Customizing a user's environment involves creating aliases, variables and keyboard mapping. Because of the level of experience this course assumes, this chapter does not go into details on these topics. For review purposes, here are the most common system variables:

Variable	Type	Purpose
cdpath	local	List of directories that are searched when the **cd** command is used.
home/HOME	local/env	The user's home directory.
path/PATH	local/env	List of directories that are searched when a command is typed.
prompt	local	Sets your prompt.
shell/SHELL	local/env	Holds the location of the C shell command
HOST	env	The name of the machine you are logged into
PRINTER	env	Used by the **lpr** command to determine what printer to print to (**lpr** is a print command).
LPDEST	env	Used by the **lp** command to determine what printer to print to (**lp** is a print command).
MANPATH	env	Used by the **man** command to search for man pages

Notes:



Notes:

9.3 Bourne Shell Initialization Files

While the Bourne shell doesn't provide many features (like the C and Korn shells do), there are a few features worth mentioning.

The `/etc/profile` file

The global initialization file for the Bourne shell is `/etc/profile`. All users on the system will be affected by the contents of this file. This file is created by the system during installation (some of the output was {omitted}):

```
# more /etc/profile
#ident  "@(#)profile    1.17    95/03/28 SMI"   /* SVr4.0 1.3   */

# The profile that all logins get before using their own .profile.
{omitted}
        if /bin/i386
        then
                TERM=AT386
        else
                TERM=sun
        fi
{omitted}
        /bin/cat -s /etc/motd
{omitted}
        umask 022
{omitted}
```

Notes:

Essentially, the default `/etc/profile` file does the following:

1. Set the terminal type
2. Display the `/etc/motd` file (message of the day file)
3. Set the user's umask setting (umask is discussed in Unit 10)

Note: The user can prevent the output of the `/etc/profile` file from being display when they log in by creating a file called .hushlogin in their home directory.

The $HOME/.profile file

The user can create their own initialization file in their home directory. This file's name starts with a dot ".", making it a hidden file.

When a user's account is created, this file is not typically created (although it can be). To make it easier, there is a template file in the `/etc/skel` directory that the user can copy to their home directory and modify.

Note: the root user's default shell is the Bourne shell and the root user's home directory is the `/` directory. Therefore, `/.profile` is the local initialization file for the root user.

The `/etc/skel/local.profile` file

This template file is never used by the system. The only purpose for this file is to provide a "starting point" for users who want to create a `$HOME/.profile` file.

If invoked, the `/etc/skel/local.profile` file will do the following:

1. Adds a path to the $PATH variable
2. Tries to start Openwindows

Notes:



Notes:

9.4 Korn Shell Initialization Files

Like the Bourne Shell, the Korn shell reads the `/etc/profile` and `$HOME/.profile` files. In this regard, there is no difference between the two shells. The Korn shell, however, does provide a feature that does not exist in the Bourne shell.

The `$HOME/.kshrc` file

While the `/etc/profile` and `$HOME/.profile` files are read only at login, the `$HOME/.kshrc` file is read **every time** a new Korn shell is opened. This is a good feature because some environment changes won't take affect when you open a new shell. For example, if you set an alias in a Korn shell and open a new Korn shell, the alias won't appear in the new Korn shell:

```
# ksh
# alias ls='ls -l'
# alias
ls='ls -l'
# ksh
# alias
{no output since there are no aliases}
```

It would be very annoying to have to reset all of your aliases each time you open a new shell. By placing commands in the `$HOME/.kshrc` file, each time a new Korn shell is opened, the command will be run in the new shell.

Notes:

Setting up `$HOME/.kshrc`

The system does not automatically look for the `$HOME/.kshrc` file. In order to tell the system to read this file when you open a new Korn shell, you must set a special variable called **ENV** in the `$HOME/.profile` file:

```
ENV=$HOME/.kshrc
export ENV
```

Note: the file doesn't have to be named ".kshrc"; it's just a traditional name for this kind of file. It also doesn't have to be in the user's home directory, it can be anywhere on the system.

Note: There is no `/etc/skel/local.kshrc` file.

9.5 C Shell Initialization Files

Because of its different syntax, the C shell has different initialization files altogether. The `/etc/.login` file is the global initialization file for the C shell while the `$HOME/.login` file is the local initialization file.

The `/etc/.login` file

The `/etc/.login` file affects all users on the system whose default shell is the C shell. By default, the `/etc/.login` file does the following:

1. Set the TERM variable (terminal type)
2. Checks the user's disk quota
3. Displays `/etc/motd` (message of the day)
4. Checks to see if the user has mail and prints a message if they do.

Note: The user can prevent the output of the `/etc/.login` file from being display when they log in by creating a file called .hushlogin in their home directory.

The `$HOME/.login` file

Just like in the Bourne and Korn shells, the user can create their own initialization file in their home directory. This file's name starts with a dot ".", making it a hidden file.

When a user's account is created, this file is not typically created. To make it easier, there is a template file in the `/etc/skel` directory that the user can copy to their home directory and modify.

Notes:

Notes:

The `/etc/skel/local.login` file

This template file is never used by the system. The only purpose for this file is to provide a "starting point" for users who want to create a `$HOME/.login` file.

If invoked, the `/etc/skel/local.login` file will do the following:

1. Sets a terminal option (see man pages for more info)
2. Tries to start Openwindows

The `$HOME/.cshrc` file

Like the Korn shell, the C shell has some features that won't be "passed into" a new C shell. The `$Home/.cshrc` file provides the same functionality for the C shell that `$HOME/.kshrc` provides for the Korn shell.

Note: There is no setup needed for the system to recognize the .cshrc file.

The `/etc/skel/local.cshrc` file

The `/etc/skel/local.cshrc` file provides an example file for users to copy and modify for their own use.

9.6 Testing Initialization Files

Many users log out and log in to test initialization files. This is unnecessary as you can test these files on the fly. All three shells support a feature called "sourcing". Sourcing will run the commands in a file and have the changes take affect on the current shell.

Sourcing Bourne and Korn shell files

The dot "**.**" command will allow you to source Bourne and Korn shell files. For example, to source the kshrc file:

```
# . $HOME/.kshrc
```

Sourcing C shell files

The command **source** will source C shell files. The following example sources the system's `/etc/.login` file:

```
sun% source /etc/.login
```

Notes:

9.7 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #12

Web sites

System Administration Guide: Basic Administration, Chapter 3. Managing User Accounts and Groups (Overview): http://docs.sun.com/app/docs/doc/817-6958/6mmafc2ub?a=view

System Administration Guide: Basic Administration, Chapter 4. Managing User Accounts and Groups (Tasks): http://docs.sun.com/app/docs/doc/817-6958/6mmafc2ul?a=view

Man pages

source

9.8 Summary of Commands and Files

Command	Description
source	A C shell command used to run (execute) a C shell script "in place"
.	A Bourne or Korn shell command used to run (execute) a Bourne or Korn shell script "in place"

File	Description
/etc/profile	Global initialization file for Bourne and Korn shell users
/etc/skel/local.profile	A template for a user's .profile file
/etc/.login	Global initialization file for C shell users
/etc/skel/local.login	A template for a user's .login file
/etc/skel/local.cshrc	A template for a user's .cshrc file

9.9 Lab Exercises

Scenario: You want to customize the environment so that **all users** (C, Bourne and Korn) on the system have the following customizations:

A. Have the cp command work like cp -i
B. Have the mv command work like mv -i
C. Have the rm command work like rm -i
D. Have the prompt display the host name and date.
E. Have the history set up to remember the past 50 commands.

Unit Ten
Permissions

Unit topics

		Page #
10.1	Basic UNIX Permissions	238
10.2	Setting Default Permissions	244
10.3	Special Permission: setuid	246
10.4	Special Permission: setgid	249
10.5	Special Permission: sticky bit	254
10.6	Additional Resources	256
10.7	Summary of Commands and Files	257
10.8	Lab Exercises	258

Notes:

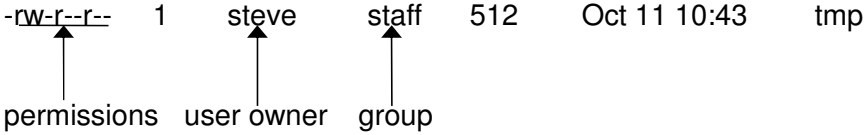
10.1 Basic UNIX Permissions

Permissions are UNIX's method of protecting files and directories. Every file or directory is owned by a user and assigned to a group. There are also default permissions set on the file (discussed in detail in a later section). The owner of a file has the right to set permissions in order to protect the file from being accessed, modified or destroyed.

Determining permissions

To determine the permissions of a file, use the **ls -l** command. The first character of the output of the **ls -l** command specifies the file type. The next nine characters represent the permissions set on the file. There are three types of permissions: r (read), w (write), and x (execute). These permissions have different meanings for files and directories.

First three permissions are for the **user owner**, second three are for people in the **group**, and last three are for everyone else (**others**).



Therefore, in the preceding example, the owner of the file (steve) has read and write permissions, the members of the group (staff) have read permission and everyone else has read permission.

Notes:

A reminder about groups

Groups were invented to provide more flexibility when issuing permissions. Every user is a member of at least one group (a primary group) and up to 15 other (secondary) groups. To see the groups you belong to, type the **groups** command.

File permissions vs. Directory permissions

Permissions have different meaning on files and directories due to the differences between a file and a directory. The following chart illustrates the differences:

Permission	Symbol	Meaning for Files	Meaning for Directories
Read	r	Can view or copy file	Can list with **ls** command
Write	w	Can modify file	Can add or delete files in the directory (if execute permission is also set)
Execute	x	Can run file like a program	Can **cd** to that directory. Can also use that directory in a path.

Notes:

Changing Permissions

Only the person who owns the file (or the root user) can change the file's permissions.

There are two methods of changing the permissions on a file: symbolic and octal. The **chmod** command is used in both cases.

Symbolic method

The symbolic method is useful for changing just a permission or two. Following the **chmod** command, you specify three items: whose permission you wish to change, whether you want to add or remove the permission, and the permission itself. The following chart illustrates the possibilities:

who	Operand:	Permission:
u (user owner)	- (remove)	r (read)
g (group)	+ (add)	w(write)
o (other)		x (execute)
a (all three)		

Notes:

For example, the following removes read permission for the group for the file myprofile:

```
# cd
# ls -l
-rw-r--r--   1   steve   staff   512   Oct 11 10:43   myprofile
# chmod g-r myprofile
# ls -l
-rw----r--   1   steve   staff   512   Oct 11 10:43   myprofile
```

You can specify multiple permissions to change; the following example will add execute permission for the user owner and remove read permission for others for the file myprofile:

```
# cd
# ls -l
-rw----r--   1   steve   staff   512   Oct 11 10:43   myprofile
# chmod u+x,o-r myprofile
# ls -l
-rwx------   1   steve   staff   512   Oct 11 10:43   myprofile
```

Notes:

Octal Method

The octal method is useful when you have to change many permissions on a file. It is based on the octal numbering system:

4=read
2=write
1=execute

By using a combination of numbers from 0 to 7, any possible combination of read, write and execute permissions can be specified. The following chart illustrates all of the possible combinations:

Value	Meaning
7	r w x
6	r w -
5	r - x
4	r - -
3	- w x
2	- w -
1	- - x
0	- - -

When the octal method is used to change permissions, all nine permissions must be specified. Because of this, the symbolic method is generally easier for changing a few permissions while the octal method is better for changes that are more drastic.

Notes:

To change the permission for the myprofile file to the permissions rwxrw-r-- use the following command:

```
# cd
# ls -l
-rw----r--   1    steve    staff    512    Oct 11 10:43    myprofile
# chmod 764 myprofile
# ls -l
- rwxrw-r--  1    steve    staff    512    Oct 11 10:43    myprofile
```

Notes:

10.2 Setting Default Permissions

When you create a file or directory, predefined permission are set on that file or directory. The current default permissions are:

files	**rw-rw-rw-**
Directories	**rwxrwxrwx**

To change the default permissions, you must change the **umask** setting. The following text and chart can be used to determine a umask entry:

#1. Know the MAXimum possible permissions for files and directories.

#2. Determine what you want your permissions to be when you either create a new file or directory. You will need to pick one (either file or directory) and understand that the umask command affects both.

#3. Determine what permissions of the MAXimum permissions need to be taken out (MASKed out).

#4. Compute the values (in octal notation) of what needs to be MASKed out for the owner, group and other. The resulting three numbers is what you will use to set your umask.

Step		**File**	**Directory**
1	MAX	rw- rw- rw-	rwx rwx rwx
3	MASK	--- -m- mmm	--- -m- mmm
4	umask=027	0 2 7	0 2 7
2	desire	rw- r-- ---	rwx r-x ---

Notes:

The **umask** command displays and changes default permissions:

```
# umask
027
# umask 026
026
```

After setting this **umask**, if you were to log out (or open a new shell) you would lose the new **umask** setting. In order for this new **umask** setting to be a permanently changed, you must place the **umask** command into your initialization file.

Notes:

10.3 Special Permission: setuid

When a user runs a command and that command accesses files, the system checks the permissions that that user has on the files the command is accessing. In some cases, this might cause problems.

Consider a command like **passwd**. When this command runs, it edits the `/etc/shadow` file. If you look at the permissions of the `/etc/shadow` file, you will see that the permissions are: r-- --- ---

If a non-root user attempted to edit the `/etc/shadow` file directly (with, for example, the **vi** command), the user would be denied access to the file. Note, however, that the passwd command allows a non-root user to access the `/etc/shadow` file and change their password.

The **passwd** command has a special permission set on it called setuid. When the **passwd** command is run and the command accesses files, the system pretends that the person accessing the file is the owner of the **passwd** command, not the person who is running the command.

```
# ls -l /bin/passwd
-r-sr-sr-x   3 root     sys        96796 Jul 15  1997 /bin/passwd
# id
uid=10051(bob) gid=1(other)
# passwd                          (accesses files as root, not bob)
```

Notes:

To set setuid permission

The setuid permission can be set using either octal or symbolic methods:

```
# ls -l /usr/bin/ls
-r-xr-xr-x   1 bin      bin        17440 Jul 15  1997 /usr/bin/ls
# chmod a+s /usr/bin/ls
# ls -l /usr/bin/ls
-r-sr-xr-x   1 bin      bin        17440 Jul 15  1997 /usr/bin/ls
# chmod a-s /usr/bin/ls
# ls -l /usr/bin/ls
-r-xr-xr-x   1 bin      bin        17440 Jul 15  1997 /usr/bin/ls
#
# chmod 4555 /usr/bin/ls
# ls -l /usr/bin/ls
-r-sr-xr-x   1 bin      bin        17440 Jul 15  1997 /usr/bin/ls
# chmod 0555 /usr/bin/ls
# ls -l /usr/bin/ls
-r-xr-xr-x   1 bin      bin        17440 Jul 15  1997 /usr/bin/ls
```

Notice the “s” character located in the owner’s permissions. This indicates that the setuid permission is set. If the “s” is lower case, it means both setuid and the execute permission is set. If the “S” is upper case, it mean only setuid (not execute) is set.

Notes:

Be careful of setuid

Commands that have the setuid permission set present a security risk on the system (especially those owned by root). Be careful of when you create setuid programs and make sure you are aware of what setuid programs are on your system

You can use the **find** command to find which programs on the system have the setuid permission set:

```
# find / -perm -4000 –ls
{output omitted}
```

10.4 Special Permission: setgid

There are actually two forms of setgid permissions: setgid on a file and setgid on a directory.

setgid on a file

This essentially means the same thing as setuid on a command. When someone runs the command, instead of accessing files as the group the person is a part of, the system pretends the person is a member of the group the file is owned by.

Notes:

Notes:

setgid on a directory

Consider the following situation: Four people from different groups in a company are working on a common project. The four users and the groups they belong to are:

User	Groups
bob	staff
steve	accounting, staff
sue	payroll
nick	admin

The company policy is for all users to have the umask 027.

All users store the files for this project in the `/export/home/beta_prog_a` directory.

After a few of the users store some file in this directory, a listing of that directory looks like this:

```
# ls -l /export/home/beta_prog_a
total 6
-rw-r-----   1 bob     staff         124   Mar  4  1998  1999_data
-rw-r-----   1 steve   accounting    575   Jul 15  1997  tax_table
-rw-r-----   1 sue     payroll       560   Jul 15  1997  salaries
-rw-r-----   1 nick    admin         560   Jul 15  1997  hr_data
```

Based upon the above information, you will note that there is problem here. While each user may be able to create files in the `/export/home/beta_prog_a` directory, no user can see another user's work.

To avoid this problem take the following steps:

1. Create a new group (called beta in the example below).
2. Place all users in the new group.
3. Give group ownership of the directory to the new group.
4. Set the setgid permission on the directory.

After taking these steps, any new file in the directory `/export/home/beta_prog_a` will be group owned by the new group. See the next page for a demonstration of these steps.

Notes:

Example:

```
# mkdir /export/home/beta_prog_a
# groupadd -g 133 beta
# usermod -G beta bob
# usermod -G beta steve
# usermod -G beta sue
# usermod -G beta nick
# chgrp beta /export/home/beta_prog_a
# chmod g+s /export/home/beta_prog_a
# ls -ld /export/home/beta_prog_a
total 0
drwxrws---   2 root     beta         512 Dec 14 12:49 /export/home/beta_prog_a
```

Notice the “s” character located in the group’s permissions. This indicates that the setgid permissions are set. If the “s” is lower case, it means both setgid and the execute permission is set. If the “S” is upper case, it mean only setgid (not execute) is set.

Notes:

Notes:

Be careful of setgid

Commands with the setgid permission set present a security risk on the system (especially those owned by system groups). Be careful of when you create setgid programs and make sure you are aware of what setgid programs are on your system

You can use the **find** command to find which programs on the system have the setgid permission set:

```
# find / -perm -2000 –ls
{output omitted}
```

10.5 Special Permission: sticky bit

Consider the following situation: You have a directory in which users can post announcements called `/export/home/pub`. In order for all users to be able to post (create files in) this directory, you need to give the permissions 777.

Unfortunately, these permissions also allow any user to remove any file from the pub directory. What if a user decides to run the command **rm -r *** on that directory?

The sticky bit permission gives you the ability to allow anyone to add to a directory, but limits who can delete files in that directory. The only users who can delete files in a sticky bit directory are:

1. root
2. The owner of the directory
3. The owner of the file

Notes:

Notes:

To set sticky bit

To set the sticky bit permission, use the **chmod** command:

```
# chmod 1777 /export/home/pub
# ls -ld /export/home/pub
drwxrwxrwt   2 root     other        512 Feb 18 18:11 /export/home/pub
```

Notice the “t” character in the place where the “x” should be for others. This “t” tell you that the sticky bit has been set on this directory.

10.6 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #10

Web sites

System Administration Guide: Basic Administration, Chapter 3. Managing User Accounts and Groups (Overview): http://docs.sun.com/app/docs/doc/817-6958/6mmafc2ub?a=view

Man pages

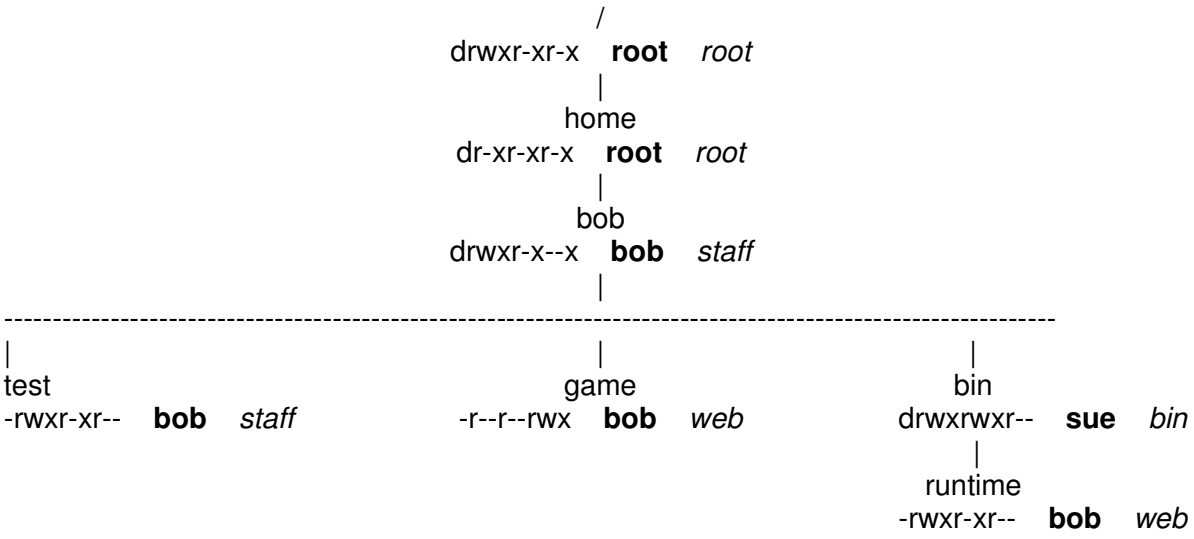
chmod
umask

10.7 Summary of Commands and Files

Command	Description
chmod	Changes the permissions of a file or directory
umask	Sets or displays default permissions

10.8 Lab Exercises

Use the following directory structure to answer the questions on the following page. Note: file/directory name is above the permissions, the owner of file is in **bold**, and group ownership is in *italics*.



Users: **Bob** (member of *staff* group)
Sue (member of *bin* group)
Jonny (member of *staff* and *web* group)
Stacey (member of *web* group)

Fill out the following grid by placing a check mark in the box where the user has permission:

User & file/dir	**Can view and cp**	**Can modify**	**Can run like prog**	**Can ls**	**Can add/remove files in dir**	**can cd & use in path**
stacey `/home/bob`						
jonny `/home/bob`						
bob `/home/bob/game`						
sue `/home/bob/bin`						
jonny `/home/bob/bin`						
sue `/home/bob/test`						
stacey `/home/bob/test`						
bob `/home/bob/bin`						
bob `/home/bob/bin/runtime`						
sue `/home/bob/bin/runtime`						

Scenario #1: You want users in your group to be able to access the files in your template directory while denying all other users. However, you don't want anyone to be able to see anything in this directory. Additionally, you don't want anyone to be able to remove or add any file in this directory. Create a directory called template under your home directory and change the permissions on the template directory accordingly.

Scenario #2: Your boss wants a bulletin board directory where anyone can post notes (but not delete other user's notes). Create a bulletin board directory `/var/board`.

Scenario #3: One of your users (username=guest) has left the company. Delete his account without deleting his home directory. A new user arrives two weeks later (username=bob). Create an account for him and give him access to the old user's directory. NOTE: Test this to make sure bob really has permissions to all of the info in his new home directory.

Unit Eleven
The Boot Process

Unit topics

		Page #
11.1	Run Levels	262
11.2	Overview of the Boot Phases	263
11.3	The init Phase	266
11.4	Modifying a Run Level	272
11.5	Changing to Different Run Levels	273
11.6	Displaying Boot Messages	276
11.7	Additional Resources	277
11.8	Summary of Commands and Files	278
11.9	Lab Exercises	280

Notes:

11.1 Run Levels

Consider run levels to be like different functional states of the system. The following chart describes the default run levels on the system:

Run Level	State of system
0	Open Boot PROM level
1,s	Single user mode: only the system administrator can log in and only at the console. Many daemons are not running and some filesystems may not be mounted.*
2	Multi-user, no shared resources: The system is at a fully functional state with the exception that it can not be an NFS server.
3	Multi-user, shared resources allowed: The system is at a fully functional state and it can be an NFS server.
4	Not defined: this run level can be created by the system administrator to make a custom run level.
5	Poweroff: will power off the system gracefully.
6	Reboot: will reboot the machine.

While this Unit will explore the entire boot process, it will focus on how to modify the run levels.

* Note: Run level s and 1 are basically the same. The primary difference is if you bring the system down from a higher run level to level s, all users will be logged off the system while if you bring the system down to run level 1 they will not.

11.2 Overview of the Boot Phases

Notes:

Open Boot PROM phase

When the system is turned on, the Open Boot PROM handles the beginning part of the boot process. Its primary tasks are:

1. Perform memory diagnostics
2. Perform self-test diagnostics
3. Read its boot parameters
4. Load the **bootblk** program – The **bootblk** program assists the PROM in its efforts to read the UNIX filesystems.
5. Loads the **ufsboot** program – The ufsboot file is like a list of instructions that tells PROM what steps to take to load the OS.
6. Loads the kernel.

Notes:

Kernel Initialization Phase

Once the kernel has been loaded into memory, it takes over from PROM and handles the rest of the boot process. Its primary tasks are:

1. Initializes itself using parameters set in the `/etc/system` file.
2. Loads device modules into memory.
3. Starts the **sched** daemon – The **sched** daemon will handle process priority.
4. Starts the **init** daemon – The **init** process handles run levels.
5. Starts the **pageout** daemon – The **pageout** daemon will handle paging and swaping functions.
6. Starts the **fsflush** daemon – The **fsflush** daemon handles flushing filesystem data out of memory and onto the hard drive (see a later Unit for more details on the fsflush daemon).

Notes:

Configuring the kernel with the `/etc/system` file

The kernel includes code that is derived from files called modules. These module files include code that is specific to a particular function or feature (such as a device or a filesystem). The module files are stored in the following directories:

- `/kernel`
- `/usr/kernel`
- `/platform/`uname -m`/kernel`
- `/platform/`uname -i`/kernel`

In most cases, the kernel will load the modules needed for the OS. In some cases you may need to specify which modules you want loaded. This can be down in the `/etc/system` file.

The `/etc/system` file can also be used to specify parameters. Parameters are used to modify the behavior of the kernel.

The default `/etc/system` file consists entirely of commented lines. Comments in this file start with an asterisk (*) and end with the end of the line.

When editing this file, consider making a backup copy. This is useful for when you make a mistake in the file; you can still use the backup copy to boot the system.

Notes:

11.3 The init Phase

The **init** daemon handles how the system moves from one run level to another. In order to determine how to perform this function, the **init** daemon reads the `/etc/inittab` file.`

Since the introduction of the System Management Facility (SMF), the init daemon no longer starts the majority of the services on the system. The **svc.startd** daemon is now used to start, monitor, and restart stopped processes.

The `/etc/inittab` file

```
# more /etc/inittab
# The /etc/inittab file controls the configuration of init(1M); for more
# information refer to init(1M) and inittab(4).  It is no longer
# necessary to edit inittab(4) directly; administrators should use the
# Solaris Service Management Facility (SMF) to define services instead.
# Refer to smf(5) and the System Administration Guide for more
# information on SMF.
#
# For modifying parameters passed to ttymon, use svccfg(1m) to modify
# the SMF repository. For example:
#
#       # svccfg
```

```
#       svc:> select system/console-login
#       svc:/system/console-login> setprop ttymon/terminal_type = "xterm"
#       svc:/system/console-login> exit
#
#ident  "@(#)inittab    1.41    04/12/14 SMI"
ap::sysinit:/sbin/autopush -f /etc/iu.ap
sp::sysinit:/sbin/soconfig -f /etc/sock2path
smf::sysinit:/lib/svc/bin/svc.startd    >/dev/msglog 2<>/dev/msglog </dev/console
p3:s1234:powerfail:/usr/sbin/shutdown -y -i5 -g0 >/dev/msglog 2<>/dev/msglog
```

Each line in this file defines an action for the **init** daemon to take. Each line has four fields:

id:rstate:keyword:process_to_run

Field	Purpose
id	A field used only by administrators to identify the line. Can be between 1-4 characters long.
rstate	States what run level to run the process.
keyword	States how to run the process (see on next page).
process_to_run	The process to run.

Notes:

Notes:

The keyword field tells the system how to run the process. Here are the most commonly used keywords:

Keyword	Meaning
initdefault	The run level to take the system to by default. This keyword should only appear once in the file. If this line is removed, the system will prompt the administrator to specify a run level during boot. (With SMF, this is no longer used)
sysinit	The processes will always be run regardless of the run level the system goes to.
wait	Start the process, but wait until it is complete before continuing to the next line in the `/etc/inittab` file.
respawn	Start the process and, if it dies out, restart it.
powerfail	Run the process only if the system receives a powerfail signal.

How init reads the inittab

The **init** daemon makes three passes through the inittab:

1. It first looks at the default run level (is:3:initdefault:) to determine what run level to take the system to. (With SMF, the effective default run level is 3)
2. It next looks for "sysinit" in the rstate field and runs those processes.
3. Lastly, it looks at all lines that have the same run level as the "initdefault" line and starts those processes.

Notes:

Examples of init reading the inittab

This first example is what occurs when a "normal" boot occurs:

1. Reads the "is" line and determines the system needs to go to run level 3. (With SMF, it defaults to a run level of 3)
2. Looks for the "sysinit" lines and runs the following processes:
   `/sbin/autopush -f /etc/iu.ap`
   `/sbin/soconfig -f /etc/sock2path`
   `/lib/svc/bin/svc.startd >/dev/msglog 2<>/dev/msglog </dev/console`
3. Looks for 3 in the rstate field and runs the following processes:
   `/usr/sbin/shutdown` (on a powerfail signal)

In this example, the command "**boot -s**" is given at the Open Boot PROM level:

1. Ignores the initdefault line since the option **-s** specifies to go to run level s.
2. Looks for the "sysinit" lines and runs the following processes:
   `/sbin/autopush -f /etc/iu.ap`
   `/sbin/autopush -f /etc/iu.ap`
   `/sbin/soconfig -f /etc/sock2path`
   `/lib/svc/bin/svc.startd >/dev/msglog 2<>/dev/msglog </dev/console`

3. Looks for s in the rstate filed and runs the following process:
   `/usr/sbin/shutdown` (on a powerfail signal)

Notes:

Run control scripts

The heart of what makes up a run level is the run control scripts (rc scripts). These scripts now start services that are not managed by SMF. These type of services are now known as legacy services. The primary functions of the `/sbin/rcs`, `/sbin/rc2` and `/sbin/rc3` scripts are:

`/sbin/rcs` - Run all files in `/etc/rcS.d` that start with a "K" with a "stop" option
Run all files in `/etc/rcS.d` that start with a "S" with a "start" option

`/sbin/rc2` - Run all files in `/etc/rc2.d` that start with a "K" with a "stop" option
Run all files in `/etc/rc2.d` that start with a "S" with a "start" option

`/sbin/rc3` - Run all files in `/etc/rc3.d` that start with a "K" with a "stop" option
Run all files in `/etc/rc3.d` that start with a "S" with a "start" option

Note: the files in the aforementioned directories are hard links to files in the `/etc/init.d` directory:

ls /etc/init.d

PRESERVE	devlinks	ldap.client	pcmcia
README	dhcp	llc2	pppd
acct	dodatadm.udaplt	lu	sendmail
acctadm	drvconfig	mipagent	slpd
apache	dtlogin	mkdtab	swupboots
autoinstall	init.dmi	ncakmod	sysetup
boot.server	init.sma	ncalogd	ufs_quota
cachefs.daemon	init.snmpdx	nfs.server	uucp
deallocate	init.wbem	nscd	volmgt

Notes:

11.4 Modifying a Run Level (Legacy)

To modify the behavior of run levels S, 1, 2, 3 or 4, perform the following steps:

#1. Add a script to the `/etc/init.d` directory.

#2. Link the file to the `/etc/rcX.d` directory.

Example of modifying a run level

```
# cd /etc/init.d
# vi runban
#!/usr/bin/sh
banner "Run" "Level 2"  #added to the file
# chmod a+x runban
# ln runban /etc/rc2.d/S99runban
```

To modify the behavior of run levels 0, 5 or 6, edit the appropriate `/sbin/rcX` script and adding the command that you wish to execute.

Notes:

Notes:

11.5 Changing to Different Run Levels

There are two methods of changing to different run levels. One method is to use the traditional UNIX commands. The other method is to use the **init** command:

To go to run level:	Traditional UNIX command	init command
0	halt	init 0
s	NONE	NONE*
1	shutdown -i 1	NONE*
2	NONE	NONE*
3	NONE	NONE*
4	NONE	NONE*
5	poweroff	init 5
6	reboot	init 6



The **init** command has an advantage over the traditional UNIX commands in that is it easy to modify how the system goes to the new run level (by modifying the rc scripts and the `/etc/rc*.d` directories).

With SMF, the concept of run levels has been replaced with "milestones". Milestones are like run levels in that represent a set of services that should be started. The following table shows the milestone equivalents of the various init commands:

Init Command	Milestone
init s	svcadm milestone svc:/milestone/single-user

init 2	svc:/milestone/multi-user
init 3	svc:/milestone/multi-user-server

There are two other special milestones: all and none. With the "none" milestone, no services are started and it is considered the SMF maintenance mode. The "all" milestone will attempt to start all enabled services.

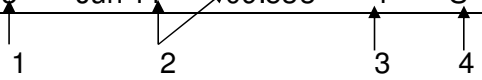
From thePROM: **boot -s** is equivalent to **boot -m milestone=single-user**. When exiting single user or maintenance mode, instead of using **init 3**, it is recommended to use **svcadm milestone all**.

Notes:

Notes:

The **who -r** command will display what run level you are currently at:

```
# who -r
.       run-level   3       Jun 11      09:533      1       S
```



1. Current level
2. Date/Time the system went to this run level
3. Number of time at this level since last boot
4. Last run level

The **svcs** command will display the status of all SMF services:

```
# svcs | tail
online         14:11:37 svc:/milestone/multi-user:default
online         14:11:37 svc:/system/boot-config:default
online         14:11:38 svc:/system/filesystem/volfs:default
online         14:11:38 svc:/milestone/multi-user-server:default
online         14:11:38 svc:/system/zones:default
online         14:11:38 svc:/application/stosreg:default
online         14:11:39 svc:/application/graphical-login/cde-login:default
online         14:11:39 svc:/application/cde-printinfo:default
online         14:12:10 svc:/system/webconsole:console
offline        14:11:10 svc:/application/management/ocm:default
```

Notes:

11.6 Displaying Boot Messages

After the boot process has completed, you can display the messages that were displayed by looking at the `/var/adm/messages` file:

```
# more /var/adm/messages
Nov 12 09:32:31 blade eri: [ID 517527 kern.info] SUNW,eri0 : No response from Ethernet network : Link down -- cable problem?
Nov 12 09:34:01 blade last message repeated 1 time
Nov 12 09:37:01 blade eri: [ID 517527 kern.info] SUNW,eri0 : No response from Ethernet network : Link down -- cable problem?
{remaining output omitted}
```

For each SMF managed service, there is also a log file located in the /lib/svc/log directory:

```
$ ls /var/svc/log | tail -5
system-system-log:default.log
system-tsol-zones:default.log
system-utmp:default.log
system-webconsole:console.log
system-zones:default.log
```

11.7 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #4

Web sites

System Administration Guide: Basic Administration: Chapter 8 Shutting Down and Booting a System (Overview): http://docs.sun.com/app/docs/doc/817-1985/6mhm8o5n1?a=view

System Administration Guide: Basic Administration: Chapter 9 Shutting Down a System (Tasks): http://docs.sun.com/app/docs/doc/817-1985/6mhm8o5nd?a=view

Man pages

consadm	ln	shutdown
halt	poweroff	who
init	reboot	

11.8 Summary of Commands and Files

Command	**Description**
consadm	Forwards system messages to the console
halt	Halts the system
init	Changes to another run level
ln	Creates a hard or symbolic link file
poweroff	Powers off the system
reboot	Reboots the system
shutdown	Brings the system down
who	With the **-r** option, shows current and previous run levels

File	Description
/dev/msglog	A link to the `/dev/sysmsg` file
/dev/sysmsg	A file that forwards data sent to it to other devices as defined by the **consadm** command
/etc/init.d	Central storage directory for boot scripts
/etc/inittab	Describes the actions for the init process to take
/etc/rc2.d	Directory that contains hard link files to scripts that are executed when system is brought to run level 2
/etc/rc3.d	Directory that contains hard link files to scripts that are executed when system is brought to run level 3
/etc/rcS.d	Directory that contains hard link files to scripts that are executed when system is brought to run level S
/etc/system	Contains kernel boot modifications
/sbin/rc2	Script used to bring system to run level 2
/sbin/rc3	Script used to bring system to run level 3
/sbin/rcs	Script used to bring system to single user run level
/var/adm/messages	Boot log file

11.9 Lab Exercises

Scenario #1: Your system hangs during the boot process. The last line that appears on the screen before it hangs is "starting rpc services". You can usually discover where the problem occurs somewhere in the init phase. Seek out the place where this problem occurs. NOTE: this is not really a problem on your system, just an example. You don't need to fix anything.

Scenario #2: When the system boots instead of taking you to the traditional command line login, it takes you to the "new" CDE login. Most of the time you don't mind going to CDE's login screen, but sometimes you want to go directly to the command line login. Create a new run level 4 that will take you to the command line login. Keep in mind that there are several steps that must be taken to create this new run level. Think about all that takes place during the init phase. Hint: you can copy the rc3 script and use that copy for your rc4 script.

Scenario #3 Modify run level 3 so that it will ping all hosts on the network during boot. What else would you like to display/test during boot?

Scenario #4 Modify run levels 5 and 6 so that they will append the output of the **last** to the file `/var/sadm/logged_in`

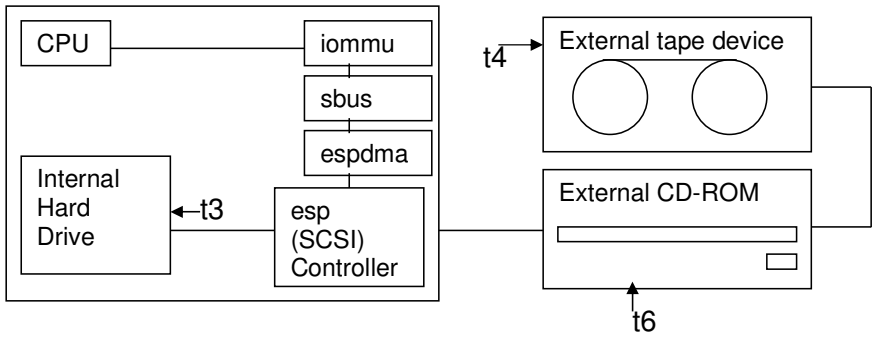
Unit Twelve
Directory Hierarchy

Unit topics

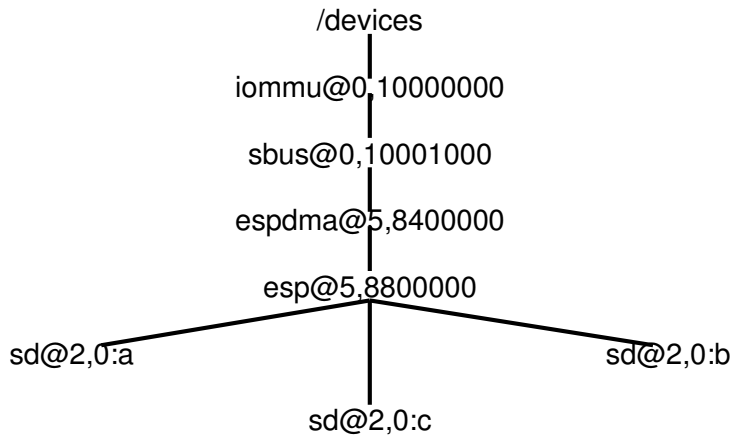
		Page #
12.1	Physical Device Names	283
12.2	Logical Device Names	285
12.3	The /etc/path_to_inst File	287
12.4	Displaying Devices	288
12.5	Virtual Filesystems vs. Physical Filesystems	289
12.6	Why Have Partitions?	291
12.7	The format Utility	292
12.8	Partitioning with SMC	301
12.9	Filesystem Types	303
12.10	The UNIX Filesystem	305
12.11	Creating a New Filesystem	315
12.12	Adding New Devices	317
12.13	Additional Resources	319
12.14	Summary of Commands and Files	320
12.15	Lab Exercises	321

12.1 Physical Device Names

When you learned the PROM command **devalias**, you learned that the system defers to devices by specifying the path it takes to get to the device starting from the CPU. The following chart illustrates the path for SCSI devices:



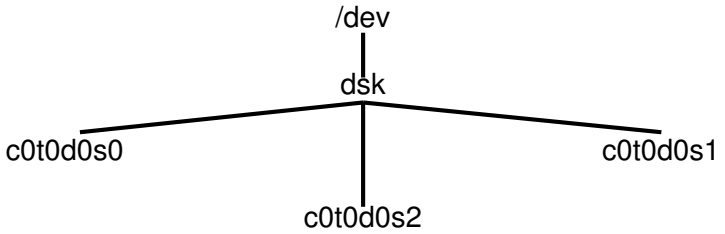
When the system performs a reconfigure boot, all devices are queried and the kernel creates a directory structure based upon the devices found. This directory structure is created under the `/devices` directory:



There are several commands which we will look at in this Unit that require you two refer directly to devices. Because the physical device directory structure can vary from systems to system and because it is difficult to type, many System Administrators prefer to refer to devices using logical device names.

12.2 Logical Device Names

Logical device names also consist of a directory structure that is created during a reconfigure boot. The purpose of logical device names is to create an easy method for System Administrators to refer to devices. The logical device structure looks like this:



A look at these logical device names will show that they are just symbolic links to the physical device names:

```
# cd /dev/dsk
# ls -l c0t3d0s0
lrwxrwxrwx   1 root     root          86 Mar  4  1998 c0t3d0s0 ->
../../devices/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@
5,8800000/sd@3,0:a
```

So, to refer to the boot partition, you can either use:

```
/devices/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/
esp@5,8800000/sd@3,0:a
```

Or you can use:

```
/dev/dsk/c0t0d0s0
```

It's your choice…

12.3 The /etc/path_to_inst File

The kernel keeps track of devices on the system in a file called `/etc/path_to_inst`:

```
# more /etc/path_to_inst
#
#       Caution! This file contains critical kernel state
#
"/iommu@0,10000000" 0 "iommu"
"/iommu@0,10000000/sbus@0,10001000" 0 "sbus"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000" 0 "dma"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000/esp@4,8800000" 0 "esp"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000/esp@4,8800000/st@5,0" 5 "st"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000/esp@4,8800000/st@4,0" 4 "st"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000/esp@4,8800000/st@6,0" 6 "st"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000/esp@4,8800000/st@1,0" 1 "st"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000/esp@4,8800000/st@0,0" 0 "st"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000/esp@4,8800000/st@3,0" 3 "st"
"/iommu@0,10000000/sbus@0,10001000/espdma@4,8400000/esp@4,8800000/st@2,0" 2 "st"
{remaining output omitted}
```

This file is solely maintained by the kernel and should not be modified directly. If this file becomes corrupt, performing a reconfigure boot will result in recreating the file.

12.4 Displaying Devices

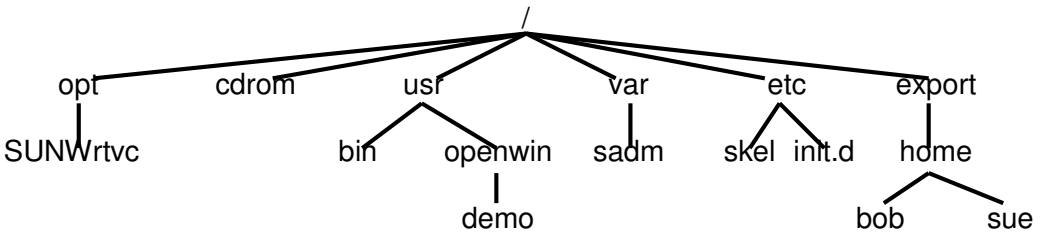
The **prtconf** command can be used to display the devices attached to the system:

```
# prtconf | grep -v not
System Configuration:  Sun Microsystems  sun4m
Memory size: 96 Megabytes
System Peripherals (Software Nodes):

SUNW,SPARCstation-4
    options, instance #0
    iommu, instance #0
        sbus, instance #0
            espdma, instance #0
                esp, instance #0
                    sd, instance #3
                    sd, instance #6
            ledma, instance #0
                le, instance #0
            SUNW,tcx, instance #0
    obio, instance #0
        zs, instance #0
        zs, instance #1
        SUNW,fdtwo, instance #0
    pseudo, instance #0
```

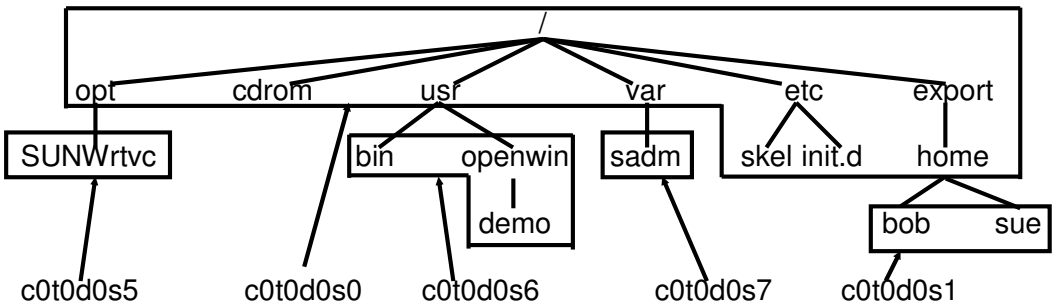
12.5 Virtual Filesystems vs. Physical Filesystems

Most users think of **the filesystem** as something like the following:



The above directory structure is called the **virtual filesystem**. The virtual filesystem is composed of **physical filesystems**. Physical filesystems are really all of the partitions merged together through a process called mounting.

Another was of looking at the directory structure is:



This directory structure illustrated how the different physical filesystems are merged together to create a virtual filesystem. The **mount** command will allow us to create the virtual filesystem and will be discussed in a later Unit.

12.6 Why Have Partitions?

Having multiple partitions (also called slices) provides additional functionality on the system and makes it easier for the system administrator to perform routine tasks.

Advantages of partitioning

- **Reliability**: By having separate slices, you reduce the likelihood of the entire system crashing. For example, if software in the /opt directory causes filesystem corruption and /opt is in its own partition, then only that partition is corrupted.
- **Quicker**: Having smaller slices reduces seek time because the head actuator arm has less distance to travel.
- **Backups**: Many backup utilities allow you to backup by slice (as opposed to directories). If the entire disk is a single partition, you lose this functionality.

Disadvantage of partitioning

- **Installations**: Most software installations require all of the software be placed within one slice. If you create many partitions on a small hard drive, you may not be able to install the required software despite the fact that you have enough overall disk space.

12.7 The format Utility

The **format** utility will allow you to partition your hard drives. When you start the format utility, it will prompt you to choose which disk you wish to format:

```
# format
Searching for disks...done


AVAILABLE DISK SELECTIONS:
       0. c0t3d0 <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>

/iommu@0,10000000/sbus@0,10001000/espdma@5,8400000/esp@5,8800000/
sd@3,0
Specify disk (enter its number):
```

format main menu

The format main menu displays all of the main options for the utility:

```
FORMAT MENU:
        disk        - select a disk
        type        - select (define) a disk type
        partition   - select (define) a partition table
        current     - describe the current disk
        format      - format and analyze the disk
        repair      - repair a defective sector
        label       - write label to the disk
        analyze     - surface analysis
        defect      - defect list management
        backup      - search for backup labels
        verify      - read and display labels
        save        - save new disk/partition definitions
        inquiry     - show vendor, product and revision
        volname     - set 8-character volume name
        !<cmd>      - execute <cmd>, then return
        quit
```

While the format utility can do many disk administration tasks, we will just focus on using it to create and modify during this Unit.

The verify command

If you choose the **verify** command from the main menu, the utility will read the hard disk's label and report the partitions currently on the hard disk:

format> **verify**

Primary label contents:

Volume name = < >
ascii name = <SUN2.1G cyl 2733 alt 2 hd 19 sec 80>
pcyl = 3500
ncyl = 2733
acyl = 2
nhead = 19
nsect = 80

Part	*Tag*	*Flag*	*Cylinders*	*Size*	*Blocks*	
0	*root*	*wm*	*0 - 101*	*75.70MB*	*(102/0/0)*	*155040*
1	*var*	*wm*	*102 - 138*	*27.46MB*	*(37/0/0)*	*56240*
2	*backup*	*wm*	*0 - 2732*	*1.98GB*	*(2733/0/0)*	*4154160*
3	*swap*	*wu*	*139 - 192*	*40.08MB*	*(54/0/0)*	*82080*
4	*unassigned*	*wm*	*193 - 337*	*107.62MB*	*(145/0/0)*	*220400*
5	*unassigned*	*wm*	*1237 - 2532*	*961.88MB*	*(1296/0/0)*	*1969920*
6	*usr*	*wm*	*338 - 1155*	*607.11MB*	*(818/0/0)*	*1243360*
7	*home*	*wm*	*1156 - 1236*	*60.12MB*	*(81/0/0)*	*123120*

The partition command

The **partition** command will take you to the partition menu:



```
format> partition


PARTITION MENU:
        0      - change `0' partition
        1      - change `1' partition
        2      - change `2' partition
        3      - change `3' partition
        4      - change `4' partition
        5      - change `5' partition
        6      - change `6' partition
        7      - change `7' partition
        select - select a predefined table
        modify - modify a predefined partition table
        name   - name the current table
        print  - display the current table
        label  - write partition map and label to the disk
        !<cmd> - execute <cmd>, then return
        quit
```

The print command

While the **print** command looks similar as the **verify** command, its function is slightly different. The **verify** command looks at the disk label and reports its contents. The **print** command displays the "current" (the in-memory copy you are working on) partition table:

partition> **print**
Current partition table (original):
Total disk cylinders available: 2733 + 2 (reserved cylinders)

Part	*Tag*	*Flag*	*Cylinders*	*Size*	*Blocks*	
0	*root*	*wm*	*0 - 101*	*75.70MB*	*(102/0/0)*	*155040*
1	*var*	*wm*	*102 - 138*	*27.46MB*	*(37/0/0)*	*56240*
2	*backup*	*wm*	*0 - 2732*	*1.98GB*	*(2733/0/0)*	*4154160*
3	*swap*	*wu*	*139 - 192*	*40.08MB*	*(54/0/0)*	*82080*
4	*unassigned*	*wm*	*193 - 337*	*107.62MB*	*(145/0/0)*	*220400*
5	*unassigned*	*wm*	*1237 - 2532*	*961.88MB*	*(1296/0/0)*	*1969920*
6	*usr*	*wm*	*338 - 1155*	*607.11MB*	*(818/0/0)*	*1243360*
7	*home*	*wm*	*1156 - 1236*	*60.12MB*	*(81/0/0)*	*123120*

Creating and Modifying partitions

To create a new partition or modify a partition, choose the number of the partition you wish to affect:

```
partition> print
Current partition table (original):
Total disk cylinders available: 2733 + 2 (reserved cylinders)

Part      Tag         Flag     Cylinders         Size            Blocks
  0       root        wm       0 -  101        75.70MB    (102/0/0)      155040
  1       var         wm       102 -  138       27.46MB    (37/0/0)        56240
  2       backup      wm       0 - 2732         1.98GB    (2733/0/0)    4154160
  3       swap        wu       139 -  192       40.08MB    (54/0/0)        82080
  4       unassigned  wm       193 -  337      107.62MB    (145/0/0)      220400
  5       unassigned  wm       1237 - 2532      961.88MB    (1296/0/0)    1969920
  6       usr         wm       338 - 1155       607.11MB    (818/0/0)     1243360
  7       home        wm       1156 - 1236       60.12MB    (81/0/0)       123120

partition> 6
Part      Tag         Flag     Cylinders         Size            Blocks
  6       usr         wm       338 - 1155       607.11MB    (818/0/0)     1243360

Enter partition id tag[usr]:
Enter partition permission flags[wm]:
Enter new starting cyl[338]:
Enter partition size[1243360b, 818c, 607.11mb, 0.59gb]: 500mb
```

```
partition> print
Current partition table (unnamed):
Total disk cylinders available: 2733 + 2 (reserved cylinders)

Part      Tag         Flag     Cylinders         Size            Blocks
  0       root        wm       0 -  101         75.70MB    (102/0/0)         155040
  1       var         wm       102 -  138        27.46MB    (37/0/0)          56240
  2       backup      wm       0 - 2732         1.98GB     (2733/0/0)        4154160
  3       swap        wu       139 -  192        40.08MB    (54/0/0)          82080
  4       unassigned  wm       193 -  337        107.62MB   (145/0/0)         220400
  5       unassigned  wm       1237 - 2532       961.88MB   (1296/0/0)        1969920
6         usr         wm       338 - 1011        500.23MB   (674/0/0)         1024480
  7       home        wm       1156 - 1236       60.12MB    (81/0/0)          123120
```

Note: the partition tag and flag are not used by the system for any purpose.

Make changes permanent

To save these changes to the disk label, use the **label** command:

```
partition> label
{output omitted}
```

Saving the partition table

You can save your partition table to the `/etc/format.dat` file for future use by using the **name** command:

```
format> partition
{output omitted}
partition> name
Enter table name (remember quotes): sample
partition> quit
{output omitted}
format> save
Saving new disk and partition definitions
Enter file name["./format.dat"]: /etc/format.dat
format> quit
```

Notes:

- The name command can only be run at the format prompt.
- The quotes are required only if there are spaces in the table name.

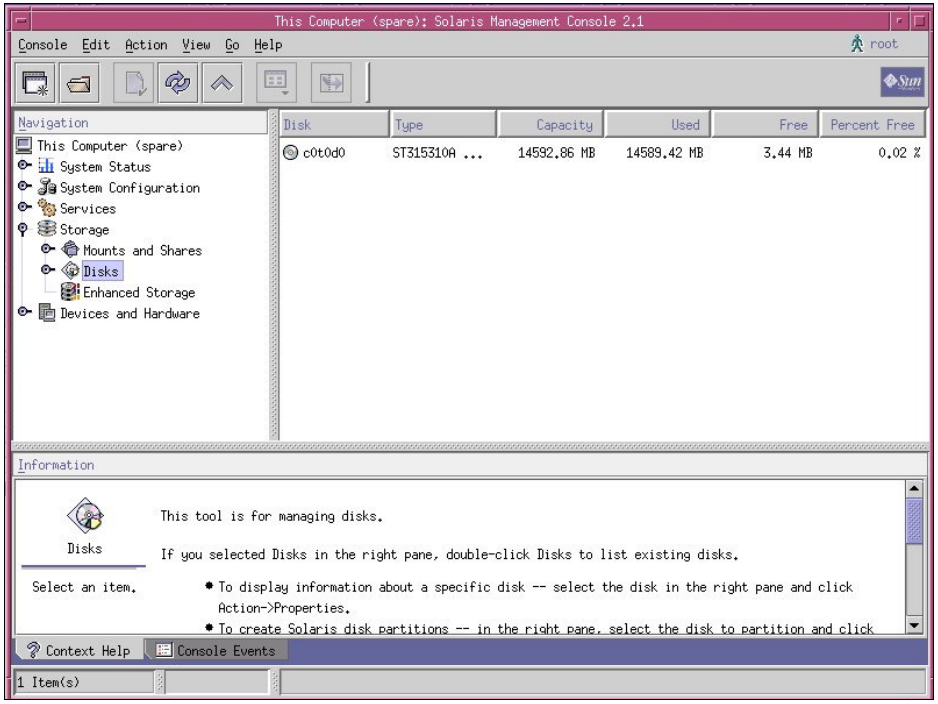
Selecting a partition table

Use the select command to choose a partition table from the `/etc/format.dat` file:

```
format> partition
{output omitted}
partition> select
        0. original
        1. sample
Specify table (enter its number)[0]: 1
```

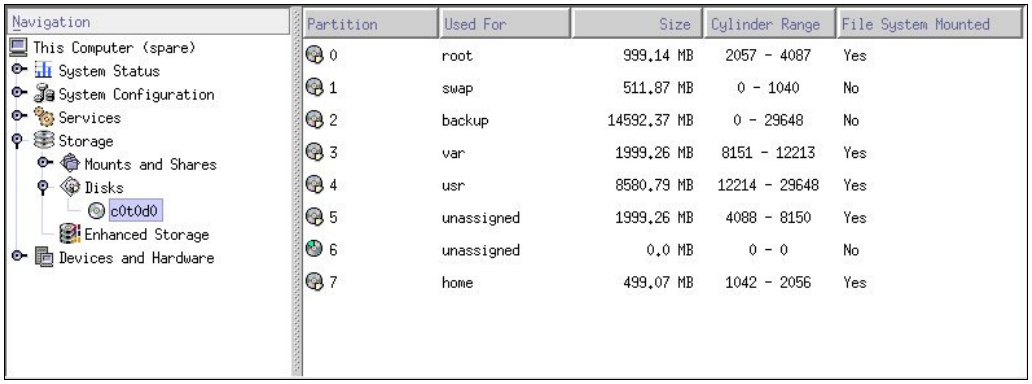
12.8 Partitioning with SMC

The Solaris Management Console can also be used to administer partitions. After starting the SMC, expand “This computer” and “Storage” and then select “Disks”:



View partitions

To view partition information, double-click on the “disk” icon or click on the “disk” icon:



Create partitions

To create a new partition, click “Action” + “Create Solaris Partitions…”.

12.9 Filesystem Types

Solaris supports three different filesystem types: Disk-based, Distributed and Pseudo. This Unit focuses primarily on one of the Disk-based filesystems, however having an understanding of the available filesystems is important.

Disk-based filesystems

UFS - The UNIX filesystem is the standard for hard disks.

HSFS - The High Sierra filesystem is designed for CD-ROMs.

PCFS - The PC filesystem is equivalent to DOS's FAT32 filesystem. If you have a floppy disk that needs to be read in both Solaris and DOS/Windows, format it as a PCFS disk.

UDFS - The Universal Disk filesystem is used on both CD-ROMs and DVDs.

Distributed filesystems

NFS - Network Filesystem provides a protocol for sharing data across a network.

Pseudo filesystems

TMPFS - A temporary filesystem is used to store files in memory instead of on the hard drive.

SWAPFS - The swap filesystem is used to store swap data on the hard drive.

FDFS - The file descriptor filesystem is used to access file by using file descriptors (`/dev/fd/1`, `/dev/fd/2`, etc.).

PROCFS - The process filesystem is used for the `/proc` directory to maintain a list of active processes.

MNTFS - The mount filesystem is used to store information for the kernel about local mounted filesystems.

Note: As a system administrator, you will rarely ever make or modify FDFS, PROCFS or MNTFS filesystems.

12.10 The UNIX Filesystem

After you have created a partition, you will need to place the UNIX filesystem (UFS) on the partition. The next section will explain the **newfs** command which is used to generate a UFS on a partition. This section will define the portions of the UFS.

The UFS consists of the following parts:

- Volume Table of Contents (VTOC)
- Boot Label
- Cylinder Groups
- Superbock
- Cylinder Group Blocks
- Backup Superblocks
- Inode Tables
- Data blocks

Volume Table of Contents (VTOC)

The VTOC (also called the "disk label") contains a table of the partitions on the hard drive. The contents of the VTOC can be displayed from within the **format** utility with the **verify** command or by using the **prtvtoc** command on the command line:

```
# prtvtoc /dev/dsk/c0t3d0s0
* /dev/dsk/c0t3d0s0 partition map
* Dimensions:
*     512 bytes/sector
*     168 sectors/track
*      20 tracks/cylinder
*    5268 cylinders
*    5266 accessible cylinders
* Flags:
*   1: unmountable
*  10: read-only
*                                First     Sector    Last
* Partition  Tag  Flags        Sector      Count    Sector  Mount Directory
       0      2    00             0     309120    309120   /
       1      7    00        309120     504000    813119   /var
       2      5    00             0   17693760  17693760
       3      3    01        813120     493920   1307039
       5      0    00       1307040    9216480  10523519   /opt
       6      4    00      10523520    5120640  15644159   /usr
       7      8    00       5644160    2049600  17693759   /export/home
```

Boot Label

The boot label is designed to contain the bootblk and ufsboot programs. These programs should only exist on the boot partition (typically partition 0), but there is an empty boot label in all other partitions as well.

Cylinder Groups

For organizational reasons, the partition is divided into cylinder groups. Typically there are 16 cylinders per group, but there is a configuration option to allow for different sizes in the **newfs** command.

Superbock

The superblock contains information about the partition itself. This information includes:

- FS state (clean, stable or active): The FS state is used by the **fsck** utility when it is run during boot to determine what filesystems to check automatically. See the Unit on **fsck** for more details.
- Size of filesystem
- Size of data blocks on filesystem
- Size of the cylinder groups on this filesystem
- Number of data blocks in filesystem
- Path name of last mount point
- Date and time of last update to the superblock

Cylinder Group Blocks

Each cylinder group has a table called a cylinder group block. This table includes information about the cylinder group including:

- The number of inodes in this cylinder group
- The number of directories in this cylinder group
- The number of data blocks in this cylinder group
- A free block map
- A used inode map

Backup Superblocks

Since the superblock is critical to the filesystem, backups are made of it in each cylinder group. The backups are made in a “cascade” method in which the first cylinder group has the most recent backup, the second cylinder group has the next most recent backup, etc.

These backup superblocks can be used by the **fsck** utility to recreate the primary superblock. A future Unit in this manual will cover more details regarding this procedure.

Inode Tables

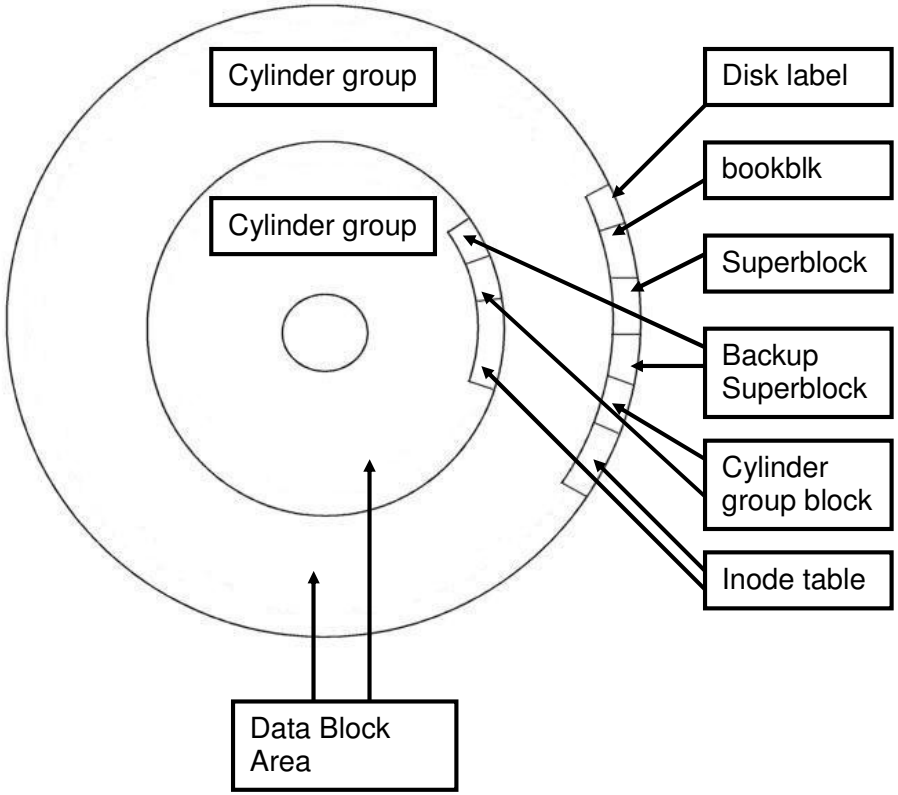
Inode tables are used to store data about files. The basically contain everything about a file except for the data itself and the file name. This information includes:

- inode number – a unique number to identify the file
- File type
- Permissions
- Hard link count
- UID of the file owner
- GID of the group owner
- Size of file
- Date/time stamp of last time file was accessed
- Date/time stamp of last time file was modified
- Date/time stamp of last time inode table was modified
- 12 Direct block pointers
- 1 single-indirect pointer
- 1 double-indirect pointer
- 1 triple-indirect pointer

Data blocks

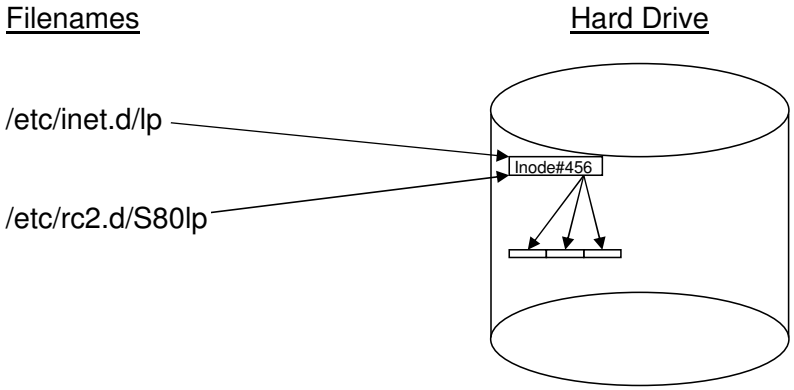
Data blocks are where the file's data is stored. The default size of data blocks is 8 kilobytes.

Visual Example of a UFS



Hard Links

When two "files" are said to be hard linked it means that they share the same inode table. A visual representation of hard linked files:



To create a hard link to an existing file, use the **ln** command:

```
# ln /etc/skel/local.login /etc/local.login
```

Finding Hard Links

Since hard links share inode tables (and inode numbers) you can find what filenames are hard linked together by looking for files with the same hard links:

```
# cd /etc/inet.d
# ls -i lp
1214 lp
# find /etc -inum 1214 -print
{output omitted}
```

Additional notes regarding Hard Links

- Since each partition (filesystem) has its own inode tables, hard links cannot "span" across filesystems:

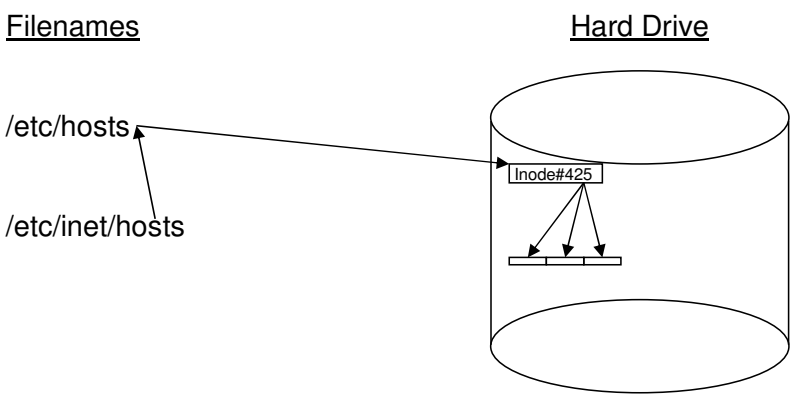
```
# ln /etc/inet.d/lp /opt/lp
{results in an error}
```

- An advantage hard links is that once you hard link two filenames together, you can remove either one and still have the file.

- You cannot hard link directories.

Soft Links

When two "files" are said to be soft linked (also known as symbolic linked) it means one file "points" to the location of another file. A visual representation of soft linked files:



Creating soft Links

To create a hard link to an existing file, use the **ln** command with the **-s** option:

```
# ln –s /etc/skel/local.login /etc/local.login
```

Finding Soft Links

The **ls -l** command displays soft links.

Soft Link Pros and Cons

- Since soft links point to files via that directory structure, soft links can span filesystems:
- If you delete the original file, the soft link is meaningless.
- You can soft link directories

12.11 Creating a New Filesystem

After creating a new partition, you must put a filesystem on it so files can be stored on it. To create a new filesystem, use the **newfs** command:

```
# newfs /dev/rdsk/c0t3d0s6
newfs: construct a new filesystem /dev/rdsk/c0t3d0s6: (y/n)? y
/dev/rdsk/c0t3d0s6: 3800 sectors in 80 cylinders of 2 tracks, 22 sectors
1.8MB in 6 cyl groups (16 c/g, 0.32MB/g, 140i/g)
super-block backups (for fsck -F ufs -b b=#) at: 32 680, 1224, 1890, 2634
```

WARNING: Running this command on an existing filesystem will PERMANENTLY delete all files on the partition.

Filesystem Maintenance Space

When you create a filesystem with the **newfs** command, some space in the filesystem is reserved for system use. This space, called "minfree" space, can not be used by regular users. The minfree space is important because if users fill a filesystem, the OS's processes can still make use of the filesystem.

According the newfs man page, the **newfs** command reserves "((64_Mbytes/partition_ size) * 100), rounded down to the nearest integer and limited between 1% and 10%, inclusively". For example, if you have a partition that is 900 MB in size, the percent of minfree space would be 7% (64/900*100=7.111).

To specify a different minfree size when creating a filesystem, use the **-m** option:

```
# newfs -m 5 /dev/rdsk/c0t3d0s6
{output omitted}
```

If you want to see the minfree size of an existing, use the **fstyp** command:

```
# fstyp -v /dev/rdsk/c0t0d0s0 | grep minfree
minfree 6%      maxbpg  2048    optim   time
```

To modify the minfree size of an existing filesystem, use the **tunefs** command:

```
# fstyp -v /dev/rdsk/c0t0d0s7 | grep minfree
minfree 5%      maxbpg  2048    optim   time
# tunefs -m 10 /dev/rdsk/c0t0d0s7
minimum percentage of free space changes from 5% to 10%
```

12.12 Adding New Devices

There are several methods to adding a new device. The easiest method is the take the following steps:

#1. Create a file called `/reconfigure` by using the touch command. If this file exists, the system will perform a reconfigure boot next type the system boots:

```
# touch /reconfigure
```

#2. Shut down the system:

```
# init 5
```

#3. Add the new device.

#4. Turn the system on.

#5. After the system has booted, log in as root and verify that the system recognizes the new device.

The advantage of a reconfigure boot is that it automatically creates all of the files that Solaris needs to recognize devices. However, if you add a new device and forget to perform a reconfigure boot, it can be a pain to have to reboot the system. Once the system has been booted, you can have the device files created by using a variety of commands:

Adding new disk devices

The **devfsadm** command can perform the same operations as a reconfigure boot. It will modify the `/etc/path_to_inst` file as well as the `/dev` and `/devices` directories.

If you have added a new CD-ROM or hard drive device, use the **-c** option with the disk argument:

```
# devfsadm -c disk
```

Other arguments to the **-c** option are tape, port, audio and pseudo.

The **devfsadm** command keeps track of the changes that it makes and these changes can be viewed with the **-v** option.

When you remove devices, you can clean up references to these devices with the **-C** option.

12.13Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #15

Web sites

System Administration Guide: Advanced Administration, 13 Managing Disk Use:
http://docs.sun.com/app/docs/doc/817-0403/6mg741bv9?a=view

Man pages

devfsadm	format	newfs	tapes
disks	fstyp	prtconf	tunefs
drvconfig	ln		

12.14 Summary of Commands and Files

Command	Description
devfsadm	For Solaris 9 and higher, generates the device files when a new device is added to the system
disks	For Solaris 8 and previous, generates the logical device when a new hard drive is added to the system
drvconfig	For Solaris 8 and previous, generates the physical device when a new device is added to the system
format	Administers partitions
fstyp	Displays filesystem attributes
ln	Creates hard and symbolic link files
newfs	Creates UNIX filesystems on partitions
prtconf	Displays devices attached to system
tapes	For Solaris 8 and previous, generates the logical device when a new tape drive is added to the system
tunefs	Modifies filesystem attributes

File	Description
/dev	Directory structure that contains links to device files
/devices	Directory structure that contains device files
/etc/path_to_inst	File used by kernel to keep track of system devices

12.15 Lab Exercises

No exercise labs for Unit 12

Unit Thirteen
Local Disk Management

Unit topics

		Page #
13.1	What is Mounting?	323
13.2	The mount Command	324
13.3	Mount Rules	327
13.4	The umount Command	328
13.5	umount Rules	329
13.6	Mounting Automatically at Boot	332
13.7	The mountall Command	336
13.8	The umountall Command	337
13.9	Review: the df Command	338
13.10	Mounting PCFS and HSFS	339
13.11	Mounting from the CD-ROM	341
13.12	Additional Resources	343
13.13	Summary of Commands and Files	344
13.14	Lab Exercises	345

13.1 What is Mounting?

Mounting is the process of "merging" separate physical filesystems (partitions) into one "virtual" filesystem.

The root filesystem is the starting point of the virtual filesystem.

All other physical filesystems are mounted under the root filesystem at mount points.

Mount points are directories that provide access to mounted filesystem.

Mounting and unmounting can be accomplished manually with the **mount**, **mountall**, **umount** and **umountall** commands.

Mounting can also be accomplished automatically at boot by placing entries in the `/etc/vfstab` (vfs stands for virtual filesystem).

Notes:

Notes:

13.2 The mount Command

Displaying mounts

Using the **mount** command with no arguments will cause the system to display the filesystems that are currently mounted:

```
# mount
/ on /dev/dsk/c0t0d0s0 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200000 on Mon Jun 16 11:17:20 2003
/usr on /dev/dsk/c0t0d0s4 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200004 on Mon Jun 16 11:17:20 2003
/proc on /proc read/write/setuid/dev=4980000 on Mon Jun 16 11:17:19 2003
/etc/mnttab on mnttab read/write/setuid/dev=4a40000 on Mon Jun 16 11:17:19 2003
/dev/fd on fd read/write/setuid/dev=4a80000 on Mon Jun 16 11:17:21 2003
/var on /dev/dsk/c0t0d0s3 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200003 on Mon Jun 16 11:17:23 2003
/var/run on swap read/write/setuid/xattr/dev=1 on Mon Jun 16 11:17:23 2003
/tmp on swap read/write/setuid/xattr/dev=2 on Mon Jun 16 11:17:27 2003
/opt on /dev/dsk/c0t0d0s5 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200005 on Mon Jun 16 11:17:27 2003
/export/home on /dev/dsk/c0t0d0s7 read/write/setuid/intr/largefiles/xattr/onerror=panic/dev=2200007 on Mon Jun 16 11:17:27 2003
```

A breakdown of the output fields:

Field	Description
Mount point	Directory where partition or device is mounted
Device name	Name of partition or device
Mount options	Options used when partition or device was mounted (see more on mount options in a later section)
Timestamp	Date and time mount process took place

Notes:

Notes:

Mounting filesystems

Use the **mount** command to manually mount filesystems:

```
# mount /dev/dsk/c0t0d0s5 /database
{no output}
```

13.3 Mount Rules

The following "rules" apply when you are trying to mount a filesystem:

1. The mount point (mount directory) must exist.
2. A filesystem (partition) can only be mounted once.
3. Each mount point can only be used to mount one filesystem at a time.
4. Any files and directories that existed under a directory are "hidden" if the directory is used as a mount point for another filesystem.
5. Only the superuser can mount a filesystem.

Notes:

13.4 The umount Command

To unmount a filesystem, use the **umount** command:

```
# umount /dev/dsk/c0t3d0s5
{no output}
```

Note: either the device name (`/dev/dsk/c0t3d0s5` in this case) or the mount point can be specified.

Notes:

13.5 umount Rules

The following rules apply to the **umount** command:

1. You can't unmount a filesystem if a user is in that filesystem. *
2. You can't unmount a filesystem if a file is opened from that filesystem.*
3. You can't unmount a filesystem has a process is running from that filesystem. *
4. Only the superuser can unmount a filesystem.
5. The command is **umount**, not unmount...:+}

* Unless you force the umount process

Notes:

Since you will need sometimes to umount a filesystem that is “busy”, there are a couple of methods that will allow you to do so:

Notes:

List and stop processes running in a filesystem

The fusers command will allow you to list processes running in a filesystem and stop these processes as well.

The **fuser** command can be used to display the processes that are accessing a filesystem:

```
# fuser /export/home
/export/home:     4167c    4139c
```

The letter after the process ID number is a code that provides information as to how the process is using the filesystem. For example, the “c” in the previous output indicates that the process is using the filesystem as its current directory. If you look that the name of the process with the **ps** command and find that it is a shell process, this would indicate that a user is in that filesystem:

```
# ps -fe | grep 4139
   root  4139  4135  0 23:06:14 pts/3    0:00 tcsh
```

Note: View the man pages of the **fuser** command to see more details regarding the process ID code.

Notes:

To display processes and owners of the processes in a filesystem, use the **-cu** options:

```
# fuser -cu /opt
/opt:    12425ctm(root)
```

To stop all processes in a filesystem, use the **-ck** options:

```
# fuser -ck /opt
/opt:    12425ctm
[1]  + Killed                  ./emacs
```

Force the umounting process (Solaris 8 and above only)

Use the **-f** option to force the umount of a busy filesystem:

```
# umount /opt
umount: /opt busy
# umount -f /opt
```

13.6 Mounting Automatically at Boot

To mount filesystems automatically at boot, edit the `/etc/vfstab` file and specify what filesystem to mount. This file is a database that contains information on where and how to mount filesystems:

```
# more /etc/vfstab
#device        device          mount           FS      fsck    mount   mount
#to mount      to fsck         point           type    pass    at boot options
#
#/dev/dsk/c1d0s2 /dev/rdsk/c1d0s2 /usr         ufs     1       yes     -
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t3d0s3       -       -       swap    -       no      -
/dev/dsk/c0t3d0s0       /dev/rdsk/c0t3d0s0      /       ufs     1       no      -
/dev/dsk/c0t3d0s6       /dev/rdsk/c0t3d0s6      /usr    ufs     1       no      -
/dev/dsk/c0t3d0s1       /dev/rdsk/c0t3d0s1      /var    ufs     1       no      -
/dev/dsk/c0t3d0s7       /dev/rdsk/c0t3d0s7      /export/home    ufs     2       yes     -
swap    -       /tmp    tmpfs   -       yes     -
/dev/dsk/c0t3d0s5 /dev/rdsk/c0t3d0s5 /opt ufs 1 yes -
```

Note: The `/root`, `/var` and `/usr` filesystems have “no” for the “mount at boot” field. This is because these filesystems are already mounted by the time the `/etc/vfstab` file is read.

Notes:

Notes:

Mount options

The last field of the `/etc/vfstab` is used to specify how to perform the mount. Here are the most common mount options:

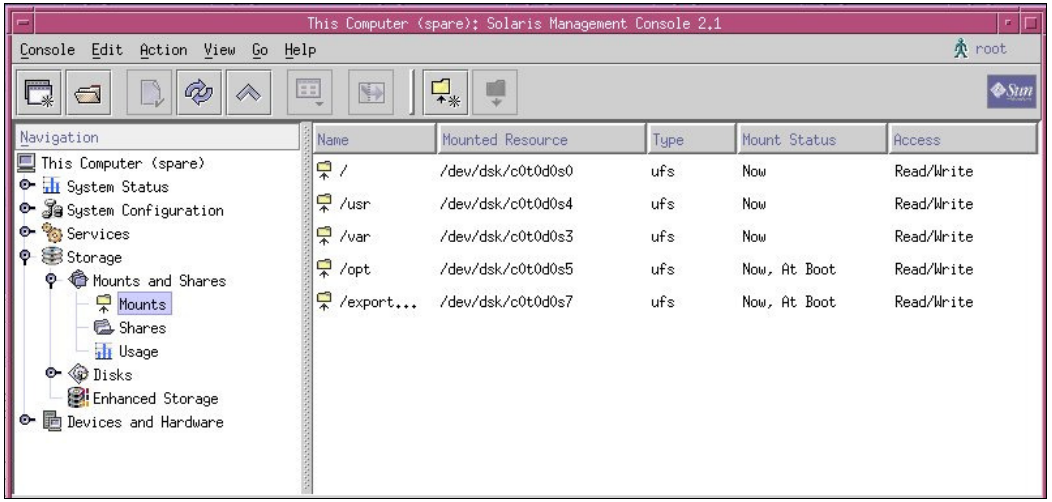
Options	Meaning
rw or ro	Mount the filesystem as read/write (rw) or read-only (ro). If not specified, **the default is rw.**
bg or fg	Mount in the background or in the foreground **The default is fg**.
soft or hard	A soft mount is one that will stop after one attempt to mount. A hard mount will continue to retry until the sever responds. **The default is hard**.
intr or nointr	intr will allow keyboard interruptions while nointr will not allow keyboard interruptions. This only applies to interrupting processes that are executed from filesystems that are locked. **The default is intr.**
setuid or nosetuid	setuid allows programs to execute with setuid permission. nosetuid disallows setuid execution. **The default is setuid.**
logging or nologging	logging enables the storing of "transactions" (changing of a file) in a log while the changes are being made. This allows for quicker filesystem recovery in the event of a system crash. **The default is nologging.**
largefiles or nolargefiles	On a ufs filesystem, largefiles allows for file size to be greater than 2 GB. If an existing file on the filesystem is large than 2 GB, the filesystem can not be mounted with the nolargefiles option. **The default is largefiles**

Notes:

onerror=action	This specifies what action to take in the event of an internal inconsistency in the filesystem. The “action” can be one of three items: panic - Forces a system shutdown lock - Locks the filesystem umount - Unmounts the filesystem
noatime	Don’t record access times in inode table. This reduces disk activity.

The Solaris Management Console

You can also use the SMC to modify existing local filesystem mounts (the SMC doesn't provide you with the ability to create new local filesystem mounts):



Notes:

13.7 The mountall Command

The **mountall** command will mount all of the filesystems that are specified in the `/etc/vfstab` file. More specifically, it looks for entries in the `/etc/vfstab` that have "**yes**" in the "**mount at boot**" field and mounts that filesystem on the designated mount point:

```
# more /etc/vfstab
#device        device         mount           FS      fsck    mount   mount
#to mount      to fsck        point           type    pass    at boot options
#
#/dev/dsk/c1d0s2 /dev/rdsk/c1d0s2 /usr        ufs     1       yes     -
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/dsk/c0t3d0s3       -       -       swap    -       no      -
/dev/dsk/c0t3d0s0       /dev/rdsk/c0t3d0s0      /       ufs     1       no      -
/dev/dsk/c0t3d0s6       /dev/rdsk/c0t3d0s6      /usr    ufs     1       no      -
/dev/dsk/c0t3d0s1       /dev/rdsk/c0t3d0s1      /var    ufs     1       no      -
/dev/dsk/c0t3d0s7       /dev/rdsk/c0t3d0s7      /export/home    ufs     2       yes     -
swap    -       /tmp    tmpfs   -       yes     -
/dev/dsk/c0t3d0s5 /dev/rdsk/c0t3d0s5 /opt ufs 1 yes -
```

Note: if the filesystem is already mounted, the **mountall** command has no affect on it.

Notes:

13.8 The umountall Command

The **umountall** command will *attempt* to unmount all filesystems:

```
# umountall
umount: /net busy
umount: /opt busy
```

If a filesystem can't be unmounted (see the section on umount rules), an error message is displayed.

Common options

-l unmount only local filesystems
-r unmount only remote (NFS) filesystems

Notes:

13.9 Review: the df Command

To determine how much space is being used by a filesystem (partition), use the **df** command:

df -k

Filesystem	*kbytes*	*used*	*avail*	*capacity*	*Mounted on*
/dev/dsk/c0t3d0s0	*72407*	*28861*	*43474*	*40%*	*/*
/dev/dsk/c0t3d0s6	*583807*	*406870*	*176354*	*70%*	*/usr*
/proc	*0*	*0*	*0*	*0%*	*/proc*
fd	*0*	*0*	*0*	*0%*	*/dev/fd*
/dev/dsk/c0t3d0s1	*25919*	*13914*	*11980*	*54%*	*/var*
/dev/dsk/c0t3d0s7	*57143*	*347*	*56739*	*1%*	*/export/home*
swap	*103972*	*196*	*103776*	*1%*	*/tmp*
/dev/dsk/c0t3d0s4	*102871*	*62056*	*40713*	*61%*	*/opt2*
/dev/dsk/c0t3d0s5	*925974*	*883640*	*40791*	*96%*	*/opt*
mvfs=/view	*72407*	*28861*	*43474*	*40%*	*/view*

Common options

-k Show in kilobytes

Notes:



13.10 Mounting PCFS and HSFS

Notes:

The vold daemon

If the **vold** daemon is running, the CD-ROM will automatically be mounted when placed in the CD-ROM drive.

If the daemon is not running, then the CD-ROM will have to be manually mounted. Since only the root user can mount filesystems, the only way other users can have access to the CD-ROM is if the **vold** daemon is running.

Mount CD-ROM manually

If the **vold** daemon isn't running, the **mount** command can be used to manually mount the CD-ROM:

```
# mount –F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
{no output}
```

Note: HSFS stands from High Sierra Filesystem, the filesystem type used for CD's.

 © 2014 One Course Source

Notes:

Mount floppies manually

The **mount** command can also be used to manually mount floppy drives:

```
# mount –F pcfs /dev/diskette /floppy
{no output}
```

Since only System Administrators can use the **mount** commands, regular users have a command that allows them to mount floppy disks:

```
sun% volcheck
```

The **volcheck** command will automatically mount the floppy under the `/floppy` directory.

The `/etc/vold.conf` file can be used to modify the behavior of the vold daemon. Consult the man pages for vold.conf for details.

Notes:

13.11 Mounting from the CD-ROM

On some occasions you may have to mount the hard drive while booted from the CD-ROM. For example, if a system file gets corrupted (such as the `/etc/passwd`) and you can't log in.

Recover root password

Take the following steps to recover the root password:

1. Boot from the installation CD in single user mode (single user mode on the installation CD doesn't start the installation program):

```
ok boot cdrom -s
{output omitted}
```

2. Mount the boot partition on the `/a` directory. This directory is placed on the installation CD just for mounting:

```
# mount /dev/dsk/c0t3d0s0 /a
{no output}
```

3. The system doesn't know what type of display device you are using, so you must set the environment variable TERM prior to editing files with the **vi** editor:

```
# TERM=sun
# export TERM
```

Notes:

4. Edit the shadow file on the hard drive with the **vi** editor and remove root's password field:

```
# vi /a/etc/shadow
```

Note: Do not attempt to edit the `/etc/shadow` file as this is the CD's shadow file.

5. Reboot the system and log in as root. Press <ENTER> if a password is requested:

```
# reboot
```

6. Create a new password immediately for root:

```
# passwd
```

13.12 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #16

Web sites

System Administration Guide: Advanced Administration, 13 Managing Disk Use:
http://docs.sun.com/app/docs/doc/817-0403/6mg741bv9?a=view

Man pages

df	mountall	umountall
fuser	smc	volcheck
mount	umount	vold

13.13 Summary of Commands and Files

Command	Description
df	Displays disk usage on a filesystem basis
fuser	Display processes "using" a filesystem
mount	Mounts a filesystem
mountall	Mounts all filesystems defined in /etc/vfstab file
smc	GUI-based admin utility
umount	Unmounts a filesystem
umountall	Unmounts all filesystems that are not "busy"
volcheck	Mounts floppy disks

File	Description
/etc/vfstab	Defines filesystems mounts
/etc/vold.conf	Configures the vold daemon

13.14 Lab Exercises

Be very careful while working on this lab!

Scenario #1: There is a new software product (still in beta testing) that you want to install and test. Because it could be unstable, you want to install it in its own partition. While there is no "free space" left on the hard drive, the /export/home partition (slice) is much larger than it needs to be. Repartition this slice so it is approximately 1/2 of the original size and create a new partition from the other half. Some things to think about:

1. You don't want the data that is currently in `/export/home` to be lost (so, back it up with **cp -r**)
2. Remember, to make use of a new partition, there are **3** steps: format it, create a new filesystem, mount it.
3. If you mess up the format portion, not only will the next two steps not work, you will most likely need to reinstall the OS.

Scenario #2 - You need more swap space on your machine. Find a partition that has 35 MB of free space and create a swap file.

Scenario #3 - Make the swap file you created in Scenario #2 a permanent swap file.

Unit Fourteen
Configuring Filesystems

Unit topics

		Page #
14.1	Why Filesystems Break	347
14.2	Fixing Filesystems with fsck	348
14.3	fsck Examples	350
14.4	Additional Resources	354
14.5	Summary of Commands and Files	355
14.6	Lab Exercises	356

14.1 Why Filesystems Break

Whenever the filesystem is modified, the superblocks, cylinder group blocks and inode tables are changed. To increase the system's speed when performing these actions, these changes are not stored directly on the hard drive.

These changes are stored in memory to make the process of filesystem modification quicker. Every 30 seconds the **fsflush** daemon flushes the memory of these filesystem changes and updates the hard drive.

If the system is improperly brought down, the **fsflush** daemon never gets the chance to update the hard drive. This results in filesystem corruption: when the superblocks, cylinder group blocks and the inode tables do not contain the correct information.

Improper shutdown

The following methods can cause filesystem corruption:
1. Power failure
2. A Stop-a
3. Turning off the power to the system

Notes:

14.2 Fixing Filesystems with fsck

The **fsck** utility checks for and will fix filesystem inconsistencies. It can be run in two different "modes": non-interactive and interactive.

Non-interactive Mode

The non-interactive mode is typically run only during the boot process. When run in this mode, **fsck** will attempt to fix "simple" problems to the filesystem. Any other problems **fsck** encounters forces it to abort because fixing such problems would require "human interaction". This mode is also called the “preen” mode.

Interactive Mode

When **fsck** is run in the interactive mode, the utility lists filesystem errors and a suggested solution. It also prompts the user if he/she wants to have the utility fix the problem.

Note: If you run **fsck** on an active filesystem, it will almost always return an error. **fsck** should always be run on a unmounted filesystem.

Notes:

Notes:

Data checked by fsck

The **fsck** utility check for inconsistencies such as:

- Data blocks that more than one inode.
- Incorrect link counts
- Incorrect inode format
- Incorrect directory size
- Unallocated data blocks
- Incorrect free block list
- Incorrect total of free or used data blocks

The aforementioned list is not a complete list of items that the **fsck** utility checks for. For more details, look at the fsck_ufs man page.

14.3 fsck Examples

Typical interactive example

When you run **fsck** in the interactive mode, it prompts the user to fix problems it encounters. The following displays a few of the most common error messages:

```
# fsck /dev/dsk/c0t3d0s0
{output omitted}
```

Responding "yes" to all inquiries

Use the **-y** option to answer "yes" to all of **fsck**'s inquiries:

```
# fsck –y /dev/dsk/c0t3d0s0
{output omitted}
```

Notes:

Notes:

Using alternate superblock

On some occasions the superblock may become so corrupted that is can not be "fixed". In these cases, an alternate superblock must be used.

To find where the superblocks are on a filesystem, use the **newfs –N** command:

```
# newfs –N /dev/dsk/c0t3d0s0
/dev/rdsk/c0t3d0s0:     155040 sectors in 102 cylinders of 19 tracks, 80 sectors
        75.7MB in 7 cyl groups (16 c/g, 11.88MB/g, 5696 i/g)
super-block backups (for fsck -F ufs -o b=#) at:
 32, 24432, 48832, 73232, 97632, 122032, 146432,
```

Once the backup superblock's locations have been determined, use the first backup to attempt to fix the filesystem:

```
# fsck –o b=32 /dev/dsk/c0t3d0s0
{output omitted}
```

If this backup is also corrupted, use the next one:

```
# fsck –o b=24432 /dev/dsk/c0t3d0s0
{output omitted}
```

Notes:

Reconnecting unreferenced files

Most of the filesystem errors can be fixed by simply answering "yes" to all questions. In the case when a file becomes "unreferenced", additional steps are necessary.

An unreferenced file is one that has an inode table and has been allocated data blocks, but isn't referenced in any directory. In other words, the file exists, but you can't access it from any directory.

If you answer "yes" when asked to reconnect an unreferenced file, the file will be placed in the lost+found directory of that filesystem:

After the files have been placed in the lost+found directory, it is the administrator's responsibility to determine where the file needs to be placed. A few suggestions:

1. View the contents of the file (if it is a text file).
2. See if the file is a system file or owned by a user.
3. If the file is a system file, compare its attributes with those found in the `/var/sadm/install/contents` file.

Notes:

Fixing the / and /usr filesystems

If either the / or /usr filesystems becomes corrupt, the system will not boot. In these cases, you must boot off the cdrom to fix these filesystems:

```
# reboot – "cdrom –s"
{system reboots off cdrom}
# fsck –y /dev/dsk/c0t3d0s[06]
{output omitted}
```

14.4 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #16

Web sites

System Administration Guide: Advanced Administration, 13 Managing Disk Use:
http://docs.sun.com/app/docs/doc/817-0403/6mg741bv9?a=view

Man pages

fsck fsck_ufs

14.5 Summary of Commands and Files

Command	Description
fsck	Fixes broken filesystems

File	Description
None	None

14.6 Lab Exercises

No exercises for Unit 14

Unit Fifteen
Administrating Processes

Unit topics

		Page #
15.1	Listing and Stopping Processes	358
15.2	CDE Process Manager (Solaris 8 and Higher Only)	359
15.3	The prstat Command (Solaris 8 and Higher Only)	361
15.4	The Solaris Management Console (Solaris 9 and Higher Only)	362
15.5	The ps and kill Commands	364
15.6	The pgrep and pkill Commands (Solaris 8 and Higher Only)	368
15.7	The xkill Command	371
15.8	Controlling Job Priority	372
15.9	Using crontab	373
15.10	Using the at Command	379
15.11	Additional Resources	384
15.12	Summary of Commands and Files	385
15.13	Lab Exercises	386

Notes:

15.1 Listing and Stopping Processes

As a system administrator you will often need to list the processes that are running on the system. In addition, you may need to stop process as well.

There are several methods of listing and/or stopping processes:

#1. CDE Process Manager (Solaris 8 and higher only)
#2. The **prstat** command (Solaris 8 and higher only)
#3. The Solaris Management Console (Solaris 9 and higher only)
#4. The **ps** and **kill** commands
#5. The **pgrep** and **pkill** command
#6. The **xkill** command

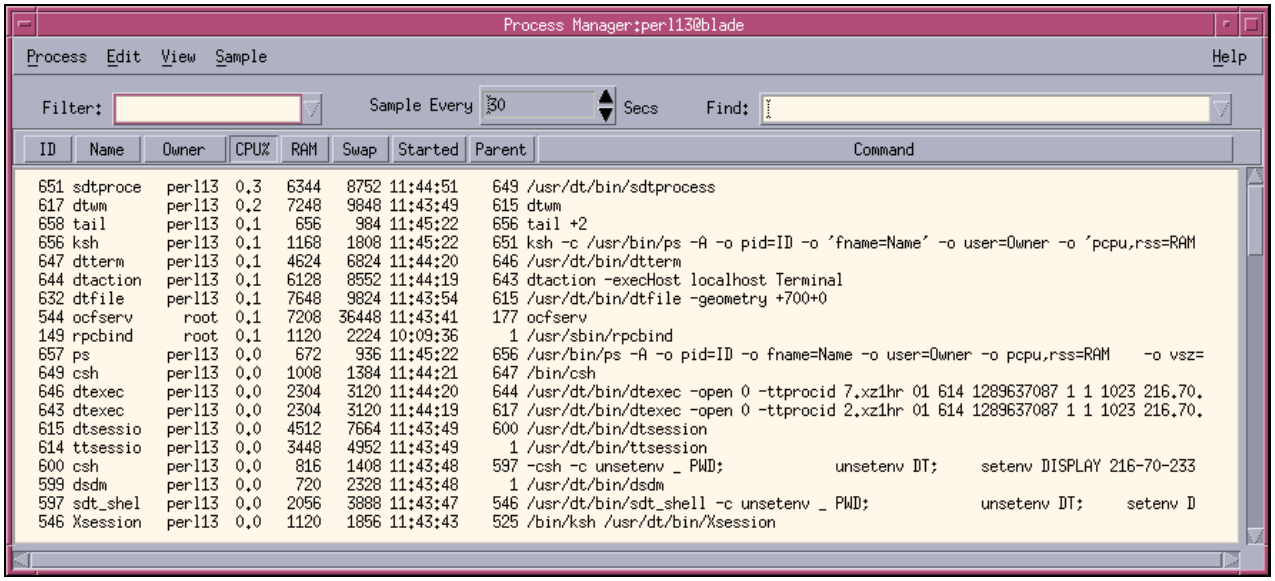
Notes:

15.2 CDE Process Manager (Solaris 8 and Higher Only)

The CDE Process Manager will allow you to list and stop processes. To start the CDE Process Manager, use the following command:

```
# /usr/dt/bin/sdtprocess
```

By default, the CDE Process Manager lists processes in order of CPU usage:



Notes:

To sort the processes

To sort the processes, click on the heading of the topic that you want to sort by.

To stop a process

To stop a process, click on the process and then click “Process + kill”.

Notes:

15.3 The prstat Command (Solaris 8 and Higher Only)

The **prstat** command will display processes that are currently running. Unlike the **ps** command, it will continue to update the display with current process information.

```
# prstat
   PID USERNAME  SIZE   RSS STATE   PRI NICE      TIME  CPU PROCESS/NLWP
   341 root     1904K 1008K sleep    10    0   0:00.01 2.3% ssld/1
   383 root       27M   68M sleep    59    0   0:00.00 1.8% Xsun/1
   277 root     2632K 1928K sleep     0    0   0:00.01 1.4% vold/6
   402 root     7544K 4720K sleep    59    0   0:00.00 0.7% dtgreet/4
    52 root     1352K  992K sleep    51    0   0:00.00 0.6% syseventd/10
{remaining output omitted}
```

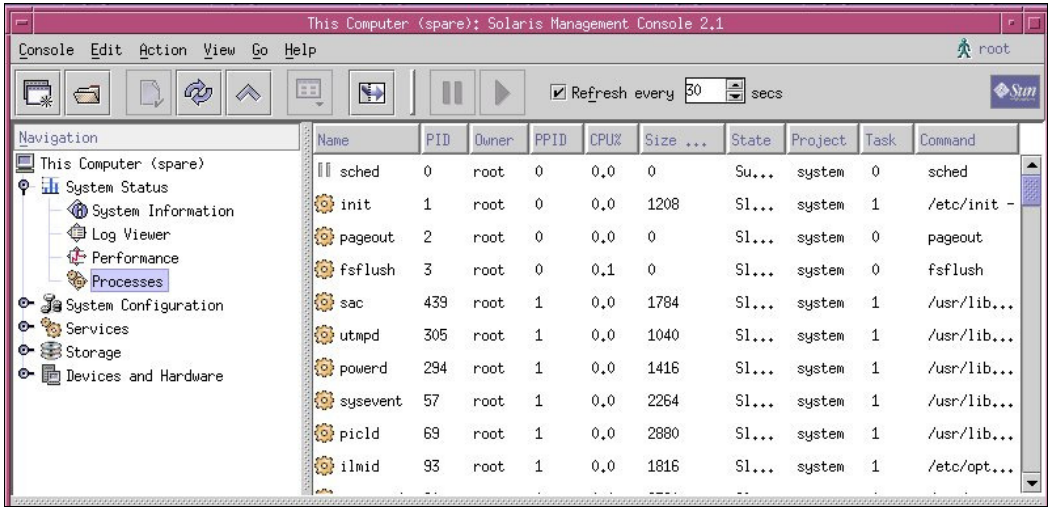
Note: To quit the **prstat** command, type the **q** key + <ENTER>.

Common **prstat** options:

Option	Purpose
-a	Separate user and system processes
-c	Print new reports below the old reports (instead of overwriting)
-n #	Specify the number of output lines
-t	Reports total usage summary for each user.

15.4 The Solaris Management Console (Solaris 9 and Higher Only)

The Solaris Management Console also provides you with a means of displaying and administering processes. After starting the SMC, expand “This computer” and “System Status” then double click “Processes”:



Note: Select a table heading (i.e. “Name” or “PID”) to sort the display.

Notes:

Notes:

Administering processes with SMC

You can perform three actions on processes with the SMC:

#1. **Pause a process** - click on the process + Action + “Suspend”.

#2. **Resume a process** - click on the process + Action + “Resume”

#3. **Stop a process** - click on the process + Edit + “Delete”

15.5 The ps and kill Commands

The **kill** command can be used to control jobs. Here are the most common **kill** options:

INT	Requests the job to cancel (like Control-c)
KILL	Forces the job to cancel
TSTP	Requests the job to freeze (like Control-z)
STOP	Forces the job to freeze
CONT	Requests the job to resume

Note: there are more options to the **kill** command. To see a complete list, run the command "**kill -l**".

A job can be stopped by using either its job or process number. The command **jobs** returns job numbers while the **ps** command returns process numbers:

```
# ps
PID        TTY   TIME CMD
 22781     pts/2 0:01  csh
```

Notes:

Notes:

By itself, the **ps** command displays all the jobs started by the shell you are currently in (including the shell itself). To see all of the processes running on the system, use the **-e** option to the **ps** command:

```
# ps -e
   PID    TTY  TIME CMD
     0    ?    0:01 sched
     1    ?    0:45 init
     2    ?    0:00 pageout
     3    ?    2:30 fsflush
   362    ?    0:00 ttymon
   109    ?    0:03 rpcbind
   359    ?    0:00 sac
   153    ?    4:05 automoun
   119    ?    0:00 kerbd
   117    ?    0:01 ypbind
   129    ?    0:09 inetd
   111    ?    0:00 keyserv
(remaining output omitted)
```

Notes:

To see more detail about the processes running, user the **-f** option:

```
# ps -f
UID        PID    PPID    C    STIME      TTY   TIME   CMD
bob        22781  22779   0    08:14:53   pts/2 0:01   -csh
```

To kill a job using its PID (process number), use the **kill** command:

```
# ps
PID          TTY   TIME CMD
 22781       pts/2 0:01  tcsh
 17332       pts/2 0:01  /usr/openwin/demo/xeyes
# kill 17332
# ps
PID          TTY   TIME CMD
 22781       pts/2 0:01  tcsh
```

Notes:

To kill a job using its job number, find the job number with the **jobs** command and use the **kill** command:

```
# /usr/openwin/demo/xeyes &
[1] 7347
# jobs
[1]  + Running                 /usr/openwin/demo/xeyes
# kill %1
#
[1]    Terminated              /usr/openwin/demo/xeyes
```

Some jobs won't die with just a **kill** command. The programmer that created the command could have "ignored" kill signals. To kill a job that won't die voluntarily, use the **-KILL** (or **-9**) option:



```
# ps
PID          TTY   TIME CMD
 22781       pts/2 0:01  tcsh
 17332       pts/2 0:01  /usr/openwin/demo/xeyes
# kill -9 17332
```

Note: The **-9** option should only be used as a last resort. Some programs ignore kill commands for important reasons. Only use the **-9** option when a normal **kill** command does not stop the job.

Notes:

15.6 The pgrep and pkill Commands (Solaris 8 and Higher Only)

The **pgrep** command allow you to specify what processes to display based upon criteria such as process owner, parent process and regular expression matches. The **pkill** command allows you to specify what processes to kill using the same criteria.

The following will list all processes own by the user bob:

```
# pgrep -u bob
383
387
416
424
470
428
505
536
473
471
488
487
490
535
```

Notes:

To see process names as well as PID numbers, use the **-l** option:

```
# pgrep -l -u perl13
  383 Xsun
  387 csh
  416 Xsession
  424 fbconsole
  470 sdt_shell
  428 speckeysd
  505 dtfile
  536 dtaction
  473 csh
  471 dsdm
  488 dtsession
  487 ttsession
  490 dtwm
  535 dtexec
  538 dtexec
  532 dtfile
  541 csh
  510 sdtvolcheck
  539 dtterm
  531 cat
  543 xeyes
```

Notes:

To list all processes of a given name, use the **-n** option:

```
# pgrep -l -n xeyes
  543 xeyes
```

The **pgrep** options can also be used with the **pkill** command:

```
# pkill -n xeyes
```

15.7 The xkill Command

The **xkill** command is good for killing processes in a GUI environment. Before running the **xkill** command, make sure that the window the holds the process that you wish to stop is in view:

```
# /usr/openwin/bin/xkill
Select the window whose client you wish to kill with button 1...
```

The next window you click on will be killed.

Notes:

15.8 Controlling Job Priority

The **nice** command allows you to control job priority. Processes can be assigned a priority from 1-19 with 1 being the highest priority. Typically, user processes all run with the same priority (10 by default).

To run a process with a priority lower than the default, specify a number higher than 10 when you run the **nice** command:

```
# nice -15 smc &
{no output}
```

To run a process with a priority higher than the default, specify a number lower than 10 when you run the **nice** command:

```
# nice -1 smc &
{no output}
```

Notes:



Notes:

15.9 Using crontab

The **crontab** is used when you wish to run a command periodically, like once a week or once a month. To set it up, you use the command "**crontab -e**" which allows you to edit a file that contains six fields. The first five of the fields specify the time the command is to run, and the sixth specifies the command itself.

Sample of a crontab file:

```
M   H   DoM   M   DoW    Command          (note: this line not shown)
15  2   1,15  *   *      /export/local/accounting/bin/print_payroll
5   *   *     2   2,4    /usr/bin/mailx bob < /export/home/bob/reminder
```

The following chart is a guide for the six fields:

Field	**Legal values**	**What * (star) means**	**Notes**
1st M (minute)	0-59	All minutes	
H (hour)	0-23	All hours	
DoM (day of month)	1-31	Not specified	Either specify a DoM or DoW, not both
2nd M (Month of year)	1-12	All months	
DoW (Day of week)	0-6	Not specified	Either specify a DoM or DoW, not both
Command	a command	Wildcard	Always use absolute pathnames.

Notes:

View crontab

To view type the crontab use the **crontab -l** command:

```
sun% crontab -l
15  2   1,15   *   *      /export/local/accounting/bin/print_payroll
5   *   *      2   2,4    /usr/bin/mailx bob < /export/home/bob/reminder
```

Explanation of the above sample output

1st line: runs the program print_payroll 15 minutes after 2 A.M. on the 1st and 15th of each month.

2nd line: mails off a reminder to Bob five minutes after every hour on every Tuesday and Thursday of the month of February.

Notes:

Edit crontab

Important note: The **crontab** looks at your EDITOR variable to determine which editor to use when you invoke **crontab**. If this is not set, **crontab** will use the **ed** editor by default. To set your EDITOR variable to the vi editor, use the following command:

```
sun% setenv EDITOR /usr/bin/vi
(no output for this command)
```

Notes about crontab

- Be careful about commands that produce output. This output should be redirected into a file; otherwise the output will be emailed to you!
- crontabs are machine specific. If you have multiple machines that you use, don't forget which machine contains the crontab. It's best to have all of your crontabs on one machine.
- To remove ALL of your crontab entries, use the command **crontab -r**

Notes:

Notes:

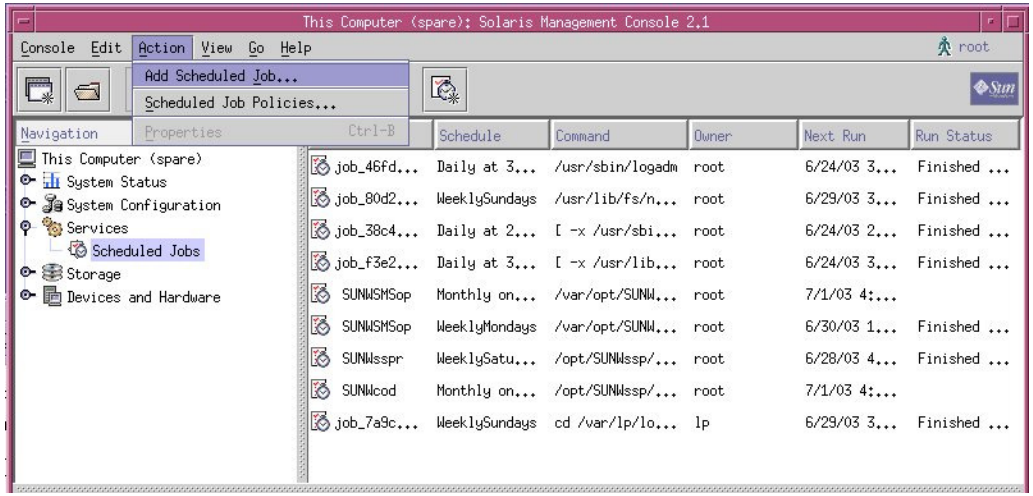
Disallowing crontabs

The System Administrator can prevent users from running crontabs by modifying either the `/etc/cron.d/cron.allow` or `/etc/cron.d/cron.deny` files. The following rules determine who can use cron:

1. If neither file exists, only the System Administrator may use cron.
2. If the user's name exists in the `/etc/cron.d/cron.allow` file, then the user can use cron.
3. If the `/etc/cron.d/cron.allow` file exists and the user's name isn't in the file, the user cannot use cron.
4. If `/etc/cron.d/cron.allow` file doesn't exist and the user's name is not in the `/etc/cron.d/cron.deny` file, then the user can use cron.
5. If the `/etc/cron.d/cron.allow` file doesn't exist and the user's name is in the `/etc/cron.d/cron.deny` file, then the user cannot use cron.

Using the Solaris Management Console to manage cron jobs

The Solaris Management Console provides an easy-to-use GUI-based tool to display and manage your crontab:



Notes:

Notes:

15.10 Using the at Command

The **at** command is used to run a command one time in the future. To use the **at** command, type **at** followed by the time and date you wish the commands to run.

The system will then display an “at” prompt at which you type in your first command. After you press RETURN, the system will once again display an "at>" prompt. When you are finished, type ^D (Control-d). The following example will e-mail a reminder to bob at 8:15 am on January 16 (note, the **echo** command "echos" back it's arguments):

```
sun% at 0815 Jan 16   (then press the RETURN key)
at>   echo "meeting today" | mailx bob
at>   ^D
```

To list the at jobs you have pending, use the **atq** command:

```
sun% atq
Rank    Execution Date       Owner    Job            Queue    Job name
1st     Jan 16, 1995 20:15   root     828333900.a    a        stdin
```

To remove an at job from the at queue (cancel the at job), use the **atrm** command:

```
sun% atrm 828333900.a
sun% atq
{no output}
```

Notes:

Running at commands in different shells

By default the **at** command will execute commands based upon your default login shell. To specify a different shell to execute at commands, use one of the following options:

Option	Meaning
-c	Run commands using the C Shell
-k	Run commands using the Korn Shell
-s	Run commands using the Bourne Shell

Sending mail confirmation

If you don't redirect the output of commands run by the **at** command, the output of these commands is mailed to you. If no output is mailed to you, then you won't receive any confirmation that the commands were executed at all.

If you want to receive an email confirmation regardless of any output, use the **-m** option:

```
sun% at -m 0815 Jan 16 (then press the RETURN key)
at> find /export/home/bob -name core -exec rm {} \;
at> ^D
```

Notes:

Specifying times with at

There is a variety of methods to enter time with the **at** command. If you specify one or two digits, the **at** command assumes you are specifying an hour to run the command. If you give four digits, the **at** command assumes you are specifying both hour and minutes.

Time Examples:

Time	Meaning
8	8 AM
8:00	8 AM
8:45	8:45 AM
1200	noon
12:00	noon
1400	2 PM
2 PM	2 PM

Key words have also been defined:

Key word	Meaning
noon	Next 12:00 PM
midnight	Next 12:00 AM
now`	Next available time

Notes:

You can also specify a date in a variety of different ways:

Date	Meaning
Jan 7	January7th
Feb 28	February 28th
Fri	Next Friday
tomorrow	Tomorrow

Lastly, you can use an increment value. You can "add to" the specified time:

at 2pm + 2 weeks

Increment options:

Option	Meaning
minute(s)	Add minutes
hour(s)	Add hours
day(s)	Add days
week(s)	Add weeks
month(s)	Add months
year(s)	Add years

Examples:

at 8:47am + 2 years
at 14:50 Mon + 5 weeks

Notes:

Disallowing the at command

The System Administrator can prevent regular users from running at commands by modifying either the `/etc/cron.d/at.allow` or `/etc/cron.d/at.deny` files. The following rules determine who has access to using at:

1. If neither file exists, only the System Administrator may use at.
2. If the user's name exists in the `/etc/cron.d/at.allow` file, then the user can use at.
3. If the `/etc/cron.d/at.allow` file exists and the user's name isn't in the file, the user cannot use at.
4. If `/etc/cron.d/at.allow` file doesn't exist and the user's name is not in the `/etc/cron.d/at.deny` file, then the user can use at.
5. If the `/etc/cron.d/at.allow` file doesn't exist and the user's name is in the `/etc/cron.d/at.deny` file, then the user cannot use at.

Note: The above rules are basically the same as **crontab**, but they deal with slightly different files.

15.11 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #8

Web sites

System Administration Guide: Advanced Administration, Chapter 15 Scheduling System Tasks (Tasks):
http://docs.sun.com/app/docs/doc/817-0403/6mg741c0n?a=view

Man pages

at	crontab	nice	prstat	smc
atq	jobs	pgrep	ps	xkill
atrm	kill	pkill	sdtprocess	

15.12 Summary of Commands and Files

Command	Description
at	Schedules processes to run once in the future
atq	Displays the user's at jobs
atrm	Removes a user's at jobs from the at queue
crontab	Schedules processes to run periodically
jobs	Lists jobs that are running in the current terminal
kill	Stops processes
nice	Executes a process with specified CPU priority
pgrep	Lists processes based on criteria provided
pkill	Stops processes based on criteria provided
prstat	Display processes that are currently running
ps	Display processes
sdtprocess	A GUI tool that allows you to manage processes
smc	A GUI administration tool
xkill	Stops a process that you click on

File	Description
/etc/cron.d/cron.allow	Allows users to use the **crontab** command
/etc/cron.d/cron.deny	Prohibits users from using the **crontab** command
/etc/cron.d/at.allow	Allows users to use the **at** command
/etc/cron.d/at.deny	Prohibits users from using the **at** command

15.13 Lab Exercises

1. Start the following job: `/usr/openwin/demo/realxfishdb`
2. Stop (do not cancel) the realfish program.

   Command: _____

3. Restart the realfish program as a background job.

   Command: _____

4. Bring the realfish program to the foreground.

   Command: _____

5. Cancel the realfish program.

   Command: _____

6. Start the following job: `/usr/openwin/demo/maze`

   Command: _____

7. Quit the command window you are working in.

8. Open a new command tool.

9. What is the PID of the maze process?

   Command: _____

10. Who has the right to kill the maze process?

    Command: _____

11. Kill the maze process.

    Command: _____

Scenario #1: The user “bob” is having a hard time creating a crontab entry. He wants to run the du command on his home directory every hour Monday-Friday from 8 AM to 5 PM and place the results into a file called “du_results.txt” in his home directory. Create this crontab entry for bob so that it will run under his account.

Scenario #2: You are concerned about disk usage for the `/var` filesystem. Create a shell script that will email the root account a warning if the /var filesystem is 85% full (or higher). Have this script run as part of the system’s crontab once per day.



Unit Sixteen
Backup and Recovery

Unit topics

		Page #
16.1	Why Perform Backups?	390
16.2	Full and Incremental Backups	391
16.3	Logical Tape Device Names	392
16.4	The ufsdump Command	394
16.5	Backing up the / and /usr Filesystems	396
16.6	The ufsrestore Command	397
16.7	The restoresymtable File	399
16.8	Restoring the / Filesystem	400
16.9	The mt Command	401
16.10	The tar Command	402
16.11	Compressing Large Files: compress	407
16.12	Compressing Large Files: gzip	412
16.13	Compressing Large Files: zip	415
16.14	Encoding Binary Files	419
16.15	Emailing Files	421
16.16	UFS Snapshots	422
16.17	Additional Resources	426
16.18	Summary of Commands and Files	428
16.19	Lab Exercises	429

16.1 Why Perform Backups?

The following are the typical reasons why administrators perform backups:

1. User error
2. Software error
3. User error
4. Hardware error
5. User error

Ok, you get the picture. File loss by user, software or hardware error can cost time and money.

Notes:

16.2 Full and Incremental Backups

There are many different backup commands that are available for Solaris. This Unit will focus on the **ufsdump** command because it is available by default on the system.

Why ufsdump?

ufsdump isn't the easiest command, nor does it have the most capabilities. The primary reason to know **ufsdump** is that it is ALWAYS available (making restores much easier)

Types of backups

When you backup a filesystem, you have a choice between two types of backups (dumps) available with ufsdump: full and incremental.

Full dumps backup everything in the filesystem. Full dumps are performed by specifying dump level "0" with the ufsdump command.

Incremental dumps back up only what has been changed since the last "lower" dump level. These dump levels are arranged in priority from 1-9.

Note: if you backup directories or files, you only have the choice of full dumps. Only when you back up a filesystem can you perform an incremental dump.

Notes:

16.3 Logical Tape Device Names

Tape devices, like other devices, can be accessed through the `/dev` directory. When the system performs a reconfigure boot, it probes the SCSI chain for tape devices. For each tape device found, the system creates file in the `/dev/rmt` directory.

```
# ls /dev/rmt
0h    0hn   0hb   0hbn
{Remaining output omitted}
```

Tape number
The first tape device on the SCSI chain is tape 0. The second tape device on the SCSI chain is tape 1.

Tape density
l=low
m=medium
h=high (default if not specified)
u=ultra
c=compressed

BSD behavior
BSD tape devices act a little differently when reading past an end-of-file mark and when using the no-rewind option. Not using "b" in the file name will use the AT&T tape behavior.

Notes:

Notes:

No rewind

By default, the system will rewind to the beginning of the tape. When backing up two volumes, use the no rewind option (n) for the first backup.

Tape name examples

`/dev/rmt/0un` Tape 0, ultra compressed, no rewind, AT&T tape behavior
`/dev/rmt/1cb` Tape 1, compressed, BSD tape behavior, rewind

Notes:

16.4 The ufsdump Command

The **ufsdump** command will back up filesystems, directories and files.

Common options of the ufsdump command:

0-9 The dump level (0=full, 1-9=incremental)
u Update the `/etc/dumpdates` file. This file is used by the system to determine when the last lower update occurred.
f Where to dump the data to. Typically this will be `/dev/rmt/0` (or some other `/dev/rmt` device)
S Don't perform the backup, just provide a size estimate.
n Notify sys group members when problems occur (via terminal window)
o Take tape drive offline, rewind the tape and eject it.

The `/etc/dumpdates` file

When you back up entire partitions and use the u option, the `/etc/dumpdates` file will keep track of when these backups occurred:

```
# more /etc/dumpdates
/dev/rdsk/c0t3d0s7      0 Thu Nov 15 10:01:51 2001
/dev/rdsk/c0t3d0s7      3 Thu Nov 22 10:02:24 2001
```

Notes:

Example of ufsdump

```
# ufsdump 0f /dev/rmt/0 /etc
DUMP: Date of this level 0 dump: Sat Feb 6 14:32:10 1999
DUMP: Date of last level 0 dump: the epoch
DUMP: Dumping /etc
DUMP: mapping (Pass I) [regular files]
DUMP: mapping {Pass II} [directories]
DUMP: estimated 254102 blocks (124.07MB) on 0.07 tape(s)
DUMP: dumping (Pass III) [directories]
DUMP: dumping {Pass IV) [regular files]
DUMP: 41.11% done, finished in 0:07
DUMP: 81.87% done, finished in 0:02
DUMP: DUMP: 254102 blocks (124.07MB) on 1 tape
DUMP: DUMP IS DONE
DUMP: level 0 dump on Sat Aug 2 04:56:03 1997
DUMP: Tape rewinding
```

Remote backups

You can back up a filesystem to a remote tape device with the ufsdump command. First set up a machine for remote backup by placing its hostname (or IP address) in the `/.rhosts` file on the tape back server. Then, use the following syntax on the machine that contains the filesystem to be backed up:

```
ufsdump options server_name:/tapedevice file_system
```

Notes:

16.5 Backing up the / and /usr Filesystems

The **ufsdump** utility performs four "passes" when it backups a filesystem:

1. Backups the directory inode tables
2. Backups the file inode tables
3. Backups the file data blocks
4. Backups the directory data blocks

This can cause problems when someone changes a file between its inode being backed-up and its data blocks being backed-up. This will create file corruption (which can also corrupt the entire backup).

To properly backup a filesystem, you must first make sure that no users (or system daemons) are writing to that filesystem. For some filesystem, such as /export/home, this can be accomplished by just unmounting the filesystem.

The / and /usr filesystems can not be unmounted after the system has been booted. Therefore, you must boot off the CD-ROM to properly back up the / and /usr filesystems:

```
# reboot -- "cdrom -s"
{system will reboot off CD-ROM}
# ufsdump 0f /dev/rmt/0 /dev/dsk/c0t3d0s0
{output omitted}
```

Notes:

16.6 The ufsrestore Command

To restore files backed up by **ufsdump**, use the **ufsrestore** command. The **ufsrestore** command has the following options:

i	Interactive restore
r	Complete restore
t	List a table of contents
x	Restore file listed on the command line
f	Used to specify the device the backup is to be restored from
v	Restore in verbose mode

Listing a Table of Contents

To list the table of contents, use the **t** option

```
# ufsrestore tvf /dev/rmt/0
{output omitted}
```

Restore an entire tape volume

To an entire tape volume, use the **r** option

```
# ufsrestore rvt /dev/rmt/0
{output omitted}
```

Notes:

Restoring an individual file

To restore an individual file, use the **x** option

```
# ufsrestore xvf /dev/rmt/0 ./etc/hosts
{output omitted}
```

Interactive restore

You can perform an interactive restore which will allow you to pick which files to restore via a "UNIX-like" interface. The **i** option performs an interactive restore:



```
# ufsrestore ivf /dev/rmt/0
ufsrestore>
```

Once in the interactive restore interface, you can run the following commands:

Command	Outcome
ls	List files in a directory
cd	cd to a tape "directory"
add	Add a file to the extract list (files on extract list will have a "*" next to them
delete	Remove a file from the extract list
extract	Extract files on the extract list

16.7 The restoresymtable File

The **r** option to the **ufsrestore** command "leaves behind" a file when restoring an incremental backup. This file is left behind in order to pass checkpoint information to subsequent restores.

Once all restores have been completed, this file can be removed. It is placed in the top-most part of the filesystem and is named "restoresymtable".

Notes:

16.8 Restoring the / Filesystem

In order to restore the / or /usr filesystems, you must boot off of the CD-ROM.

To prevent any file corruption, a new filesystem should be placed on the partition prior to performing the restore operation.

To restore the / partition, take the following steps (note: all output omitted):

```
ok boot cdrom -s
# newfs /dev/rdsk/c0t3d0s0
# mount /dev/dsk/c0t3d0s0 /a
# cd /a
# ufsrestore rvf /dev/rmt/0
# fsck /dev/dsk/c0t3d0s0
# cd /usr/platform/`uname -i`/lib/fs/ufs
# installboot bootblk /dev/rdsk/c0t3d0s0
# reboot
```

The **installboot** command places the ufsboot program on the partition. This is only needed for the / partition.

Notes:

16.9 The mt Command

The **mt** command provides a means to directly manipulate the tape drive. The most operations of the **mt** command:

Option:	Meaning:
status	Display information about the tape drive
rewind	Rewinds the tape
erase	Erases the tape
fsf	Skip forward tape files
eom	Skip to the end of recorded media

Examples

```
mt -f /dev/rmt/0 status
mt -f /dev/rmt/0 rewind
mt -f /dev/rmt/0 erase
mt -f /dev/rmt/0n fsf 1
```

Note: don't forget the "n" part of the tape device name when skipping forward tape files or the tape will rewind when the action is completed.

Notes:

16.10 The tar Command

The **tar** command will combine files together. It has some inherent advantages and disadvantages when compared to the **ufsdump** command.

Advantages of tar

- It is quick and simple to use
- It is available on almost every UNIX OS (**ufsdump** is primarily a Solaris command)
- It is the most common method of preparing multiple files for e-transport (email, ftp, floppy disks, CD-ROMS)

Disadvantages of tar

- It doesn't provide the "dump levels" that **ufsdump** provides. Everything is a "full backup"
- It doesn't work as well for full filesystems as **ufsdump** does.

Notes:

Notes:

Creating tar files

Use the **-c** option to create tar files:

```
# tar -cvf skel.tar /etc/skel
a /etc/skel/ 0K
a /etc/skel/.profile 1K
a /etc/skel/local.cshrc 1K
a /etc/skel/local.login 1K
a /etc/skel/local.profile 1K
# ls
skel.tar
```

Option	Meaning
-c	Create a tar file
-v	Verbose mode. With the **-c** option this displays how large in kilobytes each file is.
-f	Specify the tar file's name

Notes:

Listing table of contents

List the table of contents of an existing tar file with the **-t** option:

```
# tar -tvf skel.tar
tar: blocksize = 13
drwxr-xr-x  0/3       0 Mar 29 20:09 1999 /etc/skel/
-rw-r--r--  0/1     144 Mar 29 20:08 1999 /etc/skel/.profile
-rw-r--r--  0/3     124 Mar 29 20:08 1999 /etc/skel/local.cshrc
-rw-r--r--  0/3     581 Mar 29 20:09 1999 /etc/skel/local.login
-rw-r--r--  0/3     562 Mar 29 20:09 1999 /etc/skel/local.profile
```

Option	Meaning
-t	List a tar file's table of contents
-v	Verbose mode. With the **-t** option this displays details of each file within the tar file
-f	Specify the tar file's name

Notes:

Extracting tar files

Use the **-x** option to extract files from a tar file:

```
# tar -xvf skel.tar
tar: blocksize = 13
x /etc/skel, 0 bytes, 0 tape blocks
x /etc/skel/.profile, 144 bytes, 1 tape blocks
x /etc/skel/local.cshrc, 124 bytes, 1 tape blocks
x /etc/skel/local.login, 581 bytes, 2 tape blocks
x /etc/skel/local.profile, 562 bytes, 2 tape blocks
```

Option	Meaning
-x	Extract files from a tar file
-v	Verbose mode. With the **-x** option this displays how large in bytes and tape blocks each extracted file is.
-f	Specify the tar file's name

Notes:

Is it a tar file?

While it is considered good "tar etiquette" to name tar files with the ".tar" extension, this is not enforced by the command. Most users do use this convention; however, some completely ignore it.

When in doubt about a file's contents, use the **file** command to determine what kind of file it is:

```
# file skel.tar
skel.tar:       USTAR tar archive
```

Absolute vs. Relative pathnames

If files are "tarred" using relative pathnames, then when someone extracts them, the files will be extracted relative to the current directory. If files are "tarred" using absolute pathnames, then when someone extracts them, the files will be extracted back into their original locations.



Typically a good rule of thumb is that when root backs up system files, absolute pathnames should be used. In all other cases (root backs up other files, users back up files), a relative pathname should be used.

Security note: Always be wary of "tarred" files you download from the internet (or from any potentially unreliable source). It is a good practice to look at the tar file for suspicious files prior to extracting them.

Notes:

16.11 Compressing Large Files: compress

The **compress** command is used to make large files smaller. The primary method that this command uses is to find redundancy of strings and "translate" these strings into a binary character. As a result, typically large text files compress better than smaller text files or binary files.

The result of the compress command is a binary file that *replaces* the original file:

```
# ls
words
# compress words
# ls
words.Z
```

Notes:

Common compress Options

The **-v** option will display how much the file is compressed by:

```
# compress -v words
words: Compression: 48.76% -- replaced with words.Z
```

The **-c** option will write the output to STDOUT and not overwrite the existing file. This will allow you to create a compressed file without losing the original:

```
# ls
words
# compress -c words > words.Z
# ls
words    words.Z
```

Notes:

Viewing compressed files

To view the contents of a compress file, use the **zcat** command:

```
# zcat words.Z | head -10
10th
1st
2nd
3rd
4th
5th
6th
7th
8th
9th
```

The **zcat** command will uncompress the file, but the uncompressed file is only stored in memory. No changes are made to the original file.

Notes:

Uncompressing files

Use the **uncompress** command to uncompress a file:

```
# ls
words.Z
# uncompress words.Z
# ls
words
```

Notes about the compress command

- The man pages state "If the appending the .Z to the files name would make the name exceed 14 bytes, the command will fail." While this is true on some systems, it does not apply to all.
- The **compress** command uses the Lempel-Ziv coding method.
- Typically files are reduced by 50-60% using the **compress** command
- There is an older compression command available called **pack**. The **pack** command typically has less effective compression and normally takes longer than **compress**. Files that have been compressed with the **pack** command have a ".z" extension.
- The **uncompress** command will **fail** if the file name doesn't end with ".Z".

Notes:

16.12 Compressing Large Files: gzip

The **gzip** command can be used to compress files. The method that **gzip** uses to compress the files is called PKZIP, which is the same method that many other compression utilities use (such as Winzip on PC's).

Create a gzip file

```
# cd /tmp
# cp /usr/dict/words .
# ls -l words
-r--r--r--  1 root     other     206662 Jul 10 10:01 words
# gzip words
# ls -l words.gz
-r--r--r--  1 root     other      81477 Jul 10 10:01 words.gz
```

Important note: The resulting gzip file will replace the original file by default.

Notes:

Notes:

Viewing compression ratio

To see how much the file was compressed by, use the **-v** option to the **gzip** command:

```
# gzip -v words
words:                60.5% -- replaced with words.gz
```

If the file is already compressed, use the **-l** option to the **gzip** command to view compression ratio information:

```
# gzip -l words.gz
         compressed        uncompressed  ratio uncompressed_name
              81477              206662  60.5% words
```

Avoiding replacing original file

You can use the **-c** option to avoid replacing the original file with the compressed file. By default, the **-c** option will send output to STDOUT (standard output), so you need to redirect this output back into a file:

```
# gzip -c words > words.gz
# ls -l words*
-r--r--r--   1 root     other     206662 Jul 10 10:02 words
-rw-r--r--   1 root     other      81477 Jul 10 10:04 words.gz
```

Notes:

Compressing multiple files

You can have the **gzip** command compress files within directory structures by using the **-r** option:

```
# cp -r /etc/skel .
# ls skel
local.cshrc    local.login    local.profile
# gzip -r skel
# ls skel
local.cshrc.gz    local.login.gz    local.profile.gz
```

Extracting gzip files

To unzip a gzip file, use the **gunzip** command:

```
# ls -l words.gz
-rw-r--r--   1 root     other      81477 Jul 10 10:04 words.gz
# gunzip words.gz
# ls -l words
-rw-r--r--   1 root     other     206662 Jul 10 10:04 words
```

Notes:

16.13 Compressing Large Files: zip

The **zip** utility performs the same function as the **gzip** command in that it can compress files. In addition it can also merge multiple files into a single, compressed file.

Create a zip file

```
# cd /tmp
# zip skel /etc/skel/*
  adding: etc/skel/local.cshrc (deflated 12%)
  adding: etc/skel/local.login (deflated 11%)
  adding: etc/skel/local.profile (deflated 12%)
```

Creating zip directories

If you need to create a zip file that contains a directory structure, use the **-r** option:

```
# zip -r skel /etc/skel
  adding: etc/skel/ (stored 0%)
  adding: etc/skel/.profile (deflated 13%)
  adding: etc/skel/local.cshrc (deflated 12%)
  adding: etc/skel/local.login (deflated 11%)
  adding: etc/skel/local.profile (deflated 12%)
```

Additional common zip options

Option	Purpose
-c	Add a comment for each archived file
-j	Don't Include path with filename
-u	Update (replace) files in an existing zip file
-v	Verbose mode

Notes:

Notes:

Listing table of contents

List the tables of content of an existing zip file with the **unzip** command:

```
# unzip -l skel.zip
Length      Date    Time    Name
 ------     ----    ----    ----
     0   02-03-03   17:54   etc/skel/
   144   02-03-03   17:54   etc/skel/.profile
   136   02-03-03   17:54   etc/skel/local.cshrc
   157   02-03-03   17:54   etc/skel/local.login
   174   02-03-03   17:54   etc/skel/local.profile
  ------                    -------
   611                      5 files
```

To see additional information, use the **-v** option:

```
# unzip -v skel.zip
Length   Method   Size  Ratio   Date     Time   CRC-32    Name
 ------  ------  ------ -----   ----     ----   ------    ----
     0   Stored       0   0%   02-03-03 17:54  00000000  etc/skel/
   144   Defl:N     125  13%   02-03-03 17:54  13954626  etc/skel/.profile
   136   Defl:N     120  12%   02-03-03 17:54  65b068c3  etc/skel/local.cshrc
   157   Defl:N     140  11%   02-03-03 17:54  6155fc54  etc/skel/local.login
   174   Defl:N     153  12%   02-03-03 17:54  534a1f26  etc/skel/local.profile
 ------          ------ ---                              -------
   611             538  12%                              5 files
```

Notes:

Extracting zip files

Use the **unzip** command to extract files from a zip file:

```
# unzip skel.zip
Archive:  skel.zip
   creating: etc/skel/
  inflating: etc/skel/.profile
  inflating: etc/skel/local.cshrc
  inflating: etc/skel/local.login
  inflating: etc/skel/local.profile
```

16.14 Encoding Binary Files

The result of the **compress** command is a binary file. While this is fine for transporting files via ftp, it isn't good for email. Sending binary files via email will cause file corruption.

There are several methods of preparing a file for email transport, including MIME, Bin Hex and uuencode. The most common method used on UNIX systems is uuencode.

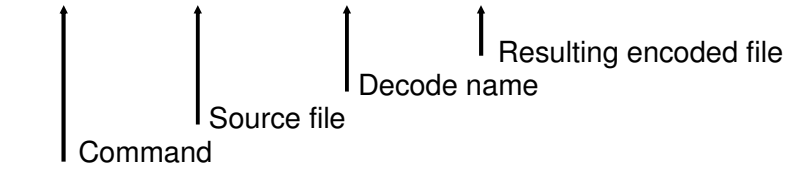
The **uuencode** command will make a binary file into a text file, allowing it to be sent via email:

```
# ls
words.Z
# uuencode words.Z words.Z > words.Z.uu
# ls
words.Z     words.Z.uu
```

Notes:

Notes:

The **uuencode** command syntax is a bit odd:



The file words.Z will be encoded and the encoded file will be called words.Z.uu. When a user decodes word.Z.uu, the resulting file will be called word.Z (the decode name). Most people use the same name for the "Source file" and the "Decode name", but they can be different.

To decode a file

Use the **uudecode** command to decode a file:

```
# ls
words.Z.uu
# uudecode words.Z.uu
# ls
words.Z     words.Z.uu
```

16.15 Emailing Files

To email files to someone, take the following steps:

1. tar the files
2. compress the file
3. uuencode the file

When the user receives the file, the following steps will extract the files:

1. uudecode the file
2. uncompress the file
3. "un-tar" the file

Notes:

16.16 UFS Snapshots

One problem that arises often when administrators use the **ufsdump** command is when changes occur on the filesystem while it is being backed up. This can result in a corrupted filesystem.

A UFS snapshot can be used to overcome this problem. You can create a snapshot while a filesystem is mounted without having to worry about losing data.

Creating a snapshot

You can use the **fssnap** command to create a snapshot:

```
# fssnap -F ufs -o bs=/opt/backups /var
/dev/fssnap/0
```

The **bs=** option is used to store a file called "backing-store". This file contains the filesystem information (but not the file data itself). This file can not be stored in the filesystem that you are backing up the data in.

The **fssnap** command will also create two virtual device files: `/dev/fssnap/0` and `/dev/rfssnap/0`. These device files are used to access the data of the filesystem that you are backing up. You can use these files to create an ufsdump backup.

Note: if you were to run **fssnap** again, the virtual device files would be `/dev/fssnap/1` and `/dev/rfssnap/1`.

Notes:

Notes:

Displaying snapshot information

To display a list of existing snapshots, use the **-i** option to the **fssnap** command:

```
# fssnap -i
  0    /var
```

To see detailed information about a snapshot, use the **-i** option to the **/usr/lib/fs/ufs/fssnap** command and specify the filesystem:

```
# /usr/lib/fs/ufs/fssnap -i /var
Snapshot number              : 0
Block Device                 : /dev/fssnap/0
Raw Device                   : /dev/rfssnap/0
Mount point                  : /var
Device state                 : idle
Backing store path           : /opt/backups
Backing store size           : 224 KB
Maximum backing store size   : Unlimited
Snapshot create time         : Thu Jul 10 10:25:06 2003
Copy-on-write granularity    : 32 KB
```

Notes:

Using the snapshot to backup the filesystem

You can use either the **tar** command or the **ufsdump** command to back up a filesystem using the snapshot:

tar example

#1. Mount the virtual device:

```
# mkdir /backup
# mount /dev/fssnap/0 /backup
```

#2. Run the **tar** command on the mount point:

```
# cd /backup
# tar -cvf /dev/rmt/0 .
{output omitted}
```

ufsdump example

When you use the **ufsdum**p command to back up a snapshot, just specify the virtual device file:

```
# ufsdump 0uf /dev/rmt/0 /dev/rfssnap/0
{output omitted}
```

Important notes about snap shots

- If you unmount a filesystem, the snapshot will be removed.
- If you reboot the system, all snapshots will be removed.
- You can limit the size of the backing-store by using the **-o maxsize=** option to the **fssnap** command.
- Users may notice slight delays while the snapshot is created.

Notes:

16.17 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #17

Web sites

System Administration Guide: Devices and File Systems, Chapter 23 UFS File System (Reference): http://docs.sun.com/app/docs/doc/817-5093/6mkisoq8u?a=view

System Administration Guide: Devices and File Systems, Chapter 24 Backing Up and Restoring File Systems (Overview): http://docs.sun.com/app/docs/doc/817-5093/6mkisoq92?a=view

System Administration Guide: Devices and File Systems, Chapter 25 Backing Up Files and File Systems (Tasks): http://docs.sun.com/app/docs/doc/817-5093/6mkisoq97?a=view

System Administration Guide: Devices and File Systems, Chapter 26 Using UFS Snapshots (Tasks): http://docs.sun.com/app/docs/doc/817-5093/6mkisoq9h?a=view

System Administration Guide: Devices and File Systems, Chapter 27 Restoring Files and File Systems (Tasks): http://docs.sun.com/app/docs/doc/817-5093/6mkisoqa5?a=view

System Administration Guide: Advanced Administration, Chapter 28 UFS Backup and Restore Commands (Reference): http://docs.sun.com/app/docs/doc/817-5093/6mkisoqai?a=view

Man pages

compress	gzip	tar	uncompress	uuencode
fssnap	mt	ufsdump	unzip	zcat
gunzip	pack	ufsrestore	uudecode	zip

16.18 Summary of Commands and Files

Command	Description
compress	File compression utility
fssnap	Freezes a filesystem in order to perform a backup
gunzip	Uncompresses files compressed by the **gzip** command
gzip	File compression utility
mt	Manipulates the tape drive
pack	Old file compression utility
tar	Creates backups of filesystems
ufsdump	Creates backups of filesystems
ufsrestore	Restores backups of ufsdumps
uncompress	Uncompresses files compressed by the **compress** command
unzip	Uncompresses files compressed by the **zip** command
uudecode	Converts uuencoded files back into their original format
uuencode	Converts binary files into text files for email transport
zcat	Uncompresses files compressed by the **compress** command and send output to terminal
zip	File compression utility

File	Description
/dev/rmt/0	Device file for tape devices
/etc/dumpdates	File that **ufsdump** uses to keep track of backups
/.rhosts	File used to permit remote backups (and other remote commands)
restoresymtable	File created by **ufsdump** command to keep track of restore operations

16.19 Lab Exercises

Scenario #1 - You just installed the OS and have decided to backup the `/opt` partition. Perform these actions using the **ufsdump** command.

Scenario #2 - The `/opt` partition has been corrupted beyond repair. Restore the partition using the backup you created in scenario #1.

Scenario #3 - You need to send a copy of the `/usr/bin` directory to another system administrator. Using relative pathnames, **tar**, **compress** and **uuencode** the directory. Test the process by mailing the resulting file to yourself. After you receive the file, decode, uncompress and untar it in the `/tmp` directory.

Unit Seventeen
Networking

Unit topics

		Page #
17.1	Setting up Remote Access Authentication	431
17.2	Remote Login	435
17.3	Remote Shell	436
17.4	Remote Copy	437
17.5	rlogin vs. telnet	438
17.6	Restricting telnet	439
17.7	Review: /etc/default/login	441
17.8	rcp vs. FTP	442
17.9	Restricting FTP	443
17.10	Disabling rlogin, rsh and rcp	446
17.11	Displaying Remote Users	448
17.12	Checking the Status of Machines	449
17.13	Additional Resources	452
17.14	Summary of Commands and Files	453
17.15	Lab Exercises	454

© 2014 One Course Source

17.1 Setting up Remote Access Authentication

Remote Access Authentication is the process of allowing users on one host to perform actions on another host without having to provide a password. Users can use the following commands on a remote system if Remote Access Authentication has been set up on the remote system:

rlogin - Allows users to log in without providing a password
rsh - Allows users to run commands on remote system
rcp - Allows users to copy files to and from the remote system

This process can either be set up "system wide" by the superuser or on an individual basis by each user.

Notes:

Commented [KW3]: dladm show-link
```
echo 'solaris' > /etc/hostname.e1000g1
echo 'wait 30' > /etc/dhcp.e1000g1

inetadm -e svc:/network/login:rlogin
inetadm -e svc:/network/shell:default
```

svcadm enable svc:/network/login:rlogin

© 2014 One Course Source

Notes:

Set up Remote Access Authentication as the superuser

To set up Remote Access Authentication, edit the `/etc/hosts.equiv` file and list the machines (and users) that will be granted access.

The format of this file is:

```
[-]hostname|+  [-][username|+]
```

Some examples:

```
# more /etc/hosts.equiv
sun bob      #grants bob access from the sun machine
moon         #grants all users access from the moon machine
+ tim        #grants tim access from any machine
+ -steve     #denies steve access from any machine
-stars       #denies all users from the stars machine access
```

Note: Placing an entry in this file will not allow root users from other machines access in any situation. To allow Remote Access Authentication for root, the file `/.rhosts` must be edited (see next page).

© 2014 One Course Source

Notes:

Set up Remote Access Authentication as a non-root user

Users can permit themselves (only) access from other machines by placing a `.rhosts` file in their home directory. This file should only contain "trusted" host names and will only affect the user (not anyone else on the system).

Example:

```
# more /export/home/bob/.rhosts
stars
sun
```

The above file will allow bob to have Remote Access Authentication from the stars and sun machines to the local machine.

© 2014 One Course Source

Notes:

Security Risks to Remote Access Authentication

Since Remote Access Authentication allows users to access the host without providing a password, there are some inherent security risks. The most common security hole exploit is through a process called "IP spoofing".

Machine names specified in the `/etc/hosts.equiv` and `.rhosts` files are "translated" into IP addresses by the system. If another machine on the network were to be given the same IP address as the machine that was granted Remote Access Authentication trust, this new machine would gain unauthorized access to the system.

Another exploit is "user spoofing". Consider the following file:

```
# more /etc/hosts.equiv
+ bob
```

In this case, the local system has granted the user bob trust to run the commands **rlogin**, **rcp** and **rsh**. If on another machine on the network a user "bob" was created, that user would now have access to this system.

© 2014 One Course Source

17.2 Remote Login

Notes:

The **rlogin** command will allow a user to log into a remote host without giving a password (provided Remote Access Authentication has been set up):

```
sun% rlogin moon
Last login: Wed Jun 16 11:31:57 from localhost
Sun Microsystems Inc.   SunOS 5.5       Generic October 1998
```

If Remote Access Authentication has not been set up on the machine the user is trying to log into, the user will be asked for a password:

```
sun% rlogin moon
Password:
```

Commented [KW4]: If the rlogin service is not enabled (connection refused error) then enable it with:

inetadm -e svc:/network/login:rlogin

© 2014 One Course Source

17.3 Remote Shell

The **rsh** command allows a user to run a command remotely on another system. The output of the command will be displayed on the local machine. Once again, Remote Access Authentication must be set up on the remote host:

```
sun% rsh moon uname -a
SunOS moon 5.5 Generic sun4m sparc SUNW,SPARCstation-10
```

If Remote Access Authentication is not set up on the remote machine, **rsh** will not work:

```
sun% rsh moon uname -a
permission denied
```

Notes:

© 2014 One Course Source

17.4 Remote Copy

The **rcp** command will allow users to copy files to and from remote hosts. Remote Access Authentication must be set up on the remote machine for **rcp** to work:

```
sun% rcp sun:/etc/inet/hosts /tmp/hosts
{no output for this command}
```

If Remote Access Authentication is not set up on the remote machine, **rcp** will not work:

```
sun% rcp sun:/etc/inet/hosts /tmp/hosts
permission denied
```

Notes:

© 2014 One Course Source

17.5 rlogin vs. telnet

The **rlogin** and **telnet** commands are similar in that they allow users to log into another host across the network. Each command has advantages and disadvantages:

Notes:

rlogin

Advantages:

- It is easier for the user to use
- The password is not sent across the network.

Disadvantage:

- The remote system is vulnerable to IP spoofing (Remote Access Authentication is vulnerable to IP spoofing).

telnet

Advantages:

- Works on all UNIX (and PC) systems, so more users are familiar with it.
- Not vulnerable to IP spoofing.

Disadvantages:

- Not as user friendly as **rlogin**
- Sends the user login information (login name and password) across network in plain text. This information can be seen by a "sniffer" (such as the **snoop** command).

17.6 Restricting telnet

As you will see in a later section, you have some flexibility when restricting ftp. When attempting to restrict telnet sessions, it's more of an "all or nothing" situation.

You can disable all **telnet** sessions (both outgoing and incoming) by editing the `/etc/services` file and commenting out the line that defines the **telnet** service:

```
#telnet         23/tcp
```

Another method you can use is to comment out the line that describes telnet sessions in the `/etc/inetd.conf` file:

```
#telnet  stream  tcp     nowait  root    /usr/sbin/in.telnetd    in.telnetd
```

Note: With this second method you need to reboot the machine or restart the **inetd** daemon before the changes take effect.

There are additional methods, such a configuring PAM or a firewall. These methods are beyond the scope of this class.

If the system is Solaris 10 or later, then the Service Management Facility(SMF) can be used to enable or disable telnet.

Notes:

© 2014 One Course Source

To enable telnet, as it is disabled by default, the administrator can execute:

```
# inetadm -e svc:/network/telnet:default
```

To disable telnet, if it is enabled, the administrator can execute:

```
# inetadm -d svc:/network/telnet:default
```

To view the status of the telnet service:

```
# inetadm | grep telnet
disabled  disabled      svc:/network/telnet:default
```

Notes:

© 2014 One Course Source

17.7 Review: `/etc/default/login`

Don't forget about the CONSOLE variable in the `/etc/default/login` file:

Variable	Meaning
CONSOLE	**CONSOLE=/dev/console** If this is set to `/dev/console`, then root can only log in from the console (no telnet). **#CONSOLE=/dev/console** If not set (or commented out), root can log in from anywhere. **CONSOLE=** If set to nothing, root CAN NOT log in at all (but you can **su** to the root account).

This variable will affect login whenever you attempt to **telnet** or **rlogin** into the machine using the root account.

Notes:

© 2014 One Course Source

Notes:

17.8 rcp vs. ftp

The **rcp** and **ftp** commands are similar in that they both allow users to copy files to and from remote systems. Each has potential advantages and disadvantages that must be considered:

ftp

Advantages:

- Works on all UNIX (and PC) systems so more users are familiar with it
- Not vulnerable to IP spoofing
- More powerful (more options) than **rcp**

Disadvantages:

- More difficult to use than **rcp**
- Sends password information across the network in plain test, allowing a sniffer to "steal" it

rcp

Advantages:

- Easier than **ftp**
- No password information sent across the network

Disadvantages:

- Vulnerable to IP spoofing
- Not as powerful as **ftp**

© 2014 One Course Source

17.9 Restricting FTP

There are several different methods to restricting ftp sessions:

Using the `/etc/ftpusers` file

You can prohibit a specific user from using ftp by placing their name in this file:

```
# more /etc/ftpusers
root
daemon
bin
sys
adm
lp
uucp
nuucp
listen
nobody
noaccess
nobody4
bob
```

Notes:

© 2014 One Course Source

Notes:

Using the `/etc/shells` file

The `/etc/shells` file is used to specify valid user shells on the system. Many applications (such as **telnet** and **rlogin**) don't use this file, however **ftp** does.

By default, this file does not exist. If you create this file and place shell's paths in it then only users who have a login shell that matches those listed in this file will be allowed to use **ftp**:

```
# more /etc/shells
/bin/sh
/bin/korn
```


© 2014 One Course Source

Notes:

Disabling ftp sessions

You can disable all **ftp** sessions (both outgoing and incoming) by editing the `/etc/services` file and commenting out the line that defines the **ftp** service:

```
#ftp          21/tcp
```

Another method you can use is to comment out the line that describes ftp sessions in the `/etc/inetd.conf` file:

```
# ftp     stream  tcp     nowait  root    /usr/sbin/in.ftpd       in.ftpd
```

Note: With this second method you need to reboot the machine or restart the inetd daemon before the changes take effect.

With SMF, the ftp service can be enabled by the administrator executing:

```
# inetadm -e svc:/network/ftp:default
```

Using SMF, the ftp service can be disabled by the administrator executing:

```
# inetadm -d svc:/network/ftp:default
```


© 2014 One Course Source

Notes:

17.10 Disabling rlogin, rsh and rcp

Since users can create their own `.rhosts` file, disabling **rlogin**, **rsh** and **rcp** can be tricky. There are basically two methods you can use:

To disable a specific user from using rlogin, rsh and rcp

Take the following steps to disable a specific user's **rlogin**, **rsh** and **rcp**:

#1. Log in as root on the machine where you want to prevent incoming **rlogin**, **rsh** and **rcp** attempts.

#2. Go to the user's home directory.

#3. Create an empty .rhosts file.

#4. Set the permissions to 600 (rw- --- ---)

To disable all users from using rlogin, rsh and rcp

Edit the `/etc/inetd.conf` file and comment out the line that contains the description of the **in.rshd** daemon:

```
#shell   stream  tcp     nowait  root    /usr/sbin/in.rshd       in.rshd
```

With SMF, the rlogin-related services can be disabled by the administrator executing:

Notes:

```
# inetadm -d svc:/network/login:rlogin
```

17.11 Displaying Remote Users

The command **rusers** will display users logged into other hosts. With no arguments, the **rusers** command lists all users on all machines in the network:

```
# rusers
Sending broadcast for rusersd protocol version 3...
sun         root bob
moon        root bob sue
Sending broadcast for rusersd protocol version 2...
```

To list only the users on a specific hosts, give the host name as an argument:

```
# rusers moon
moon        root bob sue
```

To get a more detailed listing (more like the **who** command), use the **-l** option to the **rusers** command:

```
# rusers -l sun
Sending broadcast for rusersd protocol version 3...
root        sun:pts/0          Jun 16 08:32       (199.3.19.10)
bob         sun:pts/1          Jun 16 11:31       1 (sun)
```

Notes:



© 2014 One Course Source

Notes:

To enable the rusers service using SMF, an administrator can execute:

```
# inetadm -e svc:/network/rpc/rusers:default
```

17.12 Checking the Status of Machines

ping

The **ping** command will send a request to another machine. If the machine responds, the message "Machine is alive" is displayed. If the machine doesn't respond, the message "no answer from machine" is displayed:

```
# ping sun
sun is alive
```

spray

If the **ping** command reports "no answer from machine" it doesn't always mean that the machine is not active. It is possible for a machine to be "up and running" but have it "ignore" **ping** requests (for example, during heavy network traffic).

The **spray** command sends multiple requests. If any requests are responded to, then you know the remote host is active:

```
# spray moon
sending 1162 packets of length 86 to moon ...
```


© 2014 One Course Source

Notes:

```
    941 packets (80.981%) dropped by moon
    760 packets/sec, 65381 bytes/sec
```

To enable the spray service using SMF, an administrator can execute:

```
# inetadm -e svc:/network/rpc/spray:default
```

rup (review)

The **rup** command sends a broadcast to all machines on the network asking them to report back to the local machine. If the machine's IP address is in the `/etc/inet/hosts` table, **rup** lists the machine's name. If not, **rup** lists the machine's IP address.

```
# rup
moon            up 30 days, 20:31,   load average: 0.03, 0.01, 0.02
sun             up 10 days, 15:34,   load average: 0.04, 0.01, 0.02
129.8.23.10     up 30 days, 21:01,   load average: 0.03, 0.01, 0.02
```

For the **rup** command to work, the rstatd service must be enabled by an administrator:

```
# inetadm -e svc:/network/rpc/rstat:default
```

netstat

Notes:

The **netstat** command will provide you with data regarding the network and network traffic:

```
# netstat -i
Name  Mtu   Net/Dest  Address   Ipkts    Ierrs    Opkts   Oerrs  Collis  Queue
lo0   8232  loopback  localhost 318832   0        318832  0      0       0
le0   1500  sun       sun       5700622  3862379  655427  1      424     0
```


© 2014 One Course Source

17.13 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #21

Web sites

System Administration Guide: IP Services >> TCP/IP Administration:
http://docs.sun.com/app/docs/doc/816-4554/6maoq01lb?a=view

Man pages

ftp	rcp	rup	spray
netstat	rlogin	rusers	telnet
ping	rsh	snoop	

© 2014 One Course Source

17.14 Summary of Commands and Files

Command	Description
ftp	Remotely copies files to and from a another machine
netstat	Displays network statistics
ping	Sends network packets to another machine to determine if it running
rcp	Remotely copies files to and from a another machine
rlogin	Remotely logins to another machine
rsh	Executes a command on another machine
rup	Displays machines that are running
rusers	Displays users on machines in the network
snoop	Displays network traffic
spray	Sends network packets to another machine to determine if it running
telnet	Remotely logins to another machine

File	Description
/etc/default/login	File used to control root logins
/etc/ftpusers	File used to control FTP access
/etc/hosts.equiv	Configuration file for **rlogin**, **rsh** & **rcp** commands
/etc/inetd.conf	Configuration file for the inetd daemon
/etc/services	Configuration file for network services & ports
/etc/shells	File that specified "value" login shells for some services (like FTP)

17.15 Lab Exercises

Scenario #1 - You need to set up remote access for a user "nick" from another machine. Pick a partner and set up remote access for this new user account on both machines. Only allow access for nick from your partner's machine.

Scenario #2 - You need to set up remote access for the root user from another machine. Pick a partner and set up remote access for the root account on both machines. Only allow access for root from your partner's machine. After testing to verify that root now has access, remove that access.

Unit Eighteen
Administrating Printers

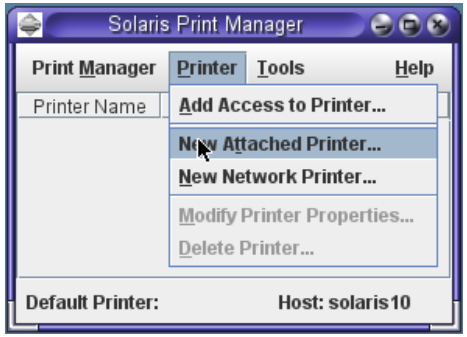
Unit topics		Page #
18.1	Using Print Manager to Set Up Local Printers	456
18.2	Using Print Manager to Set Up Network Printers	458
18.3	Printing Files	460
18.4	Monitoring the Print Queue	461
18.5	Canceling Print Jobs	462
18.6	Specifying a Default Printer	463
18.7	Printer Classes	464
18.8	Checking Printer Status	465
18.9	Managing the Print Queue	466
18.10	Printing with Different Priorities	467
18.11	Rejecting Print Jobs	468
18.12	Moving Print Jobs	469
18.13	Recovering from Common Printer Problems	471
18.14	Removing a Printer	472
18.15	Additional Resources	473
18.16	Summary of Commands and Files	475
18.17	Lab Exercises	476

18.1 Using Print Manager to Set Up Local Printers

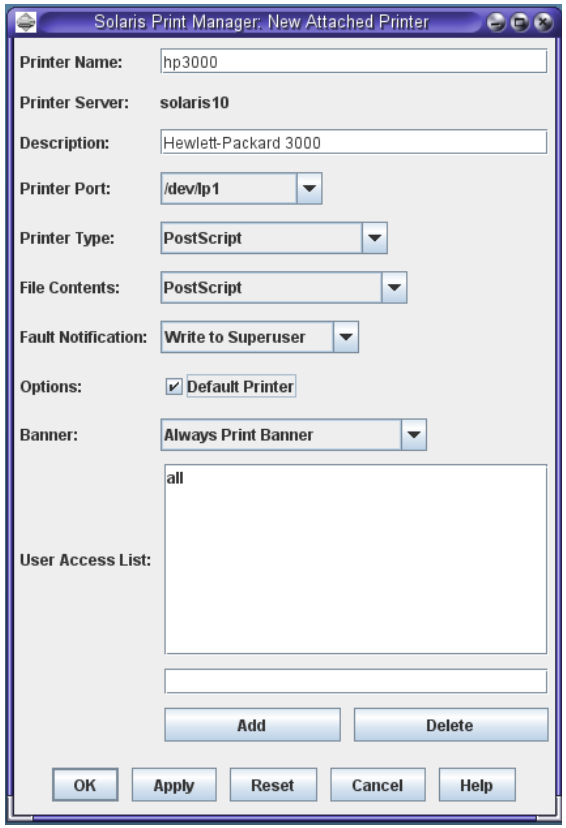
Use the /**usr**/**sbin**/**printmgr** tool.

To add a new local printer

Click Printer + New Attached Printer…



Notes:



Name of Printer is your choice

Print server is local machine

Description can be anything

Printer Port:

`/dev/term/a`	serial port a
`/dev/term/b`	serial port b
`/dev/bpp0`	parallel port

Printer Type is the name of the printer. Names are located in the `/usr/share/lib/terminfo` file

File Contents depends on printer type

Fault notification:

Write to superuser displays error messages on screen

Mail to superuser email error messages to root

Options:

Default printer makes this printer the default for all users on this host

Always print banner will print a banner (header) for each print job

Access List is who can use this printer

Notes:

18.2 Using Print Manager to Set Up Network Printers.

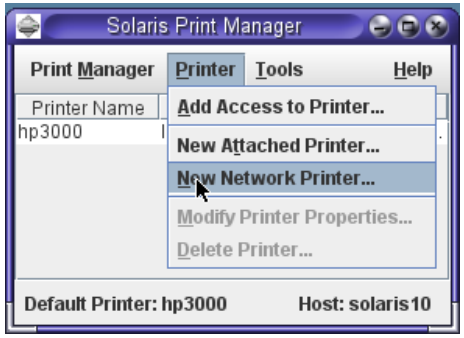
Notes:

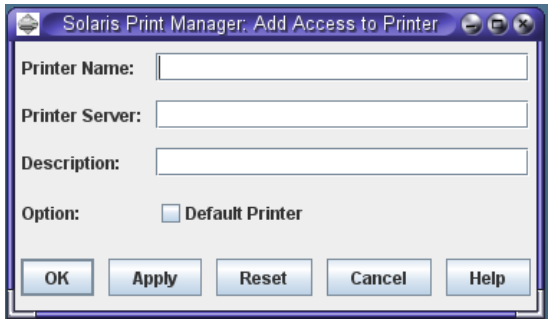
To start Print Manager

```
# /usr/sbin/printmgr &
```

To add a access from the network to a printer

Click Printer + Add Access to Printer…





Printer Name is whatever the name of the printer on the server is

Print Server is the host name of the host that is connected to the printer

Description can be whatever you want (good idea would be where the printer is located)

Option: Default printer makes this printer the default for all users on this host

Notes:

Notes:

18.3 Printing Files

The **lp** command will print files (or data past in via a pipe). Unless either the System Administer or user has set a default printer, the **-d** option must be used to specify a printer:

```
# lp -d laser1 test.data
request id is laser1-281
```

To print more than one copy of the file, use the **-n** option:

```
# lp -d laser1 -n 3 test.data
request id is laser1-282
```

You can also print the outcome of another command by piping the output into the **lp** command:

```
# who | lp -d laser1 -n 3 test.data
request id is laser1-282
```

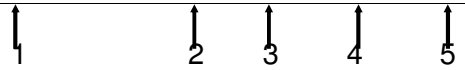
Notes:

18.4 Monitoring the Print Queue

The **lpstat** command is useful for getting information regarding the printer. It can be used to list the current print jobs, to determine the default printer (if any), or to determine the status of the printer.

List current print jobs

```
# lpstat  laser2
laser2-595          sun!bob      90 Jul 30 11:07
```



1 job number
2 machine that the print job originated from
3 User who started the print job
4 Size in bytes of the print job
5 Date/time the **lpstat** command was run

Note: If you don't have a system default printer and you run **lpstat** without specifying a printer, **lpstat** will query ALL printers to display their print jobs.

List default printer

```
# lpstat -d
system default destination: laser1
```

Notes:

18.5 Canceling Print Jobs

The **cancel** command will cancel print jobs that are in the print queue:

```
# lp -d laser1 /etc/skel/local.login
request id is laser1-738
# lpstat laser1
laser1-738          sun!bob      90 Jul 30 11:21
# cancel laser1-738
request "laser1-738" cancelled
# lpstat laser1
{no output since there are no jobs left}
```

You can delete all the print jobs of a particular user by using the **-u** option:

```
# lpstat laser1
laser1-738          sun!bob      90 Jul 30 11:21
laser1-739          sun!bob      90 Jul 30 11:21
laser1-740          sun!sue      90 Jul 30 11:21
laser1-741          sun!bob      90 Jul 30 11:21
# cancel -u bob laser1
request "laser1-738" cancelled
request "laser1-739" cancelled
request "laser1-741" cancelled
# lpstat laser1
laser1-740          sun!sue      90 Jul 30 11:21
```

18.6 Specifying a Default Printer

As the System Administrator you can specify a default printer with the **lpadmin** command:

```
# lpadmin -d laser1
# lpstat -d
system default destination: laser1
```

Notes:

18.7 Printer Classes

In cases where you have multiple printers of the same type in a central location, you might want to make printer classes. Instead of having users print to a specific printer, you will have them print to the printer class. The class will consist of 2 or more printers. Print jobs will be evenly distributed among the printers in the printer class.

The **lpadmin** command is used to create a printer class and add printers to the class. The **accept** command is used to tell the system that the printer class is ready to accept print jobs:

```
# lpadmin -p laser1 -c lasers
# lpadmin -p laser2 -c lasers
# accept lasers
destination "lasers" now accepting requests
```

Note: the act of adding a printer to a class will create the class if it doesn't already exist.

Notes:



18.8 Checking Printer Status

The **lpstat** command can be used to determine the status of a printer. It will indicate if the printer is accepting print requests, if it is currently printing and what jobs are currently printing:

```
# lpstat -t
scheduler is running
no system default destination
device for laser1: /dev/console
laser1 accepting requests since Fri Jul 30 11:48:22 PDT 1999
printer laser1 is idle. enabled since Fri Jul 30 11:48:22 PDT 1999. available.
```

Notes:

18.9 Managing the Print Queue

If you are either the owner of a print job or the System Administrator, you can control the print job while it is still in the print queue. The **lp** command provides the following options to control print jobs:

Keyword	Meaning
hold	Allows you to pause a print job
resume	Allows you to "continue" a paused print job
immediate	Allows the System Administrator to send the print job to the top of the queue

The following command will pause a print job:

```
#lp -i laser-123 -H hold
{no output for this command}
```

Notes:

18.10 Printing with Different Priorities

Users can request higher priorities for their print jobs when running the **lp** command by using the **-q** option. The **-q** option accepts a numeric option (from 0 to 39) that indicates priority of the print job. The lower the number, the higher the priority.

The following will print the `/etc/skel/local.login` file with the highest priority:

```
lp -d laser1 -q 0 /etc/skel/local.login
request id is laser1-738
```

Notes:

- If no priority is specified, 20 is used as a default
- Since all users can use this option, it's best not to let anyone know about it unless there is a good reason. Otherwise, "print wars" might break out!

Notes:

Notes:

18.11 Rejecting Print Jobs

In some cases you might want to have the printer reject print jobs. For example, the printer might be taken offline for some maintenance work.

To tell the printer to stop accepting print requests, use the **reject** command:

```
# reject -r "Printer down for repairs" laser1
destination "laser1" will no longer accept requests
```

The **-r** option is used to specify a "reply comment" that the users can see if they run the **lpstat -t** command:

```
# lp /etc/skel/local.login
laser1: queue is disabled
# lpstat -t
scheduler is running
system default destination: laser1
device for laser1: /dev/console
laser1 not accepting requests since Fri Jul 30 12:28:13 PDT 1999 -
        Printer down for repairs
printer laser1 is idle. enabled since Fri Jul 30 11:48:22 PDT 1999. available.
```

To have the printer accept print jobs again, use the **accept** command:

```
# accept laser1
destination "laser1" now accepting requests
```

Notes:

18.12 Moving Print Jobs

If you bring a printer off line for repairs, you might still have some print jobs in the queue. You can move these print jobs to another printer with the **lpmove** command.

The first step is to tell the printer to stop printing these print jobs. The command **disable** will perform this function:

```
# disable laser1
printer "laser1" now disabled
# lp /etc/skel/local.login
request id is laser1-4 (1 file(s))
# lpstat
laser1-4            bobofett!root      575   Jul 30 12:33
```

Notice that **disable** does not prevent print jobs from being added to the queue. It just prevents them from printing.

To move the print jobs, use the **lpmove** command:

```
# lpmove laser1 laser2
{no output for this command}
```

Notes:

You can also move just one print job at any time with the **lpmove** command. This is useful if a large print job is placed accidentally on a busy printer:

```
# lpmove laser1-112 laser2
{no output for this command}
```

When the printer is back on line, use the **enable** command to allow printing to continue:



```
# enable laser1
printer "laser1" now enabled
```

18.13 Recovering from Common Printer Problems

In some cases, a printer might go "haywire". If this is the case, the following actions can be taken to fix the printer:

Method #1
1. Cancel all print jobs
2. Turn off the printer
3. Turn it back on again

Method #2
1. Cancel all print jobs
2. Run the "**/etc/init.d/lp stop**" command to stop the printer daemons
3. Run the "**/etc/init.d/lp start**" command to restart the printer daemons

The **cancel** command may not be able to function when the printer is really "messed up". To manually remove print jobs, run the following command:

```
# rm -r /var/spool/lp/requests/hostname/*
```

"hostname" is the name of the machine where the printer is set up as a local printer.

Notes:



18.14 Removing a Printer

To remove a printer, use the **-x** option to the **lpadmin** command:

```
# lpadmin -x printer_name
```

If removing a local printer, first use the **reject** and **disable** commands to prevent any problems with print queues:

```
# reject printer_name
# disable printer_name
# lpadmin -x printer_name
```

Removing a printer will also remove its print queue.

Notes:

18.15 Additional Resources

Books

Solaris: The Complete Reference
by Paul Watters
Publisher: McGraw-Hill Osborne Media
ISBN: 0072229985
Chapter #18

Web sites

System Administration Guide: Advanced Administration, Managing Printing Services (Overview):
http://docs.sun.com/app/docs/doc/817-0403/6mg741bn2?a=view

System Administration Guide: Advanced Administration, Planning Printers on Your Network (Tasks):
http://docs.sun.com/app/docs/doc/817-0403/6mg741bnb?a=view

System Administration Guide: Advanced Administration, Setting Up Printers (Tasks):
http://docs.sun.com/app/docs/doc/817-0403/6mg741bnh?a=view

System Administration Guide: Advanced Administration, Administering Printers (Tasks):
http://docs.sun.com/app/docs/doc/817-0403/6mg741bog?a=view

System Administration Guide: Advanced Administration, Customizing the LP Print Service (Tasks):
http://docs.sun.com/app/docs/doc/817-0403/6mg741brf?a=view

System Administration Guide: Advanced Administration, LP Print Service (Reference):
http://docs.sun.com/app/docs/doc/817-0403/6mg741bsb?a=view

Man pages

accept	disable	lpadmin	printmgr
admintool	enable	lpmove	reject
cancel	lp	lpstat	

18.16 Summary of Commands and Files

Command	Description
accept	Allows print jobs to be sent to printer queue
admintool	GUI-based tool for administering printers in Solaris 8 and earlier
cancel	Cancels print jobs from the print queue
disable	Prevents print jobs from going from the print queue to the printer
enable	Allows prints jobs to go from the print queue to the printer
lp	Sends print jobs to the printer
lpadmin	Command-based tool for administering printers
lpmove	Moves print jobs from one printer queue to another printer queue
lpstat	Provides printer statistics
printmgr	GUI-based tool for administering printers in Solaris 9 and later
reject	Prevents print jobs from being sent to printer queue

File	Description
/dev/bpp0	Device file for the first parallel port
/dev/term/a	Device file for the first serial port
/dev/term/b	Device file for the second serial port
/usr/share/lib/terminfo	Printer definition file

18.17 Lab Exercises

No labs for this Unit