SiliconGraphics
Computer Systems

IRIX System Administration

Student Handbook

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

RESTRICTION ON USE:

U.S. GOVERNMENT RESTRICTED RIGHTS LEGEND:

PART NUMBER:

LBT111-2.0-6.2-S-SD-W, June 1996

TRADEMARKS:

SGI Trademarks

Showcase™, Silicon Graphics®, Silicon Studio™, SGI Logo®, Sirius Video™, SkyWriter™, StereoView™, System Manager™, VGXT™, VideoCreator™, VideoLab™, Virtual 24™, VoxelLab™, VTX™, WorkSpace™

Other Trademarks

Ethernet is a registered trademark of Xerox Corp.
OSF/Motif is a trademarks of Open Software Foundation
PostScript is a registered trademark of Adobe Systems, Inc.
UNIX is a registered trademark of UNIX System Laboratories.
X Window System is a trademark of the Massachusetts Institute of Technology

CONTRIBUTIONS:

Course development based on previous courses developed by Carol Hyman, Bill Mannel, John Harechmak, Adam Ginsberg and John Schmelzer

Technical review by Chuck Adams, Paul Bell, Sharon Bayne, Barbara Stamatakis, Dick Daneau, Patrick Reardon, Vicki Shreiner, Dave Williams, and Kevin Brunner

Illustrations by Chris Taylor and Kathy Gochenour

Editing and production by Chuck Hudson and Chris Taylor

FONT CONVENTIONS:

The following font conventions are used throughout this manual:

`Courier` is used for pathnames, filenames, programs, code, and system output.

`Courier bold` is used for user input.

Palatino is used for names of windows, buttons, and fields.

Palatino bold is used for menu items and keyboard keys; keyboard keys are surrounded by angle brackets.

Italics is used for first mention of a new term and to indicate mouse functions.

Contents

Module 4: User Support

Module 5: System Monitoring

Module 6: Disk Maintenance

Module 7: Filesystems

Module 8: Swap Administration

Module 9: Introduction to NFS

Module 10: Logical Volume Management

Module 11: System Startup and Shutdown

Module 12: IRIX Run Levels

Module 13: Printers & Terminals

Module 14: Installing Software on Your IRIS

Module 15: Backup and Recovery

SiliconGraphics
Computer Systems

Module 1: Introduction

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

In this module, you will find out about your instructor, the other students in the class, and the topics to be covered this week.

1-2 Welcome

Welcome

- Classroom introductions
- Goals
- Course objectives
- Course presentation/participation

IRIX System Administration: Introduction Revision 2.0 1 - 2

Student Notes

Classroom introductions: What is your name, company, responsibilities, type of computer(s) you use, and your UNIX® or IRIX experience.

Goals: What do you expect to gain from this course?

Course objectives: You will learn how to administer the IRIS, understand operating system operation, work efficiently, avoid and correct errors, and demonstrate your knowledge in system troubleshooting.

Course presentation/participation: You will learn through concept briefing, lecture, hands-on lab exercises, lab debriefing, reviews, and troubleshooting.

1-3 Prerequisites

Prerequisites

- *Introduction to IRIX* course

  or

- Six or more months of experience using IRIX

IRIX System Administration: Introduction Revision 2.0 1 – 3

Student Notes

1-4 Materials and Tools

Materials and Tools

- Yours to keep and mark up
  - Student workbook (lecture)
  - Lab book (lab exercises and review questions)
- Reference only/SGI training property
  - System–related tools
    - Release notes
    - Online books
    - Man pages
  - Classroom documentation sets

IRIX System Administration: Introduction Revision 2.0 1 – 4

Student Notes

The student workbook contains a table of contents, modules, and an index.

1-5 Course Topics

Course Topics

- Introduction
- System Setup
- System Documentation
- User Support
- System Monitoring
- Disk Maintenance
- Filesystems

IRIX System Administration: Introduction Revision 2.0 1 – 5

Student Notes

1-6 Course Topics (continued)

Course Topics (continued)

- Swap Administration
- Introduction to NFS
- Logical Volume Management
- System Startup and Shutdown
- IRIX Run Levels
- Printers
- Software Installation
- Backups and Recovery

Menu

IRIX System Administration: Introduction Revision 2.0 1 – 6

Student Notes

SiliconGraphics
Computer Systems

Module 2: System Setup

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module will discuss how to setup your system—connecting cables, setting the time and date, basic network setup, etc.

2-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Set up an IRIS
  - Connect cables
  - Set the date and time zone
  - Establish a system hostname
  - Start basic Ethernet communications
- Start up and shut down your system properly
- Maintain your system

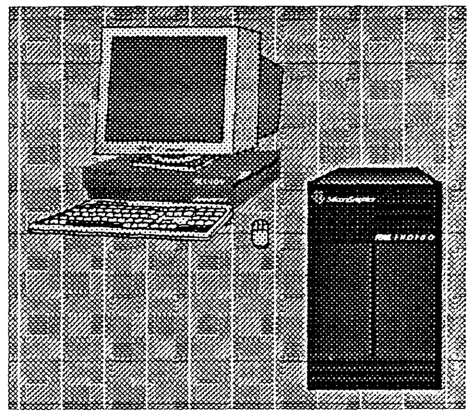
IRIX System Administration: System Setup Revision 2.0 2 - 2

Student Notes

2-3 SGI IRIS Desktop Chassis Types

SGI IRIS Desktop Chassis Types

- Indy
- Challenge S
- [Power] Indigo^2
- [Power] Challenge M



IRIX System Administration: System Setup Revision 2.0 2 – 3

Student Notes

Product Name	Processor Options	Graphics Options
Indigo	R4000	Entry XS / XS24 XZ Elan
Indy Challenge S	R4000 PC / SC R4400 SC R4600 PC / SC R5000 PC / SC	XL / XL24 XZ XGE-24 (R5000 only)

Product Name	Processor Options	Graphics Options
Indigo^2	R4000 SC	XL24
Challenge M	R4400 SC	XZ
PowerChallenge M	R8000	Extreme
	R10000	Solid Impact
		High Impact
		Maximum Impact

"PC" indicates a system with only primary cache. "SC" indicates that the system has both primary and secondary caches. R8000 and R10000 systems only support secondary cache as standard.

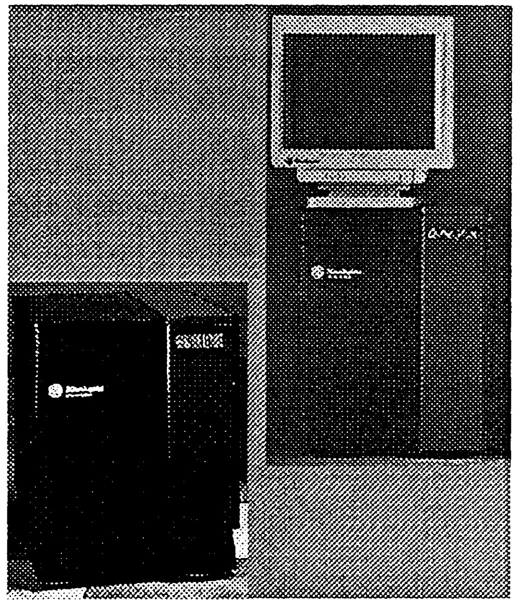
Challenge S is an Indy server with either an R4400 or an R4600 processor. Challenge M is an Indigo2 server with either an R4000 SC or an R4400 SC processor. Challenge products are servers, generally with no graphics. (Some servers may have console graphics as an add-on.)

The PowerChallenge M server supports the R8000 or R10000 processor.

2-4 IRIS Deskside Chassis Types

IRIS Deskside Chassis Types

- [Power] Onyx L
- Challenge DM, L,
  Power Challenge L



IRIX System Administration: System Setup Revision 2.0 2 – 4

Student Notes

Product Name	Processors Options	Maximum Processors	Graphics Options
Crimson	R4000 SC R4400 SC	1 1	Elan VGX / VGXT RealityEngine
Challenge DM	R4000 SC R4400 SC	4 4	
Challenge L PowerChallenge L	R4000 SC R4400 SC R8000 R10000	12 12 6 12	Extreme
Onyx PowerOnyx	Same as Challenge	Same as Challenge	VTX RealityEngine^2 InfiniteReality

Challenge DM systems support a single processor board which can contain up to four R4400 processors.

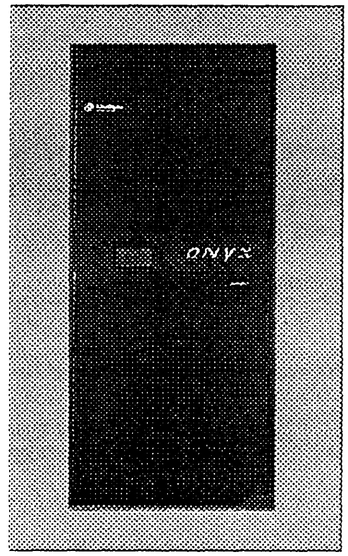
PowerChallenge and PowerOnyx systems contain only R8000 or R10000 processors.

The RealityStation is a single-processor version of an Onyx supporting RealityEngine^2 graphics. The *i*Station is an R4400-based or R10000-based system with a modified InfiniteReality graphic system.

2-5 IRIS Rack Chassis Types

IRIS Rack Chassis Types

- [Power] Onyx XL
- [Power] Challenge XL



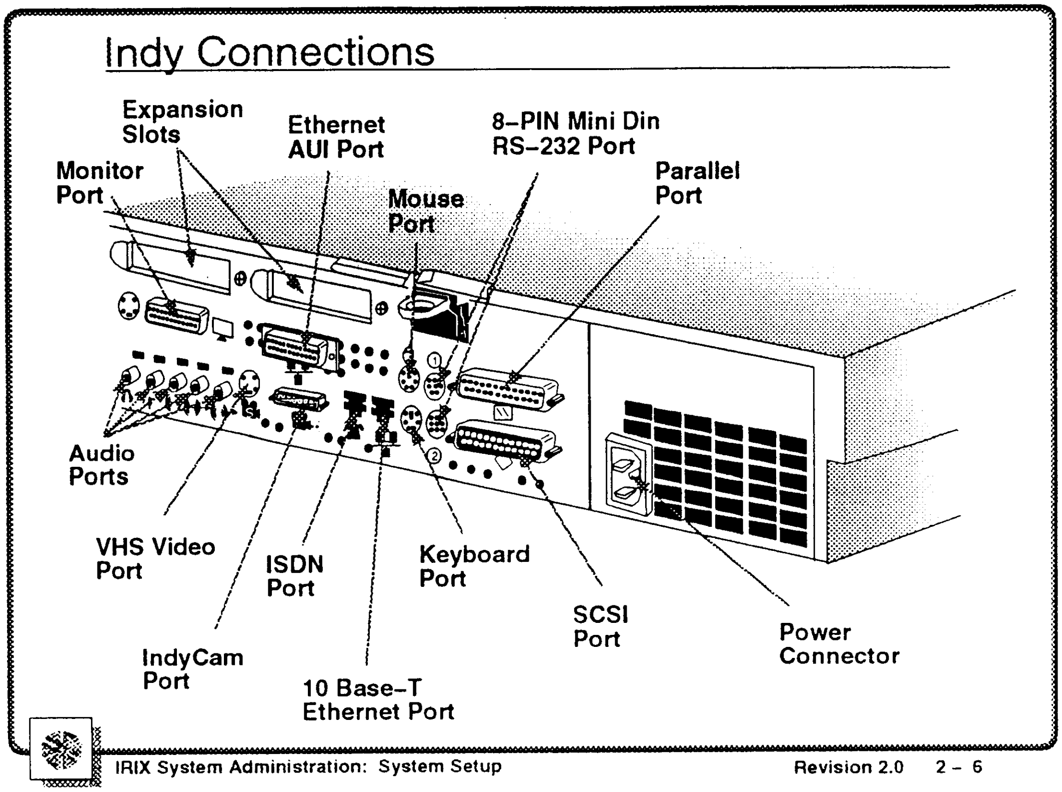
IRIX System Administration: System Setup Revision 2.0 2 – 5

Student Notes

Product Name	Processors Options	Maximum Processors	Graphics Options
Challenge XL PowerChallenge XL	R4000 SC R4400 SC R8000 R10000	36 36 18 36	
Onyx XL PowerOnyx XL	R4000 SC R4400 SC R8000 R10000	24 24 12 24	VTX RealityEngine^2 InfiniteReality
PowerChallenge Array	R8000 R10000	144	

Standard Challenge and PowerChallenge systems do not support graphics subsystems. Challenge GR and PowerChallenge GR systems can be upgraded to support RealityEngine^2 or InfiniteReality graphics.

2-6 Indy Connections



Student Notes

To connect a SCSI device to the workstation connect the flat, smaller end of the cable to the port above the SCSI icon on the back of the workstation. Connect the other end of the SCSI cable to one of the SCSI connectors on the back of the device. To connect a SCSI 1 device, such as a CD-ROM, to your Indy or Indigo2 system, you need a SCSI 1 to SCSI 2 cable. It is possible to daisy-chain an external SCSI device to another SCSI device. Ensure that the SCSI IDs are unique in the same daisy chain. Valid SCSI addresses are usually from 1 to 7. Make sure the last SCSI device is properly terminated.

If you are connecting an ASCII terminal, attach it to serial port 1. If you are connecting another device, such as a modem, you can attach it to any port.

The Ethernet® attachment unit interface (AUI) port attaches to a transceiver with a cable (drop line) connected to the Ethernet. Only one of the Ethernet ports can be active at any one time. If both ports have valid connections, the system defaults to the 10-base T interface. The Challenge S system has both ports active, because the system contains two networking cards, which act as a network gateway.

2-7 System Startup Procedure

System Startup Procedure

- Connect all components
- Power on monitor, terminals, and CD–ROM
- Power on CPU
  - System may autoboot (depending on PROM variable settings)
- If it does not autoboot:
  - Select **System Startup** at the PROM menu
- Log on

IRIX System Administration: System Setup Revision 2.0 2 – 7

Student Notes

Power on all SCSI peripherals *before* you turn on the computer; otherwise, the SCSI bus does not see the devices during its power-on diagnostics and you cannot use them until you reboot the system. This applies to DAT tape drives, CD-ROM drives, and external disk drives, but not to ASCII terminals or printers—only peripherals attached to the SCSI bus.

2-8 Determining a System Configuration

Determining a System Configuration

Use the following commands to check a new machine:

- **hinv(1M)** shows hardware capability of a system
- **/usr/gfx/gfxinfo(1G)** determines graphics capability
- **versions(1M), showprods(1M)** lists software installed
- **chkconfig(1M)** checks software configuration flags

IRIX System Administration: System Setup Revision 2.0 2 – 8

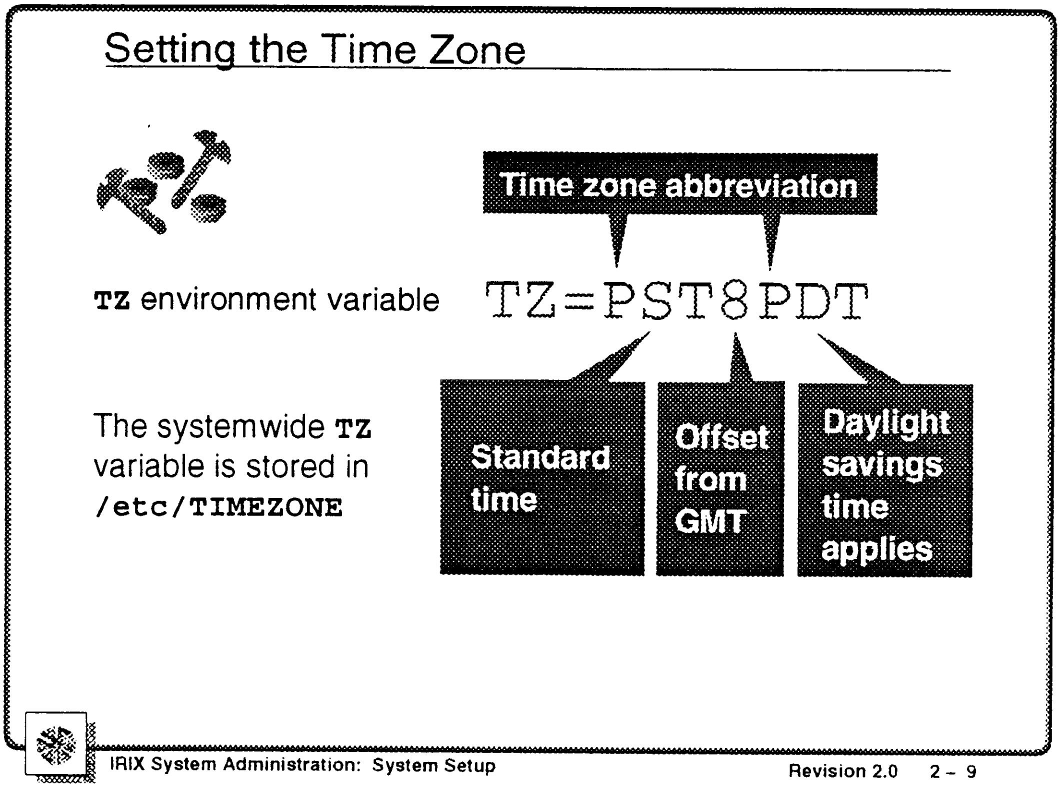
Student Notes

The hinv command displays the contents of the system hardware inventory table. Example: Indigo2

```
# hinv
Iris Audio Processor: version A2 revision 1.1.0
1 150 MHZ IP22 Processor
FPU: MIPS R4010 Floating Point Chip Revision: 0.0
CPU: MIPS R4400 Processor Chip Revision: 5.0
On-board serial ports: 2
On-board bi-directional parallel port
Data cache size: 16 Kbytes
Instruction cache size: 16 Kbytes
Secondary unified instruction/data cache size: 1 Mbyte
Main memory size: 96 Mbytes
EISA bus: adapter 0
Integral Ethernet: ec0, version 1
Integral SCSI controller 1: Version WD33C93B,revision D
```

```
Integral SCSI controller 0: Version WD33C93B,revision D
CDROM: unit 3 on SCSI controller 0
Disk drive: unit 2 on SCSI controller 0
Disk drive: unit 1 on SCSI controller 0
Graphics board: GU1-Extreme
```

2-9 Setting the Time Zone



Student Notes

Accurate timekeeping is very important on an SGI system. All SGI IRIS systems have a built-in time-of-day clock. Example:

TZ Value	Location
BST3BDT	Brazil
EST5EDT	Eastern USA
CST6CDT	Central USA
HST10HDT	Hawaii
TST7TDT	Thailand
SST8SDT	China
JST9JDT	Japan
NZT12NDT	New Zealand
GMT0BST-1,M3.5.0/1,M10.5.0/2	United Kingdom

2-10 Setting the Date

Setting the Date

- Set the date and time with the **date(1)** command:

  `date [[mmdd]HHMM | mmddHHMM[cc]yy][.ss]`

- To establish a date of 12 Oct 1996, 11:04 a.m.:

  `# date 10121104`

- **timed(1M)** daemon may synchronize date and time on your Local Area Network

IRIX System Administration: System Setup Revision 2.0 2 – 1 0

Student Notes

Brackets ([ ]) imply an optional string.

A pipe symbol (|) implies either/or.

Set the time zone first, and reboot the system to set the correct date and time.

2-11 Setting the System Hostname

Setting the System Hostname

- All systems are shipped with the name "IRIS" as the default
- Change the system hostname in main memory:

```
# hostname frodo
```

- Put a unique system hostname in the file `/etc/sys_id`:

```
# hostname > /etc/sys_id
```

IRIX System Administration: System Setup Revision 2.0 2 - 1 1

Student Notes

The `hostname` command only updates the system hostname until the next system reboot. You must save this value in the `/etc/sys_id` file for the system to remember its hostname across reboots.

2-12 Hostname/IP Address Table /etc/hosts

Hostname/IP Address Table /etc/hosts

To communicate with other hosts on a TCP/IP network, you must have a file that maps system names to IP addresses – **/etc/hosts**

- At minimum, you must have at least two entries in this table:
  - Local host for loopback testing, X Window System, and licensing
  - Your system's IP address and hostname
- Must change default Internet address 192.0.2.1 to a valid address

```
# This entry must be present or the system will not work
127.1    localhost       localhost
# 192.0.2.1 IRIS
192.26.52.128  frodo
```

IRIX System Administration: System Setup   Revision 2.0   2 – 1 2

Student Notes

Because all systems are shipped with the name "IRIS," you need to give the system a unique hostname before attaching it to the network.

2-13 Establishing Basic Ethernet Support

Establishing Basic Ethernet Support

- To do remote logins and remote file copies to other machines on the net, the hostnames and Internet addresses of other machines are placed in **/etc/hosts**
- Example of an **/etc/hosts** file:

```
# cat /etc/hosts
127.1              localhost
192.26.52.128    batman
192.26.52.55       robin
192.26.52.23       batgirl
192.26.52.16       joker
192.26.52.11       riddler
```

IRIX System Administration: System Setup   Revision 2.0   2 – 1 3

Student Notes

The numbers "192.26.52.128" are Internet addresses and are generally unique in the world. The numbers correspond to mailing addresses. The hostnames are familiar names of machines.

The `127.1 localhost` address is required.

2-14 Establishing Basic Ethernet Support (continued)

Establishing Basic Ethernet Support (continued)

- If your site has a domain naming scheme, enter the following information into **/etc/hosts**
  - Full domain name
  - Aliases

```
# cat /etc/hosts
127.1          localhost.acme.com      localhost
192.26.52.12   batman.acme.com         batman
192.26.52.55   robin.acme.com          robin      bat2
192.26.52.23   batgirl.acme.com        batgirl
192.26.52.16   joker.acme.com          joker
192.26.52.11   riddler.acme.com        riddler
```

- Either reboot, or cycle the **network** script

```
# /etc/init.d/network stop
# /etc/init.d/network start
```

IRIX System Administration: System Setup   Revision 2.0   2 – 1 4

Student Notes

You only need to reboot or cycle the network script if you add your system to the /etc/hosts file, not every time you add an entry to the hosts file.

DNS is required if you are connecting to the Internet. DNS uses domain names as an administrative area of control. The domain names must be unique across the Internet.

2-15 System Shutdown Procedure

System Shutdown Procedure

- Log in as **root** or **su** to root
- Warn all users
- Issue a shutdown command:

```
# shutdown -y
```

- Wait for PROM menu
- Power off the system and peripherals

Student Notes

The `shutdown` command issues a standard message:

```
"The system will be shut down in 60 seconds.
Please log off now."
```

2-16 System Precautions

System Precautions

- Powering up/down frequently can lead to additional wear on system components
- Pressing the reset button should only be done as a last resort
- Moving a system or removing skins and covers while the system is running can damage disk drives
- Running the system without skins and covers can lead to overheating
- Avoiding electrostatic discharge can preserve sensitive electronic components

IRIX System Administration: System Setup Revision 2.0 2 - 1 6

Student Notes

SiliconGraphics
Computer Systems

Module 3: System Documentation

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

In this module, you will learn what documentation is available to help you with the system administration task.

3-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Effectively search IRIX documentation
- Determine where to look for administration questions
- Use the online reference material:
  - Man pages
  - Release notes
  - InSight books
- Maintain the man page entires
- The **`whatis`** database
- Use an *Owner's Guide* for specific platforms

IRIX System Administration: System Documentation Revision 2.0 3 - 2

Student Notes

3-3 What Are Man Pages?

What Are Man Pages?

Man pages

- Contain reference materials, not instructional information
- Main source of information relating to individual daemons, commands, and files
- Sometimes terse and unfriendly, but very important
  - Use the **man(1)** and **xman(1)** commands
- Locate and print titled entries from the online reference manuals
- Print summaries of manual entries selected by keyword or by associated filename

IRIX System Administration: System Documentation Revision 2.0 3 – 3

Student Notes

Man pages can take 45 Mbyte or more of disk storage. If you are working in a networked environment, you might want to designate one workstation to be the reference page server and export the `catman` directories. All other workstations can remove those files and mount them remotely from the server. Because NFS mounts take up no disk space on the client workstation, that disk space is available for other uses.

3-4 IRIX Man Page Organization

IRIX Man Page Organization

- These sections contain similar information to the traditional AT&T sections:

Section 1	User commands, application programs, and programming commands
Section 2	System calls
Section 4	File formats
Section 5	Miscellaneous: **ascii(5)**, **nroff(1)**, **troff(1)** macro package information
Section 7M	Device file information: **ipi(7M)**

- To browse through introductory pages of all sections:

```
# man intro
```

IRIX System Administration: System Documentation Revision 2.0 3 – 4

Student Notes

3-5 IRIX Man Page Organization (continued)

IRIX Man Page Organization (continued)

Example sections generally unique to IRIX:

Section 1M	System administration commands
Section 1C	Network communication commands
Section 1G	Graphical applications, `jot(1G)`
Section 3	Functions such as `printf(3)` and graphics library calls (3G)
Section 6	Games—`dog(6D)`, SGI image tools — `ipaste(6D)`, references to other SGI–specific programs stored in `/usr/sbin`
Section 7	Special files about hardware peripherals and UNIX system device drivers

The notation `ls(1)` indicates that the man page for `/bin/ls` is located in Section 1

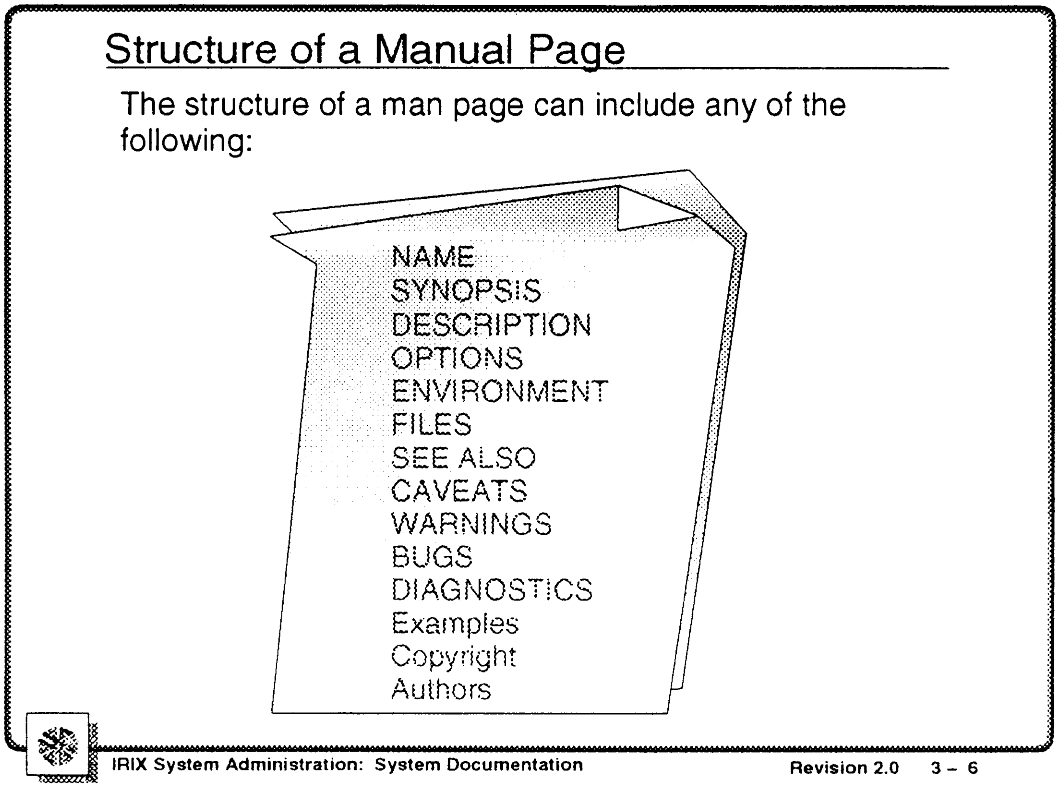
IRIX System Administration: System Documentation Revision 2.0 3 – 5

Student Notes

Section 3 is divided into some of these major collections:

Section 3C	Standard C Library
Section 3G	IRIS Graphics Library and general-purpose library routines
Section 3M	Math Library
Section 3S	Standard I/O package
Section 3B	4.3BSD system calls and library routines
Section 3N	Networking functions
Section 3Y	Remote procedure call (RPC) and NIS support routines
Section 3R	RPC services built on top of the Sun RPC protocol
Section 3P	Parallel processing library
Section 3T	General terminal interface

3-6 Structure of a Manual Page



Student Notes

3-7 Online Manual Pages

Online Manual Pages

- Located in multiple locations, referenced in the following order:

  /usr/share/catman

  /usr/share/man

  /usr/catman

  /usr/man

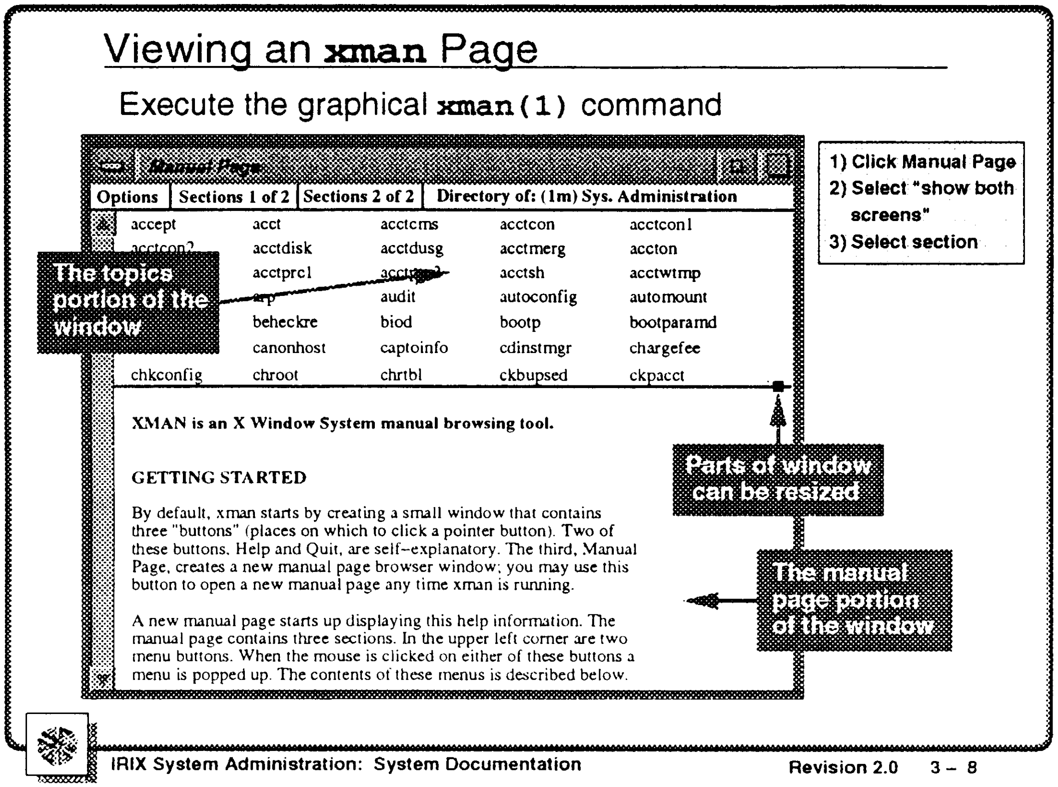
- Five categories:

a_man	System administration
g_man	Graphics Library reference, Section 3G
p_man	Contains the programmer's reference, Sections 2, 3, 4, and 5
u_man	User's reference, Sections 1 and 6D
local	Site–specific references

IRIX System Administration: System Documentation Revision 2.0 3 – 7

Student Notes

3-8 Viewing an xman Page



Student Notes

The xman program is a manual page display program for the X Window System™. The xman program lets you customize both the directories to be searched for manual pages, and the name that each directory maps to in the **Sections** menu. By using the MANDESC file, a user or System Manager can more closely control which manual pages appear in each of the sections represented by menu items in the **Sections** menu. This functionality is only available on a section-by-section basis, and individual manual pages may not be handled in this manner.

3-9 The whatis Database

The **whatis** Database

A short description of every man page

- Use **makewhatis(1M)** to create the database
- Use the **man(1)**, **apropos(1)**, and **whatis(1)** commands to access the **whatis** database
  - **man -k** and **apropos** commands locate entries by keyword lookup (name and description)
  - **whatis** and **man -f** commands describe what a command is (letter matching)
- Format

```
apropos keyword(s)
man -k keyword(s)
whatis keyword(s)
man -f keyword(s)
```

IRIX System Administration: System Documentation Revision 2.0 3 – 9

Student Notes

The makewhatis command, by default, creates the file /usr/share/catman/whatis. Every manual page on the system is read.

After installing software using inst or swmgr, the makewhatis command is automatically run the next time the system is rebooted. The makewhatis command takes 10 to 15 minutes to run.

3-10 Accessing the `whatis` Database

Accessing the `whatis` Database

```
% man -k copy
cp, ln, mv (1)  - copy, link or move files
cpio (1)        - copy file archives in and out
distcp (1M)     - copy software distribution
imgcopy (1)     - copy and convert image file
rcp (1C)        - remote file copy

% whatis ls
ls (1)          - list contents of directory
lsback (3G)     - controls whether the ends of a
                  line segment are colored
lsearch, lfind  - linear search and update
```

IRIX System Administration: System Documentation Revision 2.0 3 - 1 0

Student Notes

The `man -k` command is the same as the `apropos` command. The `man -f` command is the same as the `whatis` command.

3-11 Using Release Notes

Using Release Notes

What are release notes?

- Release notes are used to learn more about a specific IRIX release
- Use the hardcopy version to learn about any installation changes or problems *before* attempting any updates or new installations to your system
- OR, use **CDrelnotes** or **CDgrelnotes** to learn about operating system and subsystem changes *before* you install via the CD with the 6.2 software
- Use the **relnotes(1)** and **grelnotes(1)** commands to learn about operating system (OS) and subsystem changes after installation

IRIX System Administration: System Documentation Revision 2.0 3-11

Student Notes

Using a CD-ROM to mount a filesystem so that the release notes can be read is explained in the chapter about filesystem maintenance. Usually, the system daemon `mediad` takes care of mounting the CD when you insert it into the CD-ROM drive. It is mounted to the directory `/CDROM`. Do an `ls` listing to see the subdirectories on the CD. Use the commands `CDgrelnotes` and `CDrelnotes` to read the release notes from the CD-ROM and not from your system disk.

For example, read before installing IRIX 6.2:

```
$ CDgrelnotes
```

Another way to read the release notes from the CD-ROM drive is to use the following command:

```
$ grelnotes -rpath /CDROM
```

Use `relnotes` or `grelnotes` to view notes after installing a product.

3-12 Using Command-Line Release Notes

Using Command–Line Release Notes

The ASCII–based **relnotes(1)** command is hierarchical

```
# relnotes
relnotes
The following products have release notes installed:

4Dwm       c_dev          ftn77_eoe     mmail
Cadmin     c_eoe          ftn90_dev     motif_dev
IDO        compiler_dev   ftn90_eoe     motif_eoe
IRIX       compiler_eoe   ftn_dev       netscape
Performer  demos          gl_dev        nfs


# relnotes IRIX
The chapters for the "IRIX" product's release notes are:
chap    title
1       Introduction
2       Product Information
3       Software Compatibility
Use "sbin/relnotes productname chapter" to view a
chapter
```

IRIX System Administration: System Documentation Revision 2.0 3 – 1 2

Student Notes

You can also view the command-line release notes and search for a specific chapter.

The command you use is similar to the following:

```
relnotes ftn90_dev 1
```

3-13 Using the Graphical Release Notes

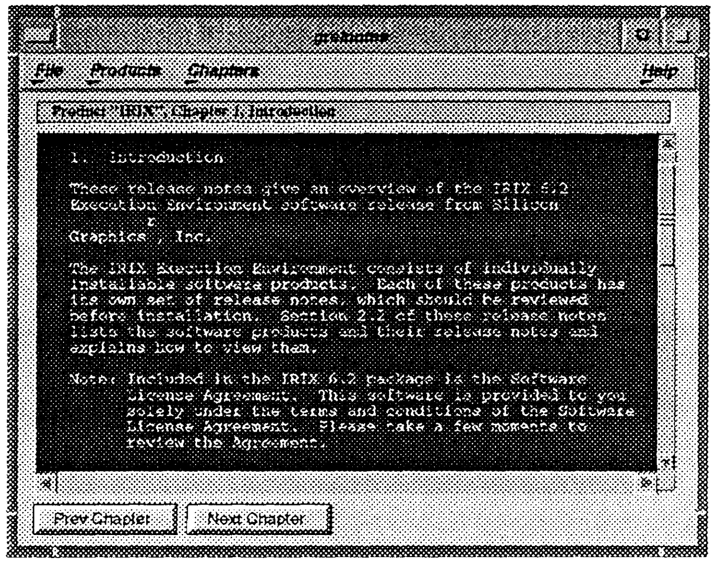
Using the Graphical Release Notes

The **grelnotes(1)** command is the graphical version of the **relnotes(1)** command

Use the **Products** menu to select various products or topics

Use the **Chapters** menu to select individual chapters for the chosen topic

Use the **File** menu to print or update the release notes



IRIX System Administration: System Documentation   Revision 2.0   3 – 1 3

Student Notes

The graphical release notes program allows browsing of the online release notes provided with each Silicon Graphics software product.

From the **Products** menu, select the software product whose release notes you want to view. The first chapter of the release notes will be displayed. After browsing this chapter, you can select a new chapter by using the **Chapters** menu.

To select chapters in sequence, use the buttons located at the bottom of the program window.

After you select a product, you can print the release notes for that product by selecting **Print** from the **File** menu.

3-14 Owner's Guides Versus Reference Manuals

Owner's Guides Versus Reference Manuals

- Guides contain general information, overviews, tutorials, and instructional materials
- Reference manuals contain reference pages for commands, function calls, and so on
  - The reference manual material is what **man(1)** accesses
- Example: How to manage disk space
  - First, you need *overview* information found in the *Personal System Administration Guide*
  - Then, get *specific* information about the **swap(1M)** command by using the **man** or **xman** reference pages

IRIX System Administration: System Documentation Revision 2.0 3-14

Student Notes

The contents of a reference manual or man page may contain the following information: how the command or function is called, its syntax and arguments, detailed explanation of what the command does, examples, typical error messages, and bugs.

For example, to learn what swap space is and the specific details about adding and deleting swap space, read the *Personal System Administration Guide*. To use the swap command, you need the information in the manual reference pages, `man 1M swap`.

3-15 What Is the InSight Tool?

What Is the InSight Tool?

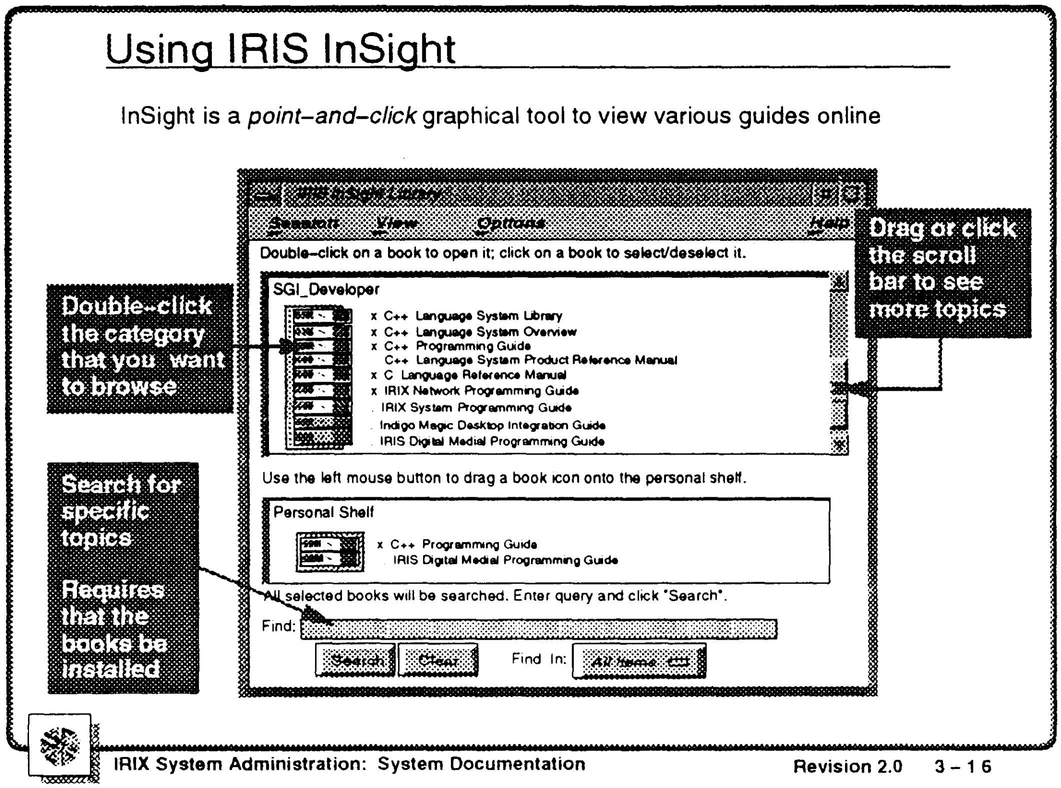
IRIS InSight Viewer

- An online information retrieval system
- Provides an easy-to-use interface that lets you search and browse through online information that is distributed from Silicon Graphics
- In some cases, replaces hardcopy user and administrative guides
  - You can find online guides in the directory **/usr/share/Insight/library/SGI_bookshelves**
- Use the **iiv(1)** or **insight(1)** commands to view or browse the online documentation, or select the **Online Books** from the Toolchest **Help** menu

IRIX System Administration: System Documentation Revision 2.0 3-15

Student Notes

3-16 Using IRIS InSight

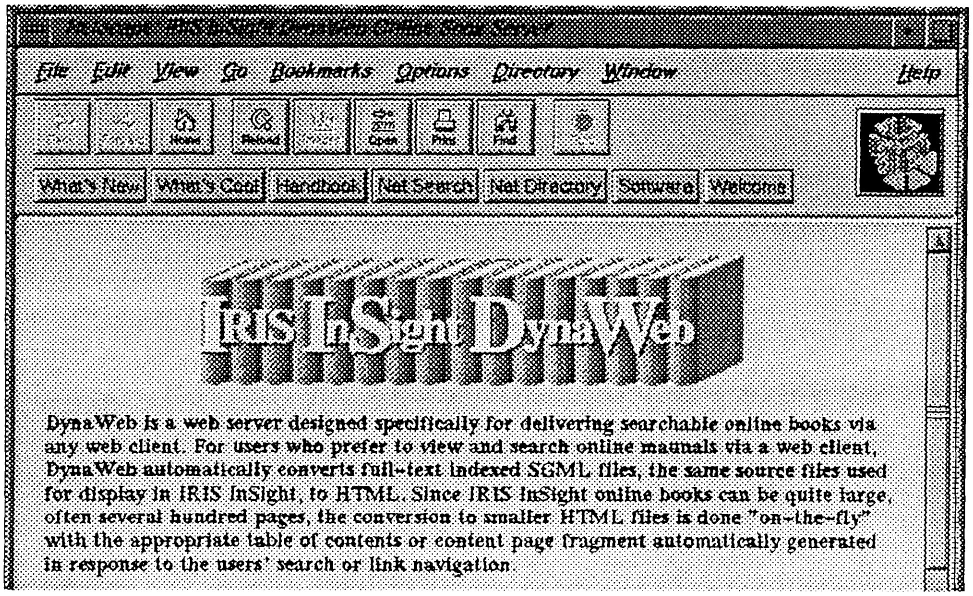


Student Notes

The IRIS InSight Viewer (`insight(1)`) is an online information retrieval system. This viewer provides an easy-to-use interface that allows the user to search and browse through online information that is distributed from Silicon Graphics.

3-17 Using DynaWeb to Access InSight

Using DynaWeb to Access InSight



IRIX System Administration: System Documentation Revision 2.0 3 – 1 7

Student Notes

DynaWeb has two major parts, the server and the SGML-to-HTML converter. The DynaWeb server accepts client requests for InSight books.

3-18 Using DynaWeb to Access InSight (continued)

Using DynaWeb to Access InSight (continued)

- DynaWeb is a tool to view InSight online documentation through a Web browser
- To start the DynaWeb server:

```
# chkconfig dynaweb on
# /etc/init.d/dynaweb start
```

- Default DynaWeb configuration from **http://hostname:88**
- InSight documents about DynaWeb:

  IRIS InSight DynaWeb Administrator's Guide

  IRIS InSight DynaWeb Client User's Guide

IRIX System Administration: System Documentation   Revision 2.0   3 - 1 8

Student Notes

The DynaWeb server can handle up to 256 simultaneous requests on the default port 88. Instead of a DynaWeb HTTP server, you can use an alternate program such as *Mindshare* or *Netscape* for your HTTP server. Then the SGML-to-HTML converter is run by the HTTP server as a CGI script.

The directory `/usr/lib/Insight/dweb/servroot` is the highest point in the DynaWeb tree that any Web client can access. Collections of InSight books are added to the DynaWeb directory structure by linking or copying to the `/usr/lib/Insight/dweb/servroot` directory.

SiliconGraphics
Computer Systems

Module 4: User Support

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module covers adding and supporting users on your SGI system.

4-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Add new user accounts
- Delete old user accounts
- Control the user's environment
- Communicate information to users
- Configure the `crontab(1)` and `at(1)` utilities

IRIX System Administration: User Support Revision 2.0 4 – 2

Student Notes

4-3 User Accounts

User Accounts

Because IRIX is a multiuser system, you need to

- Keep one user's work separate from other users
- Protect the system software from inexperienced users
- Track *who* is doing *what* on a system
- Maintain data security and integrity
- Optimize system resources between users

IRIX System Administration: User Support Revision 2.0 4 – 3

Student Notes

4-4 User Setup Methods

User Setup Methods

- System Manager graphical tool
  /usr/Cadmin/bin/cpeople(1)
- Manual method
- **passmgmt(1M)** command

IRIX System Administration: User Support Revision 2.0 4 – 4

Student Notes

The cpeople graphical tool lets you add, delete, and modify user accounts on the specified host. It allows the creation of privileged users and extends the database of information which describes a login account. The tool interacts with the system's local /etc/passwd and /etc/shadow files, object database, and (optionally) its NIS server.

The passmgmt command updates information in the password files. This command works with both /etc/passwd and /etc/shadow. If there is no /etc/shadow file, any changes made by passmgmt will only go into the /etc/passwd file.

4-5 User Account Management Process

User Account Management Process

Basic Steps

- Identify user account characteristics
- Add the account to **`/etc/passwd`**
- Establish group memberships in **`/etc/group`**
- Create user's login directory
- Set up user configuration files
- Verify file and directory permissions
- Assign a temporary password to the account
- Have user verify setup and change password

IRIX System Administration: User Support Revision 2.0 4 - 5

Student Notes

4-6 User Account Information

User Account Information

- Unique user login name
- Unique user identification number
- Default group identification number
- Unique home directory
- Login shell
- Specialized group memberships
- Configuration files

IRIX System Administration: User Support Revision 2.0 4 – 6

Student Notes

When assigning a unique user identification number, the valid range for UIDs is 100 to 2147483647. UID 0 is reserved for superuser accounts; 1-99 are reserved for other system users.

For historical reasons, certain operations are restricted for UIDs larger than 65535. Most significantly, these users cannot own files on an EFS filesystem. For these reasons, you should only use large UIDs on XFS-based systems.

By default, `cpeople` (the System Manager Tool) assigns the number "20" as a default primary group identification number and as the default location of user home directories, `/usr/people/username`.

Some UNIX SVR4 implementations use `/home` or `/d` as the default location for home directories. You might want to consider placing user home directories in a separate filesystem or on another disk.

The following login shells are provided by IRIX: `/bin/sh` Bourne shell, `/bin/csh` C shell, `/bin/ksh` Korn, and `/bin/tcsh` TC shell.

4-7 /etc/passwd Entries

/etc/passwd Entries

username:passwd:userid:groupid:comment:login dir:login shell

`guest::998:998:Guest Account:/usr/people/guest:/bin/csh`

- **username** must be unique
- **passwd** field is empty until user creates a password
- **userid** number must be unique; username assigned to UID
- **groupid** defines default group membership
- **comment** defines user's full name, phone number
- **Login dir** is home directory
- **Login shell** defines process (typically a shell)

IRIX System Administration: User Support Revision 2.0 4 - 7

Student Notes

The encoded password is produced through a one-way hashing encryption algorithm with variations intended to frustrate the use of hardware implementations of a key search.

The following is an example /etc/passwd entry with an encrypted password:

```
inky:PtFJzXps9Pua.:1000:20:Inky's Acct:/usr/people/inky:/bin/ksh
```

The pwck(1M) command verifies all entries in the /etc/passwd file, including the number of fields, login name, user ID, group ID, and whether the login directory and the program to use exists.

4-8 Securing the Password File Using /etc/shadow

Securing the Password File Using /etc/shadow

- **/etc/passwd** must have read permission for every user
- Passwords are encoded, but encoded values are visible to all users on a system
- With a fast computer, it is possible to "crack" all passwords on a system if you use a "guessing" program

So...

- Use **/sbin/pwconv** to create a more secure password system
  - Shadow password system moves encoded passwords to a nonreadable file, **/etc/shadow**
  - Similar in structure to **/etc/passwd**, but only includes usernames and encoded passwords
  - Encoded password field is replaced by an "**x**" in **/etc/passwd**

IRIX System Administration: User Support Revision 2.0 4 - 8

Student Notes

/etc/shadow is an access-restricted ASCII system file. Unlike the /etc/passwd file, /etc/shadow does not have general read permission. The fields for each user entry are separated by colons, and each user is separated from the next by a newline.

pwconv creates and updates /etc/shadow with information from /etc/passwd, such as the user's login name, password, and password aging information. If password aging information does not exist in /etc/passwd for a given user, none is added to /etc/shadow.

IRIX 6.2 complies with, but is not certified to the C2 security level. It has options for auditing of security-relevant events, additional protection of passwords through shadow passwords, password aging, and expanded login. Password aging and shadow passwords do not function when you use NIS, because there is no /etc/shadow map file. There are ways to get around this restriction to use NIS and shadow password files by using NIS and a local shadow password file.

4-9 Example /etc/passwd & /etc/shadow Files

Example /etc/shadow & /etc/passwd Files

```
21# tail /etc/passwd
nobody:x:60001:60001:SVR4 nobody uid:/dev/null:/dev/null
noaccess:x:60002:60002:uid no access:/dev/null:/dev/null
patrick:x:1110:20:Patrick Ewing,New York Knicks,:/usr/people/patrick:/bin/csh
scottie:x:1111:20:Scottie Pippen,Chicago Bulls,:/usr/people/scottie:/bin/csh
anthony:x:1112:20:Anthony Mason,,:/usr/people/anthony:/bin/csh
charles:x:1113:20:Sir Charles Barkley,Phoenix
     Suns,:/usr/people/charles:/bin/csh
chris:x:1114:20:Chris Mullin,Golden State
     Warriors,:/usr/people/chris:/bin/csh
ed:x:1115:20:Ed McCracken,SGI,, Bldg 6,415/390-XXXX:/usr/people/ed:/bin/csh
kelly:x:1116:20::/usr/people/kelly:/bin/csh
+mars::0:0:::

22# tail /etc/shadow
4Dgifts::8552::::::
nobody:*:8552::::::
noaccess:*:8552::::::
patrick::::::::
scottie::::::::
anthony::::::::
charles:Hd18QzXjOW5AI:8552::::::
chris:Bbrg8gcufIoCI:8552::::::
ed:VYo81kJyWRns6:::::::
kelly:VdpdOsKjUUoEw:8552::::::
```

IRIX System Administration: User Support   Revision 2.0   4 - 9

Student Notes

Field	Meaning
username	The user's login name
password	A 13-character encrypted password for the user, a lock string to indicate that the login is not accessible, or no string to show that there is no password for the login
lastchanged	The number of days between January 1, 1970, and the date that the password was last modified
minimum	Minimum number of days required between password changes
maximum	Maximum number of days the password is valid
warn	Number of days before password expires that the user is warned
inactive	The number of days of inactivity allowed for that user
expire	Absolute date specifying when the login may no longer be used
flag	Reserved for future use, set to zero (currently not used)

4-10 Group Account Database: /etc/group

Group Account Database: /etc/group

Format: group name:passwd:GID:member list

- Unique familiar group name
  - Typically, use organizational department or function names
- Encoded password
- Unique **Group ID** number
  - 20 is for default group ID
- List of additional users and groups who have access to group–owned files

  Example: /etc/group

  ```
  adm::3:root,adm,daemon
  payroll::609:ripley,suzanne,dmills
  ```

IRIX System Administration: User Support Revision 2.0 4 – 1 0

Student Notes

The /etc/group field assigns names to group IDs. The user's default group is defined in /etc/passwd. Users can change group affiliation to groups they are members of in /etc/group using newgrp(1) or multgrps(1).

The newgrp(1) command changes a user's group identification and creates a child shell. Type **exit** to return to the original shell.

The multgrps(1) command spawns a shell with membership in multiple groups.

The grpck(1M) command verifies all entries in the group file. This verification includes a check of the following: number of fields, group name, group ID, whether all login names appear in the password file, duplicate logname entries, and maximum number of groups per logname.

4-11 Using the `passmgmt` Tool

Using the `passmgmt` Tool

- **`passmgmt(1M)`** creates **`/etc/passwd`** entries but does not create a user's home directory
- Format

  `# passmgmt -a|m [options] username`

- Example

  `# passmgmt -a -g 20 -s /bin/csh jenny`

  - Use the **`-m`** option to modify the existing entry
  - Change jenny's username to *jennyh*:

    `# passmgmt -m -l jennyh jenny`

IRIX System Administration: User Support Revision 2.0 4 – 1 1

Student Notes

The `passmgmt(1M)` command creates entries in `/etc/passwd`, but it does not create home directory or shell configuration files.

The defaults of `passmgmt` include the following:

- Empty comment field
- Home directory of `/usr/people/`*`name`*
- UID is next available unique number greater than 99
- GID is 1 (this is probably not what group you want for users)
- Shell field is blank (system makes it `/bin/sh`)

Use options to override defaults for home directory, comment field, login shell, and so on.

4-12 Completing User Setup

Completing User Setup

- Create home directory

```
# mkdir /usr/people/username
```

- Create shell configuration files

```
# cp /etc/stdlogin ~username/.login
# cp /etc/stdcshrc ~username/.cshrc
# cp /etc/stdprofile ~username/.profile
```

- Give the user other configuration files, as desired, for mail, X Windows, etc.

IRIX System Administration: User Support Revision 2.0 4 - 1 2

Student Notes

It is a good idea to place the users' home directories in a separate filesystem.

The `/etc/stdlogin` file is the standard login configuration file. Users can modify their own `.login` files to make changes. This file is used by the C and T shells.

The `/etc/stdcshrc` file is the standard `csh` initialization command file. Users can modify their own `.cshrc` to make additional changes. This file is used by the C and T shells. If a `.tcshrc` file exists, T shell uses it; otherwise, it uses the `.cshrc` file.

The `/etc/stdprofile` file is the standard login configuration file. The file `$HOME/.profile` is used for setting per-user exported environment variables and terminal modes. This file is used by Bourne and Korn shells.

4-13 Using the System Manager to Manage Accounts

Using the System Manager to Manage Accounts

Benefits

- Easier and more consistent than manual methods
- Handles NIS accounts consistently
- Automatically assigns UIDs

Disadvantages

- Time consuming if many users need to be added to system
- Does not work on ASCII terminals

IRIX System Administration: User Support Revision 2.0 4 – 1 3

Student Notes

4-14 User Configuration Files

User Configuration Files

- Shell configuration files
  - .cshrc
  - .login
  - .profile
  - .history (or .sh_history for ksh)
  - .alias
- X11 configuration files
  - .Xdefaults
  - .Xresources
- Motif and 4DWm configuration files
  - .auxchestrc
  - .4Dwmrc
- Application configuration files
  - .mailrc
  - .exrc

IRIX System Administration: User Support Revision 2.0 4 – 1 4

Student Notes

File	Used for
.cshrc	csh and tcsh configuration file
.login	Login configuration file for terminal type and environment
.profile	User exported environment variables and terminal modes, for sh and ksh
.history	History list of user commands
.alias	List of aliases that modify commands
.Xdefaults	Resource settings to customize all X applications at startup
.Xresources	Resource settings to customize X applications
.auxchestrc	Specifications for the toolchest utility menu
.4Dwmrc	Menu, key binding, and button binding definitions
.mailrc	Mail aliases

4-15 Verify File and Directory Permissions

Verify File and Directory Permissions

- User's home directory and configuration files should be owned by the user
- Change ownership and group of user's files:

```
# chown -R username ~username

# chgrp -R defaultgrpname ~username   OR

# chown -R fizzy.payroll ~fizzy
```

```
# ls -la /usr/people/fizzy
total 9
drwxr-xr-x    3 fizzy  payroll    512 Jun 12 13:12 .
drwxr-xr-x    8 root   sys        512 Jun 12 13:12 ..
-rw-r--r--    1 fizzy  payroll    533 Jun 12 13:12 .cshrc
-rw-r--r--    1 fizzy  payroll    597 Jun 12 13:12 .login
-rw-r--r--    1 fizzy  payroll    722 Jun 12 13:12 .profile
drwxr-xr-x    2 fizzy  payroll    512 Jun 12 13:12 .workspace
```

IRIX System Administration: User Support Revision 2.0 4 – 1 5

Student Notes

You might want to explain the meaning of permissions to your end users to prevent problems.

Remember, *read* permission for *other* means anyone can read a file; *write* permission on a directory for *other* means that anyone can modify or delete a file.

4-16 Assigning a Password

Assigning a Password

- Use the **passwd(1)** program to change local passwords
  - Format

    passwd [name]

```
# /bin/passwd fizzy
New password:
Re-enter new password:
# grep fizzy /etc/passwd
fizzy:x:6135:20:Fizzy Fogg:/usr/people/fizzy:/bin/tcsh
```

- The first time users log in, they should be forced to change their password

```
# /bin/passwd -f username
```

IRIX System Administration: User Support Revision 2.0 4 - 16

Student Notes

The following guidelines apply to passwords:

- Must be at least six characters in length (only the first eight are significant)
- Must contain at least two alphabetic characters (uppercase or lowercase letters) and at least one numeric or special character
- Must differ from the user's login name and any reverse or circular shift of that login name (an uppercase or lowercase letter is considered the same value)
- *New* passwords must differ by at least three characters from the old password

4-17 Customizing a User's Login Icon

Customizing a User's Login Icon

- You can place customized user icons in three locations:
  - **/usr/local/lib/faces/$USER**
  - **/usr/lib/faces/$USER**
  - **$HOME/.icons/login.icon**
- You can create the icons from screen images using the **snapshot(6D)** program
  - Icons more than 100 x 100 pixels overlap others
  - Use **imgworks(1)** to get the size of the image and to reduce or expand it if necessary

Student Notes

The clogin command checks for the files $HOME/.icons/login.icon, /usr/local/lib/faces/$USER, and /usr/lib/faces/$USER for login images. $HOME represents the user's home directory, and $USER represents the user's login name.

The directory $HOME/.icons does not exist by default. You must create it before placing the login icon image there.

```
% mkdir $HOME/.icons
% cp /tmp/cat.rgb $HOME/.icons/login.icon
```

If more than one login icon exists for a user, the one in $HOME/.icons/login.icon is displayed.

4-18 Customize Visual Login Screen & User Manager

Customize Visual Login Screen & User Manager

- Customize the accounts that have icons displayed by **clogin(1)** in the file **/var/Cadmin/clogin.conf**
  - For example, add the following line to **/var/Cadmin/clogin.conf**:

    `burt:noicon`
  - **Burt's** icon will not show on the **clogin** screen
- Customize accounts that have icons displayed by **cpeople(1)**-graphical User Manager in the file **/var/Cadmin/cpeople.conf**
  - For example, add the following line to **/var/Cadmin/cpeople.conf**:

    `rsh:noshow`
  - **rsh**'s icon will not show on the **cpeople** screen

IRIX System Administration: User Support Revision 2.0 4 – 1 8

Student Notes

The `configClogin` command is used to customize the graphical login window. It can be run from the System Manager, `chost`.

You can use the `configClogin` command to specify the following:

- An account that should be logged into the workstation automatically when it is booted
- The size of the `clogin` window
- Whether to show icons or pictures representing login accounts
- Which login accounts to hide

Only a privileged user or administrator can run `configClogin` (Login setup—System Manager Tool).

4-19 Customizing a User's Login

Customizing a User's Login

By default, users log into the graphics terminal, using **clogin(1)**, the visual login that displays icons for users

- Turn off **clogin** to display the standard Xdm login:

  `# chkconfig visuallogin off`

- Replace individual user icons with a single image:

  `# chkconfig noiconlogin on`

- Turn the graphics monitor into an ASCII terminal:

  `# chkconfig windowsystem off`

IRIX System Administration: User Support Revision 2.0 4-19

Student Notes

If you turn the graphics monitor into the equivalent of an ASCII terminal, you need to set the variable TERM to `iris-tp` for `vi` to work properly.

`setenv TERM iris-tp`

When a system has a large number of user accounts, it may be faster to not display user icons in `clogin`. By default, the system replaces the icons with the image found in `/usr/Cadmin/images/cloginlogo.rgb` when `noiconlogin` is set to "on."

4-20 Deleting User Accounts – Manual Method

Deleting User Accounts – Manual Method

- Lock the account by inserting an asterisk (*) in the password field of the **/etc/passwd** or **/etc/shadow** file
- Back up the user's data files and directories
- Remove the user's data files and directories
- Remove the user's mailbox from **/var/mail/username**
- Remove the user's username from group lists
  - Local mail aliases — **/etc/aliases**
  - Local group lists — **/etc/group**

IRIX System Administration: User Support Revision 2.0 4 – 2 0

Student Notes

4-21 Using Tools to Delete Users

Using Tools to Delete Users

- System Manager
  - You might want to back up the user's files first
  - Use the Delete button of the User Tool
  - The system asks if you want to delete the user's files
- **passmgmt(1M)**
  - Use the **-d** option
  - Delete the user *sonia*:

```
# passmgmt -d sonia
```

IRIX System Administration: User Support Revision 2.0 4 – 2 1

Student Notes

4-22 Tools for Communicating Information to Users

Tools for Communicating Information to Users

- **/etc/motd** displays the **m**essage **o**f **t**he **d**ay file when the user brings up a command shell
  - **/etc/cshrc** or **/etc/profile** uses the **cat** command to display the contents of the **/etc/motd** file to the user's screen
- **/etc/issue** displays the message for network and terminal logins with the login prompt before the user logs in
- **/etc/wall** displays the message to all users who are logged in
- E-mail must be read frequently to be useful

IRIX System Administration: User Support Revision 2.0 4 – 22

Student Notes

The /etc/motd file displays every time a user logs in or opens a new window. Reserve this for important (and short) messages. The /etc/issue file displays when you use telnet or rlogin commands.

The /etc/cshrc and /etc/profile files are the system-wide setup files for csh/tcsh and sh/ksh.

4-23 Process Scheduler `cron`

Process Scheduler **cron**

- Allows users and administrators to execute jobs at specified intervals
- Facility is controlled by the system administrator
  - Use files **/etc/cron.d/cron.allow** and **/etc/cron.d/cron.deny**
- System daemon **/sbin/cron**
- When system boots, **cron** makes its in–memory file
- **cron** updates its in–memory file whenever
  - Users change their **cron** jobs (with **crontab** command)
  - Users issue the **at** command
- **cron** activity is recorded in **/var/cron/log**

IRIX System Administration: User Support Revision 2.0 4 – 2 3

Student Notes

`cron` executes commands at specified dates and times. You can specify regularly scheduled commands according to instructions found in `crontab` files in the directory `/var/spool/cron/crontabs`. Users can submit their own `crontab` file with the `crontab(1)` command. Commands which are to be executed only once may be submitted with the `at(1)` command.

Who Has Access	**cron.deny**	**cron.allow**
Only superuser	No file	No file
Selected deny	User names	No file
Selected allow	No file	User names
All users	Empty file	No file

4-24 `cron` Format

`cron` Format

minutes hour date month week program/command

Example `crontab` file:

Minute	Hour	Day of Month	Month	Day of Week	Event
15	2	*	*	*	`/bin/calendar`
30	8 – 17	*	*	*	`/usr/local/bin/prog1`
17	5	*	*	0,6	`/usr/local/bin/prog2`
0	23	31	2	*	`/bin/rm -r /`
0,15,30,45	*	*	*	1-5	`/usr/local/bin/progA`

IRIX System Administration: User Support Revision 2.0 4 – 2 4

Student Notes

A `crontab` file consists of lines of six fields each. The fields are separated by spaces or tabs. The first five fields are integer patterns that specify the following:

- Minute (0-59)
- Hour (0-23)
- Day of the month (1-31)
- Month of the year (1-12)
- Day of the week (0-6 with 0=Sunday)

The sixth field is the program or command to execute at the above specific time.

4-25 Using cron

Using cron

crontab(1) command

- This example uses **cron.file** to create **crontab** file with the user's name

```
mary% crontab cron.file
# ls /var/spool/cron/crontabs
adm mary rfindd root sys uucp
```

- Removes the user's **crontab** file

```
% crontab -r
```

- Lists contents of the user's **crontab** file

```
% crontab -l
```

IRIX System Administration: User Support Revision 2.0 4 - 25

Student Notes

The `cron(1M)` command executes commands using the Bourne shell. Output from `cron(1M)` jobs is mailed unless redirected.

The `crontab -l > new.cron.file` command takes the current `cron` job and creates a copy that you can edit and then resubmit with the command `crontab new.cron.file`.

Root has a `cron` file by default. You can either edit it directly and then kill and restart the `cron` daemon, or use the following command to edit that file and resubmit it:

```
crontab -l > /roots.cron.file
```

4-26 Process Scheduler `at`

Process Scheduler `at`

`at(1)` command

- Jobs are executed by **`cron`** at a specified time
- Facility is controlled by the system administrator

  `/etc/cron.d/at.allow`

  `/etc/cron.d/at.deny`
- Authorized users can add entries with the **`at`** command
- Format

  `at time [date] [+ increment] < command.script`
- Example

  `% at 2:00pm Friday < cmd.script`

IRIX System Administration: User Support　　Revision 2.0　　4 – 2 6

Student Notes

The `at(1)` command handles jobs like `cron(1M)`.

More examples:

```
$ at now < prog1
$ at 9am < cmd2.script > /tmp/out.at 2> /tmp/errs.at
```

All output messages go to the file `/tmp/out.at`, and all error messages go to the file `/tmp/errs.at` when the `cmd2.script` program runs at 9:00 a.m.

```
$ at 1900 thursday next week /etc/shutdown -y -p
```

SiliconGraphics
Computer Systems

Module 5: System Monitoring

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

In the module you will learn several tools for monitoring what is running on the system and a few performance parameters.

5-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Monitor important aspects of system utilization
- Define the term *process*
- Utilize **`syslogd(1M)`** to record system error messages and other useful information

Student Notes

Important aspects of system utilization include disk usage, number of users on the system, and processes running on the system.

5-3 System Monitoring – Disk Usage

System Monitoring – Disk Usage

- Use these commands to monitor disk usage
  - **du -s**
  - **du -s ***
  - **du -ks**
  - **df [-k]**
- Use **find(1)** to find inactive and large files

```
# find / -type f -mtime +60 -print| mail root &
# find / -size +10000 -print | mail root &
```

IRIX System Administration: System Monitoring Revision 2.0 5 – 3

Student Notes

The du(1M) command summarizes disk usage, reporting the number of blocks contained in all files and directories specified. If no names are given, the current directory is used. The du -s command specifies only the grand total for the specified name. The -k flag causes du to express all block counts in terms of 1024-byte blocks, instead of the default 512-byte blocks.

The find(1) command recursively descends the directory hierarchy for each pathname in the path-name-list (one or more path names) seeking files that match a Boolean expression.

The `find / -type f -mtime +60 -print| mail root &` command finds all files from the root directory and below that have *not* been modified or changed in the last *60 days* and mails that list of files to the root user.

The `find / -size +10000 -print | mail root &` command finds all files with a *size* greater than *10000 blocks* (512 bytes/block) and mails that list of files to the root user.

5-4 System Monitoring – Disk Usage (continued)

System Monitoring – Disk Usage (continued)

- Monitor system log files and accounting files for size, because they all grow

  /var/cron/log

  /var/adm/SYSLOG

  /var/adm/sulog

- Periodically look in **/var/adm/crash** for kernel core dumps (very large)

IRIX System Administration: System Monitoring Revision 2.0 5 – 4

Student Notes

The /var/cron/log file contains a history of all the actions taken by cron. Check this file periodically for excessive size, and reduce it if necessary.

The /var/adm/SYSLOG file contains system and daemon error messages.

The /var/adm/sulog file contains a history of su command usage. Check this file periodically for excessive size, and archive it.

The /usr/etc/savecore file saves any core dumps to the /var/adm/crash directory.

5-5 User Monitoring Tools: var/adm/sulog

User Monitoring Tools: /var/adm/sulog

Monitor the **su** log for successful (+) versus unsuccessful (-) **su**'s

```
# tail /var/adm/sulog
SU 05/12 12:53 + ttyq2 root-jls
SU 05/12 12:53 - ttyq2 jls-sconnor
SU 05/12 12:53 - ttyq2 jls-sconnor
SU 05/12 12:53 + ttyq2 jls-sconnor
SU 05/12 12:54 + ttyq2 sconnor-sconnor
SU 05/12 12:54 + ttyq2 sconnor-mars
SU 05/12 12:55 + ttyq2 sconnor-kramer
SU 05/12 12:55 + ttyq2 sconnor-root
SU 05/12 12:55 - ttyq2 sconnor-sysadm
SU 05/12 12:56 + ttyq2 adamg-root
```

Note: Must have **root** or **adm** permission to look at the **sulog**

```
# ls -l /var/adm/sulog
-rw------- 1 adm adm 641 May 12 12:56 /var/adm/sulog
```

IRIX System Administration: System Monitoring Revision 2.0 5 – 5

Student Notes

The su(1M) command allows you to become another user without logging off. The default user name is root (that is, *superuser*).

The su command attempts to write an entry each time you invoke it, regardless of outcome, to the file specified by the variable SULOG defined in the /etc/default/su file. The default file to write to is /var/adm/sulog.

The /var/adm/sulog file contains a history of su command usage. Check this file periodically for excessive size, and archive it.

5-6 User Monitoring Tools: who

User Monitoring Tools: who

who(1) displays who is on the system

```
% who
root        ttyd1       May 11 08:48
john        ttyq0       May 11 07:11
john        ttyq1       May 11 07:17
jim         ttyq2       May 11 08:00
jim         ttyq3       May 11 08:01
cynthia     ttyq4       May 11 18:14
john        ttyq5       May 12 10:32
mary        ttyq6       May 11 14:08
ian         ttyq8       May 12 08:37
beth        ttyq10      May 11 14:33
cynthia     ttyq7       May 11 14:05
jim         ttyq9       May 12 10:17
mike        ttyq11      May 11 18:10
nancy       ttyq12      May 12 09:56
mike        ttyq13      May 11 19:06
dave        ttyq14      May 11 19:30
guest       ttyq17      May 12 10:59
mary        ttyq18      May 12 11:03
```

IRIX System Administration: System Monitoring Revision 2.0 5 – 6

Student Notes

The who(1) command lists the user's name, terminal line, login time, elapsed time since activity occurred on the line, and the process ID of the command interpreter (shell) for each current UNIX system user. It examines the file /var/adm/utmp to obtain its information.

The who am i or who am I commands identify the *invoking* user.

The whoami command displays the effective current user name.

5-7 User Monitoring Tools: w

User Monitoring Tools: w

`w(1)` displays who is on the system and what they are doing

```
% w

 10:59am  up 1 day, 21:53,  17 users,  load average: 0.26, 1.18, 2.08
User     tty from            login@  idle   JCPU   PCPU  what
root     d1                  Tue 8am  1:25   8:42   2:02  nsrjbd
john     q0  192.221.179.1   Tue 7am    34  78:27     19  vi fcntl.c
jim      q2  patton.wpd.sgi  Tue 8am    17  95:39     18  -ksh
jim      q3  patton.wpd.sgi  Tue 8am    11   4:21     49  telnet
```

IRIX System Administration: System Monitoring Revision 2.0 5 – 7

Student Notes

The command `whodo(1M)` displays who is doing what on the system, which is similar to the `w(1M)` command.

5-8 User Monitoring Tools: `last`

User Monitoring Tools: `last`

`last [name]` indicates last logins of users

```
% last guest
guest ttyq5   :0.0   Sat Dec 24 16:49   still logged in
guest ttyq4   :0.0   Sat Dec 24 16:09   still logged in
guest ttyq3   :0.0   Sat Dec 24 12:17 - 15:16 (02:58)
guest ttyq2   :0.0   Sat Dec 24 12:17   still logged in
guest ttyq1   :0.0   Sat Dec 24 12:17   still logged in
guest ttyq0   :0.0   Sat Dec 24 12:17   still logged in
guest ttyq4          Fri Dec 23 12:23 - 23:25 (11:02)
```

`last [tty..]` indicates last logins of terminals

```
% last ttyq2
sara   ttyq2  :0.0   Thu Dec 22 10:53   still logged in
sara   ttyq2  :0.0   Wed Dec 21 15:09 - 10:53  (19:43)
sara   ttyq2  :0.0   Wed Dec 21 14:30 - 14:40  (00:10)
sara   ttyq2  :0.0   Mon Dec 19 08:58 - 14:19 (2+05:20)
```

IRIX System Administration: System Monitoring Revision 2.0 5 - 8

Student Notes

The `last(1)` command looks in the `/var/adm/wtmp` file, which records all logins and logouts for information about a user, a terminal, or any group of users and terminals. Arguments specify names of users or terminals of interest. Names of terminals may be given fully or abbreviated. For example "`last d1`" is the same as "`last ttyd1`." If multiple arguments are given, the information (which applies to any of the arguments) is printed. For example, `last root console` lists all of `root`'s sessions and all sessions on the console terminal.

5-9 Types of Processes

Types of Processes

- Process: An instance of a running program
- Interactive processes
  - Processes associated with a login, terminal, or window session
  - Examples: **jot**, **showcase**, **vi**, **csh**
- Batch processes
  - Processes not associated with a specific login, but which are submitted from a queue
  - Examples: **cron**, **batch**, **at**
- Daemons
  - System processes initiated at boot time that wait in the background until an active process requests their service
    - Not associated with a particular user or login
    - Perform system tasks on a periodic basis and then go to sleep
    - Examples: **inetd**, **lpsched**, **biod**, **rpc.lockd**

IRIX System Administration: System Monitoring Revision 2.0 5 – 9

Student Notes

IRIX is a multitasking operating system. This means that your system can perform many operations simultaneously. For example, you can run an application, print a file, and read your mail at the same time, without waiting for each operation to complete before beginning the next.

Every program that you run on the system is assigned a process ID that allows IRIX to keep track of what it is doing. By identifying the process numbers assigned to tasks, you can manage your operations more effectively.

5-10 Life (and Death) of Processes

Life (and Death) of Processes

- Process genealogy
  - A process that creates or *spawns* a new process is called the *parent*
  - The new process is called the *child* process
- Process generation
  - The current process *forks* a child process and then goes to *sleep*
  - The new process runs the same program as the parent, but is assigned a new PID
  - The child process then *execs* the new program, which maintains the same PID as the child
  - When the new program is completed, the child process *dies*, and the parent process *wakes up*

IRIX System Administration: System Monitoring Revision 2.0 5 – 1 0

Student Notes

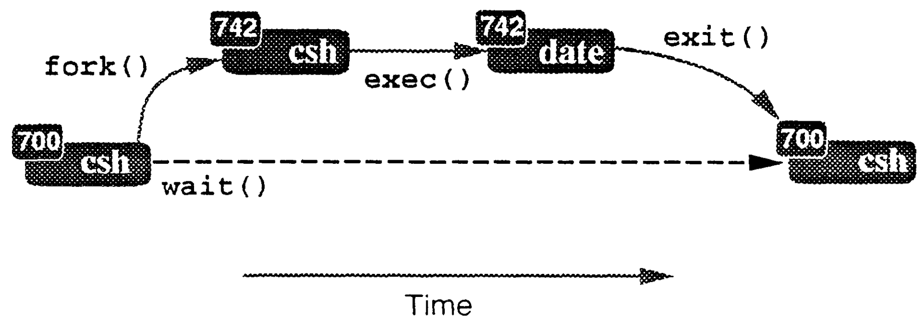
IRIX processes are tasks that IRIX does to keep the system running correctly or to complete an explicit command. Each process has a unique process ID number.

The `fork` command creates a new process. The new process (child process) is an exact copy of the calling process (parent process). This means that the child process inherits several attributes from the parent process.

The `sleep` command suspends the current process from execution.

5-11 Life (and Death) of Processes (continued)

Life (and Death) of Processes (continued)



IRIX System Administration: System Monitoring Revision 2.0 5 - 1 1

Student Notes

5-12 Special Cases

Special Cases

- Orphan processes
  - *Orphan* process is still active after the *parent* process is *terminated*
  - Orphan processes are *always* inherited by **`init`** (PID 1)
- Zombie processes
  - A process that has terminated, but has not been removed from the process table, because the parent did not **`wait()`** for it
  - *zombies* remain in the process table, occupying a slot; therefore, *zombies* may interfere with new process creation
- Process table cleanup
  - Terminated processes removed from the process table by **`wait()`**
  - *Zombies* normally removed from the process table when parent dies
  - If *zombies* are not removed when the parent dies, they are adopted by **`init`**, which then removes them from the process table

IRIX System Administration: System Monitoring Revision 2.0 5 – 1 2

Student Notes

Zombie processes go away after the parent has died or after `init` has handled the zombie process.

Background processes sometimes become zombies.

5-13 Process Table

Process Table

- What is it?
  - A table (of a fixed size) of all processes maintained by the kernel
  - Make changes with the **systune(1M)** command
- How is it used?
  - Every time a new process is created, it is assigned a unique process identification number (PID), and an entry is dedicated in the process table
- What kind of information is stored in the process table?
  - Process identification
  - Scheduling
  - Resource utilization

IRIX System Administration: System Monitoring Revision 2.0 5 - 1 3

Student Notes

The `systune` tool lets you examine and configure your tunable kernel parameters. `systune` adjusts some parameters in real time and informs you if you need to reboot your system after reconfiguration. It saves the reconfigured kernel in `/unix.install`, unless you use the `f` option. Use `systune` to change the size of the process table.

5-14 System Monitoring Tools: ps

System Monitoring Tools: ps

ps(1) lets you see a "snapshot" of your process table

- Key options to ps

-e	Every process running on the system
-l	Long listing
-f	Full listing
-u	List for a specific user only
-c	Scheduler properties or class

IRIX System Administration: System Monitoring Revision 2.0 5-14

Student Notes

The ps command prints certain information about active processes. Without options, information is printed about processes associated with the controlling terminal. The output consists of a short listing containing only the process ID, terminal identifier, cumulative execution time, and the command name. Otherwise, the information that is displayed is controlled by the selection of options.

Under the -f option of the ps command, ps tries to determine the command name and arguments given when the process was created by examining the user block.

5-15 `ps -ef` Example

`ps -ef` Example

```
UID   PID   PPID  C   STIME   TTY  TIME COMD
root  0      0    0   Dec 22  ?    0:01 sched
root  1      0    0   Dec 22  ?    0:07 /etc/init
root  2      0    0   Dec 22  ?    0:00 vhand
root  3      0    0   Dec 22  ?    0:27 bdflush
root  4      0    0   Dec 22  ?    0:27 vfs_sync
root  10     0    0   Dec 22  ?    0:00 pdflush
root  294    1    0   Dec 23  ?    0:00 /usr/etc/biod 4
root  81     1    0   Dec 22  ?    0:01 /usr/etc/syslogd
root  270    1    0   Dec 23  ?    0:00 /usr/etc/ypbind -ypsetme
root  311    1    0   Dec 23  ?    0:01 /usr/etc/inetd
root  287    285  0   Dec 23  ?    0:00 /usr/etc/nfsd 4
root  291    1    0   Dec 23  ?    0:00 /usr/etc/biod 4
root  302    1    0   Dec 23  ?    0:00 /usr/etc/rpc.statd
root  450    1    0   Dec 23  ?    4:01 /usr/bin/X11/xlock +nolock
root  310    1    0   Dec 23  ?    0:00 /usr/etc/rpc.lockd
root  240    1    0   Dec 22  ?    0:19 /usr/etc/xntpd
root  718    1    0   Dec 22  ?    0:00 /usr/etc/appletalk/epd
root  657    1    0   Dec 22  ?    0:16 /usr/Cadmin/bin/objectserver
root  519    1    0   Dec 22  ?    0:04 /sbin/cron
root  692    657  0   Dec 22  ?    0:10 /usr/Cadmin/bin/objectserver
root  693    674  0   Dec 22  ?    0:00 /usr/bin/X11/xdm
root  674    1    0   Dec 22  ?    0:00 /usr/bin/X11/xdm
root  667    1    0   Dec 22  ?    7:20 famd -t 6
lp    491    1    0   Dec 22  ?    0:00 /usr/lib/lpsched
root  510 1  0   Dec 22  ?   0:01 /usr/lib/sendmail -bd -q15m
```

IRIX System Administration: System Monitoring   Revision 2.0   5 – 1 5

Student Notes

The headings above the `ps -lfu` command have the following meanings:

Heading	Meaning
`F`	Flags (hexadecimal and additive) associated with the process
`S`	The state of the process
`UID`	The user ID number of the process owner
`PID`	The process ID of the process (this datum is necessary in order to kill a process)
`PPID`	The process ID of the parent process
`C`	Processor utilization for scheduling
`NI`	Nice value, used in priority computation
`P`	Displays the CPU number on which the process is executing (an asterisk otherwise)
`SZ`	Total size in pages of the process

RSS	Total resident size (in pages) of process
WCHAN	The address of an event for which the process is sleeping (if blank, the process is running)
STIME	The starting time of the process, given in hours, minutes, and seconds
TTY	The controlling terminal for the process (a ? is printed when no controlling terminal)
TIME	The cumulative execution time for the process
Comd	The command name (the full command name and its arguments printed with the -f option)

5-16 Controlling Runaway Processes

Controlling Runaway Processes

- Get the PID of the process with **ps(1)**, and use the **kill(1)** command:

  ```
  # kill <PID>
  ```

- If the process does not die, it might be ignoring the default signal
  - Send ***signal*** *9* or **KILL**, which cannot be ignored:

    ```
    # kill -9 <PID>
    # kill -KILL <PID>
    ```

- Use **killall(1M)** with the name of the process to kill *all* processes having that name:

  ```
  # killall prog1
  ```

IRIX System Administration: System Monitoring Revision 2.0 5 - 16

Student Notes

The `killall` command sends a signal to a set of processes specified either by name, process group, or process ID. It is similar to `kill(1)`, except that it allows processes to be specified by name and has special options used by `/etc/shutdown`.

Be careful when using `killall`. You might kill more than you want, such as *all* `csh`'s on the system.

When no processes are specified, `killall` terminates all processes that are not in the same process group as the caller. This form is used for shutting down the system and is only available to the superuser.

`man 5 signal` reports all the signals, with their numeric and mnemonic names.

5-17 System Monitoring Tools: top and gr_top

System Monitoring Tools: top and gr_top

top(1) and gr_top(1) let you see a sorted list of the top CPU utilization processes updated at a specified interval

- top For terminals or text windows
- gr_top For graphical displays only

```
IRIX wizard 6.2 01181722 IP22 Load[0.75,0.34,0.13] 14:02:52   38 procs

   user   pid  pgrp  %cpu proc  pri  size   rss   time  command
  guest  8981  8980 69.63    *   80  2804  1607   0:03  showcase
   root  5199  5199  3.04    *   60  2940  1561   2:43  Xsgi
  guest  5384  5367  1.40    *   26  1490   786   0:02  4Dwm
  guest  6424  6424  1.09    *   26   757   218   0:03  xwsh
  guest  8820  8820  0.45    0   60   402    98   0:00  top
   root     4     0  0.11    *  +39     0     0   0:01  vfs_sync
  guest  5435  5424  0.03    *   26  3973  2920   5:16  maker4X.exe
   root     3     0  0.03    *  +39     0     0   0:02  bdflush
```

IRIX System Administration: System Monitoring Revision 2.0 5 – 1 7

Student Notes

The first header gives the machine name, release and build date information, processor type, 1-, 5-, and 15-minute load average, current time, and the number of active processes. The second header line contains the following:

user	Username
pid	Process ID
pgrp	Process group ID
%cpu	CPU usage
proc	Processor number running process; * if not currently executing
pri	Process priority
size	Process size in pages
rss	Resident set size (amount of a program currently in RAM)
time	Amount of CPU time used by the process
command	Process name

5-18 Tools for System Performance: `gr_osview`

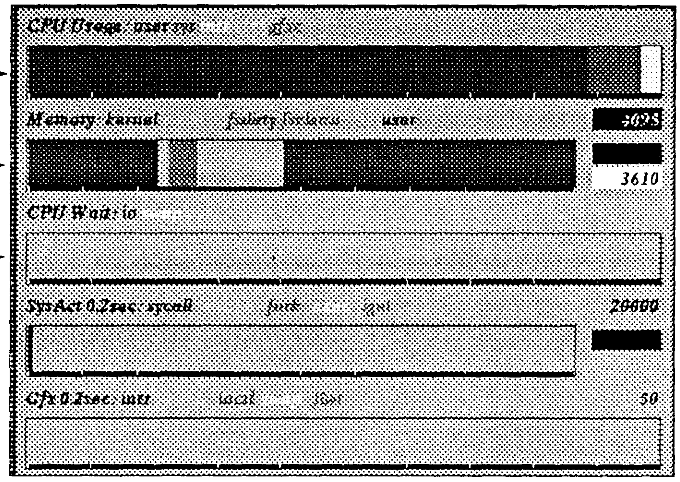
Tools for System Performance: `gr_osview`

- Graphical tool to see system activity
- Example: `gr_osview -a`

What is using the CPU?

What is using the memory?

Is the CPU idle waiting for a resource?



IRIX System Administration: System Monitoring   Revision 2.0   5 - 1 8

Student Notes

The `gr_osview` tool provides a graphical display of usage of certain types of system resources. This display provides a real-time window into the overall operation of the system.

`gr_osview` utilizes various formats such as bar, strip chart, and digital readout.

It can be customized to show different aspects using the `~/.grosview` file.

The `osview(1)` tool gives a textual output similar to `gr_osview(1)`, but is usable on ASCII terminals or across the network.

5-19 Manipulating Scheduling Priorities: `npri`, `nice`

Manipulating Scheduling Priorities: `npri`, `nice`

All processes are assigned priority values by the kernel

- Priority values
  - Range from 0 to 254
  - Are banded into subranges
  - Have meaning relative to priorities of other processes
  - May change over time (degrade or upgrade)
- You can change the priority of a process with `npri`, `nice`

Command	Type of Change
`npri`	Absolute or relative, set nondegrading, time slice, deadline scheduling
`nice`	Relative modifier, regular users can lower their priority

IRIX System Administration: System Monitoring Revision 2.0 5 – 1 9

Student Notes

The following are the scheduling (run) queues:

```
0[os]       Kernel(  0, 29)
1[rt]    Real-time( 30, 39)
2[dl]     Deadline( 40,255)
3[gn]         Gang( 40,127)
4[ts]   Time-share( 40,127)
5[bg]   Gang-batch(128,255)
6[bt]        Batch(128,255)
```

The run queue (or class names) in brackets is what the `ps -c` command shows. See the `pset(1M)` command for more information.

5-20 Being `nice`

Being **nice**

- **nice** changes the priority of a time–sharing process
- **nice** adds a value to the **nice** value of your process giving a lower scheduling priority
- Syntax

```
/sbin/nice [ -increment ] command [ arguments ]

$ /usr/demos/bin/atlantis&

$ /sbin/nice -10 /usr/demos/bin/atlantis&

#ps -el | grep atlantis
30 R    1000  9332  1 46  87 20 *  889:212    pts/2   0:16 atlantis
30 R    1000  9341  1 37  92 30 *  889:312    pts/2   0:01 atlantis
```

IRIX System Administration: System Monitoring   Revision 2.0   5 – 2 0

Student Notes

`nice` executes a command with a lower CPU scheduling priority. If the increment argument (in the range 1-20) is given, it is used; if not, an increment of 10 is assumed. For example, the following command will add 10 to the default niceness value (20), yielding a niceness of 30:

```
% nice /usr/demos/bin/flight&
```

In contrast this command will add 2 to the default niceness value of 20, yielding a niceness of 22:

```
% nice -2 /usr/demos/bin/flight&
```

A version of the `nice` command exists as a built-in command within the C shell. It has a different syntax than `/sbin/nice` and slightly different behavior. If you are running the C or T shell, it is important to type the full path `/sbin/nice` to get `/sbin/nice` behavior, otherwise, you get the built-in `nice`. See the man page on the C or T shell to get information about this behavior.

5-21 Using `npri`

Using `npri`

Superuser can use **`npri`** to change the scheduling parameters of a user's process

- Priority, making it *nondegrading* **`-h`**
- Absolute "nice–ness" value **`-n`**
- Time slice for the process **`-t`**

```
IRIX thisisit 6.2 01181722 IP22 Load[0.40,0.25,0.20] 21:21:24 56 procs
    user   pid  pgrp   %cpu proc  pri  size   rss    time  command
     pat 27695 23267  83.30    2   64  1652  1407   39:06  uftx
    root   131     0  23.61    *  +39     0     0  502:34  rtnetd
   guest 27688 27688  13.32    4  +65   441   251    0:01  top
     jmx 27705 22177   8.47    *   30   338   112    0:00  driver
    root   160     0   7.41    *   26   306    54   95:39  ypserv
     ism 27554   389   1.50    *   30   469   195    0:01  make
```

Note: User **`pat`** is using 83% of the system's CPU resources

IRIX System Administration: System Monitoring Revision 2.0 5 – 2 1

Student Notes

IRIX allows a system administrator or programmer to manipulate many aspects of a process' scheduling priorities.

You can control the following:

- Which order processes are scheduled into the CPU

  NDPRI Set or remove nondegrading priority
  RENICE Relative "nice-ness" value

- How long a process executes on the CPU

  SLICE Size of process' time slice in the CPU

5-22 Example Using `npri`

Example Using `npri`

- Use the **ps** command to find out the process id of Pat's process
- Change the priority of Pat's process with **npri** to 254

```
# npri -h 254 -p 27695
```

- Priority is automatically *nondegrading*
- Pat's process will still run, but only when the system is not busy doing other work

```
F S  UID   PID   PPID C PRI NI  P   SZ:RSS    WCHAN   TTY    TIME  COMD
30 S 1224 27695 5764 0 254 BT  2 1652:1407 801cd4f8 ttyq2  41:12 ufta
```

- Use the **ps -elc** command to see the class or run queue that Pat's process is in

IRIX System Administration: System Monitoring   Revision 2.0   5 - 22

Student Notes

Pat's process is using most of the system's CPU resources. Pat will not `nice` the process or run it at night. Many other people on the system have to get work done. Use the `npri -h` command to set the program to a nondegrading, low priority.

A `ps -el` output shows this output for Pat's process, with the `BT` to indicate nondegrading priority. Use the `ps -elc` command to see that Pat's program has been put into the `BT` or batch run queue.

5-23 System Error Logging

System Error Logging

- **syslogd(1M)** daemon reads and logs messages
- Use **/etc/syslog.conf** file to customize
- Each line has a *selector* and an *action*
  - Selectors are composed of *facilities* and *levels*
    - Facilities include **kern**, **user**, and **mail**
    - Levels include **emerg**, **alert**, and **crit**
  - Actions are composed of *destinations*
    - Destinations include filenames, user names, devices, hostnames, and filter programs

IRIX System Administration: System Monitoring Revision 2.0 5 – 23

Student Notes

The `syslogd` daemon collects messages sent by various system processes and routes these messages to their final destination based on instructions given in the configuration file `/etc/syslog.conf`.

`syslogd` reads its configuration when it starts up and whenever it receives a hangup signal. Lines in the configuration file have a *selector* to determine the message priorities to which the line applies and an action. The action field(s) are separated from the selector by one or more tabs.

Selectors are composed of *which messages* (facilities) and *what priority* (levels) those messages should be. Actions are *which destinations* to write the messages to.

Actions can be filenames, user names, hostnames, devices, or filter programs. Facilities can be `kern`, `user`, `mail`, `cron`, `news`, `uucp`, `auth`, `audit`, `daemon`, `lpr`, `syslog`, or `local0 - 7`. Levels can be `emerg`, `alert`, `crit`, `err`, `warn`, `notice`, `info`, or `debug`. See `/usr/include/sys/syslog.h` for a list of all facilities and levels.

5-24 Example `/etc/syslog.conf`

Example `/etc/syslog.conf`

```
kern.debug     |/usr/sbin/klogpp    /var/adm/SYSLOG
kern.debug     |/usr/sbin/klogpp    /dev/console
user,mail,daemon,auth.debug         /var/adm/SYSLOG
kern.debug     @vaquero
*.emerg        *
*.alert        mannel,nickd
```

IRIX System Administration: System Monitoring   Revision 2.0   5 – 24

Student Notes

```
# Formats: selector<TAB>action
#          selector<TAB>filter<TAB>action
```

You can direct the same messages to several destinations. You can filter kernel messages with `/usr/sbin/klogpp`. The file `/var/adm/klogpp` is symbolically linked to `/usr/sbin/klogpp`. Kernel messages directed to other machines are filtered on *that host*.

High-level messages (`alert`, `crit`, `emerg`) should be addressed to logged in users, system administrators, and the log file.

5-25 Utilizing a Remote Console for Logging Messages

Utilizing a Remote Console for Logging Messages

- In IRIX, any streams–based device (an **xwsh**, **xterm**, or a program of your own) can display console information
  - Example: Either of the following commands creates a window that receives console messages:

```
# xwsh -console &
# startconsole
```

- Because of the network extensibility of X, you can display a machine's console window on another machine across the net
  - Example: Display the current machine's console on machine *happy*:

```
# xwsh -console -display happy:0 &
```

IRIX System Administration: System Monitoring Revision 2.0 5–25

Student Notes

Any error messages, such as `syslogd` messages, sent to `/dev/console` appear in windows started with `startconsole` or `xwsh -console &`.

The `xhost(1)` program adds and deletes hostnames or user names to the list allowed to make connections to the X server. In the case of hosts, this provides a rudimentary form of privacy control and security.

`xhost [[+-]name ...]`

`xhost +` permits access by everyone

`xhost -` restricts access to only those on the `xhost +` list

SiliconGraphics
Computer Systems

Module 6: Disk Maintenance

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module discusses partitioning disk drives on your SGI system.

6-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Understand the difference between *system* and *option* disks and the partitions that must exist for each
- Read partition information using **`prtvtoc(1M)`**
- Read volume header information using **`dvhtool(1M)`**
- Partition a disk drive using **`fx(1M)`**
- Partition a disk drive using **`xdkm(1M)`**
- Create device files using **`mknod(1M)`**

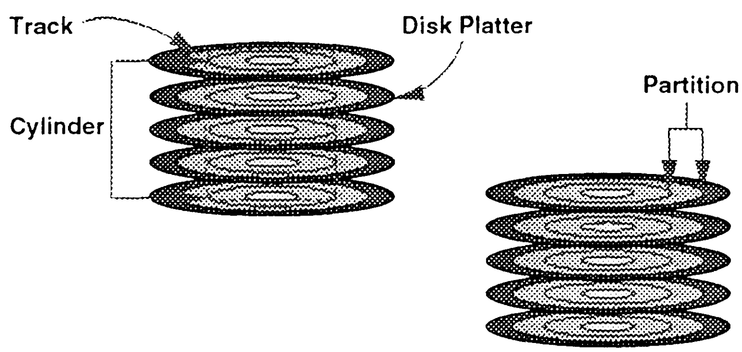
IRIX System Administration: Disk Maintenance Revision 2.0 6 – 2

Student Notes

6-3 IRIX Disk Structure

IRIX Disk Structure

- An IRIX disk is divided into sections called *partitions*
- A partition is a contiguous group of disk cylinders
- A *cylinder* is the same track on all the platters of the disk



IRIX System Administration: Disk Maintenance Revision 2.0 6 – 3

Student Notes

Although other options are available, we tend to define sizes in terms of cylinders.

A track is a single band of data divided into sectors.

Read-write heads work on each surface of a disk platter.

6-4 IRIX Partitions

IRIX Partitions

Types of IRIX partitions:

- Volume header data (**volhdr**)
- **efs** or **xfs** data
- **lvol** or **xlv** data
- **xfslog** information
- **raw** data

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 4

Student Notes

EFS or XFS partitions contain, for example, IRIX files, directories, and C programs.

Logical volumes, `lvol` or `xlv`, contain data from more than one disk partition for IRIX files, directories, C programs, and so on.

XFS requires a log area to store information about the filesystem as it is changing. The partition type `xfslog` is only used when the log is an external log.

You can create swap space with a raw partition to increase the size of virtual memory.

The volume header partition can contain programs and files such as `sash`, `ide`, `fx`, `symmon`, and `sgilabel`. There is limited space in the volume header partition, so all files might not fit.

6-5 System Disk Versus Option Disks

System Disk Versus Option Disks

	System Disk	Option Disk
Purpose:	For booting IRIX For storing system data	For storing user data
Must contain:	Volume header – **sash** Root filesystem Swap partition Partition 10 (**vol**)	Volume header Partition 10 (**vol**)
May also contain:	**usr** filesystem	Swap partition data partition Part of a logical volume

IRIX System Administration: Disk Maintenance   Revision 2.0   6 – 5

Student Notes

The system disk must contain the root filesystem, partition 0, a swap space, partition 1,the volume header, partition 8, and a special partition representing the entire disk, partition 10. Some system disks also have a `usr` filesystem, partition 6. There can only be one active system disk per system. The default SCSI system disk is either `dks0d1` or `dks1d1`.

The option disk can have many partitions to contain user data and applications. The option disk can have a volume header, partition 8, a data area, partition 7 and the special partition representing the entire disk, partition 10. Or there can be several partitions (up to 16), partition 0 through partition 15 for extra swap space, a special project area, more user data area, and an application area. An option disk can also be partitioned without the volume header, partition 8, to be used as the entire disk for data. This requires a program to manage the files, such as SYBASE.

There can be many option disks on a system. Option disks are sometimes called secondary disks.

6-6 Volume Header

Volume Header

- Special partition on each disk
- Can contain programs such as **fx**, **sash**, **ide**, and **symmon**
- Contains partition and disk parameter information
- Volume header device file on default system disk linked to **/dev/rvh**

  Example:

```
ls -i /dev/rvh /dev/rdsk/dks0d1vh
204     /dev/rvh
204     /dev/rdsk/dks0d1vh
```

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 6

Student Notes

`dvhtool` (an SGI proprietary tool) looks at the volume header information, which is stored in partition 8.

The volume header partition can contain programs and files such as `sash`, `ide`, `fx`, `symmon`, and `sgilabel`. There is limited space in the volume header partition, so all files might not fit. Volume header data also contains partition information, drive parameters, boot information, and label information.

Boot information on the volume header contains the location of `swap`, `unix`, and the `root` partition.

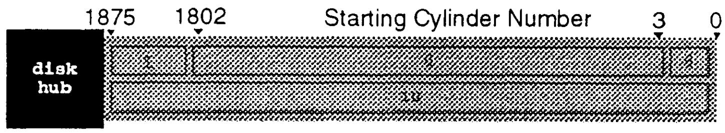
Although `ide` and `symmon` are by default in the volume header, it is possible that they are not included in the volume header as disks get reconfigured.

The example at the bottom of the page proves that `/dev/rvh` is hard linked to the volume header of the default system disk.

6-7 Reading Partition Information

Reading Partition Information

- Use **prtvtoc(1M)**
- Without an argument, **prtvtoc** prints system disk's volume header




```
# prtvtoc
Printing label for root disk

* /dev/rdsk/dks0d1s0 (bootfile "/unix")
*     512 bytes/sector
*      74 sectors/track
*      15 tracks/cylinder
*       3 spare blocks/cylinder
*    1876 cylinders
*       3 cylinders occupied by header
*    1873 accessible cylinders
*
* No space unallocated to partitions

Partition  Type   Fs   Start: sec    (cyl)     Size: sec   (cyl)  Mount Directory
 0          xfs   yes       3321   (   3)      1991493  (1799)   /
 1          raw          1994814   (1802)        81918  (  74)   swap
 8        volhdr               0   (   0)         3321  (   3)
10        volume               0   (   0)      2076732  (1875)
```


Student Notes

To list the partition information for an option disk, use the disk device name as an argument to `prtvtoc`. The `-a` flag shows all disk, whereas the `-s` flag is a shortened version.

```
# prtvtoc -s dks0d2vh

Partition Type   Fs    Start:sec   (cyl)     Size:sec    (cyl)   Mount
                         Directory
 0         xfs              3321   (  3)       32103    (  29)
 1         raw             35424   ( 32)       81918    (  74)
 6         xfs            117342   (106)     1959390    (1770)
 7         xfs   yes        3321   (  3)     2073411    (1873)    /d2
 8        volhdr               0   (   0)       3321    (   3)
10        volume               0   (   0)    2076732    (1876)
```

6-8 Using dvhtool

Using dvhtool

- Prints volume header information
- Adds or removes files from the volume header

```
# dvhtool
Volume? (/dev/rvh)<return>
 Command? (read, vd, pt, dp, write, bootfile, or quit): vd
 Command? (d FILE, a UNIX_FILE FILE, c UNIX_FILE FILE, g FILE
 UNIX_FILE or l) ?  l

Current contents:
      File name       Length       Block #
      sgilabel           512             2
      sash            140800             3
      symmon          244224           278
      ide             977920           756
```

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 8

Student Notes

The following are options from the main menu of `dvhtool`.

Options	Description
`read`	Read from a different volume header
`vd`	Volume directory: files in the volume directory area
`pt`	Partition table: partition start blocks, length, and type
`dp`	Disk parameters: controller information
`write`	Write out (save) changed information
`bootfile`	Name of IRIX kernel, location of root and swap partitions
`quit`	Quit and return to shell

6-9 Disk Utility `fx`

Disk Utility `fx`

- Two modes of operation
  - Regular
  - Expert (`-x`)
- Available in
  - Multiuser mode and single-user mode:

    `# fx -x`
  - standalone (local or network)

    `>> boot dksc(0,1,0)/stand/fx`

    `>> boot -f bootp()host:/stand/fx --x`

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 9

Student Notes

You must boot the appropriate `fx` for the system's processor type. In the past, most systems used the CPU processor number, `fx.IP22` in the case of an Indigo2, where `hinv -c` processor reveals one 150-MHz IP22 processor.

Now most systems use `fx.ARCS` (Indy, Indigo2, Onyx, Challenge, and R4K Indigo). The 64-bit systems (R8000 and above) use `fx.64` to indicate which version of `fx`.

To use `fx` from PROM, use the following command:

```
>> boot dksc(0,1,0)/stand/fx
```

From PROM, to boot a copy of `fx` in expert mode, from a similar system across the network, use the following command:

```
>> boot -f bootp()server:/stand/fx --x
```

6-10 fx Prompts

fx Prompts

When entering **fx**, you are prompted for the address of the disk

- Press the <**Enter**> key to accept the default displayed in parentheses; otherwise, type in a different selection

  `fx: "device-name" = (dksc)` <Enter>

- Select the controller:

  `fx: ctlr# = (0)` <Enter>

- Select the drive:

  `fx: drive# = (2)` <Enter>

- Select the logical unit number:

  `fx: lun# = (0)` <Enter>

Student Notes

If you booted `fx` from PROM, confirm mode of operation. Answer "`yes`" to extended mode prompt to make changes.

When you are selecting the drive type, `fx` reads from the volume header. Valid choices include the following:

Drive Type	Name
`dksc`	SCSI
`fd`	Floppy
`jag`	VME SCSI Jaguar
`rad`	SCSI RAID

6-11 fx Top Menu

fx Top Menu

```
...opening dksc(0,2,)
fx: mounted partitions detected on device
fx: devname                 seq start    end       owner    cause
fx: /dev/rdsk/dks0d1s1      2 4113408 4194815 swap     already in use
fx: /dev/rdsk/dks0d1s0      1 4608    4113407 xfs/efs  already in use

fx: Warning:  this disk appears to have mounted filesystems.
         Don't do anything destructive, unless you are sure
         nothing is really mounted on this disk.
...controller test...OK
Scsi drive type == SGI      SEAGATE ST31200N8640

----- please choose one (? for help, .. to quit this menu)-----
[exi]t           [d]ebug/        [l]abel/          [a]uto
[b]adblock/      [exe]rcise/     [r]epartition/    [f]ormat
fx>
```

IRIX System Administration: Disk Maintenance   Revision 2.0   6 – 1 1

Student Notes

A trailing / indicates a submenu. The following options are usually performed by the customer: `exit`, `badblock`, `exercise`, `label`, `repartition`, and `format`, whereas `debug` and `auto` are usually performed by SGI.

Option	Operation or Submenu
`exit`	Exit from `fx`
`debug`	Disk debugging operations
`label`	Adjust partitions, disk parameters, etc.
`badblock`	Add bad block information
`exercise`	Run drive exercising programs
`repartition`	Repartition a disk
`auto`	Automated setup of disk
`format`	Format the disk

6-12 Using fx to Show Disk Partitions

Using fx to Show Disk Partitions

- Use the **label** menu

```
---- please choose one of (? for help, .. to quit this menu)----
[exi]t          [d]ebug/        [l]abel/
[b]adblock/     [exe]rcise/     [r]epartition/
fx> l

----- please choose one (? for help, .. to quit this menu)-----
[s]how/
fx/label> s

----- please choose one (? for help, .. to quit this menu)-----
[para]meters       [part]itions       [b]ootinfo        [a]ll
[g]eometry         [s]giinfo          [d]irectory
fx/label/show> part

----- partitions-----
part  type        cyls              blocks            Megabytes
(base+size)
  7: xfs        4 + 2460       3824 + 2351760        2 + 1148
  8: volhdr     0 + 4             0 + 3824           0 + 2
 10: volume     0 + 2464          0 + 2355584        0 + 1150
```

IRIX System Administration: Disk Maintenance   Revision 2.0   6 – 1 2

Student Notes

There is an fx option called `label/show/all` that shows partition information, volume header information, hardware geometry, and boot information.

6-13 Exiting `fx`

Exiting `fx`

- Type `..` at any prompt to get out of that menu and go to the parent menu

```
fx/label/show> ..
fx/label>
```

- Type `/exit` to get out of `fx`

```
fx/label/show> /exit
#
```

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 1 3

Student Notes

Type ". ." to move from a submenu to the parent menu, which is very much like the IRIX filesystem structure. Use `/exit` to exit `fx` from any menu. To execute an option in a submenu, use the submenu name followed by a slash with the option, such as `label/sync`. Type "`?`", "`? option`", or "`? submenu name`" to get help.

6-14 Why Change Disk Partitions?

Why Change Disk Partitions?

- System runs out of swap space
- Need more room in `root`
- Need more room in user filesystems
- Prepare *additional* disk drives

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 14

Student Notes

6-15 System Disk Partitioning Procedure

System Disk Partitioning Procedure

- Collect current partition information and plan repartition
- Back up all filesystems on disk affected
- Shut down to PROM
- Boot **`fx`**
- Use **`fx`** to show, reconfigure, and check partitions
- Save changes
- Boot miniroot to remake and reload affected filesystems

Student Notes

Only one partition can be *used* for a given range of space; however, there can be several partitions *defined* for that range of space.

You must use `mknod` to create the device files for those partitions that are nonstandard partitions: 2, 3, 4, 5, 9, 11, 12, 13, and 14.

The following are predefined standard partition selections available from `fx`:

Partitions	Naming Convention
0, 1, 8, 10	root drive
0, 1, 6, 8, 10	usr root drive
7, 8, 10	option drive

6-16 More Disk Partition Tips

More Disk Partition Tips

Partition's starting cylinder number

+

size

=

starting number of next partition

Example:

Partition	Type	Fs	Start: sec	(cyl)	Size: sec	(cyl)	Mount Directory
3	xfs	yes	3321	(3)	32103	(29)	/extra
5	raw		35424	(**32**)	81918	(74)	
6	xfs	yes	117342	(106)	1959390	(1770)	/project

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 1 6

Student Notes

It is extremely important to partition the disk correctly; otherwise, you might waste space or, more devastating, you might have overlapping partitions that do not present problems until you are already using them.

6-17 SCSI Partition Examples

SCSI Partition Examples

Indy System Disk

Partition Number	Name	Type	What	Starting Cylinder	Size in Cylinders
8	vh	volhdr	volhdr	0	4
0	s0	efs	root partition	4	2615
1	s1	raw	swap	2619	107
10	vol	volume	all disk	0	2726

Option Disk

Partition Number	Name	Type	What	Starting Cylinder	Size in Cylinders
8	vh	volhdr	volhdr	0	4
7	s7	xfs	user space	4	1351
10	vol	volume	all disk	0	1355

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 1 7

Student Notes

6-18 Disk Partition Rules and Recommendations

Disk Partition Rules and Recommendations

- Plan partition changes carefully
  - Increasing one partition reduces another (if all disk space is utilized)
  - Changing one partition can affect many partitions
- Only one partition can be *used* for a given range of space
- Customization is required for the use of partitions 2,3,4,5,9,11,12,13,14
- Volume header cannot be used for data
- Partition 1 on system disk is reserved for swap space
- Partition 10 defined only for raw data; can be used for some database applications

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 18

Student Notes

If you are redoing the system disk after repartitioning with `fx`, you need to load miniroot to run the `mkfs` command and then do the file/filesystem recovery.

6-19 Option Disk Partitioning Procedure

Option Disk Partitioning Procedure

- Collect current partition information
- Plan repartition
- Back up all filesystems on disk affected
- Unmount affected filesystems
- Use `fx` to show, reconfigure, and check partitions
- Save changes
- Remake, reload affected filesystems

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 1 9

Student Notes

You may have to run `fx` with the `-x` option to get the expert mode, so that nonstandard partitions can be created and resized.

6-20 Using fx to Repartition

Using fx to Repartition

```
----- please choose one (? for help, .. to quit this menu)-----
[exi]t              [d]ebug/             [l]abel/             [a]uto
[b]adblock/         [exe]rcise/          [r]epartition/       [f]ormat
fx> r

----- partitions-----
part  type         cyls              blocks            Megabytes
(base+size)
  0: xfs        4 + 53          3824 + 50668          2 + 25
  1: raw       57 + 85         54492 + 81260         27 + 40
  6: xfs      142 + 2322      135752 + 2219832       66 + 1084
  8: volhdr     0 + 4              0 + 3824           0 + 2
 10: volume     0 + 2464           0 + 2355584        0 + 1150

capacity is 2356180 blocks

----- please choose one (? for help, .. to quit this menu)-----
[ro]otdrive            [o]ptiondrive          [e]xpert
[u]srrootdrive         [re]size
fx/repartition>
```

IRIX System Administration: Disk Maintenance — Revision 2.0 — 6 - 20

Student Notes

In this example, the megabyte counts do not add up from partition 1 to partition 6; 27 + 40 does not equal 66—the starting megabyte of partition 6. For this reason, the cylinder count is more accurate and usually preferred.

6-21 Options for Partitioning

Options for Partitioning

Option	Operation or Submenu
`rootdrive`	Repartition as system drive with **root**, **swap**, **vh**, and **vol** partitions
`usrrootdrive`	Repartition an old–style system drive with **root** partition, **swap**, **usr**, **vh, vol**
`optiondrive`	Repartition as option drive with one large **efs/xfs**, **vh**, and **vol** partitions
`resize`	Resize the *standard* partitions (depending on whether you choose **root**, **usrroot**, or option drive)
`expert`	Use expert mode for repartitioning. Must be used for creating *nonstandard* partitions

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 2 1

Student Notes

6-22 Resizing With `fx`

Resizing With `fx`

- With **`resize`** option, you specify which partition to change, and **`fx`** changes that partition, automatically subtracting or adding from adjacent partitions
  - Only works with *standard* partitions
  - When you are done repartitioning, decide if you want the new arrangement **`fx`** chose for you
- With **`resize`** option, repartition in terms of cylinders, blocks, megabytes, or percentage of total disk
  - With expert option, can only set in terms of cylinders, but it prints the resulting block and Mb counts

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 2 2

Student Notes

Resizing partitions only works for standard partitions.

6-23 Example: Resizing Standard Partitions

```
[ro]otdrive              [o]ptiondrive        [e]xpert
[u]srrootdrive          [re]size
fx/repartition>re

After changing the partition,the other partitions will be
adjusted around it to fit the change. The result will be
displayed and you will be asked whether it is OK, before the
change is committed to disk. Only the standard partitions may be
changed with this function. Type ? at the prompts for a list of
possible choices
fx/repartition/resize: partition to change = (swap) ?
----- partition to change -----
 [r]oot     [s]wap     [u]sr     [v]olume[x]fslog
fx/repartition/resize: partition to change = (swap) <Enter>
current:  type raw  base:  57 cyls,  54492 blks,  27 Mb
                     len:  85 cyls,  81260 blks,  40 Mb
fx/repartition/resize: partitioning method = (megabytes) ?
----- partitioning method -----
 [m]egabytes (2^20 bytes)[b]locks
 [p]ercentage[c]ylinders

fx/repartition/resize: partitioning method = (megabytes)m
fx/repartition/resize: size in megabytes (max 1149) = (40) 140
----- partitions -----
part  type      cyls          blocks        Megabytes (base+size)
  0: xfs        4 + 53       3824 + 50668        2 + 25
  1: raw       57 + 299     54492 + 286720      27 + 140
  6: xfs      356 + 2107   341212 + 2014372    167 + 984
  8: volhdr     0 + 4           0 + 3824         0 + 2
 10: volume     0 + 2464        0 + 2355584      0 + 1150
Use the new partition layout? (no)yes

----- partitions -----
part  type      cyls          blocks        Megabytes (base+size)
  0: xfs        4 + 53       3824 + 50668        2 + 25
  1: raw       57 + 299     54492 + 286720      27 + 140
  6: xfs      356 + 2107   341212 + 2014372    167 + 984
  8: volhdr     0 + 4           0 + 3824         0 + 2
10: volume     0 + 2464        0 + 2355584      0 + 1150

capacity is 2356180 blocks

----- please choose one (? for help, .. to quit this menu)-----
```

```
[ro]otdrive[o]ptiondrive[re]size    [e]xpert
fx/repartition> /exit
```

Student Notes

If you choose m for megabytes, the information is automatically converted to cylinders. The actual megabytes line looks like this: (megabytes (2^20 bytes)). Notice that since partition 1 increased from 40 Mbytes to 140 Mbytes, the adjacent partition 6 was automatically adjusted (decreased) in size.

When you approve the new layout, the information is automatically written to disk.

6-24 Example: Using Expert Mode to Repartition

- fx presents partition numbers one at a time. To accept existing info, press <Enter>; otherwise, type in new information

```
---- please choose one (? for help, .. to quit this menu)----
[ro]otdrive    [o]ptiondrive      [e]xpert
[u]srrootdrive     [re]size
fx/repartition> e

Warning: you will need to re-install all software and restore
user data from backups after changing the partition layout.
Changing partitions will cause all data on the drive to be lost.
Be sure you have the drive backed up if it contains any user
data.Continue? y
Enter .. when done
fx/repartition/expert: change partition = (0) <Enter>
before:type xfs base: 4 cyls, 3824 blks, 2 Mb
      len:53 cyls,50668 blks,25 Mb
fx/repartition/expert: partition type = (xfs) <Enter>
fx/repartition/expert: base cyl = (4) <Enter>
fx/repartition/expert: ncyls = (max 2460) (53) 100
before:type xfs base: 4 cyls, 3824 blks, 2 Mb
     len:100 cyls,95600 blks,50 Mb
fx/repartition/expert: change partition = (1)
```

- To skip to a particular partition, type that partition number at the `change partition` prompt:

```
fx/repartition/expert: change partition = (1) 13
before:type xfsbase: 57 cyls,  54492 blks,
           len: 85 cyls, 81260  blks,
fx/repartition/expert: partition type = (raw) xfs <Enter>
fx/repartition/expert: base cyl = (57) 290 <Enter>
fx/repartition/expert: ncyls = (max 2322)(85)624 <Enter>
```

Student Notes

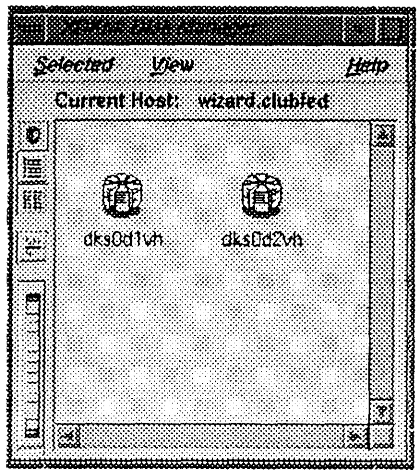
This example uses expert mode to repartition the disk increasing partition 0 currently 53 cylinders in size (25 Mb) to 100 cylinders in size (50 Mb). Because this change was made to partition 0, a standard partition, expert mode was not necessary. You could have used the `resize` option of `repartition`.

To add a nonstandard partition to your disk definition, supply the partition number at the "change partition" prompt. In this example, the nonstandard partition 13 is being added. Partition 13 is of type `xfs`, starts at cylinder 290, and is 624 cylinders in size.

6-25 Using `xdkm`

Using **xdkm**

xdkm is a graphical tool to repartition the disk or to get information about the disk

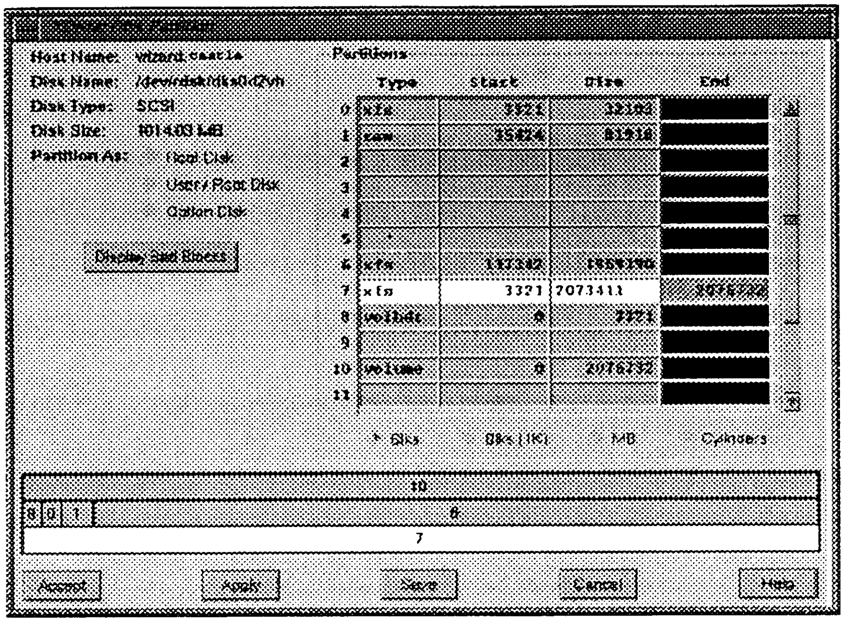


IRIX System Administration: Disk Maintenance Revision 2.0 6 – 25

Student Notes

6-26 Using xdkm to Repartition

Using xdkm to Repartition



IRIX System Administration: Disk Maintenance Revision 2.0 6 – 2 6

Student Notes

You can use the graphical tool xdkm instead of `fx` to repartition a disk when you are running in multiuser mode. The `fx` utility is still the most flexible command to repartition a disk. You can boot it from PROM, use it in single-user or multiuser mode and on server systems that do not have graphics.

6-27 Partition Device Files

Partition Device Files

- Kernel talks to partitions via the device files
- Each partition has a block and character device file
  - Block device files
    - Contained in directory **/dev/dsk**
    - Talk to the disk via the system buffer cache using blocks of data
  - Character device files
    - Contained in **/dev/rdsk**
    - Used to talk to the disk *one* byte at a time
- Example: **<ctlr type ctlr#><ddisk#><spartition#>**

Partition	SCSI	
0	dks0d1s0	dks1d3s0
1	dks0d1s1	dks1d3s1
8	dks0d1vh	dks1d3vh
10	dks0d1vol	dks1d3vol

- Use **hinv(1M)** to determine hardware addresses

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 27

Student Notes

Block device files utilize memory caching; character device files write directly to disk. In general, devices that buffer themselves are character devices; devices that want the kernel to handle the buffering are block devices. Terminals communicate using character devices.

Both partition 8 and 10 only have character device files.

The following line from `hinv(1M)` and its corresponding device file:

```
Disk drive: unit 2 on SCSI controller 0
```
`/dev/rdsk/dks0d2s#` and `/dev/dsk/dks0d2s#`

6-28 Configuring New Partitions

Configuring New Partitions

- New partitions require block and character device files
- Create device files if you use nondefault partitions
  - Only device files 0, 1, 6, 7, 15, **vh**, and **vol** are created by default with the **/dev/MAKEDEV** script
- List existing device files to view major and minor numbers

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 28

Student Notes

6-29 Creating New Device Files

Creating New Device Files

- Example: SCSI controller 0, disk 2, partition 3
  - The device files for this partition are **/dev/dsk/dks0d2s3** and **/dev/rdsk/dks0d2s3**
  - Look at the major and minor number for disk 2 partition 0

```
# ls -li /dev/dsk/dks0d2s0
165 brw----- 2 root sys  128,32 Mar 17 10:27 /dev/dsk/dks0d2s0
```

  - For the new device files, use the same major number (128)

```
165 brw------ 2 root sys 128,32 Mar 17 10:27 /dev/dsk/dks0d2s0
```

  - Add the partition number (3) to the minor number (32) of the root partition to get the minor number of the new partition (3 + 32) = 35

```
165 brw------ 2 root sys 128,32 Mar 17 10:27 /dev/dsk/dks0d2s0
```

  - Use **mknod(1M)**

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 29

Student Notes

Always refer to partition 0 of the disk when you are creating new partition device files for the major and minor numbers.

6-30 Using mknod for Partition Device Files

Using mknod for Partition Device Files

Syntax:

mknod <device filename> b/c <major no.> <minor no>

- Use the appropriate major and minor numbers
- Create both a block device file in **/dev/dsk** and a character device file in **/dev/rdsk**
  - Making the block device file

    ```
    # mknod /dev/dsk/dks0d2s3 b 128 35
    ```
  - Making the character device file

    ```
    # mknod /dev/rdsk/dks0d2s3 c 128 35
    ```

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 3 0

Student Notes

You must calculate the minor number based on the device file for partition 0 of that controller and disk number. To determine a non-standard partition's minor number, use partition zero (0)'s minor number as the base and add to it the partition number.Example:

```
# ls -l /dev/dsk/dks0d2s0
brw-r-----   2 root   sys    128, 32 Mar   1 17:47 /dev/dsk/dks0d2s0
```

The minor number is 32. If you are creating partition 3, the minor number is 32+ 3= 35

```
# ls -l /dev/dsk/dks0d2s3
brw-r-----   2 root   sys    128, 35 Mar   1 17:47 /dev/dsk/dks0d2s3
```

6-31 Identifying Filesystem and Partition Linkage

Identifying Filesystem and Partition Linkage

- Use **`ls -il`** to get the inode number of the device file

```
# ls -il /dev/dsk/dks0d1s0
436 brw------- 2 root sys 128,16 Feb 7 13:10 /dev/dsk/dks0d1s0
```

and

- Use **`find`** to get the corresponding partition device file

```
# find /dev -inum 436 -print
/dev/root
/dev/dsk/dks0d1s0
```

or

- Use **`devnm`** to get the corresponding partition device file

```
# devnm /dev/root
/dev/dsk/dks0d1s0 /dev/root
```

IRIX System Administration: Disk Maintenance Revision 2.0 6 – 3 1

Student Notes

The `find` command is the simplest and fastest way to get the linked device filenames. You must be superuser to use the `ncheck` command, but not for the `find` command.

In the following listing, 128 is a major number, that is, the type of device; 16 is a minor number, specifying the controller, port, and partition:

```
436 brw------- 2 root sys  128, 16 Feb 13 13:10 /dev/root
```

SiliconGraphics
Computer Systems

Module 7: Filesystems

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module discusses maintenance of filesystems. You will learn what filesystems are, how to create, check and mount them on to the IRIX directory tree.

7-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Describe filesystem characteristics and organization
- Plan the creation of an XFS filesystem
- Create EFS and XFS filesystems:
  mkfs, **mkfs_efs**, **mkfs_xfs**
- Mount filesystems manually and automatically
  mount, **/etc/fstab**
- Use **xfsm** to create and manage filesystems
- Maintain EFS filesystems using **fsck**, **fsr**
- Use a CD-ROM filesystem

IRIX System Administration: Filesystems Revision 2.0 7 - 2

Student Notes

7-3 Filesystems

Filesystems

- Organizes data on disks
  - Maps names to data on disks
- Provides a uniform interface to accessing disk files
- Created on a partition or a logical volume

IRIX System Administration: Filesystems Revision 2.0 7 – 3

Student Notes

A filesystem manages storing your data on a disk and provides a consistent means of addressing that data. Behind the scenes, there are data structures and algorithms that allow fast and easy access to data.

Filesystems are usually created on a single or group of disk partitions. These partitions hold the data of the filesystem and the data structures that control the layout of the data.

7-4 Organization of a Filesystem

Organization of a Filesystem

Composed of data structures and file data

- At the user level
  - User files
  - Directories
  - Device files
- At the system level
  - Superblock
  - Basic blocks
  - Inode table
  - Inode free list
  - Basic blocks/extents
  - Free block bitmap

IRIX System Administration: Filesystems Revision 2.0 7 - 4

Student Notes

Basic blocks are the fundamental unit of data storage in a filesystem. Each basic block is composed of some number of bytes, which is determined when the filesystem is created. Basic blocks may be grouped into contiguous regions called *extents*.

There are trade-offs in using basic blocks of different sizes. Considering that a basic block is contiguous on the disk, less time is spent seeking for data on a disk with larger block sizes, but this may waste space. Smaller block sizes use space more efficiently, but may cause disk fragmentation.

7-5 Organization of a Filesystem (continued)

Organization of a Filesystem (continued)

System data

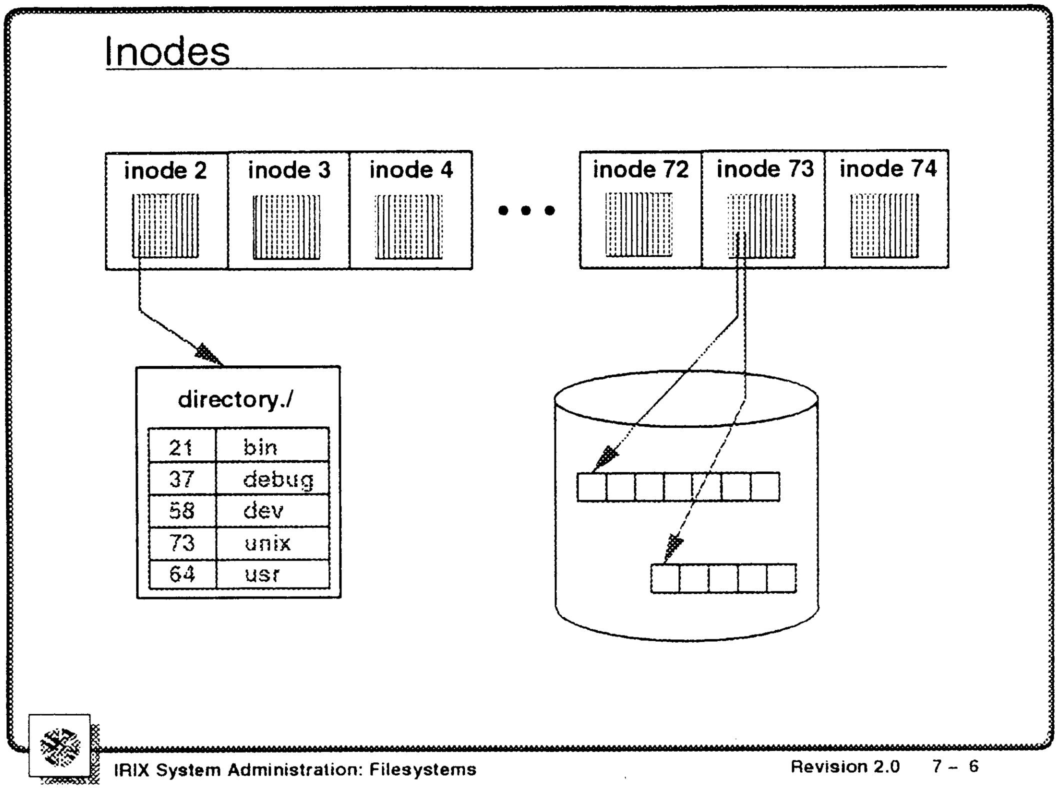
- Superblock
  - Filesystem parameters used by kernel
  - Duplicate copies of the superblock exist
- Inodes – data structures
  - One for each file/directory in the filesystem
  - Permission, ownership information, location of file data on disk
  - Inode size 128–2K bytes
  - IRIX keeps two copies of inodes
    - Working copy in kernel memory
    - Stored copy on disk in inode table
    - Sync copies kernel inodes to inodes on disk

IRIX System Administration: Filesystems Revision 2.0 7 – 5

Student Notes

The superblock of a filesystem is arguably the most important part of the filesystem. It contains information such as how to locate directories, and index nodes. Because the superblock is so important, modern filesystems usually contain duplicate copies of it. In the case that the system cannot use the original superblock, it can try to use an alternate superblock to continue its operation.

7-6 Inodes



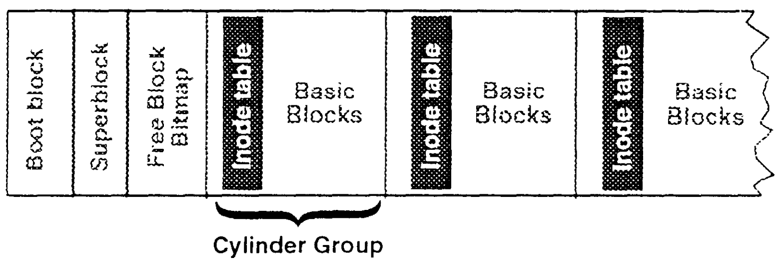
Student Notes

Inodes contain the address of all the data blocks for a particular file. In general, there may be 10 or 12 address pointers per inode, which could point to a block of file data, or a block that contains additional block addresses. When a block contains addresses of other blocks, it is called *indirect*.

7-7 SGI Extent Filesystem EFS

SGI Extent Filesystem EFS

- Free blocks indicated with a bitmap
- Inodes contained single, double, and triple indirects
- Inodes pointed to extents in the filesystem
- Extent pointers play a dual role
  - All are direct extent pointers until all 12 extent pointers are used
  - All extent pointers are converted to indirect extent pointers for larger files
- Extents can be of variable size (1–248 blocks)



IRIX System Administration: Filesystems Revision 2.0 7 - 7

Student Notes

EFS uses two copies of the superblock for better reliability, a bitmap to keep basic block allocation, which is faster. The inode table was broken into sections called cylinder groups. This allows a file's inode and data to have better locality, thereby reducing the seek latency of reading the file's data.

Basic blocks are stored in extents, and inodes point to extents. EFS can have up to 248- to 512-byte basic blocks per extent. Each inode contains pointers to 12 extents.When all of the extent pointers of an inode become allocated, the filesystem allocates an extent, copies the original extent inode pointers to this block of data, and replaces the first extent pointer with the address of this new block. This is indirect extent addressing. This will continue until the maximum file size of 2 Gb is obtained.

7-8 Creating an EFS Filesystem

Creating an EFS Filesystem

- After creating the new device files with **mknod(1M)**, run **mkfs(1M)** to make a filesystem

```
# mkfs_efs /dev/rdsk/dks0d2s3
```

or

```
# mkfs -t efs /dev/rdsk/dks0d2s3
```

IRIX System Administration: Filesystems Revision 2.0 7 - 8

Student Notes

Using the command `mkfs_efs` or `mkfs -t` does the same thing. Be careful because if you only use `mkfs /dev/rdsk/dks0d2s3`, the default filesystem created will be of type XFS.

7-9 SGI XFS Filesystem

SGI XFS Filesystem

- 64–bit file capability
- Journalled filesystem
  - Internal or external `log` required
  - `fsck(1)` not used
- High–performance filesystem
- No performance penalty as filesystem size grows
  - Recovery time does not increase with filesystem size
- Inodes are dynamically allocated
  - Variable sized from 256 bytes to 2 Kbyte
  - Default size of 256 bytes
- Small directories and symbolic links can be stored directly in inodes (inlined)

IRIX System Administration: Filesystems Revision 2.0 7 – 9

Student Notes

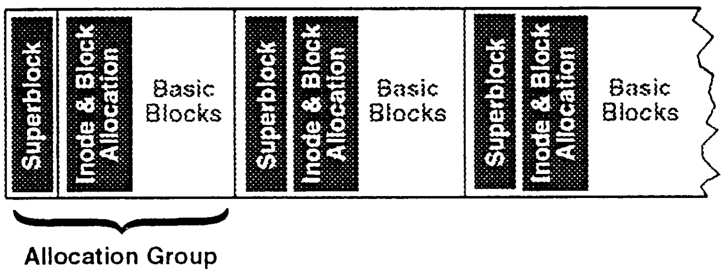
XFS is a 64-bit filesystem that increases data integrity by using *journaling* technology. Journaling is a technique that records modifications made to a file in a special area of the filesystem called a *log*. The log is a small part of the filesystem, usually a few megabytes, which is continuously updated with all modifications made to files. Periodically, as the log becomes filled, these changes are made to the disk file. This technique allows almost instant checking of a disk at boot time and much greater data integrity.

XFS also allows files and filesystems up to 1 terabyte, and for 64-bit hardware platforms files to 9 million terabytes and an unlimited size for filesystems. XFS is very flexible in its configuration, and allows inodes and basic blocks of various sizes.

7-10 SGI XFS Filesystem (continued)

SGI XFS Filesystem (continued)

- Variable basic block sizes
  - 512 bytes to 64 Kbyte
- XLV volume manager
  - Supports EFS or XFS filesystems
- Compatibility with existing applications and NFS



IRIX System Administration: Filesystems Revision 2.0 7-10

Student Notes

XFS also introduces a new logical volume manager, which replaces the EFS logical volume mechanism, and also the Volume Manager product. An XLV lets you manage large, multipartition logical volumes.

XLVs also support plexing (disk mirroring). XFS allows up to four simultaneous copies of data to exist in an XLV, which requires an additional cost license. XLVs are the topic of the Module 10, "Logical Volume Management."

7-11 Planning an XFS Filesystem

Planning an XFS Filesystem

- Block size
  - 512 bytes for small filesytems of 100 Mb or less
  - 4096 bytes default block size, use for filesytems over 100 Mb
- Log type
  - Internal log must be used for an XFS filesystem on a single disk partition
  - Internal or external log for XLV filesystem on a logical volume
- Log size
  - Depends on how the filesystem will be used, that is, how much activity/changes will be made, *not* how large or small the filesystem is
  - Minimum 512 blocks, typical 1000 blocks, high activity 2000 blocks
- Inode size
  - Default 256 bytes

IRIX System Administration: Filesystems   Revision 2.0   7 - 1 1

Student Notes

Variable-sized (set a creation time) basic blocks allow better disk usage or faster block allocation.

An important consideration in XFS filesystem construction is the size of the log. The log accumulates all the changes to the filesystem until it receives an opportunity to update those modifications to the actual files. If a filesystem has a high number of file modifications, it is beneficial to have a larger log and, conversely, a smaller log uses less disk space for a filesystem, that has few changes made to it.

7-12 Creating an XFS Filesystem

Creating an XFS Filesystem

- Creating a new XFS filesystem on a single partition
  - Use all defaults

    mkfs_xfs /dev/dsk/dks1d1s7
  - Use a block size of 2048 bytes

    mkfs -t xfs -d name=/dev/dsk/dks0d2s3 **-b size=**2048
  - Use a block size of 1024 bytes and an internal log of 2M bytes

    mkfs -d name=/dev/dsk/dks1d3s7 **-b size=**1024

    -l internal,size=2m
  - Use an inode size of 512 bytes, a block size of 8192 bytes and an internal log with a size of 1000 blocks

    mkfs_xfs -d name=/dev/dsk/dks0d4s6 **-i size=**512

    -b size=8192 **-l size=**1000b

IRIX System Administration: Filesystems Revision 2.0 7 – 1 2

Student Notes

You create an XFS filesystem using `mkfs_xfs(1M)`. Every XFS filesystem must have a log. The log partition can be internal to the filesystem, which is specified with the `-i` option. With an XLV, the log can be on another disk partition external to the filesystem data.

There are many ways to specify the creation of an XFS filesystem. The three commands are `mkfs`, `mkfs_xfs`, or `mkfs -t xfs`. There are also numerous ways to specify the options to `mkfs`: `-l internal,size=4m`, or `-l size=4m` are the same, specifying an internal log of size 4 Mbyte.

7-13 Using Filesystems

Using Filesystems

- Filesystems must be *mounted* to be used
- Mounted filesystems are part of the IRIX hierarchical directory tree
- Every filesystem needs a *mount point*
  - Typically, an empty directory where the filesystem is attached
  - Mounting a filesystem over an existing directory hides the existing file structure under the directory until the filesystem is unmounted

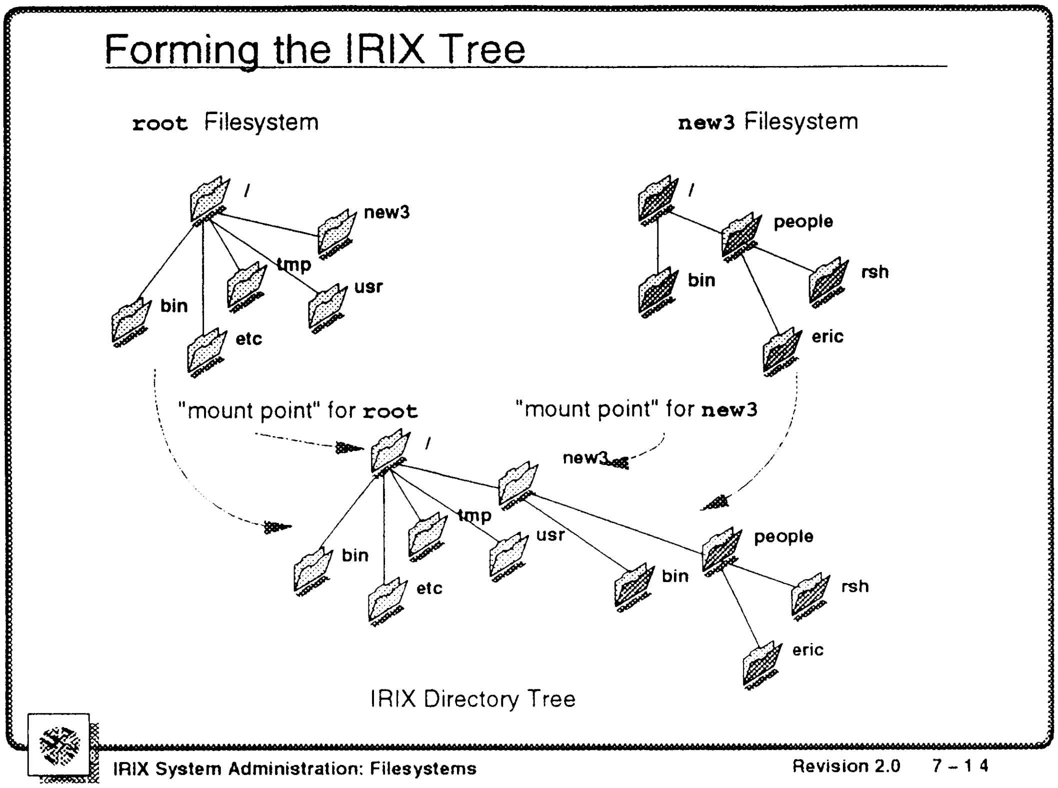
IRIX System Administration: Filesystems Revision 2.0 7-13

Student Notes

/dev/dsk/ ~~dsk0/0d~~

dks0d3s7

7-14 Forming the IRIX Tree



Student Notes

Remember, every filesystem must have a mount point.

7-15 Manual Mounting

Manual Mounting

- Filesystems can be mounted by **root**
- Mounting a filesystem at **/new3**

  `# mount /dev/dsk/dks0d2s3 /new3`
- Specifying only the filesystem or the mount point causes the system to look in **/etc/fstab** to find a corresponding entry:

  `# mount /new3`
- With the **-a** option, all filesystems in **/etc/fstab** are mounted

  `# mount -a`
- To unmount a filesystem use the mount point or the filesystem device file

  `# umount /new3` or `# umount /dev/dsk/dks0d2s3`

IRIX System Administration: Filesystems Revision 2.0 7-15

Student Notes

If only the mount point name is given to the `mount` command, it looks in the `/etc/fstab` file for the entry corresponding to `/new3` and mounts `/dev/dsk/dks0d2s3` at `/new3`.

The `mount -a` command mounts all filesystems as specified in the `/etc/fstab` file; however, the opposite command, `umount -a`, causes problems, for example, the `ps` command might fail. To fix this problem, run the `/etc/mntproc` script or the following command:

```
# mount -t proc /proc /proc
```

The `/proc` filesystem provides access to the image of each active process in the system. This filesystem was historically mounted as `/debug`. `/proc` does not consume any disk resources.

`/proc` is used by `dbx` and `CaseVision/WorkShop` (source-level debuggers) and is similar to making virtual memory a contiguous filesystem.

7-16 Automatic Mounting of Filesystems

Automatic Mounting of Filesystems

- When the system is started, it automatically mounts all filesystems specified in **/etc/fstab(4)**
- Add entries for all newly created filesystems
- Example of **/etc/fstab**:

```
/dev/root   /   xfs   rw,raw=/dev/rroot 0 0
/dev/dsk/dks0d2s3 /new3 xfs rw,raw=/dev/rdsk/dks0d2s3 0 0
```

Filesystem — Mount Point — Filesystem Type — Options — Dump Frequency — Pass

IRIX System Administration: Filesystems   Revision 2.0   7 - 1 6

Student Notes

The fields of the `/etc/fstab` file:

Field	Description
filesystem	The device file for this filesystem
mount directory	Where this filesystem is attached to the directory tree
type	Type of filesystem (`efs`, `xfs`, `nfs`)
options	Options to modify the mount: `ro` = read only `rw` = read/write `raw` = <`dev name`> = raw device file for this filesystem
freq	`dump(1M)` frequency only valid for EFS filesystems
pass	Order in which filesystems are checked parallel `fsck`

7-17 What Filesystems Are Mounted?

What Filesystems Are Mounted?

- Use the **mount(1M)** command with no arguments:

```
$ mount
/dev/root on / type xfs (rw,raw=/dev/rroot)
/proc on /proc type proc (rw)
/dev/fd on /dev/fd type fd (rw)
/dev/dsk/dks0d2s3 on /new3 type xfs (rw,raw=/dev/rdsk/dks0d2s3)
```

- Use the **df(1)** command to list mounted directories and their space usages:

```
$ df
Filesystem          Type   blocks    use      avail   %use  Mounted on
/dev/root           xfs   1939714  1818993  120721   94%   /
/dev/dsk/dks0d2s3   xfs   2019750  1683003  336747   83%   /new3
```

IRIX System Administration: Filesystems Revision 2.0 7-17

Student Notes

`mount -p` outputs in the format of the `/etc/fstab` file.

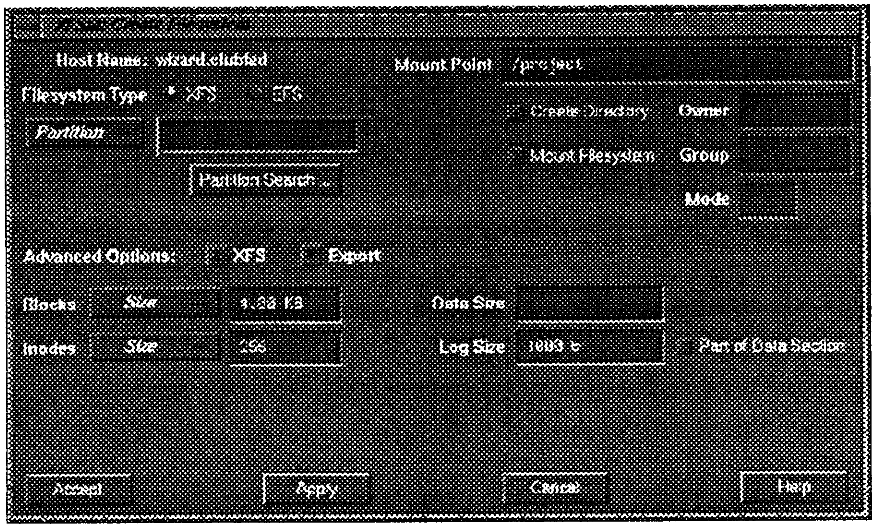
`/proc` is a filesystem that provides access to the image of each active process in the system.

`/dev/fd` refers to file descriptor files, conventionally called, for example, `/dev/fd/0`, `/dev/fd/1`. The file `/dev/fd` is available for system calls.

7-18 Using `xfsm`

Using `xfsm`

`xfsm` is a graphical tool to create a filesystem, or to manage filesystems by mounting and unmounting



IRIX System Administration: Filesystems Revision 2.0 7 - 1 8

Student Notes

`xfsm` can create EFS or XFS filesystems, create the mount point, and mount it along with setting advanced features for both types of filesystems. Because this is a graphical tool, the only information about it is through the SGI Online Help facility. Run the command, `xfsm`, and then click on Help.

For XFS filesystems, some of the advanced features are block size, inode size, log size, and whether it is internal or external.

For EFS filesystems, some of the advanced features are `suid` for superuser only, disk quotas enabled, read only, number of inodes, `fsck` pass, and dump frequency.

7-19 EFS: Filesystem Corruption

EFS: Filesystem Corruption

- Filesystem's superblock, bitmap, directory, and inode lists are on disk and in memory
- Memory version updated constantly, but disk version less frequently
- Filesystem corruption occurs when the memory and disk versions become inconsistent
- Caused by software or hardware failures
- `fsck(1M)` can clean up filesystem corruption problems
- `fsck(1M)` is only valid on EFS filesystems

IRIX System Administration: Filesystems Revision 2.0 7 – 1 9

Student Notes

Filesystem corruption can be caused by the following:

1. Brownouts and power failures
2. Improper use of the reset or power switch
3. Setting incorrect partition boundaries

7-20 EFS: When `fsck` is Run

EFS: When `fsck` Is Run

- Automatically
  - If **`fs_dirty`** bit is set, runs when booting the system
  - Runs with **`-y`** option (noninteractively, attempts to fix all problems encountered)
  - Places "orphaned" files or directories in **`lost+found`** directory for administrator to check
- Manually run without **`-y`**, on a *quiet* filesystem and answer questions on a case by case basis
  - When **`fsck(1M)`** runs automatically, and has many errors, it might give up easily. Run again manually
  - After doing disk maintenance
    - Installing new disks
    - Repartitioning disk
  - Before a backup or after a restore
  - Before *manually mounting* a dirty filesystem

IRIX System Administration: Filesystems   Revision 2.0   7 – 2 0

Student Notes

Tips for `fsck`:

On the first pass, do not run `fsck` with the `-y` option. (You can always answer *yes* later, but do not wait too long to fix problems.) Note any inode numbers reported that have problems. Manually check the inodes that are reported. On second pass, use the `y` option, or respond *yes* to

```
      SALVAGE? FIX? CONTINUE?
     RECONNECT? ADJUST?
Respond yes (with care) to

      REMOVE? CLEAR?
```

7-21 EFS: `lost+found` Files

EFS: `lost+found` Files

- *Orphaned* files and directories are stored in `lost+found` by their inode number
- Use `ls -l` or `file(1)` to determine the file type
- Use `strings(1)` on binary files
- Use `more(1)` on ASCII files
- Test executables as `guest`
  - Since `guest` cannot access `lost+found` directories by default, requires a `cp(1)` of the file to a `tmp` directory first

IRIX System Administration: Filesystems Revision 2.0 7 - 2 1

Student Notes

7-22 EFS: `fsr`

EFS: `fsr`

- `fsr(1M)` is the *filesystem reorganizer*
  - Used to reduce disk fragmentation
- `fsr` is run automatically once a week via roots cron job
  - Runs for 2 hours at a time
  - Looks at each mounted filesystem
  - `fsr` does not reorganize any inodes locked by another process
  - If `fsr` does not complete all filesystems, it records where it left off in `/var/tmp/.fsrlast` and starts there the next time `fsr` is run
  - `fsr` records fragmentation info to the SYSLOG as it completes a filesystem

Student Notes

7-23 EFS: Customizing `fsr`

EFS: Customizing `fsr`

- Edit root's **cron** job to change the default behavior of **fsr(1M)**
  - Run it more frequently
  - Run it longer with the **-t(ime)** option (time in seconds). For example, run for 4 hours:

    ```
    fsr -t 14400
    ```

  - Change the location of the **fsr** status file

    ```
    fsr -f /usr/local/adm/fsr/.fsrlast
    ```

IRIX System Administration: Filesystems Revision 2.0 7 - 2 3

Student Notes

7-24 Mounting CD-ROM Disks

Mounting CD–ROM Disks

- **mediad(1M)** controls mounting removable media (floppy, CD–ROM)
- When a inserts a CD–ROM into a drive, **mediad** attempts to mount it based on instructions from:
  - The Cadmin object server
  - **/etc/fsd.auto** (equivalent of **/etc/fstab**, but for CDs)
- **eject(1)** causes **mediad(1M)** to unmount the CD and, if successful, ejects it

IRIX System Administration: Filesystems Revision 2.0 7 – 24

Student Notes

mediad is a daemon that monitors the removable media devices on a system. When media is inserted, mediad mounts it if it makes sense for that media type and there is a valid filesystem on it. When you issue the eject command, eject sends mediad a message that causes it to attempt to unmount the media and eject it.

7-25 Mounting CD-ROM Disks (continued)

Mounting CD-ROM Disks (continued)

- Adding an entry to **`/etc/fsd.auto`**:

```
# mediad -p /dev/scsi/sc0d7l0 /CDROM
```

- Removing an entry:

```
# mediad -r /dev/scsi/sc0d7l0
```

IRIX System Administration: Filesystems   Revision 2.0   7 - 2 5

Student Notes

Use `hinv` to get the address of the CD-ROM.

Silicon Graphics
Computer Systems

Module 8: Swap Administration

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module describes how to list available swap spaces and add additional swap spaces and swap files.

8-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Administer swap space
- Create physical swap space
- Create swap files

IRIX System Administration: Swap Administration Revision 2.0 8 – 2

Student Notes

8-3 Swap Space

Swap Space

Used for temporarily saving part or all of a program when there is not enough physical memory

- To add swap space:
  - Repartition your system drive and make that swap space larger
  - Add swap space on a option drive
  - Add a swap file on a filesystem
- Having multiple swap spaces on multiple disks may help the kernel's access to swap

IRIX System Administration: Swap Administration Revision 2.0 8 – 3

Student Notes

There is no magic number for saying how much swap space is needed. The amount of swap space depends on the types of applications running on your system. You will know if there is not enough swap space configured if, in the `/var/adm/SYSLOG` file or on the `/dev/console` window, there are messages about "Out of logical swap space." Adding physical disk partitions for swap offers the best performance.

8-4 Swap Space Commands

Swap Space Commands

Use **swap(1M)** to manipulate swap spaces

- a adds additional swap space
- l lists swap space in blocks
- ln lists swap space in megabytes
- d deletes swap space

IRIX System Administration: Swap Administration Revision 2.0 8 – 4

Student Notes

The swap command provides a method of adding, deleting, and monitoring the system swap areas used by the memory manager.

The `swap -l` command lists all swap spaces in blocks. The `swap -ln` command lists all swap spaces in megabytes.

8-5 Adding Additional Swap Space

Adding Additional Swap Space

- Manually add swap space with

  swap -a swapdev [swaplow] [swaplen]

- Arguments
  - Use the block device file for **swapdev**
  - Typically, use 0 for **swaplow**, the offset into the swap space
  - **swaplen** is typically the size of the partition in blocks

IRIX System Administration: Swap Administration Revision 2.0 8 – 5

Student Notes

Usually the `swaplow` and `swaplen` values are not given, therefore swap space defaults to the beginning of the partition and uses the whole size of the partition. To get the value for `swaplen`, use the number of sectors of the partition from the `prtvtoc` command.

Example:

```
swap -a /dev/dsk/dks0d2s1.
```

8-6 Adding Physical Swap Space

Adding Physical Swap Space

- Use **fx** to create the additional partitions of type **raw**
- Use **mknod** to create device files for the swap partitions (if necessary)
- Use the **swap** command to add the physical swap space

```
# swap -a /dev/dsk/dks0d2s1 0 81534
```

or

```
# swap -a /dev/dsk/dks0d2s1
```

IRIX System Administration: Swap Administration Revision 2.0 8 – 6

Student Notes

8-7 Automatically Adding Swap Space

Automatically Adding Swap Space

- To automatically add swap space on reboot, create an entry in **/etc/fstab**. At boot time, the **/etc/init.d/swap** script adds additional swap space.
- Example **fstab** entry:

```
/dev/dsk/dks0d5s1 swap swap pri=1 0 0
```


Student Notes

8-8 Using Swap Files

Using Swap Files

- Allows the use of files within a filesystem as swap resources
- Use **mkfile(1M)** to create an empty file
  - Example, a 100 Mbyte swap file is created:

    `# mkfile 100m /swapfile1`
- Add the file as a swap resource with the **swap(1M)** command or by adding an entry to **/etc/fstab**:

  `# swap -a /swapfile1`

IRIX System Administration: Swap Administration Revision 2.0 8 – 8

Student Notes

The `/etc/fstab` entry for automatically adding the swap file is:

```
/swapfile swap swap pri=3 0 0
```

SiliconGraphics
Computer Systems

Module 9: Introduction to NFS

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module describes basic Network Filesystem (NFS) usage. You will be able to export and mount NFS filesystems.

9-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Describe the basic theory of operation between NFS servers and clients
- Configure the server system to provide access to specified directories for targeted clients
- Configure the client system to access remote files and directories

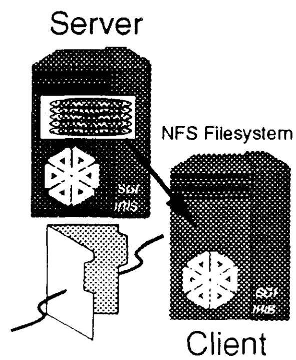
IRIX System Administration: Introduction to NFS Revision 2.0 9 - 2

Student Notes

9-3 Network Filesystem Concepts

Network Filesystem Concepts

NFS *server* system *exports* a local filesystem for use by an NFS *client* system



NFS *client* system *mounts* a remote filesystem to access informationon NFS server system

- One host system can be both client and server
- After mounting, users have transparent access to data
- NFS functions in hetergeneous environments
- Configuration is done by the system administrator

IRIX System Administration: Introduction to NFS Revision 2.0 9 – 3

Student Notes

Network Filesystem (NFS) is meant to share disk filesystems among many systems in a network environment. These systems may be of a heterogeneous environment; other systems besides Silicon Graphics machines may share files and disk filesystems.

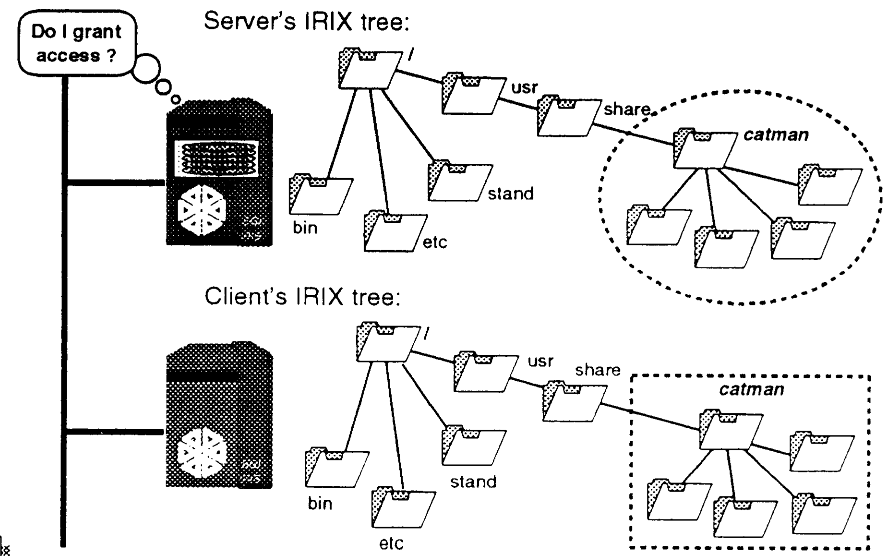
The NFS software package is an optional product.

You can export a filesystem or, more specifically, a directory.

9-4 Exporting Filesystem Structures

Exporting Filesystem Structures

Server systems give access to client systems by exporting a local filesystem or directory structure

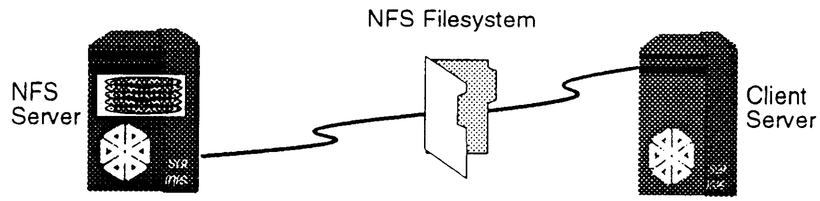


IRIX System Administration: Introduction to NFS Revision 2.0 9 - 4

Student Notes

9-5 Network Filesystem Analysis

Network Filesystem Analysis



Decide:

- Which directories or filesystem to export ?
- Which hosts can access?
- Which users can access ?
- Which clients should have read and write access ?
- Which clients have root access ?
- How should unknown users be handled ?
- Should a subfilesystem be given access to?
- Where is the mount point?
- Should remote access be read–only or read–write?
- How critical is the remote filesystem to system startup?
- Should a client process be able to timeout?
- Should a client process be able to terminate?

IRIX System Administration: Introduction to NFS   Revision 2.0   9 – 5

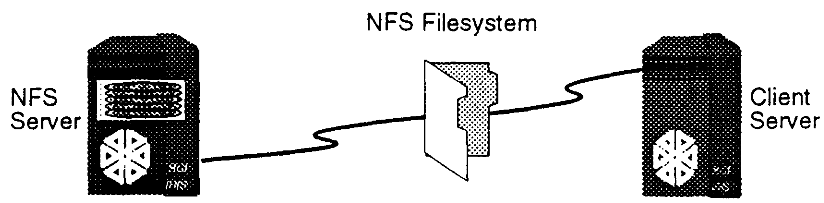
Student Notes

"Which users" really means "permissions."

Unknown users as anon are only necessary if different types of NFS platforms are being used such as PS(DOS) and SGI.

9-6 Network Filesystem Setup

Network Filesystem Setup



Steps:

- Install NFS using **`inst`**
- Activate NFS using **`chkconfig`**
- Cycle **`/etc/init.d/network`**
- Modify the **`/etc/exports`** configuration file
- Export entries using the **`exportfs`** command
- Verify status using the **`showmount`** command

- Install NFS using **`inst`**
- Activate NFS using **`chkconfig`**
- Cycle **`/etc/init.d/network`**
- Create local mount points, **`mkdir`**
- Modify **`/etc/fstab`**
- Mount remote filesystems using the **`mount`** command
- Verify status using **`df`** command

IRIX System Administration: Introduction to NFS   Revision 2.0   9 – 6

Student Notes

9-7 Enabling the NFS Server

Enabling the NFS Server

- Check NFS software installation

```
# showprods nfs.sw.nfs
```

I = Installed, R = Removed

	Name	Date	Description
I	nfs	02/03/96	Network Filesystem, 6.2
I	nfs.sw	02/03/96	NFS Software
I	nfs.sw.nfs	02/03/96	NFS Support

- Activate NFS package

```
# chkconfig nfs on
```

- Cycle the network scripts

```
# /etc/init.d/network stop
# /etc/init.d/network start
```

IRIX System Administration: Introduction to NFS Revision 2.0 9 - 7

Student Notes

NFS is an optional product, so you must first install and configure it for the networked filesystems to work.

The `chkconfig` command sets a flag so the next time the system is booted, that option will be on and valid. If you do not want to reboot, you can cycle the network script off and on to set the NFS option.

9-8 Configuring the `/etc/exports` File

Configuring the `/etc/exports` File

- Describes the filesystems that can be exported to NFS clients
- Example `/etc/exports` file:

```
# NFS exported filesystem database (see exports(4)for
more information)
# Entries in this file consist of lines containing the
following fields:
#
#filesystem  [ options ]  [ netgroup ]  [ hostname ]
...
/u1
/usr/people/guest -rw
/usr/share/catman -ro
/data  -access=frodo:nomad,rw=nomad
```

Read and write access

Read-only access

Read-only *and* read–write access

IRIX System Administration: Introduction to NFS   Revision 2.0   9 – 8

Student Notes

In the example file shown above, the filesystem `/project` has the default read/write access, and filesystem `/usr/people/guest` has read-only access for the users on the client systems `host1` and `host3` and read/write access for `host2`. The filesystem /data has read and write access for all users (all clients). The filesystems `/usr/share/catman` and `/usr/share/relnotes` have read-only access for all clients.

You may wish to export a subfilesystem (a filesystem mounted under a parent filesystem) to different clients or with different permissions than the parent filesystem is exported. To do this, you need to use the `nohide` option; otherwise, you get an error about the parent directory already being exported. The `nohide` option only works for a subfilesystem. Below is an example of its use:

```
/catman

/catman/catman2   -nohide
```

9-9 Using the exportfs Command

Using the exportfs Command

The **exportfs** command reads the **/etc/exports** configuration file to export and unexport specified directory structures

Examples:

- Export **all** filesystems listed in **/etc/exports**

```
# exportfs -a
```

- Export an individual filesystem entry

```
# exportfs /usr/share/relnotes
```

- Unexport a previously exported filesystem

```
# exportfs -u /usr/share/catman
```

IRIX System Administration: Introduction to NFS Revision 2.0 9 – 9

Student Notes

The superuser can run exportfs at any time to alter the list or characteristics of exported directories. Directories that are currently exported are listed in the /etc/xtab file.

With no options or arguments, exportfs prints the list of directories currently exported.

When you use the -a flag with the exportfs command, it exports all directories listed in /etc/exports.

9-10 Determine Export Status of Server

Determine Export Status of Server

Commands that show export status

- **showmount** display access list

```
# showmount -e
/project                (everyone)
/data                   (everyone)
/usr/share/catman       (everyone)
/usr/share/relnotes     (everyone)
/usr/people/guest       (host1,host2,host3)
```

- **exportfs** display detailed list with access *options*

```
# exportfs
/project
/data                   -rw
/usr/share/catman       -ro
/usr/share/relnotes     -ro
/usr/people/guest       -rw
```

IRIX System Administration: Introduction to NFS   Revision 2.0   9 – 10

Student Notes

The `showmount` command lists all the clients that have remotely mounted a filesystem from host (or the local host if host is not given). This information is maintained by the `mountd(1M)` server on host and is saved across crashes in the `/etc/rmtab` file.

9-11 Configuring the Client

Configuring the Client

- Determine which remote filesystem structures are available to your system
- Example from client:

NFS server name

```
nomad# showmount -x  jeeves
/u1
/usr/people/guest   -rw
/usr/share/catman   -ro
/data      -access=frodo:nomad,rw=nomad
```

- Whenever possible, use the same directory mount points on your local system

```
# mkdir /data
```

IRIX System Administration: Introduction to NFS Revision 2.0 9 – 1 1

Student Notes

9-12 Mounting and Unmounting

Mounting and Unmounting

All filesystems specified in the file **/etc/fstab** are mounted automatically when the system goes to multiuser run level

- Use the **mount** command to utilize NFS filesystems

  \# **mount** ***servername:directory*** ***mount point***

  \# mount jeeves:/data /data

- Use the **umount** command to unmount a filesystem

  \# umount ***servername:directory*** / ***mount point***

  \# umount jeeves:/data

  or

  \# umount /data

IRIX System Administration: Introduction to NFS Revision 2.0 9 – 1 2

Student Notes

`mount` attaches a named filesystem `fsname` to the filesystem hierarchy at the pathname location `dir`. The directory `dir` must already exist. It becomes the name of the newly mounted root. The contents of `dir` are hidden until the filesystem is unmounted. If `fsname` is of the form `host:path`, the filesystem type is assumed to be `nfs`.

9-13 EFS/XFS Versus NFS `/etc/fstab` Entries

EFS/XFS Versus NFS `/etc/fstab` Entries

EFS/XFS filesystem local device file	directory mount point	file system type	FS mount options	EFS dump freq.	parallel fsck pass
/dev/root	/	xfs	rw,raw=/dev/rroot	0	0
/dev/usr	/usr	xfs	rw,raw=/dev/rusr	0	0
/dev/xlv0	/u0	xfs	rw,raw=/dev/rxlv0	0	0
/dev/lv1	/u1	efs	rw,raw=/dev/rlv1	0	0

NFS server name absolute path to mount	directory mount point	file system type	NFS mount options	ignored	ignored
manserv:/usr/relnotes	/usr/relnotes	nfs	ro,bg,soft	0	0
news:/var/spool/news	/var/news	nfs	ro,bg,soft	0	0
clubted:/var/mail	/var/mail	nfs	rw,hard,intr,bg	0	0
jeeves:/data	/u/data	nfs	rw,hard,intr,bg	0	0

IRIX System Administration: Introduction to NFS   Revision 2.0   9 – 1 3

Student Notes

The file `/etc/fstab` describes the filesystems and swap partitions used by the local machine. The system administrator can modify it with a text editor. It is read by commands that mount, unmount, and check the consistency of filesystems. The file consists of a number of lines of the following form:

```
filesystem  directory  type  options  frequency  pass
```

Example:

```
/dev/root    /    xfs    rw  0 0
```

If the directory on system `jeeves` is `/data`, use the mount point `/data` on your local system.

On system `lemon`, the `/etc/fstab` file would look like this:

```
jeeves:/data /data nfs rw,intr,bg,hard 0 0
```

9-14 Mount Options

Mount Options

- `soft` mount
  - A client waits to receive an acknowledgment of a read or write to a server for a given amount of time
    - If the server is dead, client transmits again, doubling the previous wait time
    - If the server still does not answer, the client tries a default number of retransmissions before giving up and returning an error

```
zeus:/usr/share/catman  /usr/catman/zeus  nfs   ro,bg,soft   0   0
```

- `hard` mount
  - Client transmits forever until server responds
  - Interrupting a hung operation on a hard–mounted NFS filesystem is not allowed by default
- `intr` option to allow interrupts

```
skipper:/techpubs  /skipper/grp.dir  nfs   bg,intr   0   0
```

IRIX System Administration: Introduction to NFS   Revision 2.0   9 – 1 4

Student Notes

9-15 Customizing NFS Mounts

Customizing NFS Mounts

- Behavior on automatic mounts at boot
  - Foreground mount: If NFS server does not answer mount request, *hang* at this stage of the boot process, *wait* until server responds, (default)
  - Background mount: Create a *background process* to attempt to mount the NFS filesystem, and *continue* the boot process
- Set as option in **/etc/fstab**

```
zeus:/usr/share/catman /usr/share/catman/zeus nfs bg 0 0
```

IRIX System Administration: Introduction to NFS Revision 2.0 9 – 15

Student Notes

9-16 Check Mount Status With `df` and `mount`

Check Mount Status With `df` and `mount`

Both `df` and `mount` show you filesystems and mount points

```
# df
Filesystem       Type  blocks    use       avail    %use  Mounted on
/dev/root        xfs   1885177   1582869   302308   84%   /
lemon:/usr/proj  nfs   2266121   2108850   157271   93%   /usr/project
spy:/training    nfs   3309072   2831552   477520   86%   /usr/training
jeeves:/data     nfs   2058666   1412302   646364   69%   /data
```

```
# mount
/dev/root on / type xfs (rw,raw=/dev/rroot)
lemon:/usr/proj on /usr/project type nfs (ro,soft,bg)
spy:/training on /usr/training type nfs (rw,soft,bg)
jeeves:/data on /data type nfs (rw,soft,bg)
```

IRIX System Administration: Introduction to NFS Revision 2.0 9 – 1 6

Student Notes

9-17 Debugging NFS Problems

Debugging NFS Problems

- Server problems

  `# exportfs -a`

  exportfs: **/usr/people**: parent–directory (**/usr**) already exported;
  cannot export a subdirectory of an already exported directory

  `# exportfs -a`

  exportfs: unknown access list entry: stargazer exported;
  stargazer not in **/etc/hosts**

  `# exportfs -a`

  exportfs: **/usr/local/bin**: too many levels of remote in path;
  a server cannot export an NFS filesystem exported and mounted to it

- Client problems

  `# mount purple:/usr/proj /purple.proj`

  mount: (access denied for purple: **/usr/proj**
  mount: giving up on **/purple.proj**
  Filesystem not exported to client

Menu

IRIX System Administration: Introduction to NFS Revision 2.0 9 – 1 7

Student Notes

SiliconGraphics
Computer Systems

Module 10: Logical Volume Management

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module describes the creation and use of XFS logical volumes.

10-2 Module Objectives

Module Objectives

After completing the module, you will be able to

- Configure EFS and XFS logical volumes – **mklv**, **xlv_make**
- Grow filesystems – **growfs**, **xfs_growfs**
- Describe the components of an XLV
- Create an XLV from a collection of unused disk partitions
- Administer XLVs by
  - Displaying volume information
  - Adding and deleting volume elements
- Convert an existing EFS logical volume to an XLV

IRIX System Administration: Logical Volume Management Revision 2.0 10 – 2

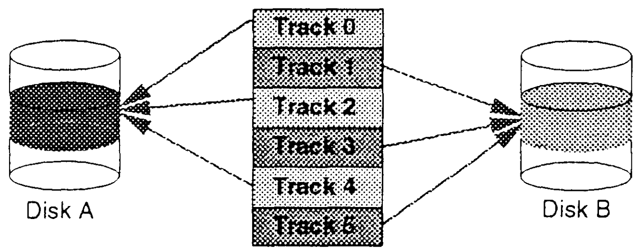
Student Notes

10-3 EFS: Types of Logical Volumes

EFS: Types of Logical Volumes

Logical Volumes – one filesystem spread across multiple disk partitions

- *Striped* Logical Volume
  - Used to create a *striped logical volume* which can give higher performance



- *Grown* Logical Volumes
  - Used to *expand* a filesystem that has outgrown its partition *without repartitioning*

IRIX System Administration: Logical Volume Management Revision 2.0 10 – 3

Student Notes

For a striped logical volume, the best performance improvement occurs if you have *more than one* controller. You are actually interleaving your data on the two disks. If the striped logical volume disks are on one controller, there is still an improvement in speed of about 20 percent.

10-4 EFS: /etc/lvtab

EFS: /etc/lvtab

- Create the `/etc/lvtab` entry:

  `vol_dev_name:[vol name]:[options]:devs=dev_pathnames`

- Sample `/etc/lvtab` entry:

```
lv0:proj one:stripes=2:devs=/dev/dsk/dks0d2s7,/dev/dsk/dks1d2s7
lv1:database:stripes=1:devs=/dev/dsk/dks0d4s2,/dev/dsk/dks0d5s3
```

Field	Description
`vol_dev_name`	Name of this logical volume `lv[0-9]`
`[vol name]`	Optional 80 character familiar name
`[options]`	Two options to striping: stripes=N (number of disks to stripe) step=Nblocks (number of blocks per write)
`devs`	Block device files of all the partitions in the logical volume

IRIX System Administration: Logical Volume Management Revision 2.0 10 - 4

Student Notes

The logical volume table is similar to `/etc/fstab`. The file `/etc/lvtab` describes the logical volumes used by the local machine. It is read by commands that create, install, and check the consistency of logical volumes.

The `lv0` example is a striped logical volume, evident from the third field, `stripes=2`. To get the best performance, each disk drive is on a different controller (`dks0d2s7` and `dks1d3s7`).

The `lv1` example is a regular (nonstriped) logical volume evident from the third field, `stripes=1`. Also notice that both disk drives are on the same controller (`dks0d4s2` and `dks0d5s3`), because this type of regular logical volume is done for convenience—expanding a filesystems space that had been limited—rather than for performance.

10-5 EFS: Striped Logical Volume Setup

EFS: Striped Logical Volume Setup

- Create an **/etc/lvtab** entry

```
lv0::stripes=2:devs=/dev/dsk/dks0d2s7,/dev/dsk/dks1d2s7
```

- Use **mklv** to create the logical volume device files

```
# mklv lv0
```

- Make a filesystem using the raw device file created by **mklv**

```
# mkfs_efs /dev/rdsk/lv0
```

- Mount the logical volume

```
# mkdir /usr1
# mount /dev/dsk/lv0 /usr1
```

- Update **/etc/fstab** to mount logical volume automatically on system boot

```
/dev/dsk/lv0 /usr1 efs rw,raw=/dev/rdsk/lv0 0 0
```

IRIX System Administration: Logical Volume Management Revision 2.0 10 – 5

Student Notes

10-6 EFS: Growing a Filesystem

EFS: Growing a Filesystem

- Unmount the filesystem that you are growing
- Create a logical volume containing that filesystem and at least one more *empty* partition
- In the **lvtab** entry, **dks0d4s2** is *existing* filesystem

  `lv1:grownfs:devs=/dev/dsk/dks0d4s2,/dev/dsk/dks0d5s3`
- Use **mklv** to create the logical volume device files

  `# mklv lv1`
- Grow the existing filesystem

  `# growfs /dev/rdsk/lv1`
- Update **/etc/fstab**

  `/dev/dsk/lv1 /grown efs rw,raw=/dev/rdsk/lv1 0 0`
- Mount the logical volume

IRIX System Administration: Logical Volume Management Revision 2.0 10 - 6

Student Notes

The `/etc/lvtab` entry for a nonstriped grown logical volume can be any of the following:

```
lv1:grown_fs::devs=/dev/dsk/dks0s4s2, /dev/dsk/dks0d5s3

lv1:grown_fs:devs=/dev/dsk/dks0d4s2, /dev/dsk/dks0d5s3

lv1:grown_fs:stripes=1:devs=/dev/dsk/dks0d4s2, /dev/dsk/dks0d5s3
```

10-7 XLVs

XLVs

- Allows filesystem to grow in sizes greater than single disks
- Provides disk striping for greater performance
- Takes advantage of XFS journaling without performance penalty
  - Creates an external log on a different disk controller
- Online reconfiguration
- Supports either **`efs`** or **`xfs`** filesystems

IRIX System Administration: Logical Volume Management Revision 2.0 10 – 7

Student Notes

XLVs can be composed of three subvolumes. The *data* subvolume is required. To increase performance or data integrity, the *log* of the XFS filesystem can be moved to its own subvolume. XLVs also support *plexing* (disk mirroring). Plexing allows much greater data availability and automatic reconstruction of damaged plexes after replacing the down volume elements. XLVs are capable of dynamic reconfiguration while still online. For example, you can add plexes to create disk mirrors and remove a plex to start with that as the base filesystem on another machine.

10-8 Associated XLV Daemons

Associated XLV Daemons

- **`xlvd`**
  - Kernel daemon
  - Handles I/O to plexes and plex error recovery
- **`xlv_labd`**
  - User space daemon
  - Responsible for updating XLV volume disk labels
- All daemons are started automatically

```
chkconfig xlv <on|off>
/etc/init.d/xlv <start|stop>
```

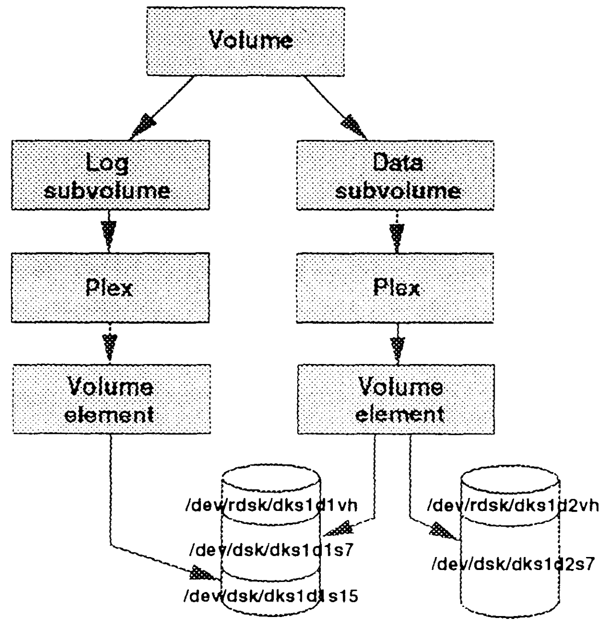
IRIX System Administration: Logical Volume Management Revision 2.0 10 - 8

Student Notes

XLVs have several system daemons which are used for handling the I/O operations and plexing.

10-9 XLV: XFS Logical Volumes

XLV: XFS Logical Volumes



IRIX System Administration: Logical Volume Management Revision 2.0 10 – 9

Student Notes

- *Volumes* are composed of *subvolumes*
- There are three types of subvolume:

Subvolume	Description
Data	Stores filesystem data
Log	Stores filesystem journal
Real-Time	Stores data using a guaranteed rate I/O stream (GRIO)

- Subvolumes are composed of *plexes.*
  - Maximum of four plexes; more than one plex requires a license
- Plexes are composed of up to 128 *volume elements*
- Volume elements can be composed of a single disk partition, multipartition (several partitions), or striped

10-10 Making an XLV

Making an XLV

- `xlv_make [script_file]`
  - Interactive
  - Support for scripting, a text file containing commands
- Creates XLV objects
  - All new components for an XLV are created with `xlv_make`
- Automates the XLV creation process
- XLV object device files reside in `/dev/[r]dsk/xlv`

IRIX System Administration: Logical Volume Management . Revision 2.0 10 - 1 0

Student Notes

XLVs are created using the interactive `xlv_make(1M)` command. This command lets you specify all parameters for the XLV and updates all necessary files, verifies disk partitions, and creates the device files.

10-11 Making an XLV (continued)

Making an XLV (continued)

xlv_make command parameters

Command	Description
vol <vol_name>	Create a new XLV volume
data	Add a data subvolume to a newly created XLV volume
log	Add an external log subvolume to a newly created XLV volume
rt	Add a GRIO subvolume to a newly created XLV volume
plex [plex_name]	Create a new plex (with a name if provided)
ve [options] <dev> ...	Create a new volume element (with a name if provided)
end	Complete the specification of a new object
show	Display all currently defined objects
exit	Exit **xlv_make**

IRIX System Administration: Logical Volume Management Revision 2.0 10 - 1 1

Student Notes

10-12 Creating an XLV From Scratch

Creating an XLV From Scratch

1. Repartition disk(s) if necessary
2. Create a new XLV object using **`xlv_make`**

Student Notes

10-13 Creating an XLV From Scratch (continued)

Creating an XLV From Scratch (continued)

```
# xlv_make
xlv_make> vol xlv0
xlv0
xlv_make> data
xlv0.data
xlv_make> plex
xlv0.data.0
xlv_make> ve /dev/dsk/dks1d1s7
xlv0.data.0.0
xlv_make> ve /dev/dsk/dks1d2s7
xlv0.data.0.1
xlv_make> end
Object specification completed
xlv_make> show
     Completed Objects
(1)  VOL xlv0 (empty)
VE xlv0.data.0.0 [empty]
         start=0, end=3928479, (cat)grp_size=1
         /dev/dsk/dks1d1s7 (3928480 blks)
VE xlv0.data.0.1 [empty]
         start=3928480, end=7856959, (cat)grp_size=1
         /dev/dsk/dks1d2s7 (3928480 blks)

xlv_make> exit
Newly created objects will be written to disk.
Is this what you want?(yes)  yes
```

IRIX System Administration: Logical Volume Management   Revision 2.0   10 - 1 3

Student Notes

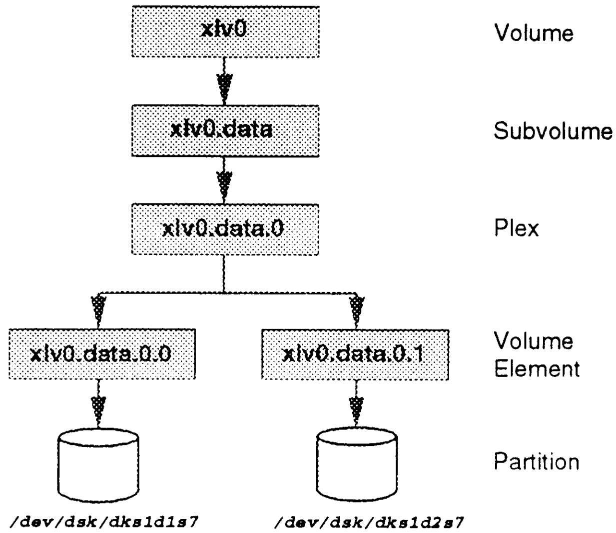
This example uses two single disk partitions concatenated versus a multipartition volume element. To create a multipartition volume element:

```
xlv_make> ve dks1d1s7 dks1d2s7
```

10-14 Creating an XLV From Scratch (continued)

Creating an XLV From Scratch (continued)

How this XLV is logically constructed:



IRIX System Administration: Logical Volume Management Revision 2.0 10 – 14

Student Notes

This XLV must use an internal log, because it was not specified in the volume definition with `xlv_make`. Use the `mkfs` command to specify the size of the internal log.

10-15 Creating an XLV From Scratch (continued)

Creating an XLV From Scratch (continued)

3. Make a filesystem on the XLV

```
# mkfs -d name=/dev/dsk/xlv/xlv0 -l internal,size=4m
meta-data=/dev/dsk/xlv/xlv0   isize=256   agcount=8,
                                  agsize=122765 blks
data     =                    bsize=4096 blocks=982120,
                                  imaxpct=25
log      =internal log        bsize=4096  blocks=1024
realtime =none                bsize=65536 blocks=0,
                                  rtextents=0
```

4. Update `/etc/fstab`

5. Mount the XLV

```
mount /dev/dsk/xlv/xlv0 /xlv0
```

IRIX System Administration: Logical Volume Management   Revision 2.0   10 - 1 5

Student Notes

If you do not specify the internal log, the default size of 1000 b (blocks) is used. The size of the internal log is a function of the filesystem block size, 1000 blocks of filesystem block size, unless the "b" is not used in the command line. Example:

```
-l internal,size=1000b means 1000 filesystem size blocks
-l internal,size=600000 means 600000 bytes
```

10-16 Administering an XLV

Administering an XLV

- `xlv_mgr`
  - Interactive
  - Support for scripting
- XLV operations
  - Showing information about XLV objects
  - Growing an XLV
  - Deleting objects from an XLV
  - Deleting an entire XLV

IRIX System Administration: Logical Volume Management Revision 2.0 10 – 16

Student Notes

Use the `xlv_mgr(1M)` command to dynamically maintain XLVs. In the previous version of XFS, the XLVs are administered by the `xlv_admin(1M)` command.

10-17 Using `xlv_mgr`

Using `xlv_mgr`

`show [-long] all`	– Display all known objects
`show [-long] labels [dks?d?vh]`	– Display XLV disk labels
`show config`	– Display XLV software configuration
`show [-verbose] object ?name?`	– Display named object
`attach ve ?src? ?dst_plex?`	– Append **ve** object "**src**" to "**dst_plex**"
`delete object ?name?`	– Delete the named object
`help or ?`	– Display this help message
`quit`	– Terminate session

IRIX System Administration: Logical Volume Management Revision 2.0 10 – 1 7

Student Notes

Only some of the commands from the `xlv_mgr` menu are shown.

`show config` - displays whether or not you have the plexing software and license installed.

10-18 Showing an XLV Configuration

Showing an XLV Configuration

- Use **xlv_mgr** to determine available volumes

```
# xlv_mgr
xlv_mgr> show -long all
Volume: xlv0 (complete)
VE xlv0.log.0.0 [active]
        start=0, end=40151, (cat)grp_size=1
        /dev/dsk/dks0d1s2 (40152 blks)
VE xlv0.data.0.0 [active]
        start=0, end=1051599, (cat)grp_size=1
        /dev/dsk/dks1d1s7 (1051600 blks)
VE xlv0.data.0.1 [active]
        start=1051600, end=2315431, (cat)grp_size=1
        /dev/dsk/dks1d2s7 (1263832 blks)
xlv_mgr>
```

IRIX System Administration: Logical Volume Management   Revision 2.0   10 – 1 8

Student Notes

There are several options to the show command. This example is one of the most useful.

10-19 Growing an XLV

Growing an XLV

- Create new volume element object with **xlv_make**
- Use **xlv_mgr** to attach additional elements to an existing XLV that is *online*
- Use **xfs_growfs** to extend the filesystem to include new elements

IRIX System Administration: Logical Volume Management Revision 2.0 10 – 1 9

Student Notes

10-20 Growing an XLV (continued)

Growing an XLV (continued)

- Original logical volume

```
% df -k
/dev/root          xfs  1105916  628284   477632   57%   /
/dev/dsk/xlv/tux   xfs  3926432     272  3926160    0%   /tux
```

- Growing the XLV
  1. Assuming the XLV is created and currently mounted
  2. Create the new volume elements to be added to the end of the XLV using **xlv_make**

IRIX System Administration: Logical Volume Management Revision 2.0 10 - 20

Student Notes

To add additional volume elements to an existing XLV, create a new volume element object, then attach it.

10-21 Growing an XLV (continued)

Growing an XLV (continued)

```
# xlv_make
xlv_make> ve disk2 /dev/dsk/dks1d4s7
xlv_make> end
Object specification completed
xlv_make> show
Completed Objects
(1) VE disk2 [empty]
    start=0, end=3928479, (cat)grp_size=1
    /dev/dsk/dks1d4s7 (3928480 blks)
(2) vol tux
VE tux.data.0.0 [active]
    start=0, end=3928479, (cat)grp_size=1
    /dev/dsk/dks1d3s7 (3928480 blks)
xlv_make> exit
Newly created objects will be written to disk.
Is this what you want?(yes) yes
Invoking xlv_assemble
```

IRIX System Administration: Logical Volume Management Revision 2.0 10 - 2 1

Student Notes

The new volume element object is called disk2 and is composed of partition `dks1d4s7`. There are now two defined objects, the original XLV, `tux`, and the new object just created, `disk2`.

10-22 Growing an XLV (continued)

Growing an XLV (continued)

3. Add this element to the volume using **xlv_mgr**

```
# xlv_mgr
xlv_mgr> attach ve disk2 tux.data.0
xlv_mgr> show object tux
VOL tux (complete)
VE tux.data.0.0 [active]
   start=0, end=3928479, (cat)grp_size=1
   /dev/dsk/dks1d3s7 (3928480 blks)
VE tux.data.0.1 [empty]
   start=3928480, end=7856959, (cat)grp_size=1
   /dev/dsk/dks1d4s7 (3928480 blks)
xlv_mgr> quit
```


Student Notes

10-23 Growing an XLV (continued)

Growing an XLV (continued)

4. Grow the filesystem using **xfs_growfs**

```
# xfs_growfs /tux
meta-data=/tux isize=256 agcount=8, agsize=61383 blks
data = bsize=4096 blocks=491060
log =internal bsize=4096 blocks=512
realtime =none bsize=65536 blocks=0, rtextents=0
data blocks changed from 491060 to 982120
```

5. Check to make sure things worked

```
# df -k
/dev/root         xfs  1105916  628284  477632  57% /root
/dev/dsk/xlv/tux xfs  5890672     400 5890272   0% /tux
```

IRIX System Administration: Logical Volume Management Revision 2.0 10 - 2 3

Student Notes

After the `xfs_growfs(1M)`, check to make sure that the size of the logical volume has increased.

10-24 Convert EFS Logical Volume to XLV

Convert EFS Logical Volume to XLV

- Rapid conversion
  - No system downtime
  - Brief logical volume downtime
  - Automated process
- Maximum filesystem size limited to EFS limit 8 Gbyte
- Data subvolume only
  - No log subvolume
- **lv_to_xlv**
  - Automatically analyzes **/etc/lvtab** and generates necessary **xlv_make** commands

IRIX System Administration: Logical Volume Management   Revision 2.0   10 - 24

Student Notes

Converting an existing EFS logical volume to XFS is simple. The end product is an XLV composed of the original EFS filesystem. The limitations of EFS (2-Gbyte files, 8-Gbyte filesystems) are still applicable to the new XLV. The benefit of creating an EFS XLV is that there is no system downtime, and archiving or restoration of files is not required. EFS logical volumes will not be supported after 6.2.

The `lv_to_xlv(1M)` command creates a script you can use with `xlv_make(1M)` to automate the process.

10-25 Convert EFS Logical Volume to XLV (continued)

Convert EFS Logical Volume to XLV (continued)

1. Run **lv_to_xlv** to generate XLV configuration
   `lv_to_xlv [-o output_file]`
2. Run **xlv_make** with output from **lv_to_xlv** to generate XLV configuration
   `xlv_make [output_file]`
3. Unmount EFS logical volume
   `umount /lv0`
4. Remove converted **efs** logical volume entry from **/etc/lvtab**
5. Modify mount line in **/etc/fstab** to reflect new device name
6. Mount XLV logical volume
   `mount /dev/dsk/xlv/lv0 /lv0`

IRIX System Administration: Logical Volume Management Revision 2.0 10 – 25

Student Notes

The `-force` option must be used in the `xlv_make` script, because the partition already contains valid data and is currently mounted.

10-26 Convert EFS Logical Volume to XLV (continued)

Convert EFS Logical Volume to XLV (continued)

```
# cat /etc/lvtab
lv0::stripes=1:devs=/dev/dsk/dks0d2s7,/dev/dsk/dks0d3s7

# lv_to_xlv -o xlv.script

# cat xlv.script
# Volume description: (null)
#
vol lv0
data
plex
ve -force /dev/dsk/dks0d2s7
ve -force /dev/dsk/dks0d3s7
end
exit
```

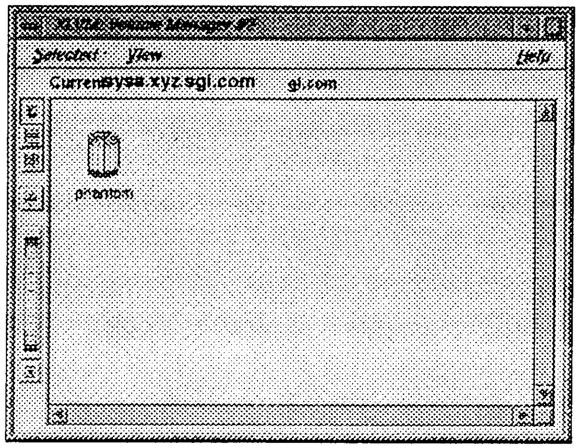

Student Notes

10-27 Using `xlvm`

Using `xlvm`

A graphical tool instead of `xlv_make` and `xlv_mgr`

- To show information about XLVs
- To add a new volume, plex, or volume element
- To attach, detach, or remove XLV components



IRIX System Administration: Logical Volume Management Revision 2.0 10 – 2 7

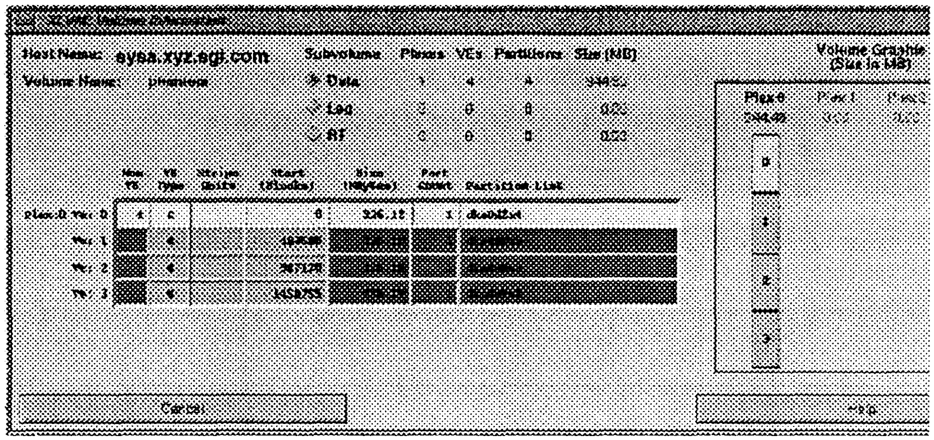
Student Notes

The graphical tool, x logical volume manager (`xlvm`), has the functionality of `xlv_make` and `xlv_mgr`.

10-28 Using `xlvm` (continued)

Using `xlvm` (continued)

To show information about XLVs



IRIX System Administration: Logical Volume Management Revision 2.0 10 – 28

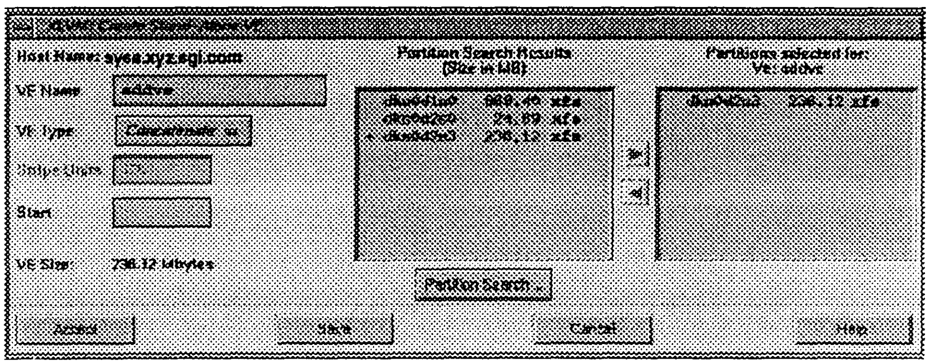
Student Notes

This example has four volume elements. Volume element 0 is highlighted in both the data size area and the graphical view.

10-29 Using `xlvm` (continued)

Using `xlvm` (continued)

Adding a new volume element



IRIX System Administration: Logical Volume Management Revision 2.0 10 - 29

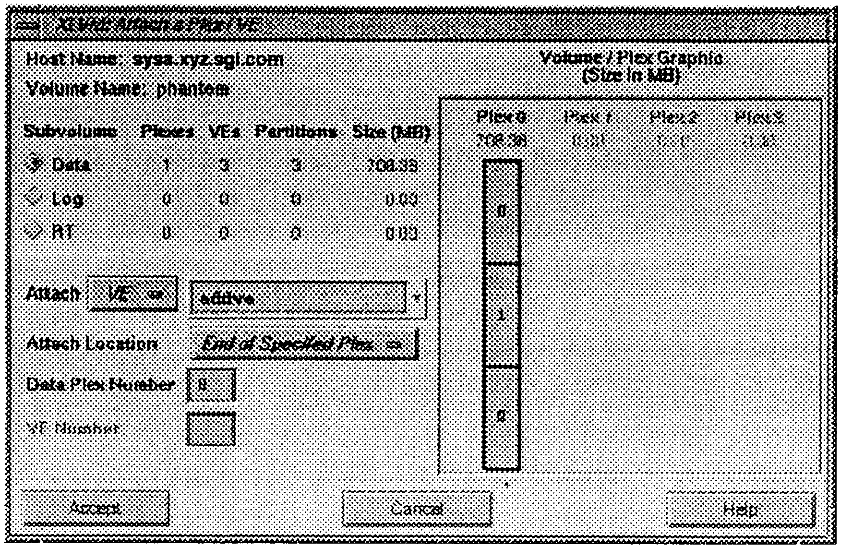
Student Notes

The `xlvm` tool works for all types of XLVs, including efs.

10-30 Using `xlvm` (continued)

Using `xlvm` (continued)

Attaching a new volume element



IRIX System Administration: Logical Volume Management Revision 2.0 10 - 3 0

Student Notes

After you create the new volume element, attach it to the end of the XLV.

SiliconGraphics
Computer Systems

Module 11: System Startup and Shutdown

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

In this module, you will learn how to startup and shutdown your system, and do some basic troubleshooting of boot problems.

11-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Describe the steps involved in the automatic boot sequence
- Stop the autoboot sequence
- Describe different reasons to manually boot the system
- Use the **boot** command to manually boot (including across the network)
- Troubleshoot hardware problems that prevent booting
- Debug the boot process in PROM, **sash**, and IRIX
- Shut down the system

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 2

Student Notes

11-3 IRIS Boot Process

IRIS Boot Process

- The three step boot process:
  - SGI/MIPS PROM monitor
  - **`sash`** – the standalone shell
  - **`/unix`** – the IRIX operating system
- Each operating environment has
  - Environment variables
  - Networking capabilities
  - Ability to move to a different environment
- Differences
  - Ability of tools
  - Ability to read IRIX filesystems
  - Location (on system disk versus PROM memory)

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 3

Student Notes

11-4 Programmable Read-Only Memory (PROM)

Programmable Read–Only Memory (PROM)

- What is PROM?
  - A program that resides in permanently programmed memory to control the startup of the machine
  - The lowest operating level on the IRIS
  - Allows access to variables and commands to change the automatic boot sequence
- What does PROM do?
  - Runs power–on diagnostics
  - Sizes and clears memory
  - Initializes hardware and graphics

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 4

Student Notes

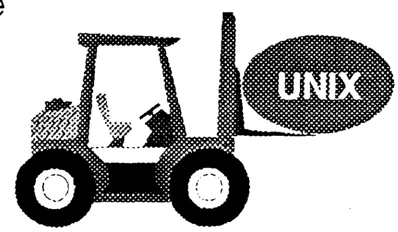
PROM runs diagnostics and boots sash. Power-on diagnostics are a series of simple tests on the CPUs and I/O interfaces. Depending on the PROM variables, this may initiate the automatic boot sequence, which then boots the program sash. Features vary from machine to machine because PROMs are not normally replaced after the manufacture of a system; therefore, newly added features will not be present on older machines.

11-5 Standalone Shell (`sash`)

Standalone Shell (`sash`)

What is **`sash`**?

- Resides on the disk as a file in the volume header, partition 8 and in **`/stand`**
- *Standalone loader* program to boot IRIX
- Understands the IRIX filesystem; reads IRIX filesystem for **`/unix`** file
- Like **`PROM`**, allows access to variables and commands to change the boot sequence



`sash` as a standalone loader

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 5

Student Notes

`sash` boots the IRIX kernel, `/unix`. The `sash` used during default booting is in the volume header, partition 8 of the system disk. A copy of `sash` is in the `/stand` directory.

11-6 IRIX - /unix

IRIX - /unix

What is **/unix**?

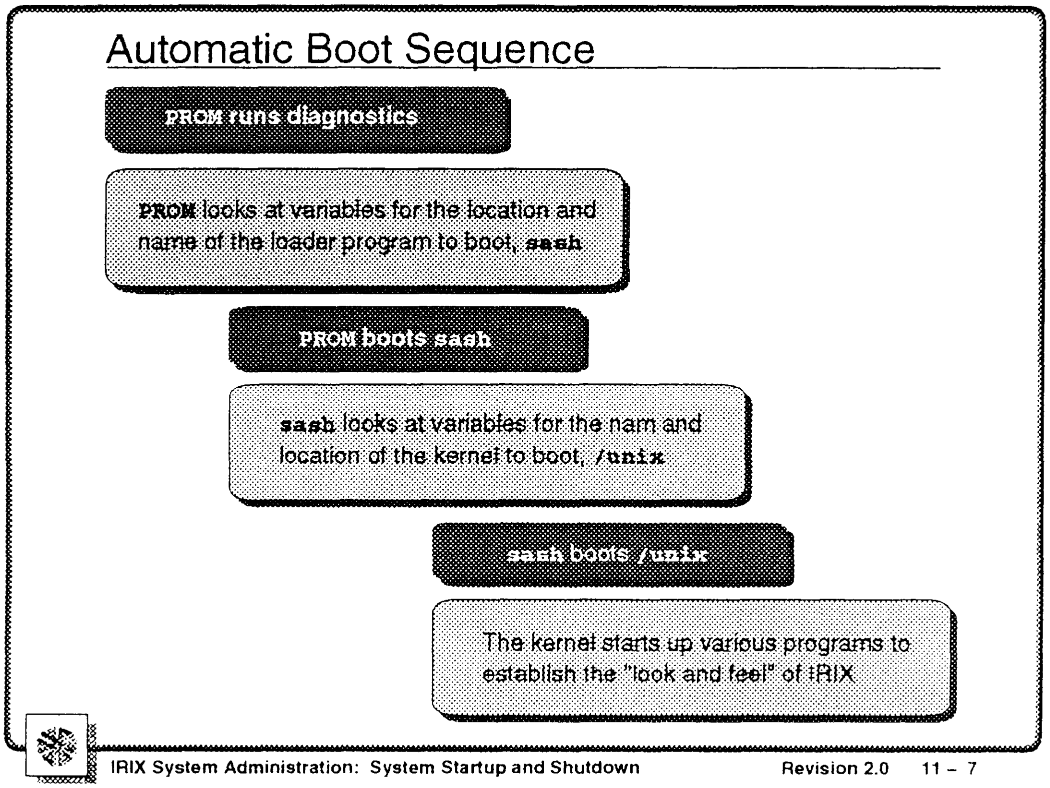
- **/unix** is the *kernel* (or the heart) of the IRIX operating system
- Resides on the system disk in partition 0 and is located in the / directory of the IRIX filesystem
- Controls access to hardware devices
- Runs programs that create multiuser, multiprocessing environment
- Allows sharing of limited resources

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 - 6

Student Notes

/unix runs programs creating IRIX's multiuser, multiprocessing environment. These programs allow limited resources (CPUs, memory, disks) to be productively shared between many programs and users. The last step of the boot process is to start the program init, which reads the configuration file /etc/inittab to start more programs.

11-7 Automatic Boot Sequence



Student Notes

The variable for the location of `sash` is `SystemPartition`; the name of the load program is `OSLoader`; the location of the IRIX kernel is `OSLoadPartition`; and the kernel name is `OSLoadFilename`. Use the `auto` command to autoboot the system manually:

```
>> auto
```

You can also manually autoboot by typing **`exit`** at the `PROM` command monitor. When the menu returns, select item 1, "`Start System`".

11-8 Stopping Autoboot

Stopping Autoboot

From power on, to stop the autoboot process:

- Click the Stop for Maintenance button

  or

- Press the <**Esc**> key

  PROM menu then displayed

From the **PROM** menu:

- Get the **PROM** prompt >> by selecting item 5, Enter Command Monitor

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 8

Student Notes

Stopping the autoboot process leaves the system in the five- or six-item PROM menu. In PROM, you can change the system configuration and manually boot the system. There are two major types of IRIS PROMs: Advanced RISC Computing Standard (ARCS) or pre-ARCS:

- ARCS - R4XXX-based machines, R5000, R8000, and R10000
- Pre-ARCS - R3XXX, R2XXX, R1XXX-based machines

The variables are different for ARCS and pre-ARCS.

11-9 PROM Commands for Variables

PROM Commands for Variables

- To display all PROM variables and values:

  `>> printenv`

- To unset a variable:

  `>> unsetenv AutoLoad`

- To change the value of a variable:

  `>> setenv AutoLoad No`

- To set variables back to the factory defaults:

  `>> resetenv`

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 9

Student Notes

```
>> printenv
SystemPartition=scsi(0)disk(1)rdisk(0)partition(8)
OSLoadPartition=scsi(0)disk(1)rdisk(0)partition(0)
OSLoader=sash
OSLoadFilename=/unix
AutoLoad=Y
netaddr=155.11.71.51
console=g
```

11-10 Modifying the Boot Sequence With Variables

Modifying the Boot Sequence With Variables

- Variables determine the boot sequence
- **AutoLoad** variable determines the default boot mode

  ```
  >> printenv AutoLoad
  AutoLoad=Yes
  ```

- Use **setenv** to modify variable values

  ```
  >> setenv AutoLoad No
  ```

- Use the **auto** command to boot the system manually using the variables

  ```
  >> auto
  ```

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 1 0

Student Notes

AutoLoad values:

Yes	Complete cold automatic boot
No	Stop in PROM

11-11 Modifying PROM Variables from IRIX

Modifying PROM Variables from IRIX

You can also change or display PROM variables from IRIX with the **nvram(1M)** command

- To list a value:

```
# /etc/nvram AutoLoad
Y
```

- To set a value:

```
# /etc/nvram AutoLoad No
```

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 1 1

Student Notes

11-12 PROM Variables

PROM Variables

The **SystemPartition** and **OSLoadPartition** variables indicate the partitions where **sash** and **/unix** will be found

```
>> printenv SystemPartition
scsi(0)disk(1)rdisk(0)partition(8)
>> printenv OSLoadPartition
scsi(0)disk(1)rdisk(0)partition(0)
```

SCSI controller

Device address

Placeholder for ARCS compliancy

Partition

IRIX System Administration: System Startup and Shutdown   Revision 2.0   11 – 1 2

Student Notes

SCSI disk controllers are the only disk devices currently supported by the ARCS PROM. sash is located by default in partition 8 (volume header) and /unix is located in partition 0 (root partition).

<disk ctlr type>(<#>)	Disk controller type and number
disk(<#>)	Disk address on the controller
rdisk(0)	Unused by SGI, place holder for ARCS compliancy
partition(<#>)	Partition number on the disk

11-13 PROM Variables (continued)

PROM Variables (continued)

The **OSLoader** and **OSLoadFilename** variables indicate the names of the operating system *loader* and the operating system *kernel*

```
>> printenv OSLoader
sash
>> printenv OSLoadFilename
unix
```

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 1 3

Student Notes

The following default values for the ARCS PROM are used on most new systems:

Variable	Default Value
SystemPartition	scsi(0)disk(1)rdisk(0)partition(8)
OSLoadPartition	scsi(0)disk(1)rdisk(0)partition(0)
OSLoader	sash
OSLoadFilename	unix

11-14 PROM Commands

PROM Commands

Use the **hinv** command from **PROM**, **sash**, or IRIX to determine what disks and tape drives you have and what the controller and drive numbers are

```
>> hinv

               System:     IP22
            Processor:     150 Mhz R4400, with FPU
 Primary I-cache size:     16 Kbytes
 Primary D-cache size:     16 Kbytes
 Secondary cache size:     1024 Kbytes
          Memory size:     96 Mbytes
             Graphics:     GU1-Extreme
            SCSI Disk:     scsi(0)disk(1)
            SCSI Disk:     scsi(0)disk(2)
            SCSI Tape:     scsi(0)tape(3)
           SCSI CDROM:     scsi(0)cdrom(4)
                Audio:     Iris Audio Processor: version
                             A2 revision 1.1.0
```

IRIX System Administration: System Startup and Shutdown   Revision 2.0   11 - 1 4

Student Notes

The above hardware inventory is from an Indigo2. No networking interfaces are shown from the PROM hardware inventory, unlike the IRIX hinv command.

11-15 More PROM Commands

More PROM Commands

>> `auto`	boots the system automatically
>> `single`	boots the system into single–user mode
>> `boot`	boots **sash** or to manually boot the system
>> `init`	restarts PROM
>> `passwd`	sets a PROM password
>> `resetpw`	removes the PROM password
>> `exit`	exits the Command Manual mode

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 1 5

Student Notes

`auto` uses the PROM variables to locate `sash` and `/unix` to boot the system.

`single` is used for maintenance activities such as backups, customizing configuration files, and recovering files and directories. If a root password exists in `/etc/passwd`, the user must provide it to enter single-user mode.

`boot` is used to manually boot the system from another disk, across the network, or by using a different `sash` or `unix` file or to boot other programs such as `fx` or `ide`. The `boot` command, with no options, boots `sash` from the PROM variables `SystemPartition` and `OSLoader`.

For added security, there is a PROM password that restricts users from entering any of the PROM menu selections except "Start System," menu item 1, unless the password is given.

`exit` or `ctrl-d` exits the PROM command monitor mode and returns to the PROM menu.

11-16 Why Boot Manually?

Why Boot Manually?

"Manual booting" means to designate which program for **PROM** or **sash** to execute:

Purpose	Program
System recovery and software installation (miniroot)	**inst**
Disk maintenance	**fx**
Use an alternate IRIX kernel for troubleshooting or tuning	Another **/unix**, either remote or local
Troubleshoot a bad default **sash**	Another **sash**, either remote or local
Hardware diagnostics	**ide**
Troubleshoot failure to completely reach multiuser mode	IRIX, with a different run level

Student Notes

11-17 Boot Command

Boot Command

Format:

`boot [-f] [device(address)file]`

- Boots the default **sash** from the **PROM** level
  `>> boot`
- Boots the default **/unix** kernel from the **sash** level
  `sash: boot`
- Can be manually modified to boot **sash**, **unix**, or other program from nondefault location

boot dksc(0,1,0)unix

load sash
device
controller
drive
partition
program (unix, fx, ide, sash, ...)

IRIX System Administration: System Startup and Shutdown   Revision 2.0   11 - 1 7

Student Notes

The device designation `dksc` applies to a SCSI disk controller device. When you are addressing the components, it is not necessary to write zeros—for example, `(,2,)` equals `(0,2,0)`, and `()` equals `(0,0,0)`.

The `boot` command loads `sash` by default. The argument to the `boot` command is passed to `sash`. To override the loading of `sash`, use the `-f` option. If the program to be booted resides in the volume header, partition 8, on the disk or on a machine across the network—not in a local filesystem—it is *not necessary* to load `sash` first. `fx`, `ide`, and `sash` may reside in the volume header of the disk. There is a copy of `sash`, `ide`, and `fx` in the `/stand` directory.

From `PROM`, you can boot a local or remote `sash` or a remote `/unix`, but you can boot a local `/unix` only if you issue a command that boots `sash` first.

11-18 Manual Boot Examples

Manual Boot Examples

```
>> boot
>> boot dksc(,1,)unix
>> boot -f dksc(0,2,8)sash
>> boot dksc(0,7,)/stand/fx --x
>> boot dksc(0,2,0)/stand/ide
sash: boot -f dksc(0,2,0)unix
```

or

```
sash: dksc(0,2,0)unix
```

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 - 1 8

Student Notes

`>>boot` boots `sash` from the default location.

`>>boot dksc(,1,)unix` boots `sash` first from `PROM` variables `SystemPartition` and `OSLoader`, then boots `/unix(IRIX)` from SCSI controller 0, disk 1, partition 0.

`>>boot -f dksc(0,2,8)sash` overrides the default loading of `sash` (`-f` option) and then boots `sash` from SCSI controller 0, disk 2, partition 8.

`>>boot dksc(0,7,)/stand/fx --x` boots `fx` from SCSI controller 0, disk 7, partition 0.

`>>boot dksc(0,2,0)/stand/ide` boots diagnostics from SCSI controller 0, disk 2, partition 0.

`sash: boot -f dksc(0,2,0)unix` overrides default loading of `sash` (because already in `sash`) and boots IRIX. Or more simply as `sash: dksc(0,2,0)unix` boots IRIX.

11-19 Manual Booting: Single-User Mode

Manual Booting: Single–User Mode

To boot into single–user mode:

- From **PROM**

  `>> single`

  or

  `>> boot dksc(0,1,0)unix initstate=s`

- From **sash**

  `sash: boot -f dksc(0,1,0)unix initstate=s`

  or

  `sash: boot dksc(0,1,0)unix initstate=s`

  or

  `sash: dksc(0,1,0)unix initstate=s`

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 - 1 9

Student Notes

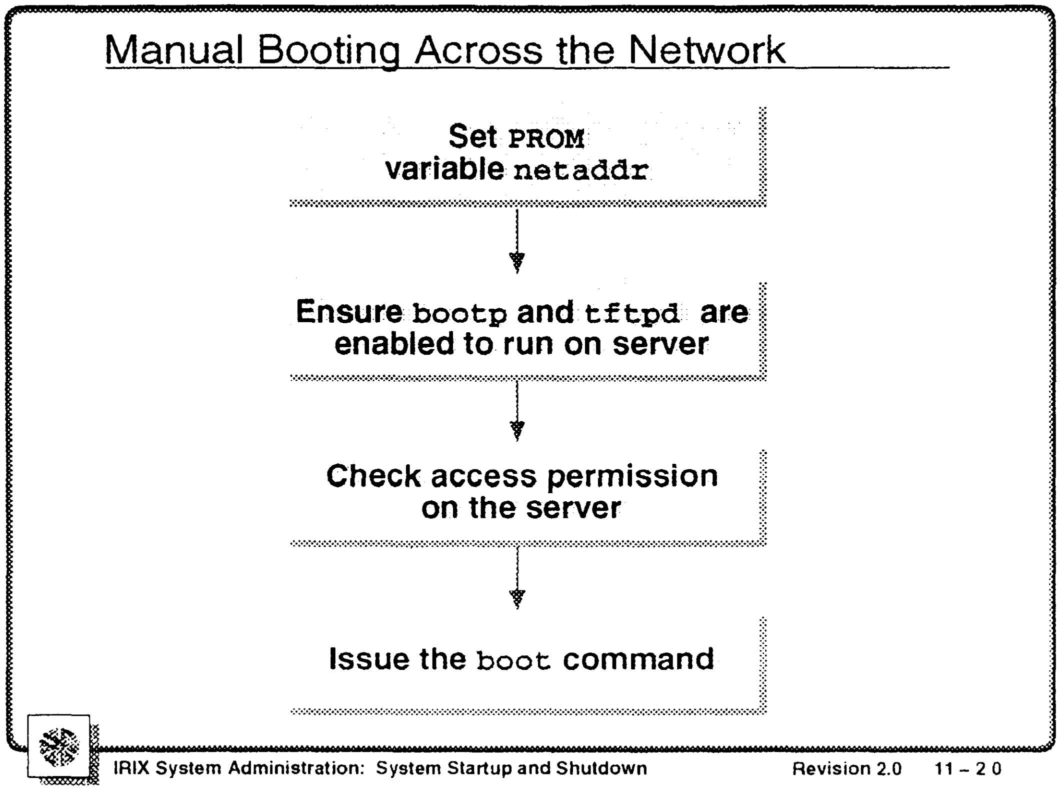
To boot into single-user mode you can also use the following commands:

`>>` **`setenv initstate s`**

`>>` **`auto`**

This variable is only set temporarily; the next time you boot, it will not use this variable.

11-20 Manual Booting Across the Network



Student Notes

`tftpd` is the trivial file transfer protocol daemon. Check to see if you have permission to access the file you are trying to boot:

- The directories listed as arguments to the `-s` option (secure mode) are the directories where another system can get files with `bootp`.
- The default open directories are `/usr/local/boot` and `/usr/etc/boot`.
- You have permission to boot `IRIX`, `sash`, and `fx` in those directories.

11-21 Manual Booting Across the Network (continued)

Manual Booting Across the Network (continued)

- Set the **PROM** variable **netaddr** to your Internet address

```
>> setenv netaddr 192.48.200.31
```

- Check **/etc/inetd.conf** for **bootp** and **tftpd** entries on the server
  - **bootp** runs on the *server* machine

```
# grep bootp /etc/inetd.conf
bootp dgram udp wait root /usr/etc/bootp bootp

# grep tftp /etc/inetd.conf
tftp dgram udp wait guest /usr/etc/tftpd tftpd \
-s/usr/local/boot /usr/etc/boot
```

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 21

Student Notes

If the lines are *uncommented* (do not having a leading # sign), the programs are enabled for this server.

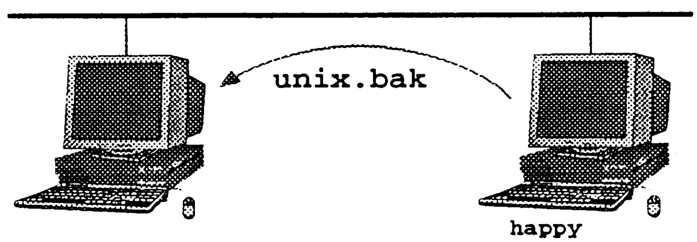
When you are checking to see if the PROM variable netaddr is set to your Internet address, you are looking on your own system, the client system.

11-22 Manual Booting Across the Network (continued)

Manual Booting Across the Network (continued)

- Issue the boot command using **bootp** instead of a disk device
- Example: Boot **unix.bak** from server **happy**

```
>> boot -f bootp()happy:/usr/local/boot/unix.bak
```



IRIX System Administration: System Startup and Shutdown Revision 2.0 11 - 2 2

Student Notes

When you are initiating a network boot, use `bootp` instead of a disk device in the `boot` command. Remember to do the following:

- Override the default loading of `sash` (use the `-f` option)
- Use the hostname of the server, not an IP address.

It is not necessary to load `sash` first, because the server is accessing the filesystem to get the `/unix` file.

11-23 Verbose Boot Information

Verbose Boot Information

- Use **showconfig** to display more verbose boot information
- Set **showconfig** variable in **PROM**, then autoboot

```
>> setenv showconfig istrue
>> auto
714 buffers configured
 64 buffer hash buckets configured

IRIX Release 6.2 1232729832
Copyright 1987-1996 Silicon Graphics Inc.
All Rights Reserved
Total real memory = 98304 kbytes
CPU Frequency = 150Mhz

SCSI disk (0,1)
SCSI disk (0,2)
SCSI tape type 2 (0,3)

ec0: 08:00:69:07:84:66
714 buffers
Available memory = 93912 kbytes
Root device 0x2000010 (fstype xfs)

Starting XLV:
```

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 - 2 3

Student Notes

More verbose information while booting the system is useful when debugging the boot process.

11-24 Boot Problems: Graphics Hardware

Boot Problems: Graphics Hardware

If the system is not displaying output to graphics monitor during boot procedure:

- Isolate suspected graphics problems by booting from an ASCII terminal on serial port 1:
  - Power down system and unhook the graphics monitor keyboard
  - Power up system
  - System switches automatically to port 1
  - A successful boot from the ASCII terminal typically indicates a graphics hardware problem
- To change the value of the **`console PROM`** variable

`d`	Console on serial port one
`g` or G	Console on the graphics monitor

```
>> setenv console d
```

- Use **`init`** to initiate the change in the console location

```
>> init
```

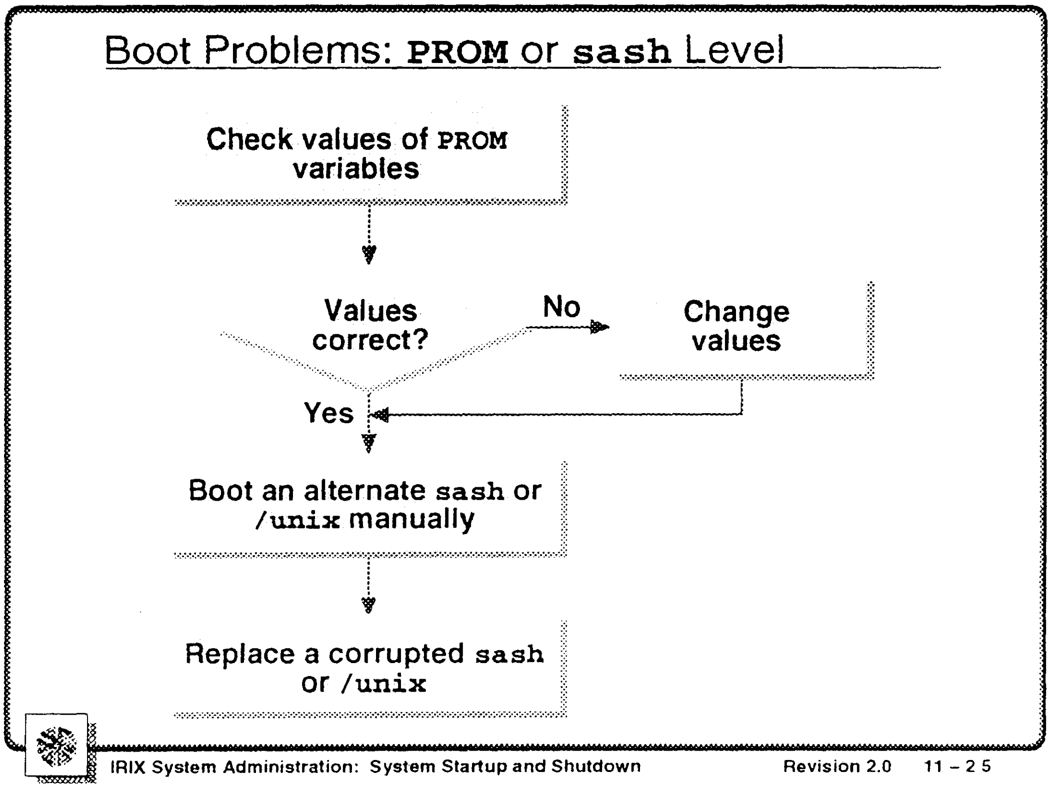

Student Notes

The console is where the system is booted from and diagnostics output is sent.

```
>> setenv console d
```

This only works on *some machines*, allowing you to switch the console in this manner. This changes the console to serial port 1, the ASCII terminal. Only boot messages go to serial port 1. The graphics monitor is still accessible in multiuser mode.

11-25 Boot Problems: PROM or sash Level



Student Notes

There is another PROM variable called `root` which is used to tell where the root filesystem is.

11-26 Debugging Boot Problems in PROM or sash

Debugging Boot Problems in PROM or sash

- Try to manually boot
  - first boot **sash** manually and if that works,
  - then boot **unix** manually
- If either one fails, try to boot **sash** or **unix** from an alternate location
  - locally

```
>> boot -f dksc(0,2,7)unix.bak
sash: dksc(0,2,7)unix.bak
```

  - remotely

```
>> boot -f bootp()happy:/usr/local/boot/unix.bak
```

Note: this assumes that prior to the problem, you created a backup unix

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 - 2 6

Student Notes

To create a backup unix, type:

```
# cp /unix /disk2/unix.bak
```

and

```
# cp /unix /usr/local/boot
```

11-27 Fixing Boot Problems: PROM or sash Level

Fixing Boot Problems: PROM or sash Level

Boot an alternate **sash** or **/unix** manually; if boot succeeds, replace the corrupted **sash** or **UNIX** by

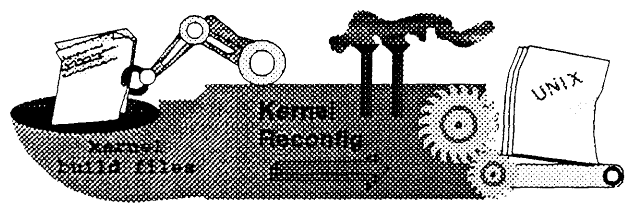
- Using **dvhtool(1M)** to copy the **sash** from an alternate location to the volume header

```
# /etc/dvhtool -v creat <alt_location>/sash sash
```

- Using **autoconfig(1M)** to *generate* a new kernel from the *build* files under **/var/sysgen**

```
# /etc/autoconfig -v
```

  - Reboot the system to use the new kernel



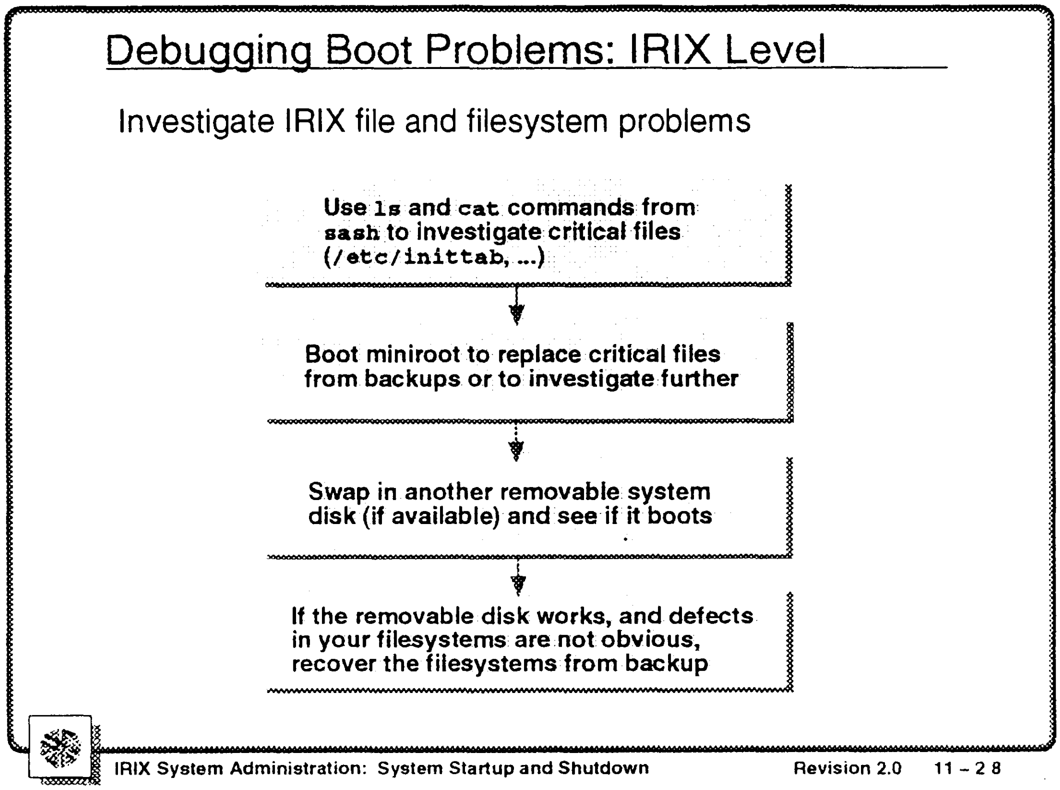
IRIX System Administration: System Startup and Shutdown   Revision 2.0   11 - 2 7

Student Notes

`dvhtool` is the disk volume header tool. It is used to modify and obtain disk volume header information. It is recommended that after software installation, you extract a copy of sash from the volume header of your system disk and place it in your UNIX filesystem (in prior versions of IRIX, it used to be placed in `/stand`. This may be a good place to keep a copy).

`/etc/autoconfig` is the recommended way to create a new `/unix` kernel file. `/etc/autoconfig` calls the `lboot` command. Before reconfiguring the kernel, make a copy of your old `/unix` file to a backup.

11-28 Debugging Boot Problems: IRIX Level



Student Notes

You can look at system files from `sash`, for example:

```
sash: cat dksc(0,1,0)etc/inittab
sash: cat dksc(0,1,0)etc/passwd
sash: ls dksc(0,1,0)
```

11-29 Shutting Down the System

Shutting Down the System

Why shut down the system?

- For power off
- For freeing up a hung system
- For single–user maintenance
- For **PROM** or **sash** maintenance

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 2 9

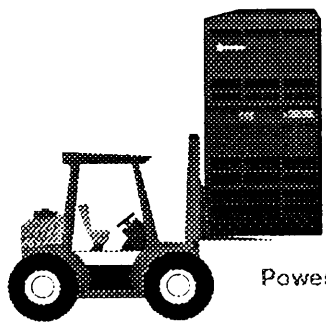
Student Notes

11-30 Power Off

Power Off

Why power off the system?

- Physically moving the system
- Adding and removing disk/tape drives, memory, boards
- Performing system hardware maintenance



Power down before moving a system

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 3 0

Student Notes

11-31 Power Off (continued)

Power Off (continued)

- Become root
- Check for local users
  - See who is on the system and what processes are running: **who**, **ps**
  - Notify specific users with the **write** command if necessary. Use **wall(1)** to notify local and remote users logged in *locally*
- **init** 0 or **halt**

  or
- **shutdown -y -g300**
- Power off the system

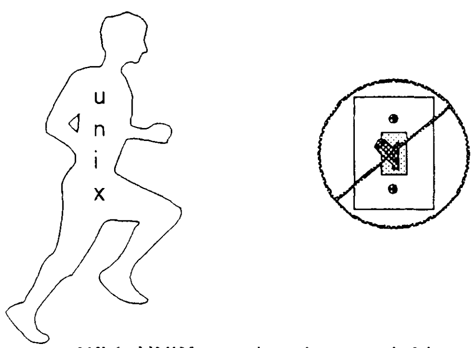
Student Notes

Check for remote (network) users. Remember that shutting down your machine may affect other machines if it is an NFS server, tape or printer server, or diskless server. To see who is on the local area network and what the status of the machines are, use rwho(1C) and ruptime(1C) (requires rwhod daemon running on all machines, which is not configured by default), rusers(1C) and rup(1C) (requires NFS software installed). Use rwall(1) to warn specific remote users. Verify that all users are off.

11-32 Freeing Up a Hung System

Freeing Up a Hung System

- Pressing Reset or power cycling when the system is hung should *only* be done as a last resort after
  - You have tried to use all other terminals
  - You have tried to remotely log in
  - You have waited 1 minute for **`bdflush`** to update the disk



With UNIX running, be careful in power cycling!

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 - 3 2

Student Notes

Indys and Indigo2s are generally safe to power down or reset using the power switch. A signal is sent to the system telling it that a shutdown or reset is imminent and that it should save cached data in memory. To be safe, it is best to safely shut down the system using the `shutdown` command.

For many other machines, it is safe to press the reset button or cycle the power switch *only* when in `PROM` or `sash`. This is a "last resort" type of action.

11-33 Single-User Mode

Single–User Mode

- Follow same procedures previously listed to make sure all the users are off the system
- `init s, init S, init 1`

  or

  `shutdown -g500 -y -iS`
- The system goes to single–user mode, prompt for the root password, and then display the root prompt # if you successfully log on
- To return to multiuser mode, `init 2`

  or
- To go to **PROM**, `init 0`

IRIX System Administration: System Startup and Shutdown Revision 2.0 11 – 33

Student Notes

Generally, you should go to single-user to recover, install, and update add-on software or user files; complete backups of the system; reconfigure networking software; or to troubleshoot problems with daemons and networking.

SiliconGraphics
Computer Systems

Module 12: IRIX Run Levels

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module discusses the IRIX run levels, used to customize the user environment on your SGI system.

12-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Utilize IRIX run levels
- Change IRIX run levels
- Describe the types of programs started by **init**
  - Starting and stopping certain daemons
  - Send options to daemons
- Describe the flow of the run–level scripts for the important system run levels

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 2

Student Notes

In this module, you will learn the purpose of IRIX run levels and the most common IRIX run levels. You will learn to recognize which run level you are using and how run levels are created with `init(1M)` and `inittab(4)`.

`init` is a general process spawner. Its primary role is to create processes from information stored in the `inittab` file. The `/etc/inittab` file supplies the script to `init`'s role as a general process dispatcher.

12-3 Purpose of Run Levels

Purpose of Run Levels

Create unique environments for different tasks:

- Multiuser run level for multiple users and multiple processes
- Single–user run level for system maintenance, software installation, and troubleshooting
- Shutdown run level for system shutdown
- Reboot run level to re–initialize the system

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 3

Student Notes

12-4 Most Common IRIX Run Levels

Most Common IRIX Run Levels

- At any given time, the system is in one of these states and can be sent to another using the **`/etc/init`** command
- **`init`** starts or stops the programs necessary to create a particular *run level*

Run Level	Name
0	Shutdown level
1	Single user
S,s	Single user
2	Multiuser
6	Reboots the system

IRIX System Administration: IRIX Run Levels Revision 2.0 12 - 4

Student Notes

Run Level 0: Shut down the machine so it is safe to turn off the power. Have the machine remove power if it can.

Run Level 1: Put the system into system administrator mode. Only a small set of essential kernel processes run. This mode is for administrative tasks such as installing optional utilities packages. All files are accessible, and no users are logged in on the system. The device file `/dev/console` is linked to the monitor or terminal, whichever is defined by the `PROM` variable console.

Run Level S or s: Enter single-user mode. When the system changes to this state as the result of a command, the terminal from which the command was executed becomes the system console. This is the only run level that does not require the existence of a properly formatted `inittab` file. If this file does not exist, then, by default, the only legal run level that `init` can enter is the single-user mode. `/dev/console` is linked to whatever terminal/monitor the initiator was on when you changed run levels.

Run Level 6: Stop the IRIX system and reboot to the state defined by the `initdefault` entry in `inittab`.

Run Level Q or q: Re-examine or re-read the `/etc/inittab` file.

Run Level a,b, or c: Process only those `inittab` entries for which the run level is set to a, b, or c. These are pseudo-states, which may be defined to run certain commands, but which do not cause the current run level to change. These are rarely used.

12-5 Determining Your Run Level: who

Determining Your Run Level: who

- Use the `-r` option to `who(1)`

How many times at this run level since last reboot

Current run level

```
# who -r
    run-level 2  Feb 27 13:52  2  0  S
```

When you got to current run level

Previous run level

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 5

Student Notes

The system administrator can send `init` a signal to change run levels:

 # init *<value>*

Valid values are 0-6, Q, q, S, s, a, b, c. Only 0-6, S, s are considered run levels. The Q, q tells `init` to re-read the `/etc/inittab` file because the system administrator has made changes.The a, b, c are pseudo-levels and are rarely used. You can use these pseudo-run levels to start daemons, but do not change the run level. The `telinit` command is linked to `/sbin/init`.

12-6 How Run Levels Are Created

How Run Levels Are Created

- `init` starts or stops the programs necessary to create a particular run level
- `init` is started at boot time to create the default run level
- `init` stays in the background to
  - Answer requests from the system administrator to change run level
  - Recreate processes that have died and need to be respawned for the current run level
- `init` consults the file `/etc/inittab` for information about how to create different run levels

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 6

Student Notes

12-7 When `init` Reads `/etc/inittab`

When `init` Reads `/etc/inittab`

- At boot time
  - **`init`** is started by the scheduler
  - Reads default run level from **`/etc/inittab`**
  - Refers to **`/etc/inittab`** and based on the default run level, starts up the necessary processes
- Changing the run level causes
  - **`init`** to re-read **`/etc/inittab`**
  - **`init`** to start and kill processes based on new run level
- When a process spawned by **`init`** dies, sending **`init`** a signal
  - **`init`** re-reads **`/etc/inittab`** to decide whether to start up a new process to replace the one that died

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 7

Student Notes

At boot time, `init` is started by the scheduler, `sched` (PID 0). The default run level is read from the `/etc/inittab` file, and the necessary processes are started based on that run level.

When the run level is changed by issuing the `init`, `reboot`, or `shutdown` commands, `init` re-reads the `/etc/inittab` file, and starts up or shuts down processes based on that new run level.

12-8 Examining /etc/inittab

Examining /etc/inittab

- First line is the **initdefault** line (default run level on booting):

  `is:2:initdefault:`

- Format of other lines:

  `id : rstate : action : process`

  - **id** field is a two– to four–character unique identification
  - **rstate** field defines run level(s) to which entry applies
  - **action** field contains keywords that tell **init** how to treat processes started by this entry
  - **process** is the shell and program to be executed

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 8

Student Notes

When `init` looks at the run-level field, the following happens: If run level equals `rstate`, `init` spawns `process`. If `rstate` field is null, `init` spawns the `process` for all run levels. If there are two digits in the `rstate` field, `init` sees *each* as a level and spawns the `process` for either. For example, `init` spawns `/etc/rc2` for levels *2 and* 3, *not* level 23.

```
s2:23:wait:/etc/rc2 >/dev/console 2>&1 </dev/console
```

Troubleshooting:

If the run level is "x," `init` ignores the entry. If `/etc/inittab` is missing, `init` starts the system in single-user mode. If there is no run level in the `initdefault` field (`is::initdefault:`), the machine uses 6 as the run level, therefore booting up and then down, up, down, etc. You need to boot into single-user mode, edit the file, and then change to multiuser mode.

If the run level is invalid, `init` prompts user for a run level:

```
is:9:initdefault
```

12-9 `inittab` Action Field

`inittab` Action Field

`init` looks for keywords in the *action* field to see how it should spawn a process

`sysinit`	Run process before `init` sends anything to the system console
`bootwait`	Start process the first time `init` goes from single–user to multiuser state after the system is booted
`wait`	When going to run level, start process and wait until it is finished before continuing with next line
`respawn`	If process does not exist, start it, wait for it to finish, and then start it again
`off`	When in level, kill process or ignore it
`once`	Run process once and do not start it again even if it finishes

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 9

Student Notes

`sysinit`: Entries of this type are executed before `init` tries to access the console (that is, before the Console Login prompt). It is expected that you will use this entry only to initialize devices on which `init` might try to ask the run-level question. These entries are executed and waited for before continuing.

`bootwait`: The entry is to be processed the first time `init` goes from single-user to multiuser state after the system is booted. (If `initdefault` is set to 2, the process runs right after the boot.) `init` starts the process, waits for its termination and, when it dies, does not restart the process.

`wait`: When `init` enters the run level that matches the entry's `rstate`, start the process and wait for its termination. All subsequent reads of the `inittab` file while `init` is in the same run level cause `init` to ignore this entry.

respawn: If the process does not exist, then start the process—do not wait for its termination—and continue scanning the `inittab` file. When this process dies, restart the process.

off: If the process associated with this entry is currently running, send the warning signal (`SIGTERM`) and wait 20 seconds before forcibly terminating the process with the kill signal (`SIGKILL`). If the process is nonexistent, ignore the entry.

once: When `init` enters a run level that matches the entry's `rstate`, start the process—do not wait for its termination. When it dies, do not restart the process. If entering a new run level (where the process is still running from a previous run-level change), the program will not be restarted.

Other valid actions are `boot`, `powerfail`, `powerwait`, and `ondemand`.

12-10 A Closer Look at /etc/inittab

A Closer Look at /etc/inittab

```
#       Copyright (c) 1984 AT&T
#       THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF AT&T

is:2:initdefault:
fs::sysinit:/etc/bcheckrc </dev/console >/dev/console 2>&1
mt::sysinit:/etc/brc </dev/console >/dev/console 2>&1
link::wait:/etc/lnsyscon > /dev/console 2>&1 < /dev/null
s0:06s:wait:/etc/rc0 >/dev/console 2>&1 </dev/console
s1:1:wait:/etc/shutdown -y -iS -g0 >/dev/console 2>&1
    </dev/console
s2:23:wait:/etc/rc2 >/dev/console 2>&1 </dev/console
s3:3:wait:/etc/rc3 >/dev/console 2>&1 </dev/console
of:0:wait:/etc/uadmin 2 0 >/dev/console 2>&1 </dev/console
RB:6:wait:/etc/init.d/announce restart
rb:6:wait:/etc/uadmin 2 1 >/dev/console 2>&1 </dev/console
# comments
tp:23:off:/sbin/getty tport co_9600              # textport
#
# Use the ttym* or ttyf* device names and the du_* or dx_* gettydefs
# tags for ports with modems.  See the getty(1M), uugetty(1M),
# init(1M),gettydefs(4), and inittab(4) man pages.
#on-board ports or on Challenge/Onyx MP machines, first IO4 board ports
t1:23:respawn:/sbin/getty ttyd1 co_9600    # alt console
t2:x:respawn:/etc/getty -N ttyd2 co_9600 # port 2
t3:23:off:/etc/getty -N ttyd3 co_9600    # port 3
t4:23:off:/etc/getty -N ttyd4 co_9600    # port 4
```

ID
RUN STATE
ACTION
PROCESS

IRIX System Administration: IRIX Run Levels — Revision 2.0 — 12 – 1 0

Student Notes

12-11 System Initialization Scripts

System Initialization Scripts

Some system initialization scripts are run by **init(1M)** only at system boot time

- **/etc/bcheckrc**
  - Checks status of root filesystem
  - Creates device files via **MAKEDEV** script
  - Initializes logical volumes
- **/etc/brc**
  - Initializes the mounted filesystem table **/etc/mtab**
  - Checks for existence of **/etc/fstab** file, creates one if not there
  - Mounts the **/proc** filesystem
- **/etc/lnsyscon**
  - Links the system console **/dev/syscon** to **/dev/console** device

IRIX System Administration: IRIX Run Levels Revision 2.0 12-11

Student Notes

The /etc/brc file is used to determine which filesystems to mount. If the file /etc/fstab does not exist, a default file is created with root and usr entries (if /usr entry is applicable).

12-12 Run-Level Scripts

Run–Level Scripts

- Run–level scripts mount and unmount filesystems, start and stop processes, and run programs from the run–level directories
- The script executed depends on the run level. For example, if going to run level 2, the script **/etc/rc2** will be run
- The **/etc/init.d** scripts typically affect the operation of system daemons
  - `start` starts daemons and mounts filesystems
  - `stop` stops daemons and unmounts filesystems

```
# /etc/init.d/network start
```


Student Notes

Run-level scripts are executed after the system initialization scripts. Daemons give a run level its particular "flavor." A daemon is generally a program that "sleeps" most of the time, rather than runs on the CPU and "wakes" on a signal to perform some action. Names of daemons typically end with a *d* to denote *daemon*. Example: `lpsched`, `rwhod`, `nfsd`.

Daemons that are started for a particular run level often depend on the settings of certain configuration flags that are read by the `/etc/init.d` scripts.

12-13 Run-Level Directories

Run–Level Directories

- Contain symbolically linked names to scripts in the **/etc/init.d** directory

Flow of Execution:

`/etc/rc2` → `/etc/rc2.d` → `/etc/init.d`

`/etc/rc2.d/S75cron --> /etc/init.d/cron`

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 1 3

Student Notes

Links in the `/etc/rc2.d` directory begin with an *S*; links in the `/etc/rc0.d` directory begin with a *K*. *S* links create programs that should be started for the current run level. *K* links remove programs that should be killed for the current run level; *K* links also kill specified daemons. *S* or *K* is followed by two digits. This allows *S* and *K* links to be executed in ASCII collating sequence. The script to which the *S* and *K* files are linked is in `/etc/init.d`.

12-14 Run-Level Directories (continued)

Run–Level Directories (continued)

```
# ls -CF /etc/rc2.d
S00announce@       S28outbox@         S50mail@           S95availmon@
S00disk_patch@     S30network@        S58rmtmpfiles@     S96fontserver@
S04usr@            S31proclaim@       S60lp@             S97cadmin@
S12filesystems@    S32aliases-ip@     S61bsdlpr@         S97mediad@
S14quotas@         S35audit@          S64dynaweb@        S98midi@
S16postinst@       S40nck@            S70uucp@           S98rfindd@
S20sysetup@        S45flexlm@         S75cron@           S98rtmond@
S21perf@           S45netls@          S88configmsg@      S98videod@
S22acct@           S48savecore@       S90chkdev@         S98xdm@
S23autoconfig@     S49swap@           S94grio@           S99cachefs@
```

- The link name is completed with the name of a script in **`/etc/init.d`** that the link points to

```
# ls -l /etc/rc2.d/S04usr
lrwxr-xr-x 1 root sys 13 Feb 7 15:27 /etc/rc2.d/S04usr -> ../init.d/usr
```

IRIX System Administration: IRIX Run Levels   Revision 2.0   12 – 14

Student Notes

12-15 Sending Flags to `/etc/init.d` Scripts

Sending Flags to **/etc/init.d** Scripts

- Done with **chkconfig(1M)** command
- Existing flags have a configuration file
  /etc/config/<flag>
- To turn *on* a flag

  `# chkconfig nfs on`
- To turn *off* a flag

  `# chkconfig nfs off`

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 1 5

Student Notes

Flags are read by the `/etc/init.d` scripts to configure the system daemons. An *on* in the configuration file turns on the flag; an *off* turns it off. Running `/etc/chkconfig` with no arguments prints the option settings (flags depend on OS release and subsystems installed).

```
# chkconfig
```

Flag	State
desktop	on
mediad	on
network	on
nfs	off
xlv	on

12-16 Supplying Options and Arguments to Daemons

Supplying Options and Arguments to Daemons

- Daemons started with an **/etc/init.d** script, read options and arguments from an options file
  - For example, if you want the **timed(1M)** daemon to run with options and arguments:

    -M -P /usr/tmp/.timetrim

- Put the options and arguments in a file called `/etc/config/<daemon/program name>.options`
  - Example:

    /etc/config/timed.options

IRIX System Administration: IRIX Run Levels Revision 2.0 12 – 1 6

Student Notes

Check the man page describing the daemon/program for the proper name of the options file and options to set.

SiliconGraphics
Computer Systems

Module 13: Printers & Terminals

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

You will learn how to configure the software to run printers and terminals from your SGI workstation.

13-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Understand the files, directories, and commands involved
- Define the spooler system and use basic spooler terminology
- Set up `lp` printers using the manual method and scripts
- Set up printer classes
- Use the spooler system to control print jobs
- Debug problems with the printer spooler
- Make and print screen images
- Set up a terminal

IRIX System Administration: Printers & Terminals Revision 2.0 13 - 2

Student Notes

13-3 Printer Setup

Printer Setup

- Manual method

  /usr/lib/lpadmin

- Graphical Printer Manager

  /usr/sbin/printers

- Scripts

  /usr/sbin/mknetpr

  /usr/sbin/mkcentpr

  /usr/sbin/mkserialpr

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 3

Student Notes

Printers can be set up using various methods. These methods include the manual procedure, using scripts, using `printers`, and the graphical printer manager available from the Toolchest. The interactive script, `mkserialpr`, adds a serial printer.

13-4 Manual Printer Installation

Manual Printer Installation

- Change ownership and permissions on device file
- Modify `/etc/inittab` to turn off `getty(1M)`
- Execute `init q` command to update the `init` process
- Make physical connection
- Configure printer
- Configure spooler
- Test printer

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 4

Student Notes

13-5 Change Ownership and Permissions

Change Ownership and Permissions

- Choose a serial port to use for the printer
- Look at the permissions of the device file for the serial port

```
# ls -l /dev/ttydN
```

- Change ownership to **lp** and group ownership to **sys**

```
# chown lp.sys /dev/ttydN
```

- Change permissions

```
# chmod 600 /dev/ttydN
```

- Check your work

```
# ls -l /dev/ttydN
# crw------- 1 lp sys 0,2 Feb  3 09:32 /dev/ttyd2
```

IRIX System Administration: Printers & Terminals Revision 2.0 13 - 5

Student Notes

Serial port one, `ttyd1`, is the alternate console.

There are two basic printer cable configurations for the DB-9 serial ports and one for the Mini-DIN8 serial port.

13-6 Modify /etc/inittab

Modify /etc/inittab

- Change second field (rstate) to **x** or third field (action) to **off**

  `t2:x:respawn:/etc/getty -N ttyd2 co_9600` # port 2

  `t3:23:off:/etc/getty -N ttyd3 co_9600` # port 3

- Re–read the configuration file **/etc/inittab**

  `# init q`

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 6

Student Notes

`init -q` causes `init` to re-examine `/etc/inittab`.

13-7 Configuring the Printer

Configuring the Printer

Printer interface programs in **/var/spool/lp/model**

- Programs copied to **/var/spool/lp/interface**
- Shell scripts
- Massages and filters data, prints the banner page
- Prepares output for different model printers

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 7

Student Notes

`/var/spool/lp` is the directory for the System V Spooling program.

Interface programs are Bourne shell scripts; examples include `dumb`, `gpsinterface`, `plotter`, `netface`, `laserwriter_model`, and `deskjet_model`.

13-8 Spooler Terminology

Spooler Terminology

- Spooler – Print scheduler, controls the jobs from the printer queue to the printer
- Printer – Unique name of interface program
- Device – Physical port
- Queue – A list of print jobs waiting to be printed on a particular printer
- Class – Group of printers or queues
- Destination – Specific printer or class
- Request ID – Unique name for the print job

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 8

Student Notes

The purpose of print spooling is to allow many users to share printer(s), to increase productivity, to allow flexible queuing of print requests, and to provide the ability to monitor and cancel print requests.

Some color printers take direct screen dumps, using the RGB output directly from the computer. These printers do *not* use the spooler.

13-9 Print Spooler Commands

Print Spooler Commands

- Commands executable by any user

– `lp`	Sends a job to a printer's queue
– `cancel`	Cancels a print job
– `lpstat`	Print LP status information

- Commands executable *only* by `root`

– `enable`	Enable printer for use by scheduler
– `disable`	Disable printer from use by scheduler
– `lpshut` `lpsched`	Turn off/on spooler
– `reject` `accept`	Reject/accept jobs to queue
– `lpmove`	Move printer requests from printer to printer

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 9

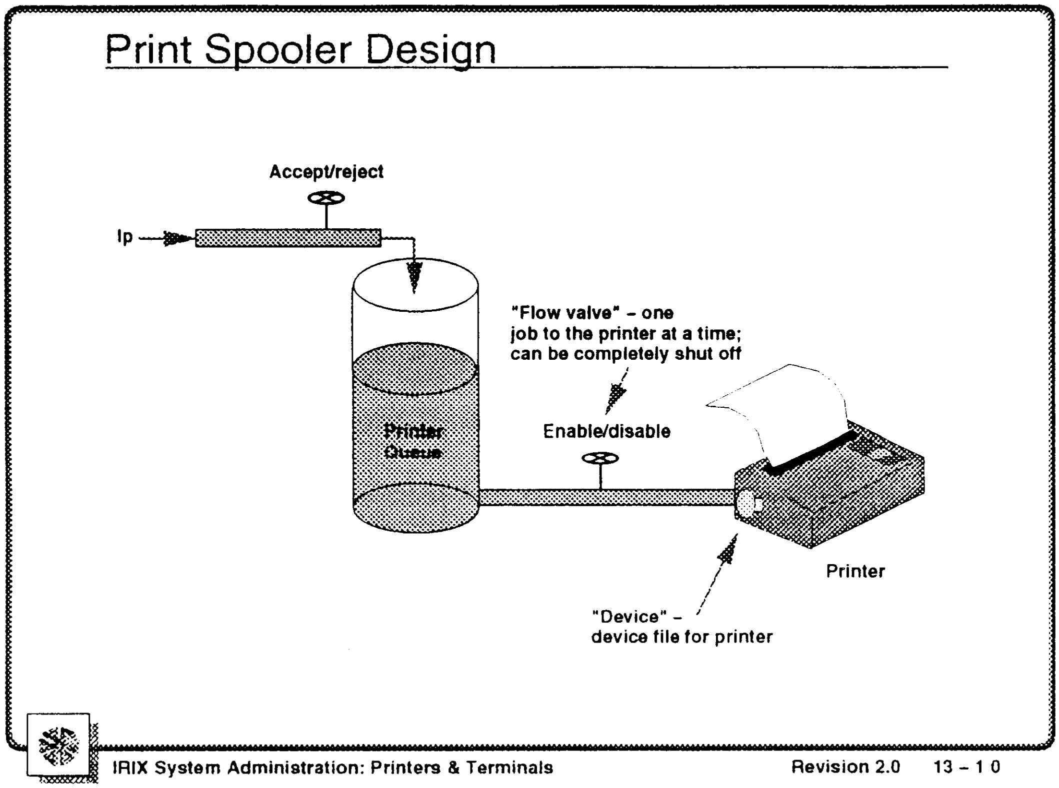
Student Notes

When you disable a printer, use the `-r` "*<some message>*" option to print an explanation message of why this printer is disabled. You can view the message from `lpstat -t` output.

When you use the `reject` command, use the `-r` option to `reject` (analogous to `disable`, but message is displayed for `lpstat -t` and when a user attempts to send a job to the printer with `lp`).

You can continue to send jobs to the printer even if it has been *disabled,* because `disable` means to not allow physical printing. Queuing of print requests is still `accepted`.

13-10 Print Spooler Design



Student Notes

`lpsched` schedules requests taken by `lp(1)` for printing on line printers.

`lpadmin` configures line printer spooling systems to describe printers, classes, and devices. It is used to add and remove destinations, change membership in classes, and change devices for printers, printer interface programs, and the system default destination.

The `accept` command allows `lp(1)` to accept requests for the named destinations. A destination can be either a line printer or a class of printers. Use `lpstat(1)` to find the status of destinations. The `enable` command activates the named printers and enables them to print requests taken by `lp(1)`.

13-11 Steps to Configure Print Spooler

Steps to Configure Print Spooler

- Stop spooler

  `# /usr/lib/lpshut`

- Create the printer using **lpadmin(1M)**

  `# /usr/lib/lpadmin -pprinter -vdevice -mmodel`

- Assign a default destination

  `# /usr/lib/lpadmin -dprinter`

- Accept jobs on printer queue

  `# /usr/lib/accept printer`

- Enable jobs to print on printer

  `# /usr/bin/enable printer`

- Turn on spooler

  `# /usr/lib/lpsched`

IRIX System Administration: Printers & Terminals   Revision 2.0   13 - 1 1

Student Notes

The system default destination can be a printer or a printer class. You can set it by using `lpadmin -d`. A destination must already exist on the `lp` spooler before you can designate it as the default destination.

The `lp` command determines a request's destination by checking for a `-d` option on the command line. If `-d` is not present, it checks to see if the environment variable `LPDEST` is set. If `LPDEST` is not set, the request is routed to the system default destination.

Setting the environment variable `LPDEST` allows a user to have a default destination other than the system default.

13-12 /usr/bin/lpstat Command

/usr/bin/lpstat Command

- Print status information

  `-t` prints status for *all* printers

```
# lpstat -t
scheduler is running
system default destination: LaserJet4
members of class Apples:
apple
device for apple: /dev/ttyd2
device for LaserJet4: /dev/plp
apple accepting requests since Jun 15 11:17
LaserJet4 accepting requests since Aug 9 18:59
Apples accepting requests since Jun 17 10:12
printer apple is idle.  enabled since Jun 15 11:17
printer LaserJet4 is idle.  enabled since Aug 9 18:59
```

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 1 2

Student Notes

To test the printer, run `lpstat -t`. To verify the printer, send some files to the printer, run `lpstat -t` again, and see if output prints.

13-13 Adding More Printers

Adding More Printers

- Stop spooler

  `# /usr/lib/lpshut`

- Create another printer

  `# /usr/lib/lpadmin -pprinter -vdevice -mmodel`

- Start the printer job queue

  `# /usr/lib/accept printer`

- Enable the printer

  `# /usr/bin/enable printer`

- Turn on spooler

  `# /usr/lib/lpsched`

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 1 3

Student Notes

Example:

```
# /usr/lib/lpadmin -plaser -v/dev/plp -mlaserwriter
```

13-14 Deleting Printers

Deleting Printers

Manual

- Stop spooler

  `# /usr/lib/lpshut`

- Delete the printer/class

  `# /usr/lib/lpadmin -xprinter/class`

- Turn on spooler

  `# /usr/lib/lpsched`

Script

`# /usr/sbin/rmprinter printer`

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 1 4

Student Notes

13-15 Moving Print Requests

Moving Print Requests

- Printer *Apple1* out for repair, printer has pending jobs

```
# /usr/lib/lpshut
# /usr/lib/lpmove Apple1 Banana1
# /usr/lib/lpsched
```

- Printer *Apple1* temporarily disabled, printer has pending jobs, particular job *Apple1-22* is needed immediately

```
# /usr/lib/lpshut
# /usr/lib/lpmove Apple1-22 Banana1
# /usr/lib/lpsched
```

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 15

Student Notes

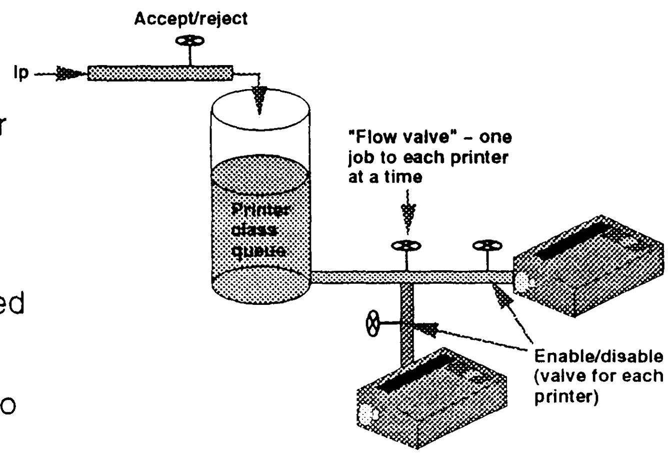
The `lpshut` command shuts down the line printer spooler. All printers that are printing at the time `lpshut` is invoked stop printing. Requests that were printing at the time a printer was shut down will be reprinted in their entirety after `lpsched` is started again.

The `lpmove` command moves requests that were queued by `lp(1)` between LP destinations.

13-16 Setting Up a Printer Class

Setting Up a Printer Class

- One queue for more than one printer
- Jobs are sent to whichever printer is not busy
- Requests to individual printer should be rejected
- Users should direct their jobs to the class, rather than the individual printers



IRIX System Administration: Printers & Terminals Revision 2.0 13 – 1 6

Student Notes

13-17 Defining Print Classes

Defining Print Classes

- Stop scheduler

  `# /usr/lib/lpshut`

- Create class, assign printer(s)

  `# /usr/lib/lpadmin -pprinter1 -cclass`

  `# /usr/lib/lpadmin -pprinter2 -cclass`

- Start the job queue

  `# /usr/lib/accept class`

- Turn on scheduler

  `# /usr/lib/lpsched`

Note: A printer can be assigned to multiple classes

IRIX System Administration: Printers & Terminals Revision 2.0 13 - 1 7

Student Notes

13-18 Network Printers

Network Printers

On system with physical printer (already set up)

- Use **addclient**

  `# /usr/sbin/addclient clientsystemname`

- Use **addclient -a** to allow any machine to use the printer

On systems without the physical printer, use **mknetpr**

`# mknetpr LocalPrinterName RemoteHost RemotePrinterName`

- **mknetpr** is a shell script that allows interactive or batch command mode
  - In interactive mode, prompts you for information about the local printer name, remote host name, printer name on the remote host

Student Notes

Execute `addclient host` for each client machine that uses the network to submit printing jobs on this machine, or use the `-a` option to allow universal access. This command modifies the `.rhosts` file in the home directory of user `lp`.

`mknetpr` provides access on a local machine to a printer that is physically connected to a networked remote machine. `mknetpr` can be run either interactively or in batch mode. `mknetpr` runs interactively if any or all of its command-line arguments have not been specified. If `mknetpr` is run interactively, it prompts for the arguments.

13-19 Parallel Printers

Parallel Printers

- Built-in parallel port uses device file **/dev/plp**
- Use the interactive script **mkcentpr(1M)** to install a parallel printer; prompts for information

```
# /usr/sbin/mkcentpr
```

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 1 9

Student Notes

13-20 Debugging `lp` Printer Problems

Debugging `lp` Printer Problems

When jobs are not printing, where do you look?

- Hardware
  - Check connections
  - Check/power cycle printer
  - Check network if network printer (`ping(1M)` to test communication)
- `lpstat -t`
  - Scheduler running?
  - Printer enabled?
  - Printer accepting requests?
  - Requests on queue?
  - Do on client and server if network printer
- Look at printer log (`/var/spool/lp/log`)
- Try cycling `enable(1)/disable(1)` on the printer or `lpshut(1M)/lpsched(1M)` on the spooler

IRIX System Administration: Printers & Terminals Revision 2.0 13 – 2 0

Student Notes

13-21 Making and Printing Screen Images

Making and Printing Screen Images

- Use **scrsave(6D)** to dump portions of screen
  - Creates SGI RGB format file
- Use **snapshot(6D)** to dump small images from screen
- Use **imgworks(1)** to alter and enhance images
- SGI RGB files are automatically printed using SGI print filters
- To print to PostScript® printers on other machines, convert RGB format to PostScript with **tops(6D)**:

```
% tops RGB.file > PS.file
% lp -or PS.file
```

IRIX System Administration: Printers & Terminals Revision 2.0 13-21

Student Notes

13-22 Terminal Setup Procedure

Terminal Setup Procedure

- Make physical connection
  - use an RS-232 "three-wired null modem" cable with either a DB9 or DIN-8 connector
- Check **/etc/inittab** to make sure that **getty(1M)** is enabled on the port. Change **off** to **respawn** for terminals you want enabled.

```
% cat /etc/inittab
[stuff deleted]
t2:23:respawn:/etc/getty -N ttyd2 co_9600 # port 2
t3:23:off:/etc/getty -N ttyd3 co_9600  # port 3
```

- If **inittab** is changed, update **init(1M)** with **init q**

IRIX System Administration: Printers & Terminals   Revision 2.0   13 – 2 2

Student Notes

By default, only the getty for port 1 is enabled (since this is the console). All other ports will have to be enabled by changing *off* to *respawn*.

Default baud rate is 9600 or less. To select a higher baud rate, choose a different set of *getty definitions* in the /etc/gettydefs file (see gettydefs(4)).

By default, terminals are assumed to be vt100-like. Users are queried on login for terminal type. By hitting **<Enter>** they select vt100, or the can type in a terminal type of their choice.To change the default terminal selection, edit the /etc/ttytype file:

```
# vi /etc/ttytype
. . .
?vt100 ttyd1
?vt100 ttyd2
. . .
```

Change `vt100` to some other terminal type (see `/usr/share/lib/terminfo`) for a list of *known* terminals). If you delete the question mark (?) before the terminal type, the user will not be queried.

SiliconGraphics
Computer Systems

Module 14: Installing Software on Your IRIS

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module discusses installing software on your SGI system.

14-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Define the terms **`inst`** and **`miniroot`**
- Use **`inst`**, the software installation tool
  - Graphical Software Manager installation tool **`swmgr`**
- Utilize naming conventions when installing software
- Differentiate between operating system releases and updates
- Determine when it is necessary to install software

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 2

Student Notes

14-3 inst

inst

inst is SGI's software installation tool

- Software shipped from SGI uses the **inst** format
  - Some software applications may be shipped in **tar** tape format using shell scripts to complete the installation process
- Can be used in
  - Single or multiuser run levels
  - **miniroot** (**Installing Software** option of the PROM menu)
- The software package being loaded determines whether or not booting **miniroot** is a requirement for software installation

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 3

Student Notes

Live installations are performed without shutting down the system. These are preferred, because other operations can be ongoing during the software installation. There are two ways to run `inst`:

- Invoke `inst` as a command from the shell. This is known as "IRIX Installation," and you must be superuser. Some software cannot be installed using IRIX Installation (release notes and `inst` warn you), because of system integrity problems that can arise from changing some software or performing certain operations while that software is running.
- Invoke `inst` in a standalone mode, known as "Miniroot Installation." Shut down the system to `PROM` and load a collection of files known as `miniroot` into the swap partition of your system disk. `Miniroot` contains an IRIX kernel, `inst`, and several other programs.

14-4 Software Organization

Software Organization

- Three–level hierarchy
  - *Product* **eoe**, **nfs**, **ftn90_eoe**, **ftn90_dev**
  - *Image* **sw** (software), **man** (man pages and documentation)
  - *Subsystem*, grouping of the image files into functional groups **eoe.sw.acct**, **java.sw.debug**
- Organization
  - Typically one product per distribution CD
  - At least two images, **sw** and **man** per product
  - Multiple subsystems per image
- Naming
  - **eoe** refers to all images (and therefore all subsystems) of **eoe**
  - **eoe.sw** refers to all the subsystems under **eoe.sw**
  - **eoe.sw.acct** refers only to the accounting subsystem of **eoe.sw**
  - **inst** metacharacters *, ?, [ ]
    ***.man** refers to all man images for all products
    eoe.*.* refers to all images and subsystems for product **eoe**

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 4

Student Notes

The *product* is a file that contains information about product requirements. An *image* is a collection of installable files that perform a similar function. Typically, each software product contains at least two images. It is possible to install the images that contain the executable programs of a product without installing the release notes image. Images are composed of one or more *subsystems*. A subsystem is the smallest installable unit of software.

14-5 Software Update CDs

Software Update CDs

- First CD, *IRIX Foundation*, holds base software such as **eoe**, **efast**,and **vino** and standalone tools for booting **miniroot** and repartitioning disk drives
- Second CD, *IRIX Layered Products*, holds **4Dwm**, **cadmin**,**demos**,**insight**, **desktop_eoe**, **mmail**, **netscape**, **print**,and **sysadmdesktop** software
- Other CDs hold optional software such as **impressario**, **dev**, or **c++_dev**

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 5

Student Notes

When you receive a software update, it is delivered on multiple CDs. Use the sequence numbers on the CD labels or the directions in "Planning the Order of Installation" to determine the order in which you should install the updates. Updates are not necessarily included for all products, because not all products are updated at the same time.

14-6 Updates Versus Releases

Updates Versus Releases

- Updates
  - May include minor/incremental changes to the operating system and its subsystems
  - Example: 6.0.1->6.0.2
  - Update existing software
- Releases
  - Usually indicate major changes to the operating system and its subsystems
  - Example: 5.3->6.2
  - Install new version of operating systems and subsystems
  - Usually implies installing the entire OS and applications, and reloading data files

IRIX System Administration: Installing Software on Your IRIS   Revision 2.0   14 - 6

Student Notes

14-7 Safely Installing Software

Safely Installing Software

Always back up your system

- Before *and* after installing update/revision software

Why?

- Errors (end user) and bugs
- Incompatibilities between the OS and applications
- May have removed a package/subsystem or data files to increase disk space

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 7

Student Notes

Backups are not always needed for minor software installations; however, until you are proficient with using the installation tools, be very careful.

14-8 Using `inst`

Using `inst`

Use the **inst** command to start the installation tool while in IRIX

- Using the default source

```
# inst
Inst>
```

- Using a remote source

```
# inst -f server.corp:/tree/5.2
Inst>
```

- Using a local CD–ROM

```
# inst -f /CDROM/dist
Inst>
```

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 8

Student Notes

14-9 The inst Main Menu

The inst Main Menu

```
# inst

For help on inst commands, type "help overview".

Inst Main Menu

 1. from [source]                  Specify location of software to be installed
 2. list [keywords] [names]        Display information about software subsystems
 3. go                             Perform software installation and removal now
 4. install [keywords] [names]     Select subsystems to be installed
 5. remove [keywords] [names]      Select subsystems to be removed
 6. keep [keywords] [names]        Do not install or remove these subsystems
 7. step [keywords] [names]        Interactive mode for install/remove/keep
 8. conflicts [choice ...]         List or resolve installation conflicts
 9. help [topic]                   Get help in general or on a specific word
10. view ...                       Go to the View Commands Menu
11. admin ...                      Go to the Administrative Commands Menu
12. quit                           Terminate software installation
Inst>
```

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 9

Student Notes

14-10 Using the `inst` Admin Menu

Using the `inst` Admin Menu

Perform many administrative tasks in the **admin** menu

```
Inst> admin

Administrative Commands Menu

 1. set [preferences]          List all preferences or set/clear a preference
 2. date [args]                Display or set the system date
 3. files [names]              List files in subsystems
 4. space                      List disk usage information
 5. recalculate                Recalculate space required for installation
 6. sh [cmd]                   Escape to a shell or run a command
 7. shroot [cmd]               Escape to a chrooted shell or run a command
 8. mount [fsname] [dir]       Show mounted filesystems or mount new ones
 9. umount [fsname]            Unmount a filesystem
10. mkfs [blockdevice]         Make a new filesystem
11. sethostname                Set name of current host, for networking
12. sethostipaddr              Set host ipaddr - Internet address
13. save filename              Save selection to file
14. load filename              Load selection from file
15. config [changed]           List all or modified config files
16. hardware                   List machine-specifiec hardware information
17. updatekeepfile             Add N(ew), unselected subsystems to keepfile
18. return                     Go to the Inst Main Menu

Admin>
```

IRIX System Administration: Installing Software on Your IRIS   Revision 2.0   14 – 1 0

Student Notes

Use item 6, sh, to run commands in miniroot's filesystem.

Use item 7, `shroot`, to run commands in your / and `/usr`, and other filesystems without prefacing all filenames with `/root`.

14-11 Using the `inst` View Menu

Using the `inst` View Menu

- Change default behavior of **`inst`** by using the **View** menu commands
- Use **`target`** to manipulate *installed* software
- Use **`distribution`** to manipulate software *to be installed*

```
Inst> view

View Commands Menu

 1. distribution                Show only products on distribution
 2. target                      Show only products on target
 3. filter [setting]            Show or hide products by attribute(s)
 4. level [1,2,3]               View (1) product, (2) image or (3) subsystem
 5. sort [size,name]            Select sorting of product list
 6. sizes                       Show total product sizes instead of deltas
 7. deltas                      Show disk space deltas instead of total sizes
 8. filesystems [on,off,fs..]   Show size/delta breakdown by filesystem
 9. show                        Show the current view settings
10. return                      Go to the Inst Main Menu
View>
```

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 1 1

Student Notes

By default, the `list` and `step` commands display all software in the distribution at the subsystem level; subsystems are sorted alphabetically by shortname. Using the **View** menu, you can change the default behavior of `list` and `step` from the top-level `inst` menu to adjust their output. You can also use the `set` command from the **View Commands** menu to display and change `inst` preferences.

14-12 `inst` Options

`inst` Options

- The **`install`** command selects items in the distribution inventory for installation
- The **`remove`** command selects items on the distribution or *target* inventory for removal
- The **`keep`** command selects items on either list for which no action should be taken
- The **`step`** command allows you to list and select software at the same time from either list (distribution or target)

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 1 2

Student Notes

The `install` command selects items in the distribution inventory for installation. This is a request only, and the installation does not occur until you use the `go` command.

Use the **View** menu to set the target in order to remove software on your system disk with the `inst remove` command.

14-13 Special Keywords

Special Keywords

- Keywords can be used with **`list`**, **`install`**, **`remove`**, **`keep`**, and **`step`**
- Examples
  - **Inst>** `keep S`

    Requests that all subsystems that are currently installed and have the same version counterparts in the software distribution be left as–is on the disk (S = same)
  - **Inst>** `list same`

    Requests that all subsystems that have an identical version installed be listed
  - **Inst>** `install required`

    Requests that all subsystems that are required for proper installation of a workstation be selected (required = q)
  - **Inst>** `remove U`

    Requests that all subsystems installed that have a newer version in the software distribution be updated, removed (U = upgrade)

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 1 3

Student Notes

There are many other keywords, and most of them have single-character references also, that is, `required` is the same as `q`.

`/var/inst/.keepfile` contains a list of products and subsystems which you do not want to even consider for installation when viewing a distribution. This can save time when installing from distributions with multiple products. Both `inst` and `swmgr` do a `keep` on those products listed in the `.keepfile`.

14-14 Choosing Specific Subsystems

Choosing Specific Subsystems

- Choose a specific subsystem for installation

```
Inst> install eoe.sw.acct
```

- Choose a specific subsystem from the distribution for removal

```
Inst> remove eoe.sw.perf
```

- Choose a specific subsystem from the target for removal (software already installed)

```
Inst> view target
Inst> remove webspace.man.relnotes
Inst> list webspace.man.relnotes
  View:      target
  Status:    I=installed, R=removed, ''=not installed
  Selection: i=install, r=remove, k=keep, u=upgrade

r I webspace.man.relnotes [d]     24- WebSpace
Disk space summary (Kbytes):     /       /disk2
Current free space             220856   216838 ...
```


Student Notes

When you choose a specific subsystem for installation or removal, you will notice an *i* or an *r* printed to the far left of subsystem when you do a listing.

Use the `install` command to identify or select subsystems that should be installed. The `remove` command requests that installed subsystems be removed from the system. The installation or removal does not occur until you enter the `go` command. Until then, you may change the requested operations for any or all subsystems. To undo a `remove` request, use the `keep` command.

14-15 Interactive Step Mode

Interactive Step Mode

Choose **step**, to interactively step through each subsystem

```
Inst> step eoe

  View:  distribution
  Status:     N=new,U=upgrade,S=same,D=downgrade,
  Selection: i=install, r=remove, k=keep

  Subsystem Types [bdrc]: b=reBoot needed, d=Default,
  r=Required, c=Client only

Step commands: i,r,k,n,p,...  Use ? for more step help.
```

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 1 5

Student Notes

You can use the `step` command instead of the `install` and `remove` commands for an interactive choice of each subsystem.

14-16 Installing Software

Installing Software

After selecting software, type go to install/remove subsystems:

```
Default distribution to install from: /CDROM/dist
Inst Main Menu

 1. from [source]                 Specify location of software to be installed
 2. list [keywords] [names]       Display information about software subsystems
 3. go                            Perform software installation and removal now
 4. install [keywords] [names]    Select subsystems to be installed
 5. remove [keywords] [names]     Select subsystems to be removed
 6. keep [keywords] [names]       Do not install or remove these subsystems
 7. step [keywords] [names]       Interactive mode for install/remove/keep
 8. conflicts [choice ...]        List or resolve installation conflicts
 9. help [topic]                  Get help in general or on a specific word
10. view ...                      Go to the View Commands Menu
11. admin ...                     Go to the Administrative Commands Menu
12. quit                          Terminate software installation

Inst> go
```

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 1 6

Student Notes

Remember, until you issue the go command, everything else has just been a request to do something.

14-17 Software Conflict Resolution

Software Conflict Resolution

inst suggests actions for resolving software conflicts

- Choose from a list of options

```
Inst> go
ERROR : Conflicts must be resolved.

media_warehouse.sw.warehouse cannot be installed because of missing
   prerequisites:
      1a. Do not install media_warehouse.sw.warehouse (1232679420)
      1b. Install media_warehouse.sw.viewers_image (1232679420 -
   2147483647).

Resolve conflicts by typing "conflicts choice choice ..."
Or try help conflicts

Inst> conflicts 1b
No conflicts
Inst> go
```

IRIX System Administration: Installing Software on Your IRIS   Revision 2.0   14 – 17

Student Notes

Sometimes, the selections that you make for installation or removal cause conflicts. For example, a conflict occurs when you select two incompatible subsystems for installation, or when you select a subsystem for installation and do not select its prerequisites. When conflicts occur, the go command does not execute, and inst posts a conflict advisory that suggests a resolution to the conflict.

14-18 Updated Configuration Files

Updated Configuration Files

- **inst** handles updates of system configuration files
  - **IF** no difference between new and old versions,
    THEN new version is not installed
  - **IF** the new version is recommended,
    THEN new version is installed as **file.N**
  - **IF** the new version contains critical features
    THEN the old file is renamed as **file.O**,
    AND THEN the new version is installed
- After installing a system update, it may be necessary to reconcile these **.O** and **.N** files
  - System prints a warning message at boot time until reconciled
- Use **diff** or **gdiff** to show their differences
- Transfer information that is needed from the **.O** version to the no–suffix version
- Add information from **.N** file to the no–suffix version

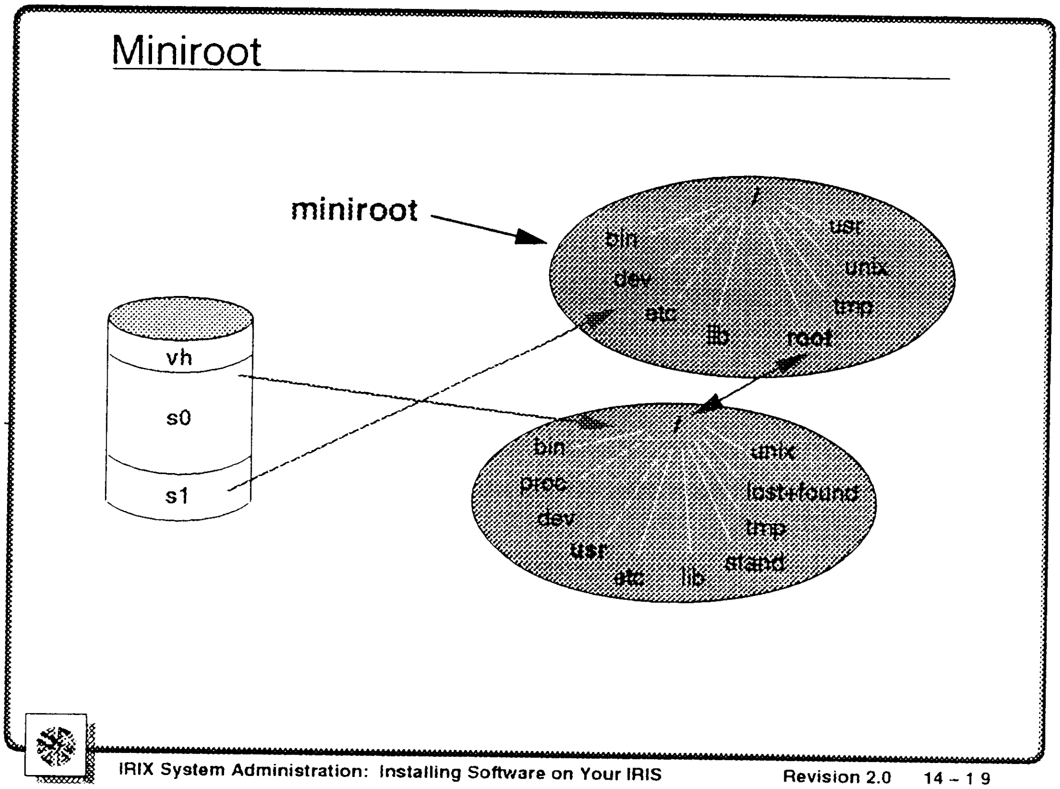
IRIX System Administration: Installing Software on Your IRIS   Revision 2.0   14 – 1 8

Student Notes

When a new version of a configuration file is created during an installation, inst posts a message about the changed files after the go processing is completed. The message is repeated the next several times the system is rebooted. Use the following command to identify the changed files:

```
# showfiles -cCH
Configuration Files
m     = modified since initial installation
?     = modification unknown
blank = file is as originally installed
m /.cshrc
  /.cshrc.N
m /etc/sendmail.cf
  /etc/sendmail.cf.N
```

14-19 Miniroot



Student Notes

Miniroot is a bare-bones mini-IRIX operating system. It is located on every distribution IRIX operating system CD, for 6.2 the IRIX Foundation CD. The miniroot kernel lives in memory, and the miniroot filesystem lives in partition one, which is swap space. The original IRIX root filesystem is mounted underneath miniroot, in `/root`. If applicable, the `usr` filesystem is mounted underneath miniroot in the location `/root/usr`.

14-20 Miniroot Partitions

Miniroot Partitions

- The root filesystem is mounted at `/root` instead of `/` while in *miniroot*

```
Admin> sh
# df -k
```

Filesystem	Type	Kbytes	Use	Avail	%Use	Mounted on
/dev/miniroot	xfs	23680	13050	10630	55	/
/dev/dsk/dks0d1s0	xfs	24244	73090	11154	54	/root
/root/dev/dsk/dks0d2s7	efs	1010250	398083	612167	39	/root/disk2
/dev/dsk/dks0d4s7	xfs	478080	467879	10201	98	/CDROM

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 2 0

Student Notes

The root filesystem is mounted at `/root` instead of `/` while in miniroot. Notice that what is normally your root partition, `/dev/dsk/dks0d1s0`, is now mounted at `/root`. All other filesystems mounted under your root partition are prefixed with `/root` in the mount point name, `/root/disk2`.

14-21 Booting `miniroot` From a Local CD-ROM

Booting `miniroot` From a Local CD-ROM

From a local CD-ROM

- Shut down to PROM monitor
- Select **Install System Software** from PROM menu
- Select the local CD-ROM icon
- Insert the first CD, *IRIX Foundation*, which contains the installation tools
- Load `miniroot`, which invokes `inst`

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 2 1

Student Notes

For installations where fundamental IRIX services, such as filesystem management, are either unavailable or unreliable, an alternative installation method is required, known as a miniroot installation.

On systems with ARCS PROMs, a menu of distribution sources is displayed after you choose *Install System Software*. Click the icon for the local CD-ROM to continue with the installation. If you are using an ASCII console, the menu of distribution sources appears as text.

14-22 REVIEW: Using `bootp` for Remote Installation

REVIEW: Using `bootp` for Remote Installation

On the *remote* workstation (server):

- Check the **`/etc/inetd.conf`** configuration file for secure access

```
bootp dgram udp wait root  /usr/etc/bootp  bootp
tftp  dgram udp wait guest /usr/etc/tftpd  tftpd -s /usr/local/boot
/usr/etc/boot
```

  - To use a source tree, add the pathname

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s /usr/local/boot
/usr/etc/boot /tree/6.2
```

  - To use a remote tape, add **`/dev/nrtape`**

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s /usr/local/boot
/usr/etc/boot /dev/nrtape
```

  - To use a remote CD, add **`/CDdir`**

```
tftp dgram udp wait guest /usr/etc/tftpd tftpd -s /usr/local/boot
/usr/etc/boot /CDROM
```

- Restart **`inetd`**

```
# killall -HUP inetd
```

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 22

Student Notes

Changing the `/etc/inetd.conf` file for booting `sash` or `/unix` is the same as for booting miniroot or loading software from another systems disk or CD-ROM.

14-23 Booting `miniroot` – Remote CD-ROM

Booting `miniroot` – Remote CD-ROM

From a CD-ROM on a remote system

On *remote workstation* with the CD-ROM (the server):

- Verify that your local system's IP address is in **`/etc/hosts`**
- Check the **`/etc/inetd.conf`** configuration file for secure access (**`bootp`** and **`tftp`**)
- Execute the **`cdinstmgr(1)`** command
  - Guides you through inserting the software installation CD and mounting it
  - Prevents other users from **`eject(1)`**'ing your CD
- When the CD is ready to be used for installation, the following message is displayed:

```
CD on <CDdir> for host <hostname>: type the word
"done" when you are finished with this CD, "quit"
if you are completely done:
```

IRIX System Administration: Installing Software on Your IRIS   Revision 2.0   14 – 2 3

Student Notes

`cdinstmgr` helps manage client installs from a CD-ROM drive on a server. It prompts for the available CD-ROM drives. If more than one drive is available, it mounts the CD, unmounts it at the end, and handles the case where more than one client is doing installs at the same time. It uses `mediad` and `eject` commands to mount, unmount, and eject the CD.

14-24 Booting `miniroot` – Remote CD-ROM (continued)

Booting `miniroot` – Remote CD-ROM (cont'd)

From a CD-ROM on a remote system

On the *local* workstation:

- Shut down to PROM
- Select **Enter Command Monitor**
- Set the **`netaddr`** variable to the IP address value (if necessary)

  `>> setenv netaddr N.N.N.N`

- Issue the following commands to boot **`miniroot`**

  ```
  >> setenv tapedevice bootp()server:/CDdir/dist/sa
  >> boot -f $tapedevice(sashARCS) --m
  ```

  or select **Install System Software** from PROM menu

IRIX System Administration: Installing Software on Your IRIS Revision 2.0 14 – 2 4

Student Notes

sa means *standalone,* the `--m` loads and boots `miniroot` in a single operation.

14-25 Booting `miniroot` From Remote Source Tree

Booting `miniroot` From Remote Source Tree

On the *local* workstation:

- Shut down to PROM
- Select **Enter Command Monitor**
- Set the **`netaddr`** variable to the IP address value

```
>> setenv netaddr N.N.N.N
```

- Issue the following commands to boot **`miniroot`**

```
>> setenv tapedevice bootp()server:/tree/6.2/sa
>> boot -f $tapedevice(sashARCS) --m
```

After a few moments, the **`inst`** menu appears

Student Notes

14-26 Creating a Software Source Tree

Creating a Software Source Tree

Why create a source tree of software on a server?

- Multiple client systems on the network can simultaneously access the source tree to load software
- Software installation may be faster across the network, depending on network traffic
- Can have source trees for many releases (5.3, 6.2)
- Use the **cp** command for a distribution on a CD–ROM

IRIX System Administration: Installing Software on Your IRIS   Revision 2.0   14 – 2 6

Student Notes

To copy from a local CD-ROM, use `cp -r`. To copy an installable CD-ROM over the network, use `rcp` with the `-r` option.

SiliconGraphics
Computer Systems

Module 15: Backup and Recovery

Part Number: LBT111-2.0-6.2-S-SD-W
June 1996

Module Overview

This module discusses how to backup and recover your system from the backup.

15-2 Module Objectives

Module Objectives

After completing this module, you will be able to

- Discuss device files for tape drives
- Discuss tools for implementing backups
- Describe the differences between full, incremental, and user backups
- Develop and implement a backup strategy
- Identify when recovery is necessary
- Determine what type of recovery to use

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 2

Student Notes

15-3 Tape Device Files

Tape Device Files

- Device files for tape drives are in **/dev/rmt** and of the form

  <tape><control #>d<drive number><option>
  - A SCSI tape on controller 0, drive 2

    tps0d2
  - No rewind device for above tape drive

    tps0d2nr
- Generic links to tape drives are created by the **/dev/MAKEDEV** script
  - **/dev/tape** – Used as default destination for most backup utilities
  - **/dev/nrtape** – With no rewind at end of operation
  - **/dev/tapens** and **/dev/nrtapens** – No byte swapping; may be needed for reading tapes from different architectures

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 3

Student Notes

Tape device files are used to access tape drives installed in or attached to your system. As with disk drive partitions, several generic links to a commonly used tape drive are created. When /dev/MAKEDEV runs, it attaches the generic link to the first tape drive it finds. If you have multiple tape drives, you may want to manually link a different drive.

To discover the tape drives on your system:

```
# hinv | grep Tape
Tape drive: unit 3 on SCSI controller 0: DAT
```

15-4 Status of the Tape Drive

Status of the Tape Drive

- To display the status of the default tape drive:

```
# mt stat
Controller: SCSI
Device: ARCHIVE: Python 25601-XXX2.63
Status: 0x20266
Drive type: DAT
Media : READY, write protected, at BOT
```

- To rewind the tape:

```
# mt rew
```

- To retension the tape:

```
# mt ret
```

- To erase the contents on a tape:

```
# mt erase
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 - 4

Student Notes

Rewinding is normally not necessary. However, occasionally you may want to create or extract from a tape without rewinding, and may need to rewind the tape later. Retensioning the tape ensures an overall level of tension for maintaining data integrity. Tapes that are stretched are difficult and more time consuming to read. Tapes stored on end are less likely to bend than if they are stored flat. A tape stored flat should always be retensioned before use.

To erase the contents on a tape, use this command:

```
# mt erase
```

This can be useful if using the backup utility `xfsdump(1M)`.

15-5 Types of Backups

Types of Backups

- Full backup – all files on all filesystems
  - Before and after system and application software has been installed
  - Before performing hardware diagnostics
  - Before repartitioning the disks
- Incremental backups – only some files, files that have changed since the last incremental or full backup
  - Daily
  - Weekly
- User backups – only a particular users directory or subdirectory of files
  - Whenever critical files change

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 5

Student Notes

Create full backup tapes to

- Recover the root filesystem and application packages if filesystems become corrupted or destroyed
- Create a "snapshot" of your system more customized than reloading directly from installation tapes

Filesystems to backup are `/dev/root` and any other filesystem containing application packages. If `/dev/usr` is a separate filesystem, you should include it in the full backup.

Create incremental backup tapes to

- Recover selected files that may have changed or been created since the full recovery tape was created
- Recover filesystems not including root

Files that change frequently include selected files from `/dev/root`, files under `/usr`, or other critical filesystems. Files under `/usr` that change frequently might include the following:

```
/usr/people
/usr/local/bin
/usr/local/lib
/var/adm
```

Encourage users to do their own backups of very important data, to provide some "double coverage" and to save time. Files may be lost before systemwide backup gets done. Lost files can be retrieved more quickly from a user's backup than from the large systemwide backups.

15-6 Pathnames – Absolute Versus Relative

Pathnames – Absolute Versus Relative

Absolute Pathnames

- Absolute pathnames begin with /

  Examples:

  `/usr/people/guest`

  `/project/proj1/data`

Relative Pathnames

- Relative pathnames contain *no* / at the beginning of the name

  Examples:

  `guest`

  `proj1/data`

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 6

Student Notes

Absolute pathnames always begin with a leading "/" to indicate to the system to begin the search at the `root` directory. Relative pathnames are *relative* to the current directory you are in, and do not begin with a leading "/."

15-7 Pathname Guidelines

Pathname Guidelines

- Use absolute pathnames when using **Backup(1)** or the System Manager tool to create full recovery backups destined for the Recovery Tool (SGI's turnkey system recovery utility)
- Use absolute pathnames for incremental backups of system configuration files, such as **/etc/passwd**
- Use relative pathnames for flexibility of user backups, and backups of nonstandard filesystems
- When producing full recovery tapes with **dump(1M)**/**xfsdump(1M)**, relative paths are used, but **dump/xfsdump** uses a different recovery method than the Recovery Tool

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 7

Student Notes

The pathnames you use determine the degree of flexibility you have when you are restoring files. By default, files backed up using absolute pathnames are restored to the same location. There are some options available with some of the tape archive commands to strip off a leading "/" on each file as the file is copied to disk. However, for greatest flexibility for user files, use relative pathnames.

15-8 Tools for Implementing Backups

Tools for Implementing Backups

- **tar(1)**
- **bru(1)**
- **Backup(1)**
- **dump(1M)**
- **xfsdump(1M)**
- System Manager **Backup and Restore** tool

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 8

Student Notes

The `tar` utility is a tape archiver that saves and restores multiple files on a single file (usually a magnetic tape, but it can be any file). `bru` is a UNIX filesystem backup utility with significant enhancements over other more common utilities such as `tar`, `cpio`, and `dd`. The `Backup` command archives the named file or directory to the tape device. The System Manager Backup and Restore tool is a graphical tool that lets you back up and restore the entire contents of a disk or selected directories and files onto a tape. This tool is based on the `bru` utility.

`dump` backs up all files in an EFS filesystem, or files changed after a certain date. `xfsdump` backs up an XFS filesystem or files in an XFS filesystem.

15-9 `bru` and `tar`

`bru` and `tar`

- `c` create a new archive
- `f` specify a different device file than default `/dev/tape`
- `v` verbose option to list files that are backed up
- `x` extract files
- `t` table of contents, list files that are backed up on the tape .
- `K` to backup files larger than 2 Gb
- `Z` to use data compression (`bru` only)

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 9

Student Notes

The data compression that bru supports is LZW. Not all vendors support LZW compression. The `K` option is necessary for files larger than 2 Gb on an XFS filesystem. Some other features of `bru` and `tar` are listed below:

Command Option	Description
n	Backup files based on modification date
e	Calculate space requirements for backup
i	Integrity check of data on tape
vvvv	Verbose (multiple levels)
dddd	Differences between archived files and current files are reported

15-10 Creating Archives With `bru` and `tar`

Creating Archives With `bru` and `tar`

- Use `tar(1)` to back up `/usr/people` to `/dev/tape`

```
# tar cv /usr/people
```

- Use `bru(1)` to back up `proj1` onto a remote tape drive on system `fido`

```
# bru -cvf guest@fido:/dev/tape proj1
```

- Use `-n` option with `bru(1)` to create an incremental tape of files modified after 12 Jan 96

```
# bru -c -n 12-Jan-96 /usr/people
```

- Use `-K` option to back up the XFS filesystem `/data`

```
# tar cvK /data
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 - 1 0

Student Notes

In the first example, `tar` recursively copies the entire contents of the `/usr/people` directory to tape. The second example shows how the `-f` flag is used to designate a tape on a remote machine connected via the network. The next example shows how to create an incremental backup of files created or modified after a certain date. The last example shows how to backup an XFS directory where a file might be larger than 2 Gb. This option may create an archive that is not supported on other vendors systems.

15-11 Restoring With `bru` and `tar`

Restoring With `bru` and `tar`

- Restore files from absolute path backup to same directory

```
# tar xv /usr/people
```

or

```
# bru -xv /usr/people/kandi
```

- Restore files to a different directory with leading slash stripping. Files are put in **/usr/local/usr/proj1**

```
# cd /usr/local
# tar xvR /usr/proj1
```

or

```
# bru -xj /usr/proj1
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 - 1 1

Student Notes

Remember to be aware of the addressing scheme on the tape and understand where the files are going to go. You can use the `t` (table of contents) option to get the addressing scheme. If you backed up using absolute pathnames and need to restore the files at a different location because you are in miniroot, for example, you can use the leading slash stripping option.

The `bru(1)` command also restores tapes made with `Backup(1)` and the System Manager tool.

15-12 Creating a Backup Script

Creating a Backup Script

- Uses absolute addresses
- Example

```
# cat backup.script
#!/bin/sh
echo backup date: `date`
/usr/sbin/bru -cv /etc/passwd /etc/group \
/etc/inittab /etc/hosts /etc/fstab /etc/lvtab
/etc/exports /etc/hosts.equiv /usr/people \
/usr/local/bin /usr/local/lib /usr/proj1/data
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 1 2

Student Notes

If multiple tapes are not required, you can automate with `cron(1M)`. Example `crontab(1)` entry:

```
0 2 * * 1-5 /usr/local/bin/adm/backup.script > /usr/local/bin/adm/backups/backup.log 2> /usr/local/bin/adm/backups/backup.errors
```

Note: If multiple tapes are required, do half on one night, and half the next.

15-13 Creating Backups With `Backup`

Creating Backups With `Backup`

- Uses the **`bru(1)`** command
- Uses absolute addresses
- Example of a backup of all files starting at /

```
# Backup /
```

- Example of a backup to the default remote tape drive **`/dev/tape`** on system *sysB*

```
# Backup -h sysB /
```

- Example of a backup to a remote tape drive **`/dev/rmt/`** on system *sysB*

```
# Backup -h sysB -t /dev/rmt/tps0d4v /
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 1 3

Student Notes

The `Backup` command uses `bru` to perform the backup function. The command used is

```
/usr/sbin/bru -cvR
```

where the `R` flag means that *R*emote files are to be excluded from the archive.

15-14 Restoring From `Backup`

Restoring From `Backup`

- Use the **`Restore(1)`** command to recover files saved with the **`Backup(1)`** command
- Uses **`bru(1)`**
- Example from local tape **`/dev/tape`**

```
# Restore /usr/people/jo
```

- Example from a remote tape **`/dev/rmt/tps0d6`** on system *ralph*

```
# Restore -h ralph -t /dev/rmt/tps0d6 /usr/people/jo
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 1 4

Student Notes

15-15 EFS: Using dump

EFS: Using dump

- dump(1M) is a standard UNIX tool for doing system backups
  - dump(1M) added to IRIX to provide compatibility in multiarchitecture environments
- Added for more flexibility
- Only works for EFS filesystems
- Backs up *filesystems*, starting in the top directory of the filesystem
- Uses a different format than tar(1) or bru(1)

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 1 5

Student Notes

dump is a standard filesystem backup utility used on many UNIX systems. The dump program makes incremental backups of entire filesystems. Use restore to retrieve files from a dump archive. With restore, you can restore an entire filesystem or specific files. It also has an interactive mode that lets you browse the contents of an archive, select specific files and restore them.

The dump(1M) command is only valid on EFS filesystems; use xfsdump(1M) for XFS filesystems.

15-16 EFS: dump Keys

EFS: dump Keys

- 0-9 dump level

  0 = full backup of all files

  1 = backup all files modified since the 0 level backup

  2 = backup all files modified since the 1 level backup
- u to update **/etc/dumpdates** with the backup date
- f for destination file other than local **/dev/tape**
- c for cartridge tapes
- s for size of dump tape in feet (9 track or cartridge)
- d for tape density in BPI

If no keys given, uses 9u and local device **/dev/tape**

Student Notes

Level 0 dumps everything in the filesystem. Level 1 dumps everything modified or created since the date of the last level 0 dump. Level 2 dumps everything modified or created since the date of the last level 1 dump. Dump date information on a per filesystem basis is written to the file /etc/dumpdates if the u key is specified with dump(1M).

It is vital to perform full, level 0 dumps at regular intervals. When performing a full dump, bring the system down to single-user mode using shutdown. While preparing for a full dump, it is a good idea to clean the tape drive and heads. Incremental dumps allow for convenient backup and recovery on a more frequent basis of active files, with a minimum of media and time.

15-17 EFS: Using dump

EFS: Using dump

- Full backup of **/disk2** filesystem using a local tape drive and writing to **/etc/dumpdates**

```
# dump 0u /disk2
```

- Level 1 dump to remote tape drive

```
# dump 1uf guest@host_name:/dev/tape /disk2
```

- Level 2 dump to local QIC−150 (18 tracks, 600 ft.), high−density tape

```
# dump 2ucs 10800 /disk2/data
```

- When using multiple keys with arguments, make sure that the order of keys matches the order of arguments

```
# dump 0ufsc guest@thisisit:/dev/tape 10800 /disk2
```

NOT

```
# dump 0ufsc 10800 guest@thisisit:/dev/tape /disk2
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 − 17

Student Notes

When using multiple keys having arguments, make sure that the order of keys matches the order of arguments. For cartridge tapes, dump(1M) assumes a default length of 5400 ft. (good for 9-track, 600-ft. QIC-24 low-density tape). The u -option means if the dump completes successfully, write the date of the beginning of the dump in the file /etc/dumpdates. This file records a separate date for each filesystem and each dump level. The format of /etc/dumpdates consists of one free format record per line: filesystem name, increment level, and format dump date. You can edit /etc/dumpdates to change any of the fields, if necessary.

To help you keep track of which filesystems to dump, use the fifth field of /etc/fstab(4).

```
/dev/root / efs rw,raw=/dev/rroot 0 0
```

15-18 EFS: Restoring Data for dump

EFS: Restoring Data for dump

- **restore(1M)** reads tapes created by **dump(1M)**
- Restores files relative to current working directory
- **restore(1M)** options

  r – all files
  x – some files
  i – interactively choose the files to restore

- Extract everything from the tape in **/dev/tape**

  `# restore r`

- Extract **/disk2/data** from a remote tape

  `# restore xf guest@lizard:/dev/tape /disk2/data`

- Extract interactively

  `# restore i`

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 1 8

Student Notes

`restore -i` reads in the directory information from the tape, and provides a shell-like interface that allows the user to move around the directory tree selecting files and directories to be extracted.

Use the `r` option to restore all files on the archive or `x` to restore the files you designate. Remember that using the `r` flag also restores filesystem information, thereby restoring fragmentation, corruption, etc. You may need to `fsck` after a `restore r`.

The `restore(1M)` command is not valid for XFS filesystems; use `xfsrestore(1M)`.

15-19 XFS: Using `xfsdump`

XFS: Using `xfsdump`

- Only works for mounted XFS filesystems
- Uses a different format than **tar**, **bru**, or **dump**
- Requires root privilege
- Dump levels `0-9` specified with the `l` option

  0 = full backup of all files
  1 = backup all files modified since the 0 level backup
  2 = backup all files modified since the 1 level backup
- `f` Destination device
- `s` Subtree
- `L` Session label
- `M` Media label
- `v` Verbose level (**verbose** , **silent**, or **trace**)

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 1 9

Student Notes

`xfsdump` is similar to `dump(1M)` but works on XFS filesystems. You must be superuser to run `xfsdump` except for the inventory mode. Verbose mode is the default. Use the session label to identify each part of a backup if multiple archives are on one tape. Use the media label to identify the actual tape used for the backup.

15-20 Using `xfsdump` to Create Backups

Using **xfsdump** to Create Backups

- To create a full backup of **/disk2** on a local tape drive

```
# xfsdump -f /dev/tape /disk2
```

or

```
# xfsdump -f /dev/tape -l 0 /disk2
```

- To create an incremental backup of **/disk2** on a remote tape device

```
# xfsdump -f sysA:/dev/tape -l 2 /disk2
```

- To create a full backup of / on a local tape device with no messages

```
# xfsdump -f /dev/tape -l 0 -v silent /
```

- To create a backup of a subdirectory in a filesystem **/disk2/project/data**

```
# xfsdump -f /dev/tape -s project/data /disk2
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 - 2 0

Student Notes

Use the `-F` option to force overwriting of a tape that contains non-XFS data, or use `mt -f /dev/tape erase` to erase tapes with preexisting backups. The defaults are level 0 (full backup) with verbose level messages.

Use the `-s` option to back up just a directory or a file rather than the whole filesystem. Only the files under `/disk2/project/data` will be saved in the last example.

15-21 Example: xfsdump

```
# xfsdump -f /dev/tape -s tmp /
xfsdump: version 2.0 - type ^C for status and control
 ==================== dump label dialog =====================
please enter label for this dump session (timeout in 300 sec)
 -> temporary
session label entered: "temporary"
------------------------ enddialog------------------------
xfsdump: level 0 dump of sysA:/
xfsdump: dump date: Tue Mar  5 08:11:04 1996
xfsdump: session id: d2508a45-b21d-1001-8bd7-080069078466
xfsdump: session label: "temporary"
xfsdump: ino map phase 1: parsing subtree selections
xfsdump: ino map phase 2: constructing initial dump list
xfsdump: ino map phase 3: pruning unneeded subtrees
xfsdump: ino map phase 4: estimating dump size
xfsdump: ino map phase 5: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 5235392 bytes
xfsdump: /var/xfsdump/inventory created
xfsdump: preparing drive
xfsdump: estimated dump size: 5235392 bytes
xfsdump: /var/xfsdump/inventory created
xfsdump: preparing drive
xfsdump: bad media file header at BOT indicates foreign or
corrupted tape
xfsdump: WARNING: media contains non-xfsdump data or a corrupt
xfsdump media file header at beginning of media
 ================= media overwrite dialog =================
overwrite non-xfsdump data on media in drive 0?
1: don't overwrite (timeout in 3600 sec)
2: overwrite (default)
 -> 2
media will be overwritten
 ------------------------ end dialog  --------------------
=================== media label dialog ===================
please enter label for media in drive 0 (timeout in 300 sec)
 -> temp media 1
media label entered: "temp media 1"
------------- end dialog  ----------

 -------------------------- end dialog --------------------------
xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
```

```
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 8388608 bytes
xfsdump: dumping session inventory
xfsdump: beginning inventory media file
xfsdump: media file 1 (media 0, file 1)
xfsdump: ending inventory media file
xfsdump: inventory media file size 4194304 bytes
xfsdump: writing stream terminator
xfsdump: beginning media stream terminator
xfsdump: media file 2 (media 0, file 2)
xfsdump: ending media stream terminator
xfsdump: media stream terminator size 2097152 bytes
xfsdump: I/O metrics: 3 by 2MB ring; 10/18 (56%) records
streamed; 123362B/s
xfsdump: dump complete: 669 seconds elapsed
```

Student Notes

The subdirectory `/tmp` will be archived. Because the `-L` and `-M` options were not given on the command line, you are prompted for a session label and a media label. The session label is "temporary." You are prompted for confirmation before overwriting the tape. The `-F` option would have prevented the question.

The media label is `"temp media 1"`.

15-22 Example: `xfsdump` Inventory Information

```
# xfsdump -I
file system 0:
        fs id:          d256a4c1-b21d-1001-8839-080069078466
        session 0:
                mount point:        sysA:/
                device:                 sysA:/dev/rroot
                time:                    Tue Mar  5 08:11:04 1996
                session label:      "temporary"
                session id:           d2508a45-b21d-1001-8bd7-
080069078466
                level:                     0
                resumed:             NO
                subtree:               YES
                streams:              1
                stream 0:
                        pathname:           /dev/tape
                        start:                    ino 2181599
offset 0
                        end:                     ino 4292545
offset 0
                        interrupted:         NO
                        media files:         2

       media file 0:
                                mfile index:      0
                                mfile type:        data
                                mfile size:        8388608
                                mfile start:        ino 2181599
offset 0
                                mfile end:         ino 4292545
offset 0
                                media label:     "temp media 1"
                                media id: d2508a47-b21d-1001-
8bd7-080069078466
                   media file 1:
                                mfile index:      1
                                mfile type:        inventory
                                mfile size:        4194304
                                media label:     "temp media 1"
                                media id: d2508a47-b21d-1001-
8bd7-080069078466
session 1:
                mount point:    sysA:/
```

```
                device:                 sysA:/dev/rroot
                time:                    Wed Mar  6 08:59:17 1996
                session label:  "full_bkup"
                session id:       d24f03a3-b21d-1001-84be-
080069078466
                level:                    0
 ...                            media label:
"full_backup_tape1"

                subtree:        NO
```

Student Notes

Each dump session updates an inventory database in /var/xfsdump/inventory. This inventory is used to determine the base of an incremental backup or a resumed backup. The media and session labels are recorded in the inventory.

To view the inventory database, use `xfsdump -I`. To limit the amount of information displayed from the inventory, use the option `depth=#`, where # is from 1 to 4, or specify a particular filesystem or media label. The most verbose information is a `depth=4`, which is the default if no depth is specified.

This example has two backups listed. The first is session label "temporary" with a media label of "temp media 1". This is a full level 0 backup of a subtree. The second backup listed has a session label "full_bkup" with a media label "full_backup_tape1".

From the rest of this inventory output, the second backup does not use a subtree, and the second tape has a media label full_backup_tape2.

```
session 1:
        mount point:     sysA:/
        device:          sysA:/dev/rroot
        time:            Wed Mar  6 08:59:17 1996
        session label:   "full_bkup"
        session id:      d24f03a3-b21d-1001-84be-080069078466
        level:           0
        resumed:         NO
        subtree:         NO
        streams:         1
        stream 0:
                pathname:        /dev/tape
                start:           ino 132 offset 0
```

```
                end:               ino 8319152 offset 0
                interrupted:       NO
                media files:       5
                media file 0:
                        mfile index:     0
                        mfile type:      data
                        mfile size:      545259520
                        mfile start:     ino 132 offset 0
                        mfile end:       ino 1466251 offset 0
                        media label:     "full_backup_tape1"
                        media id:        d24f03a4-b21d-1001-84be-
080069078466
                media file 1:
                        mfile index:     1
                        mfile type:      data
                        mfile size:      545259520
                        mfile start:     ino 1466251 offset 0
                        mfile end:       ino 4272002 offset 0
                        media label:     "full_backup_tape1"
                        media id:        d24f03a4-b21d-1001-84be-
080069078466
                media file 2:
                        mfile index:     2
                        mfile type:      data
                        mfile size:      224395264
                        mfile start:     ino 4272002 offset 0
                       mfile end:       ino 5609609 offset 0
                       media label:     "full_backup_tape1"
                       media id:        d24f03a4-b21d-1001-84be-
080069078466
               media file 3:
                        mfile index:     0
                        mfile type:      data
                        mfile size:      505413632
                        mfile start:     ino 5609609 offset 0
                        mfile end:       ino 8319152 offset 0
                        media label:     "full_backup_tape2"
                        media id:        d24f03a5-b21d-1001-84be-
080069078466
                media file 4:
                        mfile index:     1
                        mfile type:      inventory
                        mfile size:      4194304
                        media label:     "full_backup_tape2"
                        media id:        d24f03a5-b21d-1001-84be-
080069078466
```

15-23 Restoring Data for `xfsdump`

Restoring Data for **xfsdump**

xfsrestore(1M) reads backups created by **xfsdump(1M)**

- To restore a full backup of **/disk2** from a local tape device

```
# xfsrestore -f /dev/tape /disk2
```

- To restore only one file from the backup with the session label **full_bkup**

```
# xfsrestore -f /dev/tape -L full_bkup -s xlvm.1.pres
```

- To restore several files

```
# xfsrestore -f /dev/tape -L full_bkup -s xlvm.1.pres
  -s xlvm.2.pres -s xlvm.3.pres
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 2 3

Student Notes

In the first example, the source is identified by `-f /dev/tape`, and the destination by `/disk2`. Wildcards are not allowed as file identifiers in the `xfsrestore` command line. Use the `-s` option to specify multiple files to restore if you do not want to recover the whole filesystem or directory.

15-24 Example: xfsrestore

Example: xfsrestore

- To restore files to a different directory. Files saved as **var/adm/crash** are restored to **/usr/people/guest/var/adm/crash**

```
# xfsrestore -f /dev/tape -L multiple -s
  var/adm/crash/minfree /usr/people/guest
# ls -FC /usr/people/guest/var/adm/crash
  minfree
```

- To restore from incremental backups

```
# xfsrestore -f /dev/tape -r .        (level 0)
# xfsrestore -f /dev/tape -r .        (level 1)
# xfsrestore -f /dev/tape -r .        (level 2) ...
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 2 4

Student Notes

`xfsrestore(1M)` uses the same leading slash stripping as `bru(1)` and `tar(1)`. This may not be exactly where you want the files, but it gets around the absolute path versus relative path problem.

15-25 Example: Interactive `xfsrestore`

```
# xfsrestore -f /dev/tape -i .
xfsrestore: version 2.0 - type ^C for status and control
xfsrestore: searching media for dump
xfsrestore: preparing drive
xfsrestore: examining media file 0
 =================== dump selection dialog ====================
the following dump has been found on drive 0
hostname: sysA
mount point: /
volume: /dev/rroot
session time:         Wed Mar  6 08:59:17 1996
level:                     0
session label:         "full_bkup"
media label:           "full_backup_tape1"
file system id:         d256a4c1-b21d-1001-8839-080069078466
session id:              d24f03a3-b21d-1001-84be-080069078466
media id:                d24f03a4-b21d-1001-84be-080069078466
restore this dump?
1: skip
2: restore (default)
 ->
this dump selected for restoral

----------------------- end dialog -------------------------   --
---------------------------------------
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: directory post-processing
================== subtree selection dialog ==================
the following commands are available:
        pwd
        ls  [ <path> ]
        cd  [ <path> ]
        add  [ <path> ]
        delete  [ <path> ]
        extract
        quit
        help
 -> ls
        1818411    disk2/
                234    stand/
          369705    unix
```

```
        6291587     sbin/
        2097280     var/
        1048704     usr/
        2171805     tmp/

         2097283 opt/
             233 lib/
             131 etc/
-> add passwd.sgi
 -> add opasswd
 -> ls
    3145868     rc0.d/
                 794     inittab.O
       *         805    passwd.sgi
                 830    syslog.conf
       *     20052    opasswd
             20054    passwd
                 189    uadmin
-> extract
 --------------------- end dialog ---------------------
xfsrestore: restoring non-directory files
xfsrestore: I/O metrics: 3 by 2MB ring; 259/265 (98%) records
streamed; 175777B/s
xfsrestore: restore complete: 3127 seconds elapsed

# ls -l /etc/*pass*
-rw-r--r--    1 root     sys           347 Feb 19 23:57 /etc/
ORIG.passwd.sgi
-rw-r--r--    1 root     sys          1315 Feb  5 11:27 /etc/
opasswd
-rw-r--r--    1 root     sys          1467 Feb  7 15:33 /etc/
passwd
-rw-r--r--    1 root     sys           347 Feb 19 23:57 /etc/
passwd.sgi
```

Student Notes

The interactive `xfsrestore(1M)` command allows you to specify the particular files to recover, and to look at the naming of files.

Use the interactive add command to selectively specify which files to recover. Then use `extract` to do the restore.

15-26 List Contents of Backup Tapes

List Contents of Backup Tapes

- **tar -t**
- **bru** formats
  - **List_tape**
  - **/usr/lib/vadmin/backup_restore**
  - **bru -tv**
    - Enable verbose mode **-vv**, **-vvv**, or **-vvvv**, to get more verbosity
- **dump** format

```
# restore -i
 restore>  ls
    proj1
 restore> cd proj1
```

- **xfsdump** format

```
# xfsrestore -f /dev/tape -t
  examine this dump?
  1: skip
  2: restore (default)
   -> 2
  this dump selected for restoral
```

IRIX System Administration: Backup and Recovery   Revision 2.0   15 – 2 6

Student Notes

Use the above commands and options to list the contents of tapes. `bru` is very flexible and displays more verbose levels of information. `xfsrestore(1M)` allows you to skip over multiple archives to select the one you want to display.

15-27 Appending to Tapes

Appending to Tapes

To add another archive:

- Using **tar(1)** or **bru(1)**

```
# tar rv /usr/people/guest
```

- Using **xfsdump(1M)** automatically forwards beyond the existing archive to add the next archive

```
# xfsdump -f /dev/tape /disk3
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 2 7

Student Notes

`tar` adds another archive to a tape by using the `r` option. `xfsdump` automatically handles multiple archives to a tape with no special options.

15-28 Restoring in Miniroot

Restoring in Miniroot

- The root filesystem is mounted at **/root** instead of **/** while in *miniroot*
- Remember that your tapes may have been created with *absolute addresses* (start with /)
- Be careful where you restore your files

```
Admin>sh
# df -k
```

Filesystem	Type	Kbytes	Use	Avail	%Use	Mounted on
/dev/miniroot	xfs	23680	13050	10630	55	/
/dev/dsk/dks0d1s0	xfs	24244	73090	11154	54	/root
/root/dev/dsk/dks0d2s7	efs	1010250	398083	612167	39	/root/disk2
/dev/dsk/dks0d4s7	xfs	478080	467879	10201	98	/CDROM

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 2 8

Student Notes

Use the `shroot` option so that your `root` filesystem *appears* to be mounted at `/`.

Always use the PROM *Recover System* option for full recoveries.

If you have a separate `usr` filesystem, it is mounted at `/root/usr` instead of `/usr` while in *miniroot*.

```
/root/dev/usr  xfs  945675  881633  64042  93  /root/usr
```

15-29 Partial Recovery From Miniroot

Partial Recovery From Miniroot

- Type **sh** or **shroot** at the **Inst>** menu (if your backups have been done with absolute addresses)
- This places you in miniroot shell

```
Inst> shroot
# ls
<contents of your root filesystem>
```

- Restore those critical files from the incremental backup tapes needed to boot the system
- Type `exit` at the **root** prompt (#)
- Type `quit` at the **Inst>** menu
- Reply `y` to reboot the system

IRIX System Administration: Backup and Recovery   Revision 2.0   15 – 2 9

Student Notes

15-30 Partial Recovery From Miniroot (continued)

Partial Recovery From Miniroot (continued)

Using **xfsrestore** from miniroot to recover files:

- Use **sh** or **shroot** and **cd** to the proper directory where the filesystem or files should be restored

```
Inst> sh
# cd /root
# /sbin/xfsrestore -f /dev/tape /
```

or

```
Inst> shroot
# /sbin/xfsrestore -f /dev/tape /
```

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 3 0

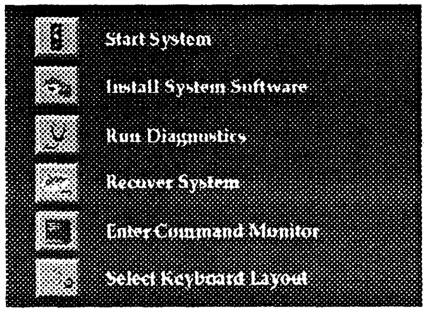
Student Notes

If this procedure fails to bring the system into a bootable state, you may need to follow Full Recovery procedures.

15-31 Full Recovery From Miniroot

Full Recovery From Miniroot

- Place the *IRIX Foundation* CD into the CD–ROM
- Select **Recover System** from the PROM menu:



System responds with this message:

```
System Recovery...
Press <esc> to return to menu.
Insert the installation tape, then press
<enter><enter>
```

- Press <**Enter**> even if it is a CD

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 3 1

Student Notes

15-32 Full Recovery From Miniroot (continued)

Full Recovery From Miniroot (continued)

System then says:

```
Type 'sh' to get a shell prompt
Remote or local restore ([r]emote,
[l]ocal):[l]<enter>
```

- Make proper choice, verify tape drive location, then at next prompt, place the first tape of the recovery tape set into the tape drive
  - If the recovery tape set has more than one tape, place each tape into the tape drive when prompted
  - When your full backup has been completely restored, the recovery tool asks you if you want to enter more tapes:
    - You can install any incremental tapes, or user backups now
    - Or you can answer **no** and stop restoring
  - Respond **yes** when system asks you if you want to reboot the system
  - As system reboots, you can choose to boot into single–user run level and recover incrementals or user backups

IRIX System Administration: Backup and Recovery Revision 2.0 15 – 3 2

Student Notes

You can escape into a miniroot shell, if necessary, at any prompt (type **sh**).