
**HP Computer Systems
Training Course**

**— HP-UX System and Network
Administration I**

Student Workbook

**Version A.01
H3064S Student
Printed in USA 2/99**

Notice

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD PROVIDES THIS MATERIAL "AS IS" AND MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS IN CONNECTION WITH THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL WHETHER BASED ON WARRANTY, CONTRACT, OR OTHER LEGAL THEORY).

Some states do not allow the exclusion of implied warranties or the limitations or exclusion of liability for incidental or consequential damages, so the above limitations and exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior consent of Hewlett-Packard Company.

OSF, OSF/1, OSF/Motif, Motif, and Open Software Foundation are trademarks of the Open Software Foundation in the U.S. and other countries.

UNIX® is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Limited.

X/Open is a trademark of X/Open Company Limited in the UK and other Countries.

HP Education
100 Mayfield Avenue
Mountain View, CA 94043 U.S.A.

© Copyright 1999 by the Hewlett-Packard Company

Contents

Overview

Course Description	1
Student Performance Objectives	1
Student Profile and Prerequisites	6

Module 1 — Introduction to HP-UX Administration

Objectives	1-1
1-1. SLIDE: The Role of the System Administrator	1-2
1-2. SLIDE: Hardware Responsibilities	1-3
1-3. SLIDE: Software Responsibilities	1-4
1-4. SLIDE: Responsibilities to the Users	1-6
1-5. SLIDE: The System Administrator's Tool Kit	1-8
1-6. REVIEW: Check Your Understanding	1-11

Module 2 — Overview of SAM

Objectives	2-1
2-1. SLIDE: Why Use SAM?	2-2
2-2. SLIDE: Using the SAM GUI	2-4
2-3. SLIDE: Using the SAM Terminal Interface	2-6
2-4. LAB: SAM	2-8

Module 3 — Creating and Managing User Accounts

Objectives	3-1
3-1. SLIDE: Users and Groups—Access to System and Data	3-2
3-2. SLIDE: What Defines a User Account?	3-4
3-3. SLIDE: The /etc/passwd File	3-5
3-4. SLIDE: The /etc/group File	3-8
3-5. SLIDE: Creating User Accounts with SAM	3-10
3-6. SLIDE: Creating User Templates in SAM	3-13
3-7. SLIDE: Deactivating and Removing Users with SAM	3-15
3-8. SLIDE: Managing Group Membership with SAM	3-18
3-9. SLIDE: Managing User Accounts from the Command Line	3-19
3-10. SLIDE: Changing User Passwords from the Command Line	3-21
3-11. REVIEW: Check Your Understanding	3-23
3-12. LAB: Hands-On Adding Users	3-24

Module 4 — Customizing User Accounts

Objectives	4-1
4-1. SLIDE: Why Customize a User Account?	4-2

4-2.	SLIDE: Some Sample Customizations	4-4
4-3.	SLIDE: What Happens When a User Logs In?	4-7
4-4.	SLIDE: What Happens at CDE Login?	4-10
4-5.	SLIDE: The /etc/skel Directory	4-12
4-6.	LAB: Customizing User Accounts	4-13

Module 5 — Guided Tour of the HP-UX File Hierarchy

Objectives	5-1
5-1. SLIDE: Introducing the File System Paradigm	5-2
5-2. SLIDE: The File System Layout	5-4
5-3. SLIDE: Application Directories	5-8
5-4. SLIDE: Commands to Help You Navigate	5-9
5-5. LAB: HP-UX File System Hierarchy	5-11

Module 6 — Connecting Peripherals

Objectives	6-1
6-1. SLIDE: The HP 9000 Product Family	6-2
6-2. SLIDE: I/O Architecture Terminology	6-4
6-3. SLIDE: I/O Expansion	6-6
6-4. SLIDE: Device Adapters	6-8
6-5. SLIDE: Types of SCSI	6-9
6-6. SLIDE: SCSI Device Power Up Guidelines	6-11
6-7. SLIDE: Viewing the Configuration with ioscan	6-12
6-8. LAB: Viewing the Configuration with ioscan	6-16

Module 7 — Configuring Device Files

Objectives	7-1
7-1. SLIDE: What Is a Device File?	7-2
7-2. SLIDE: Listing Device Files with ll	7-4
7-3. SLIDE: Listing Device Files with ioscan	7-6
7-4. SLIDE: Listing Device Files with lssf	7-7
7-5. SLIDE: The Layout of /dev	7-9
7-6. SLIDE: Device File Naming Convention	7-10
7-7. SLIDE: Disk Device File Names	7-13
7-8. SLIDE: Tape Device File Names	7-15
7-9. SLIDE: Terminal and Modem Device File Names	7-18
7-10. SLIDE: How Device Files are Created	7-20
7-11. SLIDE: Autoconfiguration	7-21
7-12. SLIDE: Creating Device Files with SAM	7-23
7-13. TEXT PAGE: Creating a Device File with mksf	7-24
7-14. TEXT PAGE: Creating Device Files with insf	7-25
7-15. LAB: Device Files	7-26
7-16. REVIEW: Check Your Understanding	7-29

Module 8 — Configuring Disk Devices

Objectives	8-1
----------------------	-----

8-1.	SLIDE: Disk Partitioning	8-2
8-2.	SLIDE: Whole Disk Partitioning	8-4
8-3.	SLIDE: LVM Disk Partitioning	8-6
8-4.	SLIDE: LVM Device Files	8-8
8-5.	SLIDE: LVM Extents	8-11
8-6.	SLIDE: Creating Physical Volumes	8-13
8-7.	SLIDE: Creating Volume Groups	8-15
8-8.	SLIDE: Creating Logical Volumes	8-19
8-9.	SLIDE: What's Next?	8-23
8-10.	LAB: Logical Volume Manager	8-24
8-11.	REVIEW: Check Your Understanding	8-29

Module 9 — File System Concepts

Objectives	9-1
9-1. SLIDE: What Is a File System?	9-2
9-2. SLIDE: File System Types	9-5
9-3. SLIDE: What's in a File System?	9-7
9-4. SLIDE: Accessing a File System	9-9
9-5. SLIDE: HP-UX Hard Links	9-11
9-6. SLIDE: HP-UX Symbolic Links	9-13
9-7. SLIDE: HFS Structural Overview	9-15
9-8. CHALK TALK: HFS File System Updates	9-17
9-9. SLIDE: HFS Blocks	9-19
9-10. SLIDE: HFS Fragments	9-21
9-11. SLIDE: HFS Implications	9-23
9-12. SLIDE: JFS Structural Overview	9-25
9-13. CHALK TALK: JFS File System Updates	9-27
9-14. SLIDE: JFS Blocks and Extents	9-30
9-15. SLIDE: JFS Implications	9-31
9-16. REVIEW: Check Your Understanding	9-33

Module 10 — File System Creation

Objectives	10-1
10-1. SLIDE: Overview of File System Creation	10-2
10-2. SLIDE: Creating a New File System	10-4
10-3. SLIDE: The <code>newfs</code> Command	10-5
10-4. SLIDE: Mounting the New File System	10-9
10-5. SLIDE: The <code>umount</code> Command	10-12
10-6. SLIDE: Automatically Mounting File Systems	10-14
10-7. SLIDE: CD-ROM File Systems (CDFS)	10-16
10-8. LAB: Creating File Systems	10-18

Module 11 — File System Repair

Objectives	11-1
11-1. SLIDE: File System Maintenance	11-2
11-2. SLIDE: File System Updates	11-3
11-3. SLIDE: Flushing Buffer Cache	11-5
11-4. SLIDE: Introducing <code>fsck</code>	11-7

11-5. SLIDE: Running <code>fsck</code>	11-8
11-6. SLIDE: Checking <code>lost+found</code>	11-12
11-7. LAB: <code>fsck</code>	11-14

Module 12 — File System Management

Objectives	12-1
12-1. SLIDE: Monitoring Disk Usage	12-2
12-2. SLIDE: Routine Management	12-4
12-3. SLIDE: Extending a Volume Group	12-7
12-4. SLIDE: Extending a Logical Volume	12-10
12-5. SLIDE: Extending a File System	12-12
12-6. TEXT PAGE: Summary of LVM Commands	12-14
12-7. LAB: File System Management	12-16
12-8. REVIEW: Check Your Understanding	12-19

Module 13 — System Backup

Objectives	13-1
13-1. SLIDE: Why Back Up?	13-2
13-2. SLIDE: What Do You Back Up?	13-4
13-3. SLIDE: How Often Do You Back Up?	13-6
13-4. TEXT PAGE: System Backup Worksheet	13-9
13-5. SLIDE: How do you Perform the Backup?	13-10
13-6. SLIDE: Using <code>fbackup</code>	13-14
13-7. SLIDE: Using <code>frecover</code>	13-18
13-8. SLIDE: Network Backup and Recovery	13-21
13-9. SLIDE: Being Prepared with <code>make_recovery</code>	13-23
13-10. SLIDE: Creating a Recovery Tape	13-25
13-11. SLIDE: Updating the Recovery Tape	13-27
13-12. TEXT PAGE: System Recovery Checklist	13-29
13-13. LAB: Backup and Recovery	13-30

Module 14 — Scheduling cron Jobs

Objectives	14-1
14-1. SLIDE: The <code>cron</code> Daemon	14-2
14-2. SLIDE: <code>cronfile</code>	14-4
14-3. SLIDE: Managing <code>cronfile</code> with <code>crontab</code>	14-6
14-4. SLIDE: What Happens When a Job is Scheduled?	14-8
14-5. LAB: <code>cron</code>	14-10

Module 15 — Managing Swap Space

Objectives	15-1
15-1. SLIDE: System Memory	15-2
15-2. SLIDE: What Is Swap Space?	15-4
15-3. SLIDE: Types of Swap Space	15-7
15-4. SLIDE: Enabling Swap from the Command Line	15-9
15-5. SLIDE: Enabling Swap via <code>/etc/fstab</code>	15-12

15-6.	SLIDE: Monitoring Swap Space Usage	15-14
15-7.	SLIDE: Guidelines for Selecting Device Swap Areas	15-17
15-8.	SLIDE: Guidelines for Selecting File System Swap Areas	15-19
15-9.	LAB: Swap	15-21

Module 16 — Printer Management

Objectives	16-1
16-1. SLIDE: Overview of the Spooling System	16-2
16-2. SLIDE: Types of Printers	16-4
16-3. SLIDE: Adding Local Printers Using SAM	16-6
16-4. SLIDE: Adding a Remote Printer Using SAM	16-9
16-5. SLIDE: Adding a Network-based Printer	16-12
16-6. TEXT PAGE: Configuring a Network Printer Using JetAdmin	16-14
16-7. SLIDE: Directory Overview	16-18
16-8. SLIDE: What Happens when a File Is Submitted with <code>lp</code>	16-20
16-9. SLIDE: Managing Print Queues	16-22
16-10. SLIDE: Priorities and Fences	16-25
16-11. SLIDE: Troubleshooting the Spooler	16-27
16-12. LAB: Hands-On Adding Printers	16-30
16-13. REVIEW: Check Your Understanding	16-32

Module 17 — Shutdown and Reboot

Objectives	17-1
17-1. SLIDE: HP-UX Operation States	17-2
17-2. SLIDE: Changing State with Shutdown and Reboot	17-3
17-3. SLIDE: System Boot Introduction	17-6
17-4. SLIDE: System Boot Players	17-7
17-5. SLIDE: System Boot Process Overview	17-9
17-6. SLIDE: Autoboot versus Manual Boot	17-12
17-7. SLIDE: Initiating the Boot Sequence	17-13
17-8. SLIDE: Interacting with the PDC/BootRom	17-14
17-9. SLIDE: Interacting with the ISL	17-17
17-10. SLIDE: What Happens after the Kernel Is Loaded?	17-19
17-11. SLIDE: Run Levels	17-21
17-12. SLIDE: Changing Run Levels with <code>init</code>	17-23
17-13. SLIDE: Configuring <code>init</code> via <code>/etc/inittab</code>	17-24
17-14. LAB: Shutting Down and Rebooting Your System	17-26

Module 18 — Reconfiguring the Kernel

Objectives	18-1
18-1. SLIDE: Why Reconfigure the Kernel?	18-2
18-2. SLIDE: Static Kernel Modules	18-4
18-3. SLIDE: Dynamic Kernel Modules	18-5
18-4. SLIDE: Using SAM for Kernel Configuration	18-6
18-5. SLIDE: Moving the New Kernel into Place	18-8
18-6. SLIDE: What If the New Kernel Won't Boot?	18-10
18-7. TEXT PAGE: Manually Tuning an HP-UX 10.x Kernel	18-12
18-8. TEXT PAGE: Some Configurable Parameters	18-14

18-9. LAB: Kernel Configuration 18-19

Module 19 — Managing Software with SD-UX

Objectives	19-1
19-1. SLIDE: Introducing SD-UX	19-2
19-2. SLIDE: SD-UX Software Structure	19-4
19-3. SLIDE: SD-UX Software Depots	19-6
19-4. SLIDE: SD-UX IPD	19-8
19-5. SLIDE: SD-UX Daemon/Agents	19-9
19-6. SLIDE: <code>swinstall</code> Main Menu	19-11
19-7. SLIDE: Select Software to Install	19-13
19-8. SLIDE: Starting the Update	19-15
19-9. SLIDE: Installing Protected Software	19-17
19-10. SLIDE: Listing Software	19-19
19-11. SLIDE: Removing Software	19-21
19-12. SLIDE: SD-UX Command Summary	19-23
19-13. LAB: Hands-On, Using the Software Distributor	19-24

Module 20 — Patch Management (SD-UX)

Objectives	20-1
20-1. SLIDE: Why Install Patches?	20-2
20-2. SLIDE: Patch Naming Conventions	20-4
20-3. SLIDE: Obtaining Patches	20-5
20-4. SLIDE: Retrieving Patches from the Web Patch Database	20-8
20-5. SLIDE: Retrieving Patches from Tape or CD	20-13
20-6. SLIDE: Installing Patches with <code>swinstall</code>	20-14
20-7. SLIDE: Listing Patches	20-17
20-8. SLIDE: Removing Patches	20-19
20-9. LAB: Patch Management	20-21

Module 21 — Connecting to a Network

Objectives	21-1
21-1. SLIDE: Setting an IP Address and Subnet Mask	21-2
21-2. SLIDE: Setting a Default Route	21-4
21-3. SLIDE: Setting a System Hostname	21-6
21-4. SLIDE: Resolving Hostnames to IP Addresses	21-8
21-5. SLIDE: Configuring /etc/hosts	21-9
21-6. SLIDE: Configuring a DNS Client	21-11
21-7. SLIDE: Choosing a Look-Up Service	21-13
21-8. SLIDE: Troubleshooting Tools	21-16
21-9. LAB: Connecting to the Network	21-18

Solutions

Overview

Course Description

The 5-day HP-UX System and Network Administration I course is the first of two courses that prepares the student to be a successful system and network administrator of an HP 9000 workstation or server system.

Student Performance Objectives

Module 1 — Introduction to HP-UX Administration

- Identify responsibilities of a system administrator.
- Identify three sources of information for system administrators.

Module 2 — Overview of SAM

- List two advantages and limitations of SAM.
- Start SAM in either graphical or terminal mode.
- Successfully navigate between menus in SAM.
- Use SAM to explore the system's current configuration and resources.
- View and manipulate SAM object lists and dialog boxes.
- View the SAM log file.
- Use the restricted SAM builder to allow non-root users access to SAM.

Module 3 —Creating and Managing User Accounts

- List the minimum requirements for a user account.
- Identify each field in the `/etc/passwd` file.
- Identify each field in the `/etc/group` file
- Create, modify, and delete a user account from SAM or the command line.
- Deactivate and reactivate a user account from SAM or the command line.

Module 4 — Customizing User Accounts

- List configuration files read during the login process.

- Change the user's default path.
- Change the user's default terminal type.
- Change the user's prompt string.
- Change the user's command line editor.
- Change the user's default printer.
- Manage default configuration files in `/etc/skel`.

Module 5 — Guided Tour of the HP-UX File Hierarchy

- Describe the reasons for separating dynamic and static file systems.
- Describe the key contents of `/sbin`, `/usr`, `/stand`, `/etc`, `/dev`, `/var`.
- Describe the key contents of `/opt`, `/etc/opt`, and `/var/opt`.
- Use `find`, `whence`, and `whereis` to find files in the HP-UX file system.

Module 6 —Connecting Peripherals

- Describe the difference between workstations and servers.
- Define the following terms: bus, device adapter/interface, and hardware path.
- Compare and contrast the purpose of SCSI, serial, MUX, and parallel interfaces.
- View the current hardware configuration with `ioscan`.
- Use `ioscan` output to create a hardware diagram of the system.
- List the steps required to install a new interface card.
- List the steps required to connect a new SCSI, serial, or parallel device.

Module 7 —Configuring Device Files

- Explain the purpose of a device file.
- Explain the significance of major and minor numbers.
- Differentiate between block and character i/o.
- Use `lsdev` to list kernel driver major numbers.
- Use `ll` to determine a device file's major and minor numbers.
- Use `ioscan` to list device files associated with a specific device.
- Use `lssf` to interpret the characteristics of a device file.

- Given a disk, tape, or CD device filename, determine the controller card instance number and target address of the associated device.
- Given a modem or terminal device filename, determine the controller card instance number and port number of the associated device.
- Describe the autoconfig process.
- Create device files using SAM.
- Create device files using `insf`.

Module 8— Configuring Disk Devices

- Describe the reasons for disk partitioning.
- Partition a disk using the whole disk layout approach.
- Describe the features and benefits of LVM.
- Create physical volumes, volume groups, and logical volumes from the command line.

Module 9 — File System Concepts

- List file system types available in HP-UX .
- Describe the difference between "user data" and "metadata."
- Describe the structure of an HFS file system.
- Describe the structure of an JFS file system.
- Compare the process used to update HFS versus JFS metadata.
- Define the terms: **superblock** and **inode**.
- Define the terms: **block**, **fragment**, and **extent**.
- Compare hard and soft links.
- Create hard and soft links.

Module 10 —File System Creation

- Create a file system from the command line and SAM.
- Mount or unmount a file system from the command line.
- Automatically mount a file system via `/etc/fstab`.

Module 11 — File System Repair

- Describe how HFS and JFS handle file system updates.

- Describe how `sync` prevents file system corruption.
- List three causes of file system corruption.
- Check and repair an HFS file system with `fsck`.
- Check and repair a JFS file system with `fsck`.

Module 12 —File System Management

- Monitor space available in file systems with `bdf` and `du`.
- Clean up file systems by purging unused files and core files.
- Clean up the `/var` file system by trimming log files.
- Extend a volume group from the command line.
- Extend a logical volume from the command line.
- Extend a file system from the command line.

Module 13 — System Backup

- Explain why backups are necessary.
- Create a graph file to determine which files are included in the system backup.
- Perform a full backup with `fbackup`.
- Perform an incremental backup with `fbackup`.
- Perform a backup and restore across the network with `fbackup`.
- Create a system recovery tape with `make_recovery`.
- List steps needed to document the system configuration to ensure a smooth recovery in case of a system crash.

Module 14 —Scheduling cron Jobs

- Submit, list, and remove time-scheduled jobs with `cron`.
- Schedule full and incremental backups to run automatically.
- Grant non-root users access to `cron`.

Module 15 — Managing Swap Space

- Explain the concept of demand paging.
- Define physical, available, and lockable memory.

- Determine the amount of configured physical, available, and lockable memory.
- Determine the amount of swap space currently configured and in-use.
- Configure device swap from the command line.
- Configure file system swap from the command line.
- Deactivate swap space.
- List considerations for selecting appropriate file system and device swap areas.

Module 16 —Printer Management

- Distinguish between local, remote, and network printers.
- Add and remove local, remote, and network printers using SAM.
- Start and stop the LP spooler from the command line.
- Check the LP spooler status from the command line.
- Manage print queues from the command line.
- Troubleshoot basic problems with the LP spooler.

Module 17 —Shutdown and Reboot

- Describe the differences between single- and multi-user mode.
- Properly shut down the system using `shutdown` and `reboot`.
- Describe how the PDC, ISL, and HP-UX utilities load the kernel in memory.
- Boot from the primary boot device.
- Boot from an alternate boot device.
- Boot from an alternate kernel
- Boot to single-user mode to perform system maintenance.
- Describe what occurs once the kernel is loaded in memory.
- Describe the purpose of system run levels.
- Change the system run level with `init`.

Module 18 — Reconfiguring the Kernel

- List three reasons for reconfiguring the kernel.
- Define and contrast static versus dynamic kernel modules.

- Describe the structure of the `/stand` directory
- Add and remove device drivers via SAM.
- Add and remove a kernel subsystem via SAM.
- Change a configurable kernel parameter via SAM.
- Boot from a backup kernel.

Module 19 — Managing Software with SD-UX

- Define the term **depot**
- Define the terms **bundle**, **product**, and **fileset**.
- Install, list, and remove software using the SD-UX GUI interface

Module 20 — Patch Management with SD-UX

- List five sources for obtaining HP-UX patches.
- Retrieve and install patches from the patch database.
- Retrieve and install patches from a tape or CD patch depot.
- List currently installed patches.
- Remove unneeded patches.

Module 21 — Connecting to the Network

- Physically connect an HP-UX machine to an existing LAN or SAM.
- Configure a hostname using `set_parms` or SAM.
- Configure an IP address using `set_parms` or SAM.
- Configure a default route using `set_parms` or SAM.
- Configure a host as a DNS client using `set_parms` or SAM.
- Test connectivity with `ping`, and `nslookup`.

Student Profile and Prerequisites

This course is designed for the individual cdxwho is responsible for performing the system administration tasks on an HP 9000 server or workstation. The student should be an experienced HP-UX user and have a working knowledge of shell programming.

The student should have completed *HP-UX Fundamentals*, Course number 51434S.

Module 1 — Introduction to HP-UX Administration

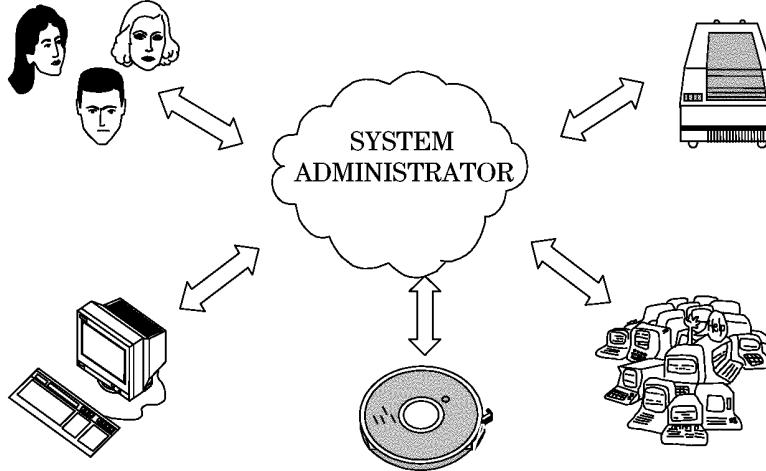
Objectives

Upon completion of this module, you will be able to do the following:

- Identify the responsibilities of a system administrator.
- List three HP-UX documentation sources for system administrators.

1-1. SLIDE: The Role of the System Administrator

The Role of the System Administrator



a5381

Student Notes

The system administrator is responsible for setting up and maintaining the system. Not only must the administrator understand both hardware and software, but he or she must also understand the needs of the user community.

Since many of the tasks associated with these responsibilities require access to commands that should not be available to everyone, the system administrator needs special access to the system. This access is called **superuser** or **root access**.

1-2. SLIDE: Hardware Responsibilities

Hardware Responsibilities

- Create and maintain a hardware diagram of the system.
- Verify that peripherals are installed correctly and tested.
- Monitor performance of hardware components.
- Arrange for repair in event of hardware failure

a5382

Student Notes

The system administrator of an HP-UX system is responsible for configuring and managing the system hardware. The administrator may not be the person who actually installs the hardware. Often a Hewlett-Packard customer engineer will perform the installation of the hardware. Once the system is operational, the administrator must monitor the performance of the various hardware components. If a hardware failure occurs, the administrator should attempt to isolate the problem as much as possible. Depending upon the service agreements in place, the administrator may schedule a customer engineer to make necessary repairs.

The system administrator must know some basic things about the system hardware to be effective in the job.

1-3. SLIDE: Software Responsibilities

Software Responsibilities

- Install and configure the HP-UX operating system.
- Create file systems.
- Manage the integrity of file systems.
- Monitor system resource usage.
- Design and implement backup and recovery routines.
- Configure and maintain printer spooler software.
- Install and maintain network communication software.
- Update the HP-UX operating system for new releases.
- Install and update application software.

a5383

Student Notes

You may need to install the HP-UX operating system software if it is not preinstalled on your system. The operating system is supplied on a distribution medium that can be one of many forms. The software is distributed on a series of magnetic tapes or CD-ROM disks.

The distribution media may be:

- compact disk - read only memory (CD-ROM) disk
- digital data storage (DDS) tapes based on digital audio tape (DAT) technology
- quarter-inch cartridge tape (QIC), (servers only)

The type of media used varies based on the system type; we will discuss this in detail later. The system administrator must install the system software onto a hard disk.

Once the HP-UX system has been installed, it is necessary to configure the software by altering certain values or parameters. This is done to allow the operating system (HP-UX) to recognize additional devices or to increase the efficiency of the system.

HP-UX utilizes a data organization scheme called a file system. Since the file system is where all of the system and user data is stored, it is important to ensure that the integrity of the file system is maintained. The administrator should implement procedures that will detect any errors or corruption in the file system. If problems are found, the administrator must ensure that corrective action is taken.

A file system is of a finite size and usually resides on a locally connected disk drive. The system administrator must monitor the available space in a file system. Procedures should be employed to archive and remove obsolete and unused files so the available free space is not completely consumed. Files that tend to grow in size should also be monitored.

System resources include not only disk storage space, but memory, peripheral devices, and kernel data structures, all of which can be customized and monitored to some extent by the system administrator.

It is the administrator's job to ensure the security of data on the system. Regular data backups are created and maintained. If there is a loss of data due to either user error or hardware failure, recovery procedures can be employed.

Most HP-UX systems include peripheral devices such as line printers and laser printers for hard copy output. The system administrator must manage the software that sends output to the printing devices.

Different HP-UX systems communicate with one another across an electronic communications mechanism called a network. The network allows electronic mail and files to be transmitted from machine to machine. The system administrator must install, configure and monitor networking software.

Hewlett-Packard periodically releases an update to the HP-UX operating system software, and to many of the subsystems and applications programs. An update may enhance or modify existing system features or add new capabilities. The system administrator is responsible for installing each software update so the HP-UX system available to the user community contains the latest version of the software.

A new HP-UX release is not necessarily installed as soon as it is received. It is system administrator's responsibility to ensure that interactions and dependencies among applications and system software are maintained. A new HP-UX release is often followed several months later by releases for applications that make them compatible with the new HP-UX release. Installing the new HP-UX release too soon may break an application that worked on an older release. Obtaining and sharing this information is your responsibility.

1-4. SLIDE: Responsibilities to the Users

Responsibilities to the Users

- Allow user access to the system as required.
- Evaluate user needs.
- Plan for future system growth/change.
- Provide assistance to the user community.
- Implement the policies and procedures of your company/organization regarding the use of the computer system and network.

a5384

Student Notes

Once the HP-UX system has been installed, certain modifications are required to allow a user to access the system. The system administrator must perform these modifications.

The administrator must, to the greatest extent possible, tailor the system to the needs of the user community. The system administrator should analyze the intended use of the system, and should be aware of the number of users on the system, the characteristics of each user, the system resources and peripherals required by each user, and the data and programs that must be shared by various user groups.

As system administrator you will be looked upon as the resident HP-UX expert. Many users will assume that you know everything about the system and will view you as an expert. This occurs independent of your knowledge level. To many users, the fact that you have been trained, in their minds, means that "you must know" more than they do.

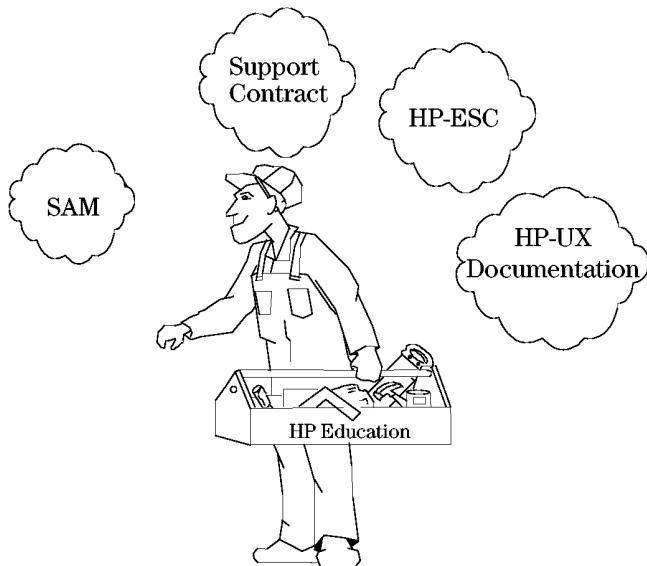
You may be asked many questions such as, "How do I do ...?" and receive comments such as, "My terminal is broken," or "I forgot my password." The problem solving and consulting aspects of the administrator's role can be the most challenging, as well as the most enjoyable

parts of the job. An important message to convey early on is that the HP-UX reference manuals are available on-line on an HP-UX system. Get the users used to at least trying to "look it up for themselves" before coming to you. This can minimize interruptions in your daily activities, and give the users greater confidence in using the system.

Since you will implement the policies and procedures of your company or organization, be aware that these policies and procedures take precedence over the things that HP tells you in this class. We can only recommend certain administration practices. If our recommendations are in conflict with your company practices, clearly you must follow your company guidelines.

1-5. SLIDE: The System Administrator's Tool Kit

The System Administrator's Tool Kit



a6691

Student Notes

The system administrator is responsible for the smooth day-to-day operation of the system, as well as for responding to and correcting large and small emergencies that may occur. In addition, the system administrator is usually the person responsible for making sure that there is a plan in place to recover the system and the data in the event of a disaster of small or large magnitude.

Although this can seem like a monumental task, there are several tools available that can make the job easier. Some of them are noted on the slide.

SAM

The System Administration Manager is a user interface for performing most routine administrative tasks without using the underlying HP-UX commands. SAM can save you keystrokes. However, it is important that you understand the nature of the task you are performing, whether you use SAM or the HP-UX commands. This course will teach the

concepts needed to administer an HP-UX system, and the HP-UX commands used to perform those tasks. You will also be allowed to explore using SAM to perform these same tasks.

Support Contract

Your support contract covers the hardware as well as the software on your system. There are several levels of support, which specify such things as the standard on-site hardware support, software and network assistance from the HP Response Center, and possibly other services such as assigned Account Teams, patch management assistance, operational reviews, system release planning, and assistance with software updates. If you are not familiar with the terms and features of your HP System Support contract, contact your local HP office and ask for the Contracts Coordinator.

HP Electronic Support Center

HP's Electronic Support Center (HP-ESC) provides access to valuable information over the World Wide Web. The URLs for the HP-ESC home page are shown below:

```
http://us-support.external.hp.com      # Asia and North America
http://europe-support.external.hp.com  # Europe
```

The home page contains a problem-searching database, patches, and information on HP products. A support contract is required to access much of the information on HP-ESC.

HP-UX Documentation

Often during the administration of your system, you will need to reference the documentation. HP-UX documentation is available in several forms:

Online man Pages

The man pages (as they are popularly known) are intended for all HP-UX systems. You may view reference material on any command, system call, subroutine, device file, or file format by using the **man** command. The **man** pages are divided into eight sections:

- | | |
|-----|--------------------------------|
| 1: | User Commands |
| 1m: | System Administration Commands |
| 2: | System Calls |
| 3: | Subroutines |
| 4: | File Formats |
| 5: | Miscellaneous Facilities |
| 7: | Device Files |
| 9: | Glossary |

Section **1m** contains information on those commands that are used primarily by a system administrator. Section **4**, File Formats, is also invaluable to the system administrator as it contains information on most of the configuration files that you will be responsible for maintaining.

Other HP Manuals

There are numerous manuals available from HP that describe various aspects of system and network administration. Several that you may want to look at early in your career as an HP-UX administrator are listed below:

- *Managing Systems and Workgroups* provides step-by-step instructions for most common 11.x system administration tasks.
- The *System Administrator's Tasks Manual* is the equivalent HP-UX 10.x manual.
- *Configuring HP-UX for Peripherals* provides more detailed instructions for configuring peripherals, such as terminals, printers, plotters, and other devices both via SAM and the command line.
- *Installing and Updating HP-UX* covers the installation and initial configuration of the HP-UX operating system.

Model-specific owner's manuals are provided with every HP 9000 system.

Instant Information and LaserROM

The manuals listed above (as well as Release Notes, white papers, and much more) are available in electronic format, as well. Instant Information is a CD-ROM product that contains all of HP's documentation for HP-UX 11.x. To launch the Instant Information graphical user interface, type: `dynatext`. The interface provides a powerful search mechanism for finding useful information quickly. Instructions for installing the product and mounting the CD-ROM are included with the Instant Information CD.

Documentation for HP-UX 10.x is available on a similar CD-ROM product called "LaserROM". To launch LaserROM on a 10.x system, type `lrom`. This product, too, includes a powerful search feature for quickly finding information you need.

HP's Documentation Web Site: <http://www.docs.hp.com>

If you don't have access to the Instant Information or LaserROM CD-ROMs, try visiting HP's documentation web site at <http://www.docs.hp.com>. This site contains a wide assortment of HP-UX documentation in standard HTML format.

1-6. REVIEW: Check Your Understanding

Directions

Write the answers to the following questions in the space provided.

1. Describe the role of the system administrator.
 2. What does the system administrator need to understand in order to perform his or her duties appropriately?
 3. What are the three main categories of system administration responsibilities?
 4. What are two items in the system administrator's tool kit?

Module 2 — Overview of SAM

Objectives

Upon completion of this module, you will be able to do the following:

- List two advantages and limitations of System Administration Manager (SAM).
- Start SAM in either graphical or terminal mode.
- Successfully navigate between menus in SAM.
- Use SAM to explore the system's current configuration and resources.
- View and manipulate SAM object lists and dialog boxes.
- View the SAM log file.
- Use the restricted SAM builder to allow non-root users access to SAM.

2-1. SLIDE: Why Use SAM?

Why Use SAM?

Advantages

- Simplifies complex administration tasks
- Minimizes potential for errors
- Provides a built-in help utility

Disadvantages

- Is less flexible than manual configuration
- Is not helpful in some troubleshooting situations

a6692

Student Notes

The System Administration Manager (SAM) is a menu-driven tool designed to perform typical system administration tasks without using the underlying HP-UX commands.

SAM has two user interfaces, an X Window System interface and a text terminal interface. The differences are the screen appearance and the keyboard/mouse interactions.

There are many benefits of using SAM:

- Instead of executing commands from a shell, you work through menus that guide task selection and facilitate data entry.
- Tasks are easier to perform because you do not need to remember (or type) complex commands.
- You get a rich set of functions, and those functions provide significant options and control.
- You can use SAM on any HP 9000 system without relearning anything.

While SAM may make it easier to perform certain tasks, the HP-UX commands can be more flexible and more powerful. As a System Administrator, it is very important that you understand the "manual" way of doing things so that when you need to do something that SAM cannot do, you have the skills. The following items suggest a strategy for using SAM:

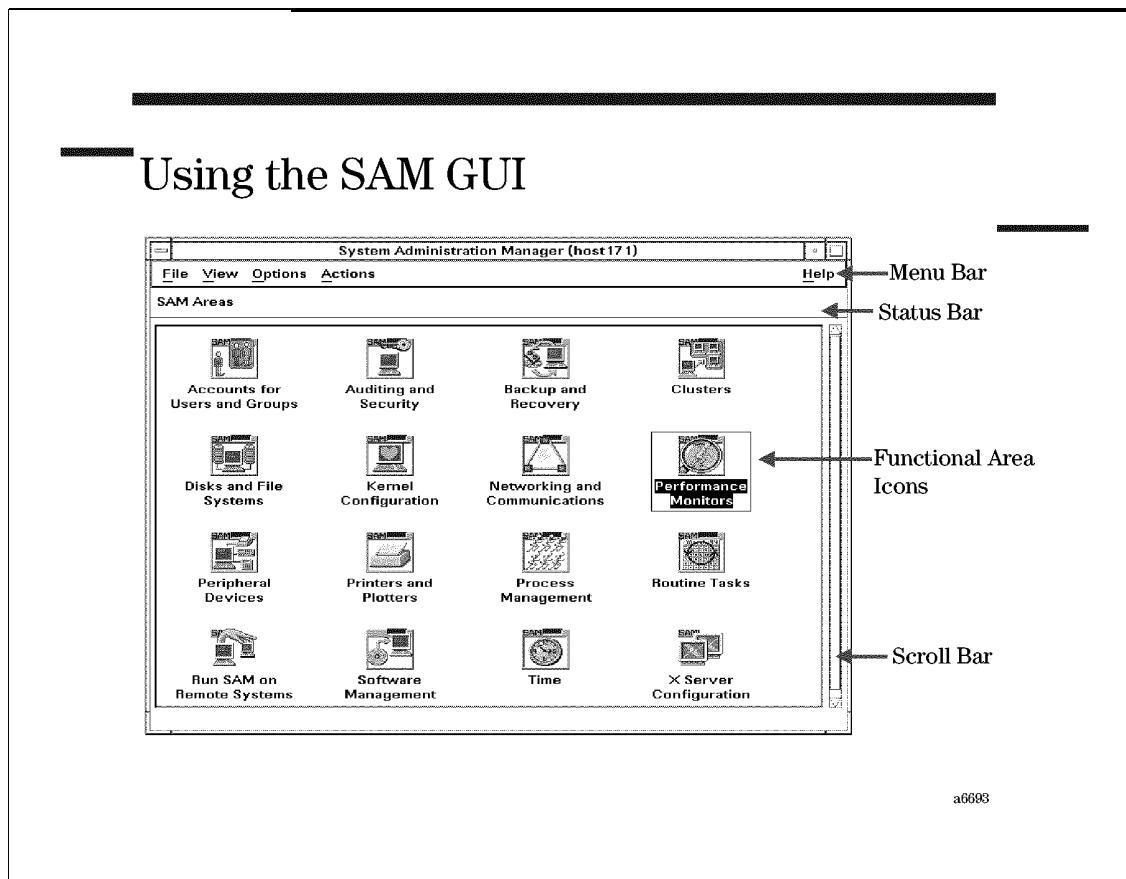
- Use SAM to administer your system whenever it has the capability you need.
- Since SAM does not accommodate every task you need to perform, learn the HP-UX commands for performing a task.
- While performing a task with SAM, if you encounter a situation that SAM cannot accommodate, exit SAM and perform the task using HP-UX commands.
- Use HP-UX commands when SAM cannot perform a task or you know (as an expert) how you want to customize a functionality.

Remember, administering a system requires problem solving skills. The more you understand about your system, the better equipped you will be to solve the problems.

NOTE:

SAM is an optionally loadable part of HP-UX. If you have not loaded SAM onto your system, you will not be able to use it. To use SAM on a workstation or X-terminal that is running the X Window System, you must also have loaded the necessary X11* file sets.

2-2. SLIDE: Using the SAM GUI



Student Notes

Administrators who have access to a terminal running the X Window system can use SAM's graphical user interface (GUI).

To use SAM on the X Window System, the `DISPLAY` environment variable must be set correctly to reflect the display on which you want SAM to appear. The value of the `DISPLAY` variable should be `hostname :0 .0`, where `hostname` is the name returned when you type the `/usr/bin/hostname` command. The `DISPLAY` environment variable is typically set and exported in the user's environment file at logon.

To view your current environment variable values, type `env`. This is how to set the correct values for the `DISPLAY` variable, depending on your shell:

Shell	Environment Variable	Environment File
POSIX, Korn or Bourne shells	<code>export DISPLAY= hostname :0.0</code>	.profile or .dtprofile
C shell	<code>setenv DISPLAY= hostname :0.0</code>	.login

After your DISPLAY variable has been set, you can run SAM simply by typing: `sam`

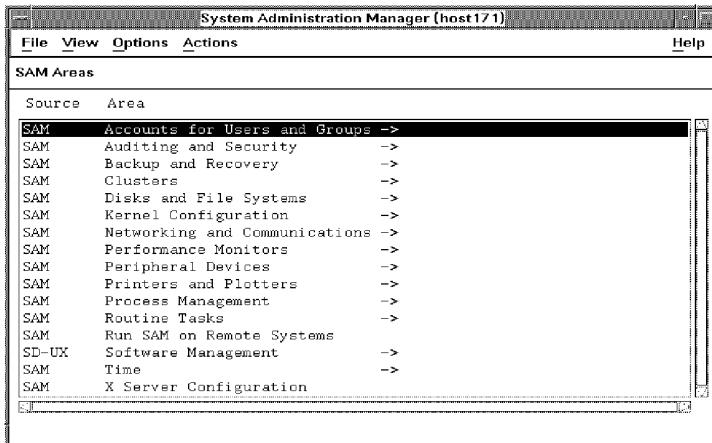
The opening screen encountered when running SAM's GUI contains several components:

- The Menu Bar at the top of the SAM screen provides a series of pull-down menus. The pull-down menus provided vary from screen to screen within SAM. Single-click on any menu to view the available menu options.
- Icons in the middle portion of your SAM window list the available functional areas within SAM. SAM provides tools for most common system administration tasks. To manage user accounts, single-click on the **Accounts for Users and Groups** functional area icon. To add a peripheral to your system, single-click the **Peripheral Devices** functional area icon.
- The Status Bar at the top of each SAM window identifies your current functional area. In some functional areas, SAM will provide additional status and configuration information in the Status Bar.

Some SAM windows may also provide horizontal and vertical scroll bars.

2-3. SLIDE: Using the SAM Terminal Interface

Using the SAM Terminal Interface



a6694

Student Notes

SAM has a special interface for use on character terminals. You use specific keys (or combinations of keys) to move from one part of the screen to another and to move among screens. The structure of SAM is identical for both character and X Window system interfaces; instead of the point-and-click approach, you will use the keyboard to control SAM's actions.

The slide shows how the Control window looks on a character terminal.

Use the up and down arrow keys to highlight different items on the list.

Use **Tab** and **Shift** + **Tab** to move forward and backward to different control buttons.

Activate a highlighted control button (execute that control button function) by pressing the **Spacebar**. You can also use the mnemonic that is underlined on the control button. For example, notice that the **O** on the **Options** is underlined. You can activate the **Options** control button by typing **O** on the keyboard.

To turn a checkbox on or off, use the **Tab** key to move to the checkbox, then press **Spacebar**. An **x** in the checkbox indicates the "on" state. In the "off" state, the checkbox is empty. The space bar toggles the state of the checkbox.

To move to the Menu Bar:

1. Press **F4** (or **Tab**).
2. Use the left arrow and right arrow keys to move to the menu you wish to open, then press **Spacebar**.
3. Use the up arrow key and the down arrow keys to move the highlight to the desired menu item, then press **Spacebar** or type the mnemonic.

In addition, many terminals have function keys which can be used to perform some of these maneuvers.

What the Function Keys Mean

Help	Context-sensitive help
Alt	Modifier key (press first, but don't hold down)
Select	Select or deselect the item, press a button, etc.
Menubar	Activate or deactivate the menu bar
OK	Accelerator for the OK button in a dialog
Apply	Accelerator for the Apply button in a dialog
Shell	Suspends the user interface and provides a shell
Cancel	Accelerator for the Cancel button in a dialog
Close	Accelerator for the Close button in a dialog
Exit	Accelerator for Exit in an object-list screen or control window

2-4. LAB: SAM

Part I: Exploring SAM's Functional Areas

Our goal in this first portion of the lab exercise is to become more familiar with the SAM interface, and to explore some of SAM's functional areas. The exercises reference terms such as **swap** and **file systems**, which we will discuss in detail in later modules. The goal at this point is to explore SAM — don't worry about the details of swap and file system configuration, yet.

1. PART I

Log onto your workstation and start SAM. The first screen in SAM lists a number of functional areas. The following are just a few of the functional areas you may see listed:

accounts for users and groups
backup and recovery
disks and file systems
peripheral devices

2. Which functional area would you choose to view a list of user accounts on your system? Select the appropriate area, and determine the number of accounts on your system. (Note: In some cases, a functional area may contain one or more sub-areas. You may have to drill down through several menus to get to the information you are looking for.) Explore some of the other SAM functional areas to find answers to the questions that follow.
3. Are any tape drives configured on your system?
4. How many disk devices are attached to your system? Do you have any "Unused" disk devices?
5. How many file systems are configured on your system?

6. Are there any currently scheduled automated backups on your system?

Part II: Manipulating SAM Object Lists

You had an opportunity in the first part of this lab to navigate some of the SAM functional areas, and you probably discovered that every functional area eventually leads to a SAM object list. The **Users** object list contains a list of user accounts on the system. The **Disk devices** object list contains a list of disk drives attached to the system. Each object list contains a different type of object that can be managed in SAM. In this part of the lab, you will have an opportunity to manipulate these object lists.

1. PART II

Go to the **Users** object list in SAM. What type of information does the object list display for each user account? List four fields.

2. SAM doesn't always show all the information available for a given object list. For instance, by default, SAM doesn't list user home directory names in the **Users** object list.

You can modify the types and order of columns shown in the object list by launching the **Column editor** from the **Menu Bar** at the top of the screen: **View --> Columns**.

The **Home Directory** field is probably currently marked **Ignore**. Change **Ignore** to **5**, click **OK**, and note the change to your object list .

3. Now use the column editor to put the **Start-up program** in the sixth column of the object list, and hide the **Office location** and **Office phone** fields.

4. You can also customize your object lists by changing the order in which they are sorted. Sort your Users object list by User ID in Descending order. To change the sort order, go to the Menu Bar at the top of the SAM screen and choose: **View --> Sort**. After selecting your sort order in the dialog box that appears, click **OK** and see what happens.

5. On large systems, even after defining a sort order, you may still find it tiresome to scroll through hundreds of user accounts to find an account that you need to modify. SAM allows you to filter object lists to show a subset of the available objects.

For the purpose of this lab exercise, define a filter in the Users object list that lists only user accounts with UIDs greater than 99. In the menu bar, choose: **View --> Filter**.

In the dialog box that appears, change the Operator for UID to Greater than, then type 99 in the Value box to the right. Click **OK** to apply the filter. Now look at the status bar at the top of the object list screen. How does SAM indicate that a filter has been applied?

6. Once you have customized your object list to your liking, save the current sort, column, and filter configuration as the default by selecting: **View --> Save view as default**

7. Go to SAM's **File systems** functional area. It would be helpful to be able to view the amount of space in use in each of your file systems so you could determine if and when new disks may be needed on your system. Add the Percent used column to the object list, and sort the objects in descending order based on this new field. Save this new object list configuration as the default.

Part III: Performing Actions on Objects

So far, we've seen how to view and manipulate object lists. In this portion of the lab, you will have an opportunity to perform actions on some SAM objects. Typically when using SAM you will use the following procedure:

1. Run SAM.
2. Navigate functional areas to the proper object list.
3. Sort and filter the object list to your liking.
4. Select an object or objects with the mouse or space bar.
5. Choose an action from the actions menu to perform on the selected object(s).

1. PART III

Return to the Users functional area in SAM. Without selecting a user from the object list, pull down the Actions menu. Can any actions be performed here without selecting a user?

2. The **Add** action allows you to create a new user account. Select **Actions -->Add**. This should bring up a dialog screen requesting information about the new account. Set **operator** as the user name for the new account, then click the **OK** button so SAM will take the defaults for the rest of the screen. After SAM prompts you for a password for the new account, check the object list to see what happened.
3. Choose **Actions --> Add** again and enter user name **dbmgr**. Instead of pressing **OK**, try the **Apply** button. What is the difference between the **OK** and **Apply** buttons in a dialog box? Can you think of a situation where **Apply** might be a more efficient choice than **OK**?
4. Now try performing an action on an existing object. Using the mouse (GUI) or space bar (TUI), select **user1** from the object list. Now pull down the Actions menu. Now that you have

selected a user, there should be many more actions to choose from. Choose **Modify user's password** and answer the questions that follow.

5. In some cases, the dialog boxes that result from SAM actions may reference terms or concepts you haven't encountered before. Select **user1** from the object list again, then choose **Actions --> Modify**. This should open the **Modify a User** dialog box. The dialog box asks you to enter the user's Primary group and Start-up Program. For more information about terms and concepts you encounter in any SAM window, look for a **Help** button or Help menu on the menu bar. Click on the **Help** button at the bottom of the dialog box window. Skim the text, and try clicking on a couple of the underlined phrases (In the TUI, select one of the "See also" topics instead.) Any underlined phrase in a SAM help window is a link to additional information. When you have finished with the Help window, click **Close** to return to the **Modify a user** dialog box. We didn't really want to modify **user1**'s account after all, so click on **Cancel**.

6. Occasionally, SAM will complain about an action you request. Try an experiment: From the **Users** object list, select **View--->Filter**. Set the User ID (UID) operator to **Any** and click on **OK**. Now select the root user account. Choose **Actions--->Remove**. Can you explain SAM's response to this action? Do you consider this to be a *feature* or a *bug*?

7. In other situations, SAM simply provides a warning message when you attempt a risky action. Try another experiment: From the **Users** screen, select the root user account, then choose **Actions --> Modify**. You will get a warning box, to which the most prudent response at this point is "No"— do not modify root's account.

Part IV: Using the Restricted SAM Builder

By default, SAM may only be run by user root. Administrators in large shops often have the luxury of a staff of operators to assist with system administration tasks. So how can we allow multiple operators, and perhaps even regular users, to access some of the functionality in SAM? Sharing the root password probably isn't the best solution. The **restricted SAM builder** makes it possible to grant non-root users access to selected functional areas in SAM. Our goal in the exercise that follows is to allow the operator user that you created earlier in

the lab to manage processes via SAM and run automated backups, while denying them access to all other areas of SAM.

1. PART IV

Choose **File** --> **Exit**. to exit your current SAM session. Then type **sam -r** to start the restricted SAM builder. (Note: you must be logged in as root to run the restricted SAM builder.)

2. The SAM builder should display a list of user names and templates. Just accept the defaults for now and click **OK**.

3. Next, you will see a window listing each of the SAM functional areas. Note that some functional areas are marked to be enabled (indicated by a green icon), some are disabled (indicated by a red icon), and some are partially enabled (indicated by a yellow icon). In the TUI interface, the status of a functional area is displayed in text form (enabled, disabled, partially enabled). Start by choosing **Actions** --> **Disable all**. What happens to the functional area icons?

4. Next, select the Process management icon with a single-click (GUI) or the space bar (TUI). Choose **Actions** --> **Enable**. What happens to the Process Management icon?

5. Next, we need to enable automated backups. Go to the **Backup and recovery** functional area, and select the Automated backup icon. Again go to the Actions menu and choose: **Actions** --> **Enable**. Go back up to the main functional area window. What happened to the Backup and recovery icon? Why is this icon marked as partially enabled?

6. You can enable and disable as many functional areas as you wish in the restricted SAM builder. Once you have enabled all the desired icons, choose **Actions** --> **Save privileges**.

7. The "Save privileges screen defines which user(s) will have access to the functional areas you have selected. Select the **operator** user from the list , and click **OK**. (If you had multiple operators, you could select multiple user names from the list while holding the shift key. If the names are not together in the list, hold the control key and select the names.) SAM saves the privileges you selected, then returns you to the icon window. Exit the restricted SAM builder, then log off your workstation or server.
8. Log in as operator and try running SAM by typing: `/usr/sbin/sam`. Which functional areas are available for use by your operator user?
9. Exit SAM, then try running SAM again by just typing `sam`. You will probably get an error message, `sh: sam: not found`. Try typing `/usr/sbin/sam`. Can you explain why SAM wouldn't start in the first case, but would start in the second?
10. Log out as user operator, then log back in again as user root.

Part V: Viewing the SAM Log

While going through the exercises so far, have you found yourself wondering what SAM is actually executing on your system on your behalf? SAM logs every action performed so you can review changes that you have made to your system configuration.

1. PART V

Restart SAM. You will be in the functional area launcher. To view the SAM log, go to the Options menu, and choose **View SAM Log**.

2. Experiment with some of the buttons in the dialog box that appears. Which Message level provides the most detail? Have any users other than root used SAM on your machine? When does your SAM log begin?
3. On an active system, the SAM log can grow quickly. You may want to automatically trim the SAM log by choosing **Options --> SAM Log Options..** Configure your system so SAM automatically trims the SAM log to 100000 bytes.

Part VI: Customizing SAM (Optional)

Although SAM has a fair amount of built-in functionality, you may have some data base utilities or custom shell scripts that you would like to be able to execute directly from SAM. SAM allows you to do this. In the exercise that follows, you will create a custom SAM icon that could be used. In the exercise that follows, you will create a custom SAM icon that could be used to execute the `/usr/sbin/whodo` command to view a description of who is doing what on your system.

1. PART VI

From the functional area launcher, choose **Actions --> Add Custom Application.**

2. Choose a name for your icon in the Label field.
3. Enter the full pathname for the `whodo` command, `/usr/sbin/whodo`, in the Command field.

4. `whodo` is not a graphical utility, so choose Terminal Environments as the user interface.
5. Skip the optional fields and click **[OK]**.
6. Try your new icon.
7. Create another custom SAM application that automatically runs the `/usr/bin/cal` program for you to display a calendar of the current month.

Put your custom icon in the SAM's **Time** functional area.

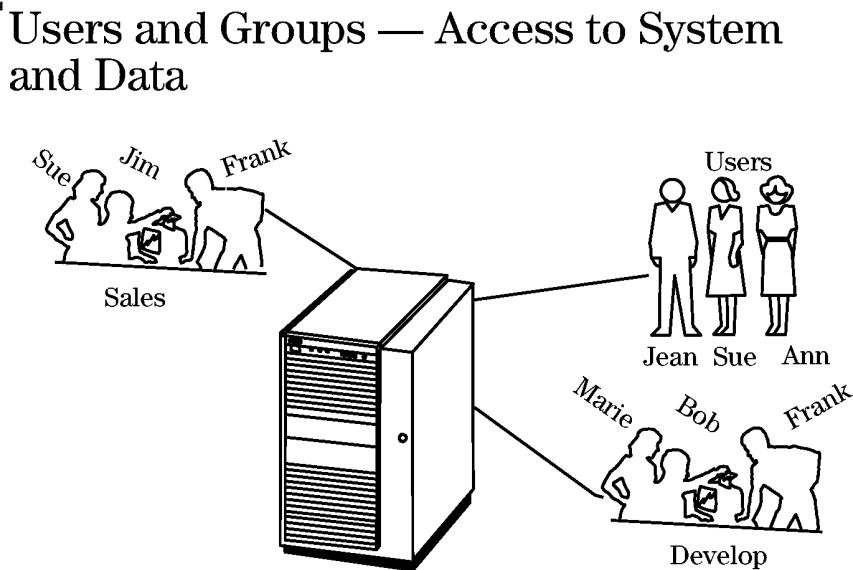
Module 3 — Creating and Managing User Accounts

Objectives

Upon completion of this module, you will be able to do the following:

- List the minimum requirements for a user account.
- Identify each field in the `/etc/passwd` file.
- Identify each field in the `/etc/group` file.
- Create, modify and delete a user account from SAM or the command line.
- Deactivate and reactivate a user account from SAM or the command line.

3-1. SLIDE: Users and Groups—Access to System and Data



Student Notes

In order to gain access to the system and its resources, users are required to log in. By controlling access to your system, you can prevent unauthorized users from running programs that consume resources, as well as control access to the data stored on your system.

Users on the system are assigned to one or more groups. Files can typically be accessed by members of the owner's group, yet they are protected from access by users outside the owner's group. Each user is granted access to files and directories based on the groups to which the user belongs.

You choose a unique user name (or login name) for each person who will be accessing the system. You also choose group names and decide which user names will be assigned to which groups.

You can perform these tasks either by using HP-UX commands, or by using SAM. SAM automatically updates system files and calls appropriate commands for you.

Users can determine the userid and the group membership you have assigned for them by typing `/usr/bin/id` without any arguments.

3-2. SLIDE: What Defines a User Account?

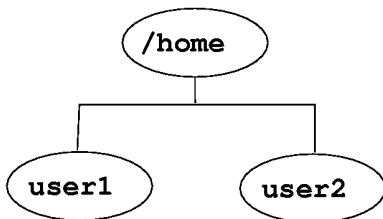
What Defines a User Account?

/etc/passwd

```
user1: . . .
user2: . . .
user3: . . .
user4: . . .
```

/etc/group

```
accounts: . . .
research: . . .
finances: . . .
admin: . . .
```



a6695

Student Notes

For users to successfully log in, they must have a valid user name, user ID, password, and other account information defined in the `/etc/passwd` file. The user may be granted access to additional groups on the system via the `/etc/group` file. Finally, most users have a home directory under `/home`, beneath which they can store their personal files and directories.

This chapter will discuss the structure and purpose of each of these, then consider several approaches for managing user accounts:

- via SAM
- from the command line with `useradd`, `userdel`, and `usermod`
- manually, via the vi editor

3-3. SLIDE: The /etc/passwd File

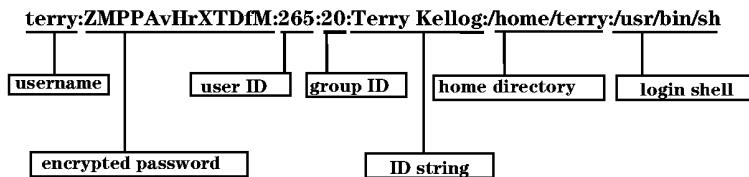
The /etc/passwd File

Example:

```
root:mAj8as.,8as.,ofads:0:3:::/bin/sh
daemon:*:1:5:::/sbin/sh
date:r.c7.0x4/,hGJ:20:1:::/usr/bin/date
erik:.r.ca8/,f2i5y:204:20::/home/erik:/usr/bin/sh
```

Use /usr/sbin/vipw to edit

Use /usr/sbin/pwck



a6971

Student Notes

The /etc/passwd file contains essential information required during login. It contains one entry per line for each valid user of the system. All fields are delimited by a colon (:).

username The user name that is used when a user logs in. It should be between one and eight characters in length and the first character should be alphabetic. If the name contains more than eight characters, only the first eight are significant.

password The encrypted password. It is encrypted by the system when the user sets the password using the **passwd** command. The password should be six to eight characters, one of which is a number or a special character. If the password field is empty, no password is associated with the login name. Never leave the password field empty. Leaving it empty makes it very easy to break into a system.
 An asterisk (*) in the *password* field deactivates an account. Nothing you can type will encrypt to an asterisk, so, no one can log in using the associated login name.

<i>user ID</i>	Each user must be assigned a userid. Userid 0 is reserved for root, and UIDs 1-99 are reserved for other predefined accounts required by the system. Version 10.20 of HP-UX introduced support for User IDs as large as 2,147,483,646. Prior to HP-UX 10.20, UIDs greater than 60,000 were not supported. When users are added using SAM, SAM begins assigning UIDs starting with UID 101. You can choose blocks of UIDs for particular groups:
	<ul style="list-style-type: none"> • 100-199 – marketing • 200-299 – engineering • 300-399 – managers

CAUTION: There may be incompatibilities when sharing files owned by large UIDs with systems that don't support large UIDs.

<i>group ID</i>	The group ID (GID). This number corresponds with an entry in the /etc/group file.
<i>ID string</i>	The comment field. It allows you to add extra information about the users, such as the user's full name, telephone extension, organization, or building number. This field is used by the line printer spooler system and by the finger command.
<i>home directory</i>	The absolute path to the directory the user will be in when they log in. If this directory does not exist or is invalid, then the user is unable to log in.
<i>command</i>	The absolute path of a command to be executed when the user logs in. Typically, this is a shell. The shells that are usually used are /usr/bin/sh, /usr/bin/ksh, and /usr/bin/csh. For system UIDs the shell is /sbin/sh, which is a special (POSIX) shell for the superuser. It should not be changed to another shell. If the field is empty, the default is /usr/bin/sh.

The *command* entry does not have to be a shell. For example, you can create the following entry in /etc/passwd:

```
date:rc70x.4.hGJdc:20:1:::/usr/bin/date
```

The command is /usr/bin/date. If you type date at the login prompt, then type the appropriate password, the system will run the /usr/bin/date command, and then log you out.

NOTE: The permissions on the passwd file should be read only (r--r--r--) and the owner must be root.

Required Entries in /etc/passwd

```
root:rZ1lps2JYh3iA:0:3:::/sbin/sh
daemon:*:1:5:::/sbin/sh
```

```
bin:*:2:2::/usr/bin:/sbin/sh
sys:*:3:3:::
adm:*:4:4::/var/adm:/sbin/sh
uucp:*:5:3::/var/spool/uucppublic:/usr/lbin/uucp/uucico
lp:*:9:7::/var/spool/lp:/sbin/sh
nuucp:*:11:11::/var/spool/uucppublic:/usr/lbin/uucp/uucico
hpdb:*:27:1:ALLBASE:/:/sbin/sh
nobody:*:-2:60001:::
```

Editing /etc/passwd

If you are using vi to edit /etc/passwd and a user attempts to change a password while you are editing, the user's change will not be entered into the file. To prevent this situation, use **vipw** when editing /etc/passwd.

```
# vipw
```

This command puts a lock on the /etc/passwd file by copying /etc/passwd to /etc/passwd.**.tmp**. If a user attempts to change a password, he or she will be told that the passwd file is busy. When you leave **vipw**, some automatic checks are done, and if your changes are correct, the temporary file is moved to /etc/passwd. Otherwise, /etc/passwd will remain unchanged.

Checking the /etc/passwd file

The consistency of the /etc/passwd file can be checked with the /usr/sbin/pwck command. It will check for the number of fields in each entry, and whether login directory and optional program name exist, and validate the number of fields, login name, user ID and group ID.

3-4. SLIDE: The /etc/group File

The /etc/group File

group_name:password:group_id:group_list

Example:

```
other::1:root,daemon,uucp,sync
users::20:
develop::101:bugs,daffy
sales::102:bugs,daffy,elmer,marvin
```

Use **/usr/sbin/grpcck** to check.

a538353

Student Notes

The /etc/group file is used to define groups. The fields are delimited by a colon (:).

group_name is the mnemonic name associated with the group. If you do an **ll** on a file, you will see this name printed in the group field.

password is typically not used, so it is blank. It can contain an encrypted group-level password if you implement privileged groups.

group_id is the group ID (GID). This is the number that should be placed in the /etc/passwd file in the *group_id* field. This number is shared by all group members. It is recommended that the system administrator create groups with IDs of greater than 100 to ensure they do not collide with current or future system needs.

group_list is a list of user names of users who are members of the group. At version 11.0 of HP-UX, not all members of a given group are listed in the /etc/

group file. A user's initial login group is defined in the fourth field of **/etc/passwd**, not in the **/etc/group** file.

Note that a user can be a member of more than one group. A user can use the **newgrp** command to change to a different group.

```
$ newgrp group_name
```

The new group is referred to as the **effective group** of the user. Changing to a new group does not alter the user's primary group entry in the **/etc/passwd** file, it only alters the user's group association for any files he creates after executing the **newgrp** command. Executing the **newgrp** command with no parameters returns the user to the group to which he is assigned in the **/etc/passwd** file.

Required Entries in **/etc/group**

```
root::0:root
other::1:root,lpdb
bin::2:root,bin
sys::3:root,uucp
adm::4:root,adm
daemon::5:root,daemon
mail::6:root
lp::7:root,lp
tty::10:
nuucp::11:nuucp
nogroup::*:-2:
```

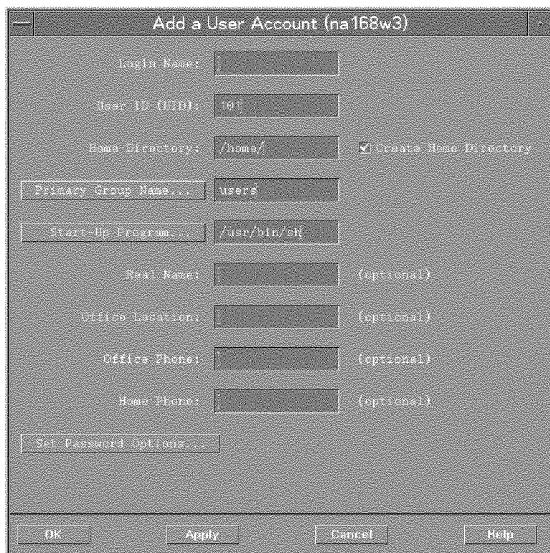
For more information on the **/etc/group** file, see **group(4)** in the *HP-UX Reference* manual.

Checking the **/etc/group** file

The consistency of the **/etc/group** file can be checked with the **/usr/sbin/grpcck** command. It will check for the number of fields in each entry, and whether all login names appear in the password file.

3-5. SLIDE: Creating User Accounts with SAM

Creating User Accounts with SAM



a6696

Student Notes

SAM provides a menu driven interface to add user accounts. In order to add an account using SAM, choose **Accounts for Users and Groups** from the SAM control window.

Choose **Users** from the next menu. SAM will display a list of users. If there are more than 500 users on your system, you will be asked to select a subset of users.

Choose **Add..** from the **Actions** menu. SAM will display a form to be completed in order to add a user.

You will need to fill in the following information:

Login Name	The user name that is used when a user logs in. It should be between one and eight characters in length. The first character should be alphabetic. If the name contains more than eight characters, only the first eight are significant.
------------	---

User ID	The user ID (UID). UID zero (0) is reserved for root. Typically, values 1-99 are reserved for the system. User IDs as large as 2,147,483,646 can be used with HFS file systems and with JFS Version 3 file systems. JFS Version 3 is installed by default with HP-UX 10.20; file systems created with earlier versions of JFS must be upgraded to Version 3 to support large user IDs (IDs larger than 60,000). SAM will use UIDs starting from 101. You can choose override the UID SAM has assigned. You may want to do this to ensure unique UIDs in a network.
Home Directory	The absolute path to the directory the user is in when logging in. If this directory does not exist or is invalid, the user is unable to log in. By default SAM will append the Login Name to /home. There is a check box to determine whether the home directory is to be created. In the case where users will be sharing a home directory you may want to turn this off.
Primary Group Name	This field defines the primary group that is assigned to the user at login. The default is users. To choose another group, type over the word users . Or, to choose from a list of groups, highlight the Primary Group Name and SAM will display a list for you to choose from.
Start-Up Program	The absolute path of a command to be executed when the user logs in. Typically, this is a shell. The shells that are usually used are /usr/bin/sh, /usr/bin/ksh, /usr/bin/keysh, and /usr/bin/csh. Other choices for restricted shells include /usr/bin/rsh and /usr/bin/rksh. It is not required that the start-up program be a shell. You can enter the full pathname of any valid command here. When the user logs in, the command will be executed. When the command terminates, the user is logged off.
Set Password Options...	Selecting this push-button opens a new dialog window for defining which of the following password behavior or aging characteristics are in force for the selected user:
No Restrictions	The user account being added or modified can have the same password indefinitely or for the duration of the account without changing it.
Force Password Change at Next Login	The user must modify his or her own password the next time they log in on the system, then the password aging reverts to No Restrictions.
Allow Only Superuser to Change Password	Only the super-user (user root) can change a user password. Individual users are not allowed to select their own password.
Set Up Periodic Password Aging	Set up password expiration with a minimum required time between password changes. This option leads to

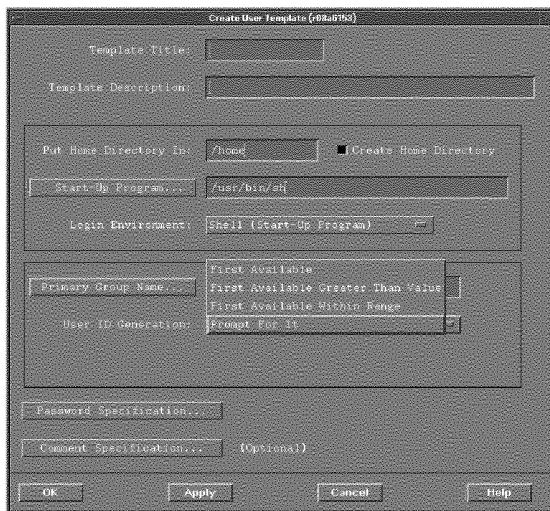
additional fields for setting up the time intervals for expiration and minimum time between changes.

When you have completed filling out the menu, press **OK**, if you do not want to add additional users, or **APPLY**, if you do.

You will be prompted for a password for the new user. SAM will create the **/etc/passwd** entry, update **/etc/group**, and create the home directory if requested.

3-6. SLIDE: Creating User Templates in SAM

Creating User Templates in SAM



a669145

Student Notes

If you plan to add many users with the same requirements you can simplify the process by creating a template.

From the **User Accounts** screen select **User Templates** from the **Actions** menu.

When you select **Create**, SAM will display the menu shown.

You must supply a template name. Names can be up to 16 characters in length and any combination of letters, numbers, and the underscore.

Most fields are the same as those on the Add User menu. You may choose to handle user ID generation in one of four ways:

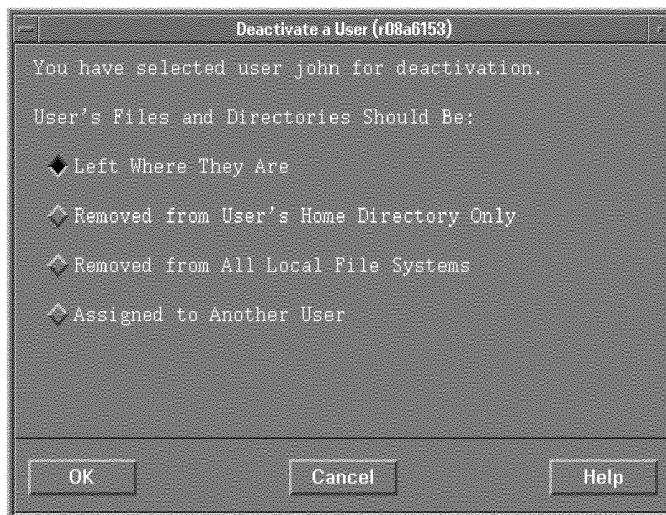
- first available
- first available greater than one
- first available within range
- prompt for it

When you have completed the menu press **OK** if you do not wish to create additional templates or **Apply** if you do.

To use a template when adding users select **User Templates** from the **Actions** menu of the User Accounts screen. When you choose **Select** SAM will display a list of templates. Highlight one and press **OK**. The selected template will be in effect until it is unselected. The template will be used for all new users added until you **Unselect** the template.

3-7. SLIDE: Deactivating and Removing Users with SAM

Deactivating and Removing Users with SAM



a6697

Student Notes

If you no longer want a user to have access to the system you can either remove the user or deactivate the account. Removing the user deletes the user's /etc/passwd and /etc/group entries. Deactivating the user causes SAM to place an * in the password field of /etc/passwd. You can either use SAM to reactivate the account or simply change the user's password using the passwd command.

Choose **Users** from the next menu. SAM will display a list of users. If there are more than 500 users on your system, you will be asked to select a subset of users.

Choose either **Remove** or **Deactivate...** from the **Actions** menu. SAM will display a list of choices for handling the user's files.

The reason you are removing a user will guide you in determining whether to keep that user's files on the system, reassign them to someone else, or remove them entirely.

SAM provides two features to help protect against inadvertent removal of users or files when removing users:

- Editable list of users to exclude from removal.

When prompting for the name of a user to remove from the system, SAM checks the name given against a list of names specified in the file `/etc/sam/rmuser.excl`. If the name matches one within the file, SAM does not remove the user.

- Editable list of files to exclude from removal when a user is removed from the system.

When SAM removes a user, all files (or a subset thereof) for that user are also removed, unless the ownership is given to another user. Before removing a file belonging to the user, SAM checks to see if the file resides in a path that has been excluded from removal. SAM uses the file `/etc/sam/rmfiles.excl` to determine which paths have been excluded from removal. So, for example, if the path `/home/joe/test` is named in the file, SAM will not remove any files residing beneath that directory. SAM logs a list of all files it removes in the file `/var/tmp/sam_remove.log`.

NOTE: You can edit the files `/etc/sam/rmuser.excl` and `/etc/sam/rmfiles.excl` to contain users and directories that you want to exclude from removal from SAM.

Here is a sample `/etc/sam/rmuser.excl` file:

```
root
daemon
bin
sys
adm
uucp
lp
nuucp
hpdb
nobody
```

Here is a sample `/etc/sam/rmfiles.excl` file:

```
/dev
/etc
/export
/mnt
/opt
/sbin
/stand
/usr
/var
```

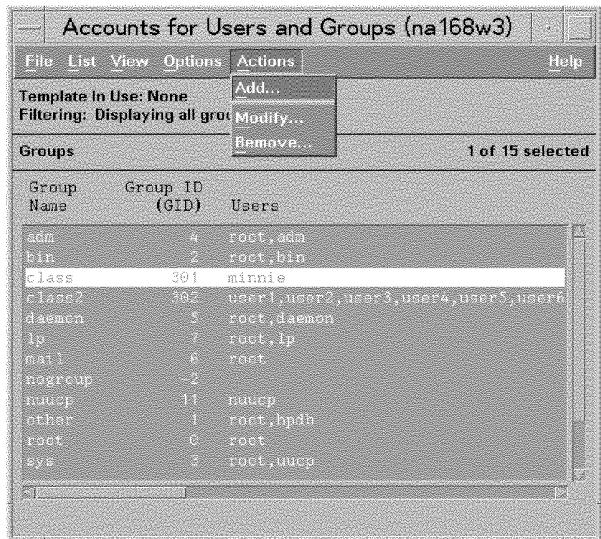
Here is a sample `/etc/sam/rmgroup.excl` file:

```
root
other
bin
sys
admin
daemon
```

mail
lp
nogroup

3-8. SLIDE: Managing Group Membership with SAM

Managing Group Membership with SAM



a6699

Student Notes

SAM provides the most intuitive method for managing group membership. After selecting any group from the object list, go to the Actions menu and choose:

- Add: To create a new group
- Modify: To modify the group membership list
- Remove: To remove the group entirely

3-9. SLIDE: Managing User Accounts from the Command Line

Managing User Accounts from the Command Line

`useradd, usermod, userdel`

`groupadd, groupmod, groupdel`

a66910

Student Notes

Another method for managing accounts and groups is using the command line interface. The `useradd` command creates `/etc/passwd` and `/etc/group` entries. It also optionally creates the user's home directory and copies the files from a skeleton directory into the home directory. The syntax is

```
useradd [-u uid [-o]] [-g group] [ -G group  
[ , group ... ] ] [-d dir] [-s shell] [-c comment] [-m [-k skel dir]] login
```

The `-o` option allows non-unique UIDs. The `-m` option causes the home directory to be created. It looks intimidating at first, but many of these options have defaults.

Examples

Create an account for user `renay` with a home directory of `/home/renay`, which is to be created. Use `/etc/skel` as the skeleton directory and `/usr/bin/sh` as the shell. Take the next highest UID.

```
useradd -m -s /usr/bin/sh renay
```

Create an account for a user named Tracy. Tracy's primary group is staff. Tracy is also in the groups pe and chemistry. Create a home directory. Take all the rest of the defaults.

```
useradd -m -g staff -G pe,chemistry tracy
```

Display the **useradd** defaults.

```
useradd -D
```

Change the default group to others.

```
useradd -D -g others
```

To delete a user use the **userdel** command. The syntax is

```
userdel -r login
```

The **-r** option is needed to delete the user's home directory.

The syntax of the **groupadd** command is

```
groupadd [-g gid [-o]] group
```

Example:

Add a group called economics. Use the next available group ID greater than 100.

```
groupadd economics
```

3-10. SLIDE: Changing User Passwords from the Command Line

Changing User Passwords from the Command Line

\$ passwd *Users can change their own password
(must know current password)*

Changing password for bugs

Old password:

New password:

Re-enter new password:

\$

passwd bugs *Root can change any user's password*

New password:

Re-enter new password:

#

passwd *Root can change the root password*

Changing password for root

New password:

Re-enter new password:

#

a6972

Student Notes

Any time a password needs to be changed, whether by a regular user or the superuser, the **passwd** command is used. Normally an ordinary user on the system can change his or her own password (but nobody else's). When invoked, the user is prompted to enter the existing password. Upon entering the correct password, the user is prompted to enter the new password. After the new password is entered, the user is prompted to enter it again. This is done for verification purposes and to ensure the user didn't make a typing error. If the second password does not match the first, the password is not changed and the user is returned to the shell.

If a user forgets his or her password, the user must seek the assistance of the administrator. The administrator can change any other user's password by invoking **passwd** with an argument of the user's login name.

As mentioned previously, there is an option when creating an account to disallow user changes to his or her password. In this case only the administrator, as super-user, can change the user password.

If, for some reason the root password needs to be changed, the administrator should invoke **passwd** while logged in as superuser.

When changing or assigning a password as a user, note the following:

- Passwords must contain at least 6 characters. Though a password may be assigned more characters, only the first eight are significant.
- Passwords must contain at least two alpha characters (upper or lower case) and at least one numeric or special character. This enforces a certain level of security within the password structure.

NOTE: When you use the **passwd** command, a copy of the old **/etc/passwd** file is saved in **/etc/opasswd**.

You can also use the **passwd** command to change password aging for an account.

```
passwd -f -n min -x max name
```

min and *max* are expressed in days but will be rounded up to the nearest week. The **-f** option forces a user to change his or her password at the next login.

Examples:

Enforce password aging for the account **fontana**:

```
passwd -n 7 -x 35 fontana
```

Force the user **buddy** to change his password at the next login:

```
passwd -f buddy
```

3-11. REVIEW: Check Your Understanding

Directions

Write the answers to the following questions.

1. What steps is SAM taking when performing the following tasks?

Adding a new user

De-activating a user

Modifying a user's information

Adding a new group

2. Describe the 7 fields of the `/etc/passwd` file

3. Describe the fields of the `/etc/group` file.

4. What does it mean to set up groups? Explain.

3-12. LAB: Hands-On Adding Users

Directions

Perform the following tasks. Write the commands you use, and the answers to any questions that are asked.

1. Invoke **SAM** and add a user to your system. (You must be superuser to invoke SAM.) Use your name as a user name. Assign the user to a group called **class** and give him or her the POSIX shell.

Now, exit SAM and look at the **/etc/passwd** and **/etc/group** files. Do you see the user you added?

2. Add a user to the system using HP-UX commands. This time, use your partner's name as the user name. (If you don't have a partner, pick any name.) Use a group called **class** and give the new user the C shell.

Look at the **/etc/passwd** and **/etc/group** files. Do you see the user added? Assign a password for the new account.

3. Run the commands to check the integrity of the **/etc/passwd** and the **/etc/group** files. Discuss your findings with the instructor.

4. Add a user called **date** that executes the **date** command. What would happen if you tried to log in using the user name **date**?

5. Use SAM to deactivate one of the new accounts you set up using SAM. Is the account still listed in **/etc/passwd**?

Module 4 — Customizing User Accounts

Objectives

Upon completion of this module, you will be able to do the following:

- List configuration files read during the login process.
- Change the user's default PATH.
- Change the user's default terminal type.
- Change the user's prompt string.
- Change the user's command line editor.
- Change the user's default printer.
- Manage default configuration files in `/etc/skel`.

4-1. SLIDE: Why Customize a User Account?

Why Customize a User Account?

- Set the user's terminal type.
- Customize the user's prompt.
- Set the user's default printer.
- Customize the user's PATH variable.
- Define a command line editor.

a66912

Student Notes

Simply creating an entry for a user in `/etc/passwd` and `/etc/group` may not give the user all the functionality needed.

- You may need to define the user's terminal type so applications can properly write to the user's screen.
- You may want to customize the user's prompt. Many users like the present working directory to appear in the prompt string.
- Although the system administrator defines a system default printer, individual users may choose to select a different default destination printer.
- If the user accesses third party applications, you may need to modify their PATH variable so their shell can find the application executables.

- Some special configuration is required if the user wishes to use command line editing and the command history mechanism. You may wish to configure this functionality for new user accounts.

4-2. SLIDE: Some Sample Customizations

Some Sample Customizations

```
export TERM='vt100'  
export PS1=' $PWD '$  
export LPDEST='laser'  
export PATH=$PATH:/usr/local/bin  
export EDITOR=vi  
export HISTSIZE=50  
export HISTFILE=~/sh_history
```

a6973

Student Notes

All of the features mentioned on the previous slide are configured via "environment variables" that are set during the login process. Some of the most commonly modified environment variables are listed below:

TERM: The TERM variable defines the user's terminal type. If the TERM variable is set incorrectly, applications may not be able to write to the user's terminal properly. Valid terminal types are listed in the /usr/lib/terminfo/* directories. You can explicitly set an appropriate TERM value using a command similar to the following:

```
export TERM=vt100      # for a vt100 type terminal  
export TERM=hp        # for an HP ASCII terminal  
export TERM=dtterm    # for a dtterm terminal emulator window
```

More commonly, however, the TERM variable is set using the **ttytype** command, which can usually automatically determine your terminal type. The

following portion of code can be included in one of the scripts that runs at login to set your terminal type for you:

```
if [ "$TERM" = "" -o \
      "$TERM" = "unknown" -o \
      "$TERM" = "dialup" -o \
      "$TERM" = "network" ]
then
    eval `ttypype -s -a`
fi
export TERM
```

PS1: The PS1 variable defines your shell prompt string. This, too, can be changed by the user. Some useful sample PS1 values are shown below:

```
export PS1='\$ '          # Use a simple "$ " prompt
export PS1='\$PWD \$'      # Include the user's pwd in the prompt
export PS1='\$PWD (\$LOGNAME) \$' # Include the user's username, too
```

LPDEST: LPDEST defines the user's default printer. The printer named in LPDEST takes precedence over the system-wide default printer configured by the system administrator. Examples:

```
export LPDEST=laser      # use "laser" as the default printer
export LPDEST=printera   # use "printera" as the default printer
```

PATH: Every time the user enters a command, the shell must find the executable associated with the requested command. The PATH variable contains a ":" separated list of directories that the shell should search for executables. If users need access to new applications and utilities, you may need to modify their PATH variables. You can append a new directory to the user's PATH using syntax similar to the following syntax:

```
PATH=$PATH:/usr/local/bin # adds /usr/local/bin
                           # to the existing PATH
```

The initial PATH variable value usually taken from the /etc/PATH file. Oftentimes installing an application automatically updates the /etc/PATH file for you, so it may not be necessary to update individual users' PATHs.

EDITOR: Three variables must be defined if your users want to use command line editing:

```
export EDITOR=vi
export HISTFILE=~/sh_history
export HISTSIZE=50
```

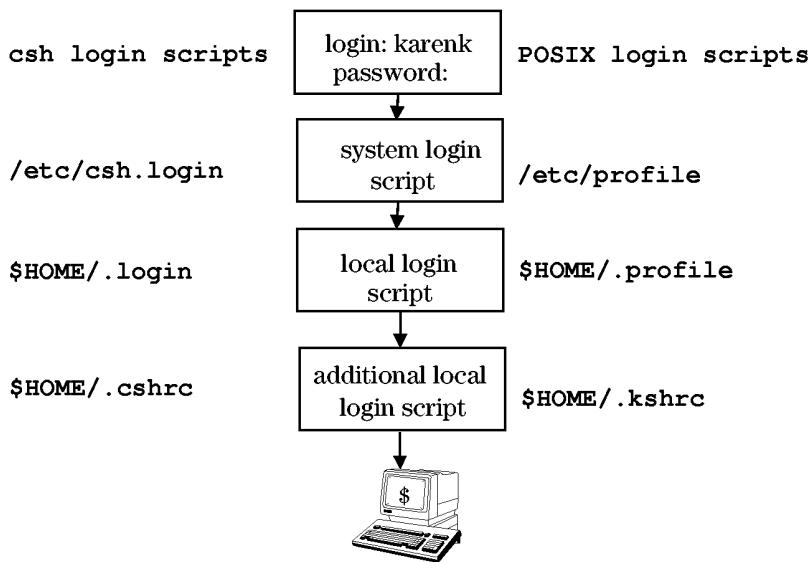
EDITOR defines the user's preferred command line editor. `emacs` and `vi` are the only allowed values. HISTFILE determines the file that should be used to log commands entered by the user. HISTSIZE determines the number of commands retained in the shell's command buffer.

These are just some of the more commonly defined environment variables that you can define for your users. Other environment variables are defined in the man page for the POSIX shell ([man 1 sh-posix](#)), and still others may be required by your applications.

Environment variables can be set from the command line, but are more commonly defined in the login configuration files, which will be covered later in this chapter.

4-3. SLIDE: What Happens When a User Logs In?

What Happens When a User Logs In?



a66914

Student Notes

Although environment variables used to customize a user's login environment can be defined from the command line, they are more commonly defined by several scripts that execute during the login process. Typically, three scripts execute during the login process to initialize the user's environment.

The first is a system-wide script maintained by the system administrator to define general environment variables required by all users.

The second is a local login script maintained by each user. Local login scripts can override or modify the system defaults on a user-by-user basis.

The optional additional local login script can define additional environment variables or set additional shell features. In the POSIX shell, this additional script is often used to define shell aliases.

The login script names vary from shell to shell. A detailed discussion of the login scripts used by the POSIX, Korn, and C shells follows:

The Shell Environment Initialization Sequence

1. The shell runs the appropriate system login script, which initializes the user's environment. The system login scripts define a **default** environment, and can be customized by the system administrator.

If the Shell is...	The System Login Script is...
Bourne (/usr/old/bin/sh)	/etc/profile
Korn (/usr/bin/ksh)	/etc/profile
POSIX (/usr/bin/sh)	/etc/profile
Restricted (/usr/bin/rsh , /usr/bin/rksh)	/etc/profile
C (/usr/bin/csh)	/etc/csh.login

As shipped, these scripts define and export for shell use the environment variables **PATH**, **TZ**, and **TERM**. Inside these scripts, the files **/etc/PATH**, **/etc/MANPATH**, **/etc/TIMEZONE**, **/etc/SHLIB_PATH** are sourced. Since the system login scripts are run for all users at login, the system administrator can modify these files to set global defaults for all users. This is useful for ensuring that each user runs essential commands at login.

2. Displays the contents of the **/etc/copyright** and **/etc/motd** file.
3. Notifies the user of unread news with the prompt:

`news: news_filename`

4. The shell runs the user's *local login script* (if it exists) in the user's home (login) directory:

If the Shell is ...	The Local Login Script is ...
Bourne (/usr/old/bin/sh)	.profile
Korn (/usr/bin/ksh)	.profile
Posix (/usr/bin/sh)	.profile
Restricted (/usr/bin/rsh , /usr/bin/rksh)	.profile
C (/usr/bin/csh)	.login

NOTE:

Typically, the system administrator initially creates a local login script for each user. If SAM or **useradd** is used to add a user, it copies the default local login script (the **/etc/skel/.profile**) to the user's home directory. Users can further customize their environments by modifying these files to suit their needs.

In addition to the above scripts, the POSIX, Korn and C shells may (and usually do) have additional local login scripts:

- POSIX and Korn shell – if the ENV variable is defined, the shell runs the file defined by ENV (typically, **.shrc** whenever a new shell is started. Many programs (for

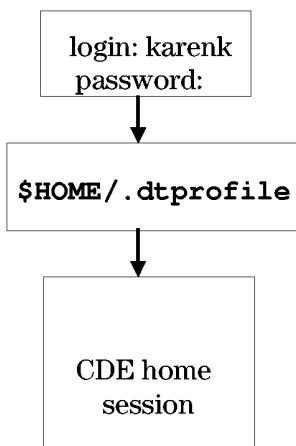
example, `vi` and `mailx`) allow users to start a shell from within the program; this is called a *shell escape*. The ENV file is re-run for a shell escape, whereas `.profile` is run only at login. The ENV file (either `.kshrc` or `.shrc`) will be executed *after* `.profile` executes upon login.

- C shell – Runs the `.cshrc` file whenever a new C shell is started. This is similar to how the Korn shell ENV file works. The `.login` file is run only at login, whereas `.cshrc` is rerun for every new C shell. The `.cshrc` file will be executed *before* the `.login` upon login.

5. Once all initialization is complete, the shell displays a prompt and waits for input from the user.

4-4. SLIDE: What Happens at CDE Login?

What Happens at CDE Login?



a66915

Student Notes

CDE has its own login manager. There are many ways to customize a CDE session.

Using Login Profile Scripts

Login profile scripts `$HOME/.profile`, and `$HOME/.login` are normally not used by CDE as they often contain terminal-I/O-based commands which could cause problems with a non-terminal, graphical interface. However, you may force `$HOME/.profile` (sh/ksh users) or `$HOME/.login` (csh users) to be run by setting the following environment variable in `.dtprofile`:

```
DTSOURCEPROFILE="true"
```

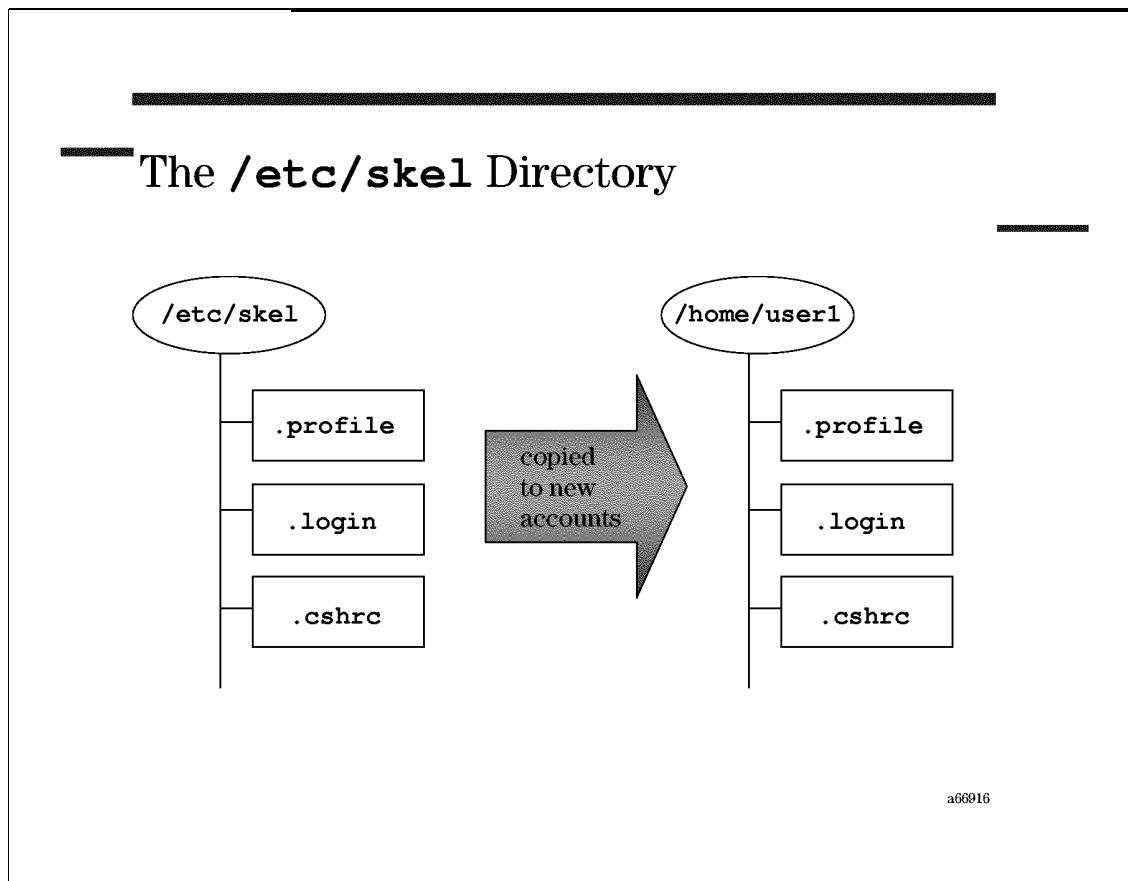
If you plan to source your `.profile` or `.login` script, you should first modify it so that it can be used in both CDE and non-CDE environments by enclosing CDE-only and non-CDE-only commands in *if* blocks that test the variable `SDT`. If the script is run by CDE, the environment variable `SDT` will be defined as "true"; if the script is run by a character-based login, `SDT` will not be defined. For example,

```
if [ ! "$DT" ]; then  
  
# Commands and environment variables used when logging  
# into a non-CDE session  
  
stty ...  
tset ...  
  
fi  
# Commands and variables common to both CDE and non-CDE sessions  
PATH=$HOME/bin:$PATH  
DISPLAY=mydisplay:0  
MAIL=/usr/bin/mail/$USER  
EDITOR=/usr/bin/vi
```

When a user logs into CDE for the first time, an initial **.dtprofile** is copied from **/usr/dt/config/sys.dtprofile**. If you wish to change the template **sys.dtprofile**, first copy it to **/etc/dt/config**.

Each new user's initial **.dtprofile** contains comments on how to run the local **.profile**.

4-5. SLIDE: The /etc/skel Directory



Student Notes

When a new user account is created with SAM or `useradd`, default configuration files are copied from the `/etc/skel` directory to the new user's home. Several files are included in `/etc/skel` by default:

```
/etc/skel/.profile      # ksh posix local login script  
/etc/skel/.login        # csh local login script  
/etc/skel/.cshrc        # csh additional login script  
/etc/skel/.exrc         # vi startup configuration file
```

If you wish to change the default configuration files that are copied to new users' home directories, modify the files in `/etc/skel`. Note that changes made in `/etc/skel` won't affect existing users' home directories.

Additional files can be copied into `/etc/skel` as well, if your applications require configuration files in users' home directories.

4-6. LAB: Customizing User Accounts

Part I: Customizing Accounts via .profile

1. PART I

Modify the appropriate configuration file so that root will have access to command line editing at next login.

2. Modify the appropriate configuration file so that the system administrator's shell prompt displays the present working directory and user name after the next login.

3. If your system is running CDE, modify root's CDE login script to ensure that the system consults your `.profile` at next login.

4. Over the course of this week, you will run several scripts in the `/labs` directory. There should be a program in your `/labs` directory called `xroach`. What message do you get when you run `xroach` from your home directory by typing: `xroach`

5. Do whatever is necessary to ensure that you can run `xroach` and the other executables in `/labs` from any directory.

6. Log out, then log back in again to see if your changes were successful.

Part II: Customizing New Accounts via /etc/skel

1. PART II

New users added to your system may appreciate having access to the same functionality you configured for `root` in Part I of this lab. How can you ensure that all new user accounts have the same custom functionality you configured for `root` in Part I? Make it so.

2. `.profile` will be read only if your new users have a modified `.dtprofile`, too. How can you ensure that new home directories automatically get a copy of the modified `.dtprofile` that you created in Part I? Make it so.

3. Create a new user account using SAM or `useradd`; then log in as that new user and see if your customizations worked.

4. Do changes to `/etc/skel` affect already-existing accounts? Try logging in as `user24` to find out. Does `user24` have the custom prompt you configured in `/etc/skel/.profile`?

Module 5 — Guided Tour of the HP-UX File Hierarchy

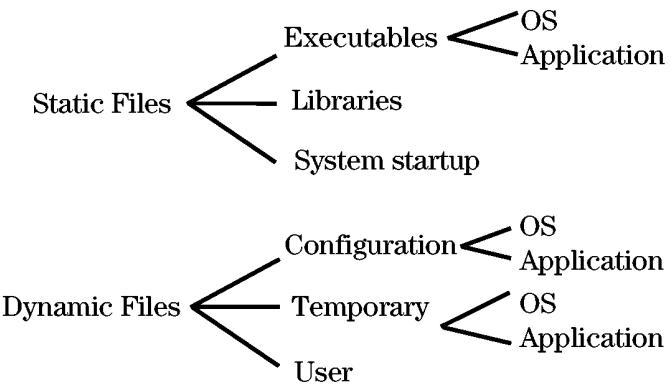
Objectives

Upon completion of this module, you will be able to do the following:

- Describe the reasons for separating dynamic and static file systems.
- Describe the key contents of `/sbin`, `/usr`, `/stand`, `/etc`, `/dev`, `/var` (OS-related directories).
- Describe the key contents of `/opt`, `/etc/opt`, and `/var/opt` (application-related directories).
- Use `find`, `whence`, and `whereis` to find files in the HP-UX file system.

5-1. SLIDE: Introducing the File System Paradigm

Introducing the File System Paradigm



a66917

Student Notes

Many HP-UX system administration tasks require the administrator to find and manipulate system and application configuration and log files. Understanding the philosophy behind the organization of the file system will ensure that you can successfully find the resources you need to perform administration tasks.

Files in the HP-UX file system are organized by various categories. **Static** files are separated from **dynamic** files. **Executable** files are separated from **configuration** files. This philosophy provides a logical structure for the file system and simplifies administration as well.

HP-UX Separates Static and Dynamic Portions of the File System

Files and directories in HP-UX may be categorized as **static** or **dynamic**. The contents of static files and directories rarely change, except when patching or installing the operating system or applications. Executable files, libraries, and system start-up utilities are all considered to be **static**.

Dynamic files and directories change frequently. They are stored in a separate portion of the file system. Configuration, temporary, and user files are all considered to be **dynamic**.

Separating dynamic and static data offers the following advantages:

- System backups are easier.
- Disk space management is simplified.

HP-UX Separates Executable Files from Configuration Files

Configuration data is kept separate from the executable code that uses that data. Separating executable files from configuration files offers the following advantages:

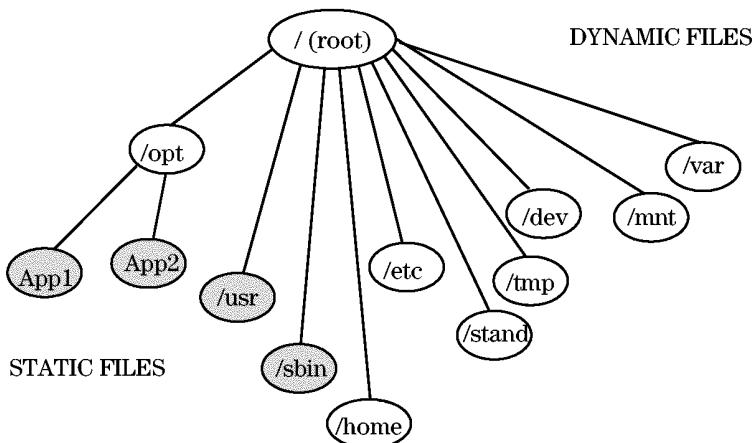
- Changes made to configuration data are not lost when updating the operating system.
- Executable files can be easily shared across the network, while host-specific configuration data is stored locally on each host.

HP-UX Follows the AT&T SVR4 Standard File System Layout

Though there are minor differences from vendor to vendor, the file system layout used in HP-UX is very similar to that used in other flavors of UNIX . This simplifies administration tasks for administrators with responsibilities on multiple vendors' machines.

5-2. SLIDE: The File System Layout

The File System Layout



a66918

Student Notes

The shaded directories in the diagram on the slide contain **static** data, while unshaded directories in the diagram contain **dynamic** data. The sharable portion of the operating system is located beneath `/usr` and `/sbin`. Only the operating system can install files into these directories. Applications are located beneath `/opt`.

The directories `/usr`, `/sbin`, and the application subdirectories below `/opt` can be shared among networked hosts. Therefore, they must not contain host-specific information. The host-specific information is located in directories in the Dynamic area of the file system.

General definitions for these directories are:

Directory	Definition
/usr	Sharable operating system commands, libraries, and documentation.
/sbin	Minimum commands needed to boot the system and mount other file systems.
/opt	Applications.
/etc	System configuration files. No longer contains executable files.
/dev	Device files.
/var	Dynamic information such as logs and spooler files (previously in /usr).
/mnt	Local mounts.
/tmp	Operating System temporary files.
/stand	Kernel and boot loader.
/home	User directories.

A Closer Look at /usr

The /usr directory contains the bulk of the operating system, including commands, libraries and documentation. The /usr file system contains operating system files, such as executable files and ASCII documentation.

The allowed subdirectories in /usr are defined below; no additional subdirectories should be created.

Examples of files that live here are

/usr/bin	Operating system user commands
/usr/conf	Kernel configuration
/usr/contrib	Contributed software
/usr/lbin	Back-ends to other commands
/usr/local	User-contributed software
/usr/newconfig	Default operating system configuration data files
/usr/sbin	System administration commands
/usr/share	Architecture independent sharable files
/usr/share/man	Operating system man pages
/usr/share/doc	White papers on technical topics

A Closer Look at /var

The `/var` directory is for multipurpose log, temporary, transient, variable sized, and spool files. The `/var` directory is extremely *variable* in size, hence the name. In general, any files that an application or command creates at runtime, and that are not critical to the operation of the system, should be placed in a directory that resides under `/var`. For example, `/var/adm` will contain log files and other runtime-created files related to system administration. `/var` will also contain variable size files like `crontabs`, and print and mail spooling areas.

In general, files beneath `/var` are somewhat temporary. System administrators that wish to free up disk space are likely to search the `/var` hierarchy for files that can be purged. Some sites may choose not to make automatic backups of the `/var` directories.

Examples of files that reside here are

<code>/var/adm</code>	Common administrative files and log files.
<code>/var/adm/crash</code>	Kernel crash dumps.
<code>/var/mail</code>	Incoming mail.
<code>/var/opt/</code>	Application-specific runtime files (e.g. logs, temporary files). Each application will have its own directory.
<code>/var/spool</code>	Spooled files used by subsystems such as <code>lp</code> , <code>cron</code> , software distributor.
<code>/var/tmp</code>	Temporary files generated by commands in the <code>/usr</code> hierarchy

A Closer Look at /var/adm

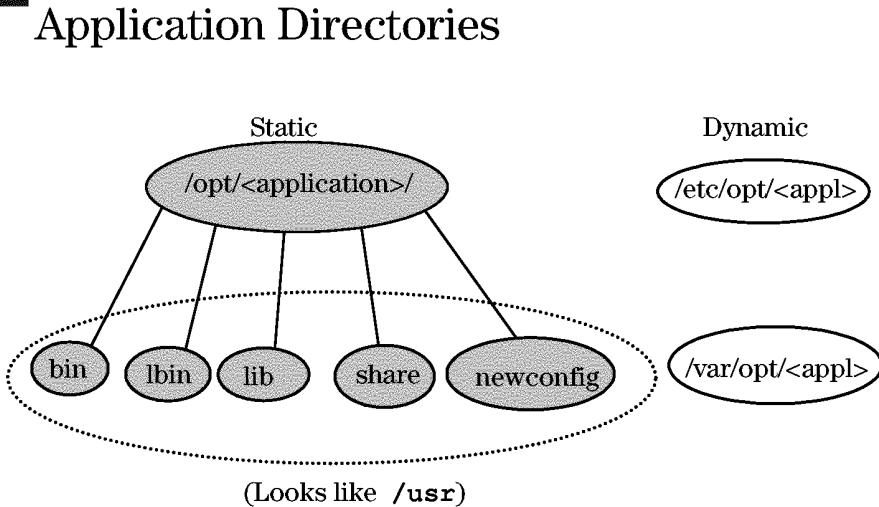
This directory hierarchy is used for common administrative files, logs, and databases. For example, files generated by `syslog(3C)`, files used by `cron(1M)`, and kernel crash dumps will be kept here and in subdirectories.

Examples of files that reside here are

<code>/var/adm/crash</code>	Kernel crash dumps will be located in this directory.
<code>/var/adm/cron</code>	Used for log files maintained by <code>cron</code> . <code>cron</code> is a subsystem that allows you to schedule processes to run at a specific time or at regular intervals.
<code>/var/adm/sw</code>	Used for log files maintained by the Software Distributor.
<code>/var/adm/syslog</code>	System log files. Applications as well as the kernel can log messages here. The <code>syslogd</code> daemon is responsible for writing the log messages. The behavior of the <code>syslogd</code> daemon can be customized with the <code>/etc/syslog.conf</code> file. The name of the default log file is <code>/var/adm/syslog/syslog.log</code> . At boot time this file is copied to <code>OLDsyslog.log</code> , and a new <code>syslog.log</code> is started. The <code>syslog.log</code> file is an ASCII file.

/var/adm/sulog	This file contains a history of all invocations of the switch user command. sulog is an ASCII log file.
/var/adm/wtmp	This file contains a history of successful logins. This file is not ASCII. The last command is used to display this information. The wtmp file will continue to grow and should be trimmed by the administrator from time to time.
/var/adm/btmp	This file contains a history of unsuccessful logins. This file is not ASCII. The lastb command is used to display this information. The btmp file will continue to grow and should be trimmed by the administrator from time to time.
/var/adm/utmp	This file contains a record of all users logged onto the system. This file is used by commands such as write and who . This file is not an ASCII file and can not be directly viewed.

5-3. SLIDE: Application Directories



a538250

Student Notes

Each application will have its own subdirectory under /opt, /etc/opt, and /var/opt. The sharable, or static, part of the application is self-contained in its own /opt/*application* directory, which has the same hierarchy as the operating system layout:

/opt/application/bin	User commands
/opt/application/share/man	man pages
/opt/application/lib	Libraries
/opt/application/lbin	Back end commands
/opt/application/newconfig	Master copies of configuration files

The application's host-specific log files are located under /var/opt/*application*, and host-specific configuration files are located under /etc/opt/*application*.

5-4. SLIDE: Commands to Help You Navigate

Commands to Help You Navigate

find	Searches the file hierarchy
whereis	Locates source, binaries, and man pages
which	Locates an executable in your PATH
file	Determines file type

a66919

Student Notes

As a system administrator, you will need to reference files in directories all over the HP-UX file system. HP-UX offers several tools for finding the files and executable files you need to perform administration tasks.

The **find** Command

The **find** command is a powerful tool for system administrators. It searches the file hierarchy starting at a specified point and finds files that match the criteria you select. You can search for files by name, owner, size, modification time, and so on. **find** also allows you to execute a command with the files found used as an argument.

Examples

- Find all files belonging to the user **greg**:

```
# find / -user greg
```

- Find files in /tmp that have not been accessed in 7 days:

```
# find /tmp -type f -atime +7
```

- Remove core files:

```
# find / -name core -exec rm {} \;
```

The whereis Command

The **whereis** command is useful when you receive "not found" error messages. It searches a predefined list of directories. By default, **whereis** looks for source, binaries, and **man** pages. You can limit the search to binary files by using the **-b** option.

Example

```
# whereis -b sam
sam: /usr/sbin/sam
```

The which Command

The **which** command is useful for determining which version of a command will be used. Some commands have multiple homes. Which version you execute is determined by the order of the directories in your PATH variable.

The file Command

The **file** command performs a series of tests on a file and attempts to classify it. It can be useful for determining if a command is a shell script or a binary executable.

Examples

```
# file /sbin/shutdown
/sbin/shutdown: 800 shared executable

# file /sbin/rc
/sbin/rc: ascii text
```

The strings Command

The **strings** command is useful when trying to find information in a binary file. It will print any printable characters in the file.

5-5. LAB: HP-UX File System Hierarchy

Directions

Answer all the questions below.

1. Which of the following directories are dynamic?

/etc

/usr

/sbin

/dev

/tmp

2. Viewing a report on your disk space usage, you note that /usr, /var, and /opt are all nearing 90% capacity. Which of these directories should you be most concerned about? Why?

3. Match the directory with its contents:

- | | |
|-------------------|--|
| 1. /usr/share/man | A. kernel, boot loader |
| 2. /stand | B. system configuration files |
| 3. /var/adm | C. shareable operating system commands |
| 4. /etc | D. man pages |
| 5. /usr | E. application directories |
| 6. /opt | F. common admin files and logs |

4. Where would you expect to find the cp and rm OS user executables? See if you are correct.

5. Where would you expect to find the `sam`, `useradd`, and `userdel` executables? See if you are correct.
6. The `pre_init_rc` utility executes in the early stages of the system start-up procedure to check for file system corruption. Where would you expect to find this executable? See if you are correct.
7. There is a system log file that maintains a record of system shutdowns. Where would you expect to find the shutdown log file? See if you are correct.
8. In which directory would you expect to find the "hosts" configuration file, which contains network hostnames and addresses? See if you are correct.
9. Though many utilities and daemons maintain independent log files, many daemons and services write their errors and other messages to a log file called `syslog.log`. See if you can find the path for this file, then check to see if any messages have been written to the file in the last day.
10. Use the `whereis` command to search for the `xclock` executable. The executable is actually under `/usr/bin/X11/xclock`. Did `whereis` find this executable? Explain.

11. Find all of the files (if any) under `/home` that are owned by root.

12. (Optional) Find all the files under `/tmp` that haven't been accessed within the last day.

13. (Optional) Find all the files on your system that are greater than 10000 bytes in size. If you needed to make some disk space available on your system, would it be safe to simply remove these large files?

Module 6 — Connecting Peripherals

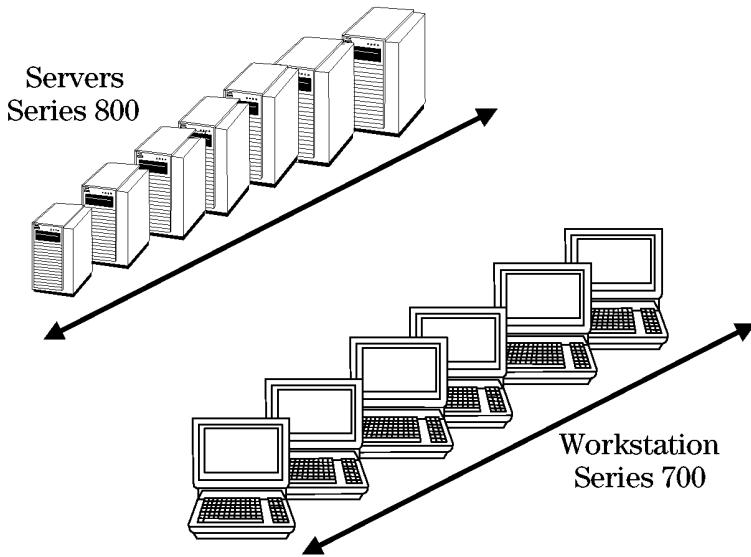
Objectives

Upon completion of this module, you will be able to do the following:

- Describe the difference between workstations and servers.
- Define the following terms: **bus**, **device adapter/interface**, **hardware path**.
- Compare and contrast the purpose of SCSI, serial, MUX, and parallel interfaces.
- View the current hardware configuration with **ioscan**.
- Use **ioscan** output to create a hardware diagram of the system.
- List the steps required to install a new interface card.
- List the steps required to connect a new SCSI, serial, or parallel device.

6-1. SLIDE: The HP 9000 Product Family

The HP 9000 Product Family



a538183

Student Notes

The HP 9000 product family encompasses both server and workstation platforms. Each platform has a wide range of models. Some of the differentiating factors between models are:

- number of CPUs
- processor speed
- I/O expandability
- graphics capabilities

For assistance in choosing the right model visit the HP Web Site at <http://www.hp.com>, or contact your local HP sales representative.

HP 9000 Servers

The HP 9000 server product line contains the broadest range of RISC systems in the UNIX market. The servers are designed to meet diverse business needs and to provide customers with a long-term growth path. All servers use the same HP-UX operating system and provide object-code compatibility across the entire product line.

HP 9000 Workstations

The HP 9000 workstation family includes a full range of workstations—from entry-level price/performance leaders, to mid-range workhorses, to high-end performance superstars and a complete family of X stations and terminals, as well as a full range of graphics systems.

HP 9000 workstations are built on open systems standards, and they are fully upward- and downward-compatible—within the workstation family as well as with HP's servers. That's because both the families are based on HP's powerful PA-RISC processor architecture and the UNIX-based HP-UX operating system.

Upgrade Paths

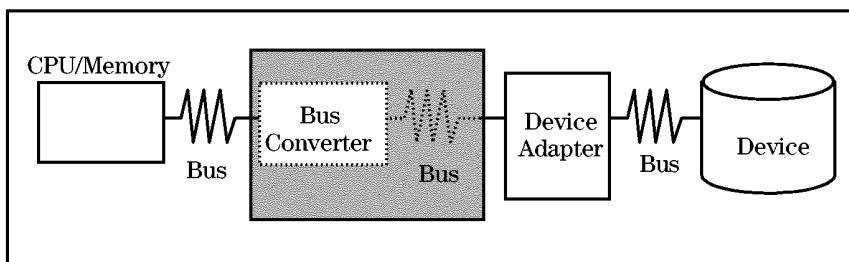
Customers often find that as their business grows, their transaction volumes demand greater capacity and performance. Hewlett-Packard provides a comprehensive upgrade program that protects customers' investments in hardware, software and training. The upgrade program includes simple board upgrades, system swaps with aggressive trade-in credits, and 100 percent return credit on most software upgrades.

PA-RISC

All HP 9000 systems use HP's Precision Architecture RISC (PA-RISC) technology to provide high performance and reliability. PA-RISC is built upon Reduced Instruction Set Computing (RISC) principles, a design approach that delivers greatly simplified computers that are optimized to provide the highest performance for a given integrated circuit technology. The inherent simplicity of PA-RISC implies that computer systems can be implemented with fewer components to achieve superior reliability when compared to older Complex Instruction Set Computer (CISC) systems.

6-2. SLIDE: I/O Architecture Terminology

I/O Architecture Terminology



a538184

Student Notes

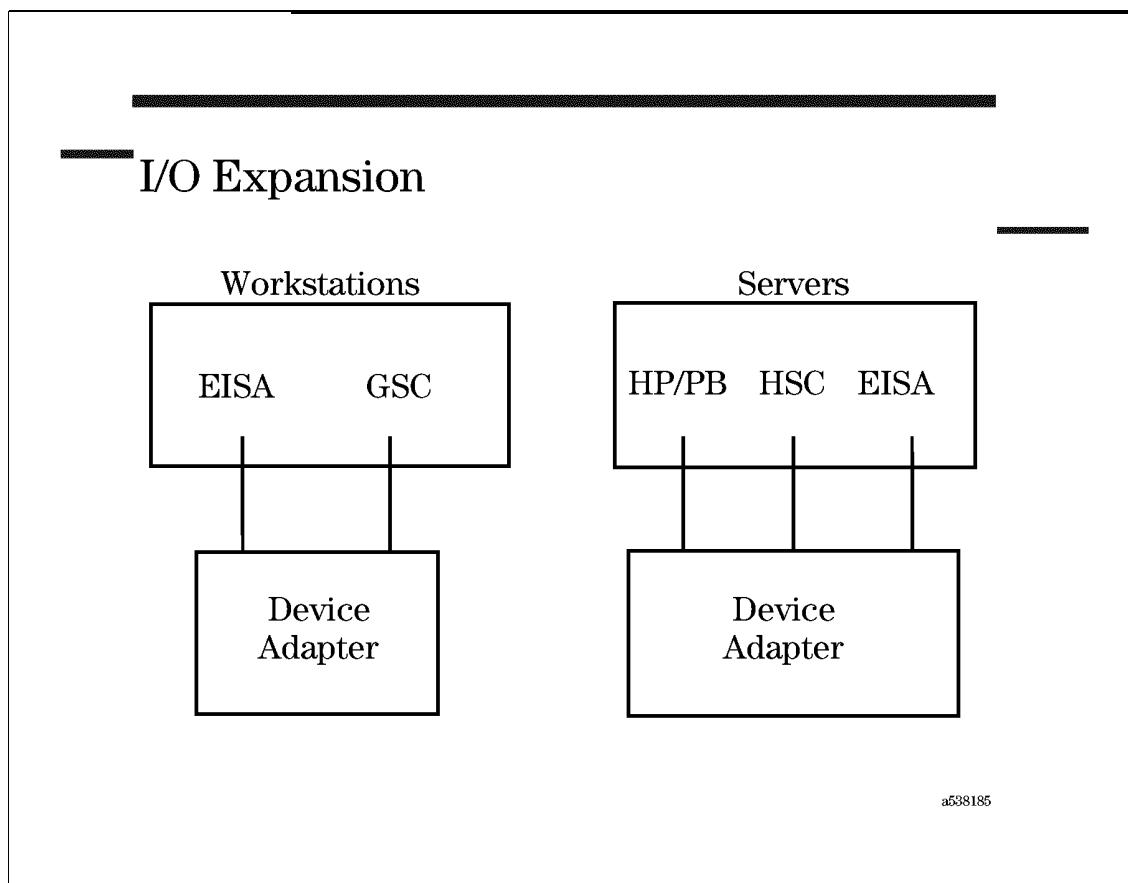
In order for data to move between a device and memory it must travel through several different types of hardware. Devices are connected to the system via device adapters, also known as interface cards. The processor and memory use buses to communicate with the device adapters. In many systems the bus that the processor and memory sit on is much faster than the device adapters can handle. In these systems a bus converter is used to step down to a bus that operates at a speed the device adapter can handle.

Definitions

Device Adapter Also known as an interface card or host adapter. This is an accessory card that is either integrated into the system or plugs into slots on the I/O system. Device adapters make the link that enables systems to communicate with external peripherals. Some device adapters accommodate multiple devices, others do not.

Bus	A bus is circuitry through which data travels between hardware modules (CPU, memory, device adapters) within the system. Some buses are internal buses, while others are external.
Internal Bus	The electronic path that connects the various areas of the SPU and allows data to flow throughout. Some models have several different internal buses joined by bus converters to facilitate efficient data flow between areas that operate at different speeds.
External Bus	A mechanism (like a cable) that can connect many like devices to one interface card. This is how devices like disks and tape drives are attached to the system.
Bus Converter	An internal interface between higher and lower speed buses. Some models have multiple levels of buses with varying speeds. Bus adapters are similar to bus converters, but is designed for non-PA-RISC architecture buses.

6-3. SLIDE: I/O Expansion



Student Notes

We have seen that devices connect to the system via device adapters. All systems come with an integrated multifunction I/O Card (**MFIOC**). The MFIOC (also known as Core I/O, or personality card) contains multiple device adapters. Examples of interfaces that may be included are:

- SCSI
- LAN
- Parallel Centronics
- FDDI
- Serial (RS-232)
- Audio

- MUX
- HIL (Keyboard/Mouse)

If you need to add more devices than can be accommodated by the MFIOC you must purchase expansion cards. The cards plug into slots. On the servers you can see these slots from the backplane. On the workstations you must open the cabinet. The HP 9000 series systems utilize several different types of I/O architectures. When ordering expansion cards you must choose the card for the correct type of I/O bus. The following are different types of I/O buses for which expansion cards can be ordered:

EISA	Extended Industry Standard Architecture. Until recently this type of I/O bus was available on workstations only. Some of the newest models of servers now support EISA. The number of EISA slots varies by model. When cards are plugged into the EISA slots they must be configured using the <code>eisa_config</code> utility.
GSC	General System Connect. This type of I/O bus is available only on the newer workstations (beginning with the J-Series). This is a very fast bus, well suited for the graphics cards available on the workstations.
HP-PB	Hewlett Packard Precision Bus. This type of I/O expansion is available only on the servers. The number of HP-PB expansion slots varies widely by model, from 2 to over 100.
HSC	High Speed System Connect. This type of I/O bus is available only on the newer servers (beginning with the K-series). As the name implies the HSC bus is faster than an HP-PB bus. There are a limited number of device adapters that can run on this bus. The HSC bus is the servers version of the GSC bus.
PCI	PCI is an industry standard bus architecture. The newer machines in both the workstation and server family support PCI. This includes the B-class, C-class, X-class, S-class families. PCI support is introduced in HP-UX release 10.20.

6-4. SLIDE: Device Adapters

Device Adapters

SCSI	Small Computer System Interface. There are three SCSI-2 interfaces available: Single-ended, Differential, and Fast and Wide. Supports SCSI disks, DDS tape drives, CD-ROM drives, MO drives, quarter inch cartridges (QIC), 8mm tapes and IBM 3480-compatible drives
MUX	Multiplexer. Supports serial terminals, printers, plotters, modems, and Access Port.
LAN/9000	Local Area Network. For IEEE 802.3 and Ethernet networks.
FDDI	Fiber Distributed Data Interface. High-speed local area network

a538186

Student Notes

I/O interfaces (or device adapters) connect external devices to the bus. This slide lists some commonly used I/O interfaces (or device adapters).

We will be taking a closer look at the different types of SCSI interfaces.

6-5. SLIDE: Types of SCSI

Types of SCSI

Other names bus is known by	'Standard SCSI' 'SCSI', 'SCSI-2' 'SE-SCSI'	'Fast' 'Fast/Narrow' 'Diff-SCSI'	'Differential-Wide' 'F/W-SCSI' 'Fast and Wide'
Compatible with the other SCSI bus types?	NO!	NO!	NO!
Bus Transfer Rate	5 MBytes/s	10 MBytes/s	20 MBytes/s
Data Bus Width	8 bits	8 bits	16 bits
# Connector Pins	50 pin	50 pin	68 pin
Max. cable length	6 meters	25 meters	25 meters
Maximum Devices on Interface	7	7	15

a538187

Student Notes

Fast/Wide SCSI is the fastest of the SCSI types. It is standard on some models and is also available as an expansion card. Fast/Wide cards can be purchased for EISA, GSC, HSC and HP-PB buses. Only Fast/Wide disks and disk arrays are supported on a F/W SCSI card.

Single Ended SCSI is slower than fast wide and supports fewer devices per interface card. It supports disks, tapes, and CD-ROM. Single-ended SCSI comes standard on all models. Expansion cards can be purchased for EISA and HP-PB.

Differential SCSI, also known as Fast/Narrow, is used for connecting Models 420SA and 1350SA disk arrays and digital linear tape drives.

SCSI Guidelines

- There are 7 SCSI device addresses (0-6) available for each Single-ended, and Differential SCSI card, and 15 devices addresses (0-6,8-15) available for each Fast and Wide SCSI card.
- On Single-ended and Differential SCSI address 6 is the highest priority and 0 is the lowest

- When Fast and Wide SCSI were introduced, the goal was to maintain that priority scheme. This makes for a somewhat confusing priority sequence with 15 devices. For Fast and Wide SCSI interfaces, the priority scheme is as follows:

A horizontal timeline diagram illustrating priority levels. It consists of a dashed line with numerical markers at 6, 0, 15, and 8. Below the line, the text '(highest priority)' is aligned with the value 6, and '(lowest priority)' is aligned with the value 8.

- All SCSI controller interfaces use bus address 7, so this cannot be used for a device.
 - Most SCSI devices use one SCSI address, but the Optical Disk Library System (HP C1700A) uses 3 addresses (two for the MO drives and one for the AutoChanger). The HP 5000 printer requires its own SCSI card, with one device per card.
 - Different SCSI devices (disks and tapes) can be on the same SCSI bus.
 - The last SCSI device on a SCSI bus must have a terminator. This terminator provides matching impedance on the bus circuits. Without a terminator, the devices on the bus will not work properly. There are two types of SCSI terminators: high-density terminators and low-density terminators. These terminators are electrically the same, but mechanically different. In other words, if the high-density terminator won't fit on your device, you are safe if you use a low-density terminator and vice versa.
 - Do not connect, disconnect, or power off any SCSI devices while the system is running. Doing so may corrupt data being transmitted on the SCSI bus.

CAUTION: HP does not support mixing Fast/Wide, Differential or Single Ended devices on the same bus.

6-6. SLIDE: SCSI Device Power Up Guidelines

SCSI Device Power Up Guidelines

- DO NOT
 - Connect or disconnect any device while the system is running
 - Turn power on or off to any device while the system is powered-up
- DO
 - Power on and complete self-tests on all peripherals before powering on SPU
 - Change bus addresses with device powered off

a538188

Student Notes

You will need to follow these SCSI device power up guidelines when changing the bus address on a SCSI device:

- *Do not* connect or disconnect any device while the system is running.
- *Do not* turn power on or off to any device while the system is powered-up.
- *Do* power on and complete self-tests on all peripherals before powering on SPU.
- *Do* change bus addresses only with device powered off.

6-7. SLIDE: Viewing the Configuration with **ioscan**

Viewing the Configuration with **ioscan**

```
# ioscan          # short listing of all devices
# ioscan -f       # full listing of all devices
# ioscan -fH 8/12.2.0 # full listing of device at 8/12.2.0
# ioscan -fC disk    # full listing of "disk" class devices

Class I H/W Path      Driver  S/W State   H/W Type  Description
=====
disk 0 8/12.2.0      sdisk   CLAIMED     DEVICE    SEAGATE
disk 1 8/12.3.0      sdisk   CLAIMED     DEVICE    SEAGATE
disk 2 8/12.4.0      sdisk   CLAIMED     DEVICE    SEAGATE
disk 3 8/12.5.0      sdisk   CLAIMED     DEVICE    SEAGATE
disk 4 8/12.6.0      sdisk   CLAIMED     DEVICE    SEAGATE
disk 5 8/16/5.3.0    sdisk   CLAIMED     DEVICE    Toshiba CD
```

a538190

Student Notes

During the system boot process, the kernel scans the system hardware to determine which devices are available. This scan should recognize any new interface cards or SCSI devices that have been connected to your system. You can check to see if your newly added device or card was recognized by executing the `/usr/sbin/ioscan` command. `ioscan` displays a list of all devices connected to the system.

When executed without any options or arguments, `ioscan` displays

- each device's hardware path
- each device's "class" (disk, tape, lan, etc.)
- a brief description of each device

Additional options on `ioscan` allow you to view even more information, as shown on the slide.

Useful Variations on **ioscan**

# ioscan	Scans hardware and lists all devices and other hardware devices found. Shows the hardware path, class, and a brief description of each component.
# ioscan -f	Scans and lists the system hardware as before, but displays a "full" listing including several additional columns of information. The "S/W State" column, in particular, is useful when trying to determine if the proper kernel drivers are installed.
# ioscan -fH 8/12.2.0	Shows a full listing for the device at the specified hardware address. This is useful on a large system if you just need to view information about a single device.
# ioscan -fC disk	Lists devices of the specified class only. Two other common classes are "tape" and "lan".
# ioscan -fn	Lists device file names associated with each device. Device files are discussed at length in the next chapter.

Fields in the **ioscan** Output

Interpreting **ioscan** output

Class	A device category, defined in the files located in the directory /usr/conf/master.d . Examples are disk, printer, and tape.
Instance	The instance number associated with the device or card. It is a unique number assigned to a card or device within a class. If no driver is available for the hardware component or an error occurs binding the driver, the kernel will not assign an instance number and a (-1), is listed.
H/W Path	A numerical string of hardware components, notated sequentially from the bus address to the device address. Typically, the initial number is appended by slash (/), to represent a bus converter (if required by your machine), and subsequent numbers are separated by periods (.). Each number represents the location of a hardware component on the path to the device.
Driver	The name of the driver that controls the hardware component. If no driver is available to control the hardware component, a question mark (?) is displayed in the output.
S/W state	The result of software binding. CLAIMED means the driver for this device was successfully bound to the device. UNCLAIMED means no driver was found in the kernel for this device. See the ioscan man page for further explanation.
Hardware Type	Entity identifier for the hardware component.
Description	Description of the device.

Troubleshooting with **ioscan -f**

After adding an interface card or SCSI device to your system, you should do an **ioscan** to see if your system recognizes the device.

First, simply check to see that your new device appears in the **ioscan** output. If not, shutdown your machine and check to ensure that all the cables are connected properly. In the case of an interface card, ensure that the card is firmly inserted in the interface card slot in the backplane of your machine.

Next, ensure that the hardware path is correct. Did you set the correct SCSI address? Add the device and its hardware path and description to the hardware diagram in your system log book.

Assuming your device has been connected with the proper SCSI address, check the "S/W State" column in the **ioscan -f** output. In order to communicate with your new device or interface card, your kernel must have the proper device drivers configured. If the proper driver already exists in your kernel, the "S/W State" column should say "CLAIMED". If this isn't the case, you will have to add the driver to the kernel via

```
SAM --> Kernel Configuration --> Drivers
```

If your new device appears to be "CLAIMED" by the kernel, proceed to the next chapter and learn how to create and use device files to access your new device.

Example: Complete **ioscan -f Output from a B132**

```
# ioscan -f
```

Class	I	H/W Path	Driver	S/W State	H/W Type	Description
bc	0		root	CLAIMED	BUS_NEXUS	
bc	1	8	bc	CLAIMED	BUS_NEXUS	Pseudo Bus Converter
ext_bus	0	8/12	c720	CLAIMED	INTERFACE	GSC Fast/Wide SCSI
target	0	8/12.2	tgt	CLAIMED	DEVICE	
disk	0	8/12.2.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
target	1	8/12.3	tgt	CLAIMED	DEVICE	
disk	1	8/12.3.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
target	2	8/12.4	tgt	CLAIMED	DEVICE	
disk	2	8/12.4.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
target	3	8/12.5	tgt	CLAIMED	DEVICE	
disk	3	8/12.5.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
target	4	8/12.6	tgt	CLAIMED	DEVICE	
disk	4	8/12.6.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
target	5	8/12.7	tgt	CLAIMED	DEVICE	
ctl	0	8/12.7.0	sctl	CLAIMED	DEVICE	Initiator
ba	1	8/16	bus_adapter	CLAIMED	BUS_NEXUS	Core I/O Adapter
ext_bus	2	8/16/0	CentIf	CLAIMED	INTERFACE	Built-in Parallel
audio	0	8/16/1	audio	CLAIMED	INTERFACE	Built-in Audio
tty	1	8/16/4	asio0	CLAIMED	INTERFACE	Built-in RS-232C
ext_bus	1	8/16/5	c720	CLAIMED	INTERFACE	Built-in SCSI
target	6	8/16/5.0	tgt	CLAIMED	DEVICE	
disk	6	8/16/5.3.0	sdisk	CLAIMED	DEVICE	Toshiba CD ROM
target	6	8/16/5.7	tgt	CLAIMED	DEVICE	
ctl	1	8/16/5.7.0	sctl	CLAIMED	DEVICE	Initiator

lan	0	8/16/6	lan2	CLAIMED	INTERFACE	Built-in LAN
ps2	0	8/16/7	ps2	CLAIMED	INTERFACE	Built-in Kbd/Mouse
ba	2	8/20	bus_adapter	CLAIMED	BUS_NEXUS	Core I/O Adapter
hil	0	8/20/1	hil	CLAIMED	INTERFACE	Built-in HIL
tty	2	8/20/2	asio0	CLAIMED	INTERFACE	Built-in RS-232C
ba	3	8/20/5	eisa	CLAIMED	BUS_NEXUS	EISA Bus Adapter
lan	1	8/20/5/1	vglan0	CLAIMED	INTERFACE	EISA card HWP1990
graphics	0	8/24	graph3	CLAIMED	INTERFACE	Graphics
processor	0	62	processor	CLAIMED	PROCESSOR	Processor
memory	0	63	memory	CLAIMED	MEMORY	Memory

6-8. LAB: Viewing the Configuration with **ioscan**

Directions

The **ioscan** command is a powerful tool for exploring your system's hardware configuration.

The sample **ioscan** output shown below was taken from a D-Class server. Use the sample **ioscan** output to answer the questions that follow.

H/W Path	Class	Description
<hr/>		
8	bc	I/O Adapter
8/0	graphics	Graphics
8/4	ext_bus	GSC add-on Fast/Wide SCSI Interface
8/4.5	target	
8/4.5.0	disk	SEAGATE ST32550W
8/4.6	target	
8/4.6.0	disk	SEAGATE ST32550W
8/4.7	target	
8/4.7.0	ctl	Initiator
8/16	ba	Core I/O Adapter
8/16/0	ext_bus	Built-in Parallel Interface
8/16/4	tty	Built-in RS-232C
8/16/5	ext_bus	Built-in SCSI
8/16/5.0	target	
8/16/5.0.0	tape	HP C1533A
8/16/5.2	target	
8/16/5.2.0	disk	TOSHIBA CD-ROM XM-4101TA
8/16/5.7	target	
8/16/5.7.0	ctl	Initiator
8/16/6	lan	Built-in LAN
8/16/7	ps2	Built-in Keyboard/Mouse
8/20	ba	Core I/O Adapter
8/20/2	tty	Built-in RS-232C
8/20/5	ba	EISA Bus Adapter
8/20/5/1	lan	EISA card HWP1990
8/20/5/2	unknown	EISA card
10	bc	I/O Adapter
32	processor	Processor
34	processor	Processor
49	memory	Memory

1. Which hardware path on this machine might be used for a serial (RS-232) modem?

2. Which hardware path on this machine might be used for a parallel printer?
3. How many LAN interfaces does this machine have? What are their hardware paths?
4. How many SCSI interfaces are available on this system? How many Fast/Wide SCSI interfaces are available?
5. How many disks are connected to the system shown in the ioscan above?
6. What are the hardware addresses of the SEAGATE disks? What are the SCSI addresses of the SEAGATE disks?
7. If you were to add another Fast/Wide SCSI disk to this system, what SCSI target address could you use for the new disk?
8. If you were to attach a new FW-SCSI disk at target address 4, what would be the resulting hardware path for the newly attached disk?

9. Is the tape drive on this system connected to the Fast/Wide SCSI chain, or the single-ended SCSI chain? What is the hardware address of the tape drive? What is the SCSI address of the tape drive?

Module 7 — Configuring Device Files

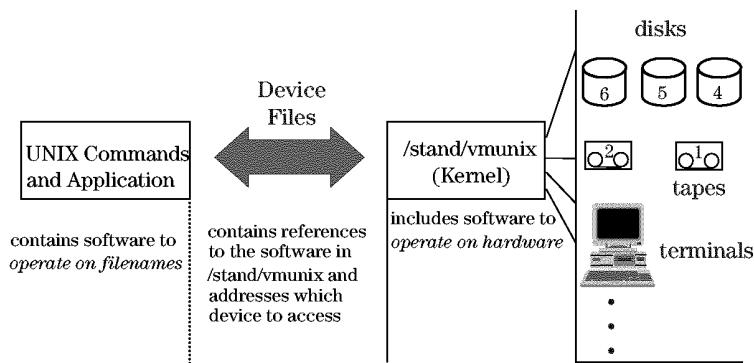
Objectives

Upon completion of this module, you will be able to do the following:

- Explain the purpose of a device file.
- Explain the significance of major and minor numbers.
- Differentiate between block and character i/o.
- Use **lsdev** to list kernal driver major numbers.
- Use **ll** to determine a device file's major and minor numbers.
- Use **ioscan** to list device files associated with a specific device.
- Use **lssf** to interpret the characteristics of a device file.
- Given a disk, tape, or CD device filename, determine the controller card instance number and target address of the associated device.
- Given a modem or terminal device filename, determine the controller card instance number and port number of the associated device.
- Describe the **autoconfig** process.
- Create device files using SAM.
- Create device files using **insf**.

7-1. SLIDE: What Is a Device File?

What Is A Device File?



a66920

Student Notes

HP-UX communicates with peripheral devices (such as tape drives, disk drives, printers, terminals, and modems) through files called **device files**. HP-UX treats I/O to a peripheral device in the same manner as I/O to a file. Before HP-UX can communicate with a peripheral device, the device must have a device file. For example, each terminal has its own device file through which HP-UX writes data (which appears on the terminal screen) and reads data (typed by the user at the keyboard).

A device file does not contain data, as a regular file does.

Instead, a device file simply specifies how HP-UX is to communicate with a device. Device files are stored in the **/dev** directory.

NOTE: Device file is synonymous with **special file**. You will see these terms used interchangeably.

Because almost all device files you will need are created for you by the HP-UX system, you need to know which device files to use when you access a peripheral device. Occasionally you will need to create device files. A device file should be removed if you permanently disconnect a peripheral device.

The following examples demonstrate the use of device files by HP-UX commands:

```
# tar -cvf /dev/rmt/0m /usr
```

The tar application opens the file specified by the **-f** option for output. It then writes the tar archive to that file. **tar** can write to an ordinary file or directly to a device. The **tar** application does not need to know the difference.

```
# echo hello > /dev/tty0p1
```

In this example the standard output of the echo command is redirected to a terminal via its device file.

7-2. SLIDE: Listing Device Files with ll

Listing Device Files with ll

Listing device files with ll /dev

```
brw-r----- 1 root sys 31 0x005000 Feb 10 1997 /dev/dsk/c0t5d0
brw-r----- 1 root sys 31 0x006000 Feb 10 1997 /dev/dsk/c0t6d0
crw-r----- 1 root sys 31 0x005000 Feb 10 1997 /dev/rdsk/c0t5d0
crw-r----- 1 root sys 31 0x006000 Feb 10 1997 /dev/rdsk/c0t6d0
crw--w---- 2 root tty 17 0x000001 Jan  9 09:25 /dev/ttyp1
crw--w---- 2 root tty 17 0x000002 Mar  6 17:46 /dev/ttyp2
```

↑
device file type ↑
major# ↑
minor# ↑
device file name

a66921

Student Notes

Device files typically reside in the /dev directory. You can list the device files in this directory with the ll command, just as you would list files in any other directory. Note, however, that the ll output for device files is a little different.

Device File Types

The very first character in the ll output for a device file indicates the device file type.

Character Device Files	A "c" in the first character position identifies a character device file. Character device files transfer data to the device one character at a time. Devices such as terminals, printers, plotters, modems, and tape drives are typically accessed via character device files. Character device files are sometimes called "raw" device files.
Block Device Files	A "b" in the first character position identifies a block device file. When accessing a device via a block device file, the system reads and writes data through a buffer in memory, rather than transferring the data directly to

the physical disk. This can significantly improve I/O for disks and CD-ROMs. Block device files are sometimes called "block" device files.

Terminals, modems, printers, plotters, and tape drives typically only have character device files. Disks and CD-ROMs may be accessed in either character or block mode, and thus typically have both types of device files.

Some applications and utilities prefer to access disks directly via a character device files. Other utilities require a block device file. Read the application or utility documentation to determine which device file is required.

Device File Major Numbers

Every device file has a "major number" that appears in the fifth field of the `ll` output. The major number identifies the "kernel driver" that should be used when the device is accessed. A kernel driver is a portion of code in the HP-UX kernel that controls I/O for a particular type of device. Most HP-UX machines have multiple drivers; the major number on every device file determines which driver should be used. The `lsdev` command lists the drivers configured in your kernel, and their associated major numbers.

Device File Minor Numbers

Every device file has an associated minor number. The "minor number" is a 24-bit hexadecimal number that identifies:

- The physical location of the device on the system.
- Device-specific access options. Tape drives, for instance, have special access options that enable/disable hardware compression and define the density format used when writing to the tape.

Minor numbers are formulated differently for various types of devices. See the "Configuring HP-UX for Peripherals" manual for more information about formulating and interpreting major and minor numbers.

Device File Names

Device file names follow a standard naming convention that makes it fairly easy to determine which device files are associated with which devices. Later slides in this chapter discuss the naming convention in detail.

7-3. SLIDE: Listing Device Files with **ioscan**

Listing Device Files with **ioscan**

```
# ioscan -fun          list all devices and device files
# ioscan -funC disk   list all disk devices and device files
# ioscan -funC tape   list all tape drives and device files

Class I H/W Path  Driver  S/W State   H/W Type   Description
-----
tape  1 2/0/1.1.0 stape  CLAIMED    DEVICE     HP C1553A
                                              /dev/rmt/lm      /dev/rmt/c0t1d0BESTb
                                              /dev/rmt/lmb     /dev/rmt/c0t1d0BESTn
                                              /dev/rmt/lmn     /dev/rmt/c0t1d0BESTnb
                                              /dev/rmt/lmnrb   /dev/rmt/c0t1d0BEST
```

a66922

Student Notes

Although the **ll** command lists the device files on your system, it does not indicate which device each device file accesses.

The **ioscan -fun** command provides a convenient mechanism for determining which device files are associated with each hardware path on your system. Under each hardware path, **ioscan -fun** lists all the device files associated with each hardware path. Since some devices have multiple access options, **ioscan** can list multiple device files for a single device.

Examples

# ioscan -fun	<i># list all devices, and their associated device files</i>
# ioscan -funC disk	<i># only list disk class devices and their device files</i>
# ioscan -funC tape	<i># only list tape drives and their device files</i>
# ioscan -funH 2/0/1.6.0	<i># only list device files for the device at 2/0/1.6.0</i>

7-4. SLIDE: Listing Device Files with lssf

Listing Device Files with **lssf**

- Lists characteristics of device files

Syntax

```
/usr/sbin/lssf path [path ...]
```

Examples

```
# lssf /dev/rdsk/clt6d0
disc3 card instance 1 SCSI target 6 SCSI LUN 0
section 0 at address 52.6.0 /dev/rdsk/clt6d0
# lssf /dev/rmt/0mn
tape2 card instance 1 SCSI target 0 SCSI LUN 0 at&t no
rewind best density available at address 52.0.0 /dev/rmt/0mn
```

a6974

Student Notes

Many devices have multiple associated device files. For instance, the tape drive shown on the previous slide had eight device files. This is because many devices may be accessed with a variety of combinations of access options. Each combination of access options requires a separate device file.

ioscan lists the device files for each device, but does not indicate which device-specific options each device file enables. The **lssf** command is the utility of choice for determining exactly what features each device file provides. **lssf** tells you:

- Which driver the device file uses
- The device's hardware address information
- Any device-specific access options used by the device file

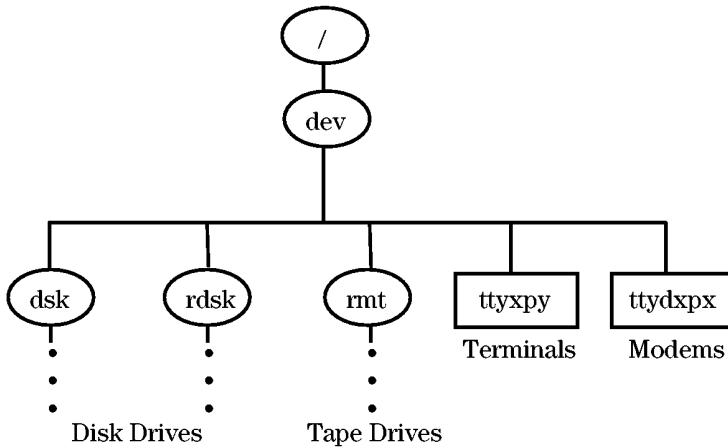
Questions

We have seen three commands for viewing device files: **11**, **lssf**, and **ioscan**. Each command has a somewhat different purpose. Determine which command would be appropriate for each of the following situations:

1. Which command would you use to list the device files for the tape drive at hardware address 2/0/1.0.0?
2. Which command(s) could you use to list all of your disk device files?
3. Which command would you use to determine the hardware path of the device accessed by **/dev/rmt/c0t0d0BEST**?
4. Which command will tell you the device-specific options used by the **/dev/rmt/c0t0d0BESTnb** device file?

7-5. SLIDE: The Layout of /dev

The Layout of /dev



a66924

Student Notes

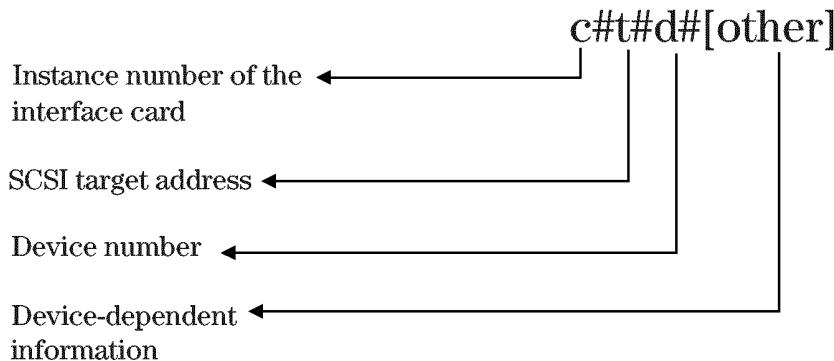
The next few slides will introduce the naming convention used by the system when assigning names to device files. An understanding of the device file naming convention will allow you to more easily choose and use device files on your system.

On most HP-UX systems, all device files live under the `/dev` directory. Some device files sit directly in `/dev`, while others are grouped in subdirectories under `/dev`. The list below highlights some of the more important device file directories.

<code>/dev</code>	Contains all terminal, modem, lan, and printer device files
<code>/dev/dsk</code>	Contains all block disk device files
<code>/dev/rdsk</code>	Contains all raw disk device files
<code>/dev/rmt</code>	Contains all tape device files

7-6. SLIDE: Device File Naming Convention

Device File Naming Convention



a66925

Student Notes

By default, HP-UX names your system device files following a standard naming convention. Although you can arbitrarily assign any device file names you wish, using this standard convention makes it easier to locate your device files. Most device file names adhere to the convention shown on the slide. The notes below explain each component of these device file names.

The Interface Card Instance Number

The kernel automatically assigns an **instance number** to every device and interface card on an HP-UX system. Instance numbers are shown in the "I" column of the `ioscan -f` output.

The "c" in a disk, tape, or CD ROM device file name identifies the instance number of the interface card to which the device is attached. All of the disks shown in the `ioscan` below would have device files beginning with "c0", since the instance number of the SCSI interface card is "0".

Note that each device has an instance number as well. Instance numbers for individual devices are used internally by the OS, but are not used to formulate device file names.

```
# ioscan -fun
```

Class	I	H/W Path	Driver	S/W State	H/W Type	Description
ext_bus	0	8/12	c720	CLAIMED	INTERFACE	GSC Fast/Wide SCSI
disk	0	8/12.2.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
disk	1	8/12.3.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
disk	2	8/12.4.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
disk	3	8/12.5.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
disk	4	8/12.6.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W

The SCSI Target Address

The "t#" portion of the device file name identifies the SCSI target address of the device file's associated device. The SCSI target address is set via jumper pins or DIP switches on the device itself. Look at the second to last digit in a SCSI device's hardware path to determine the SCSI target address. For instance, in the `ioscan` shown above, the disk at 8/12.3.0 has SCSI address "3". The SCSI target address for the disk at 8/12.6.0 is "6".

The SCSI Logical Unit Number

The **Logical Unit Number** (LUN) can be used to identify the robotic mechanism in a tape changer, or a logical unit in a disk array. For most SCSI devices, however, the LUN number will simply be "0". The LUN number for each SCSI device appears in the last digit of the device's hardware path. Note that all the disks in the sample `ioscan` above have a "0" in the LUN digit.

Device-Dependent Access Options

The last part of the device file name lists device-specific access options enabled by the device file. Tape drive device file names may have a variety of options listed in this portion of the device file name. Access options vary from device to device.

Questions

Consider the following `ioscan` output:

```
# ioscan -fun
```

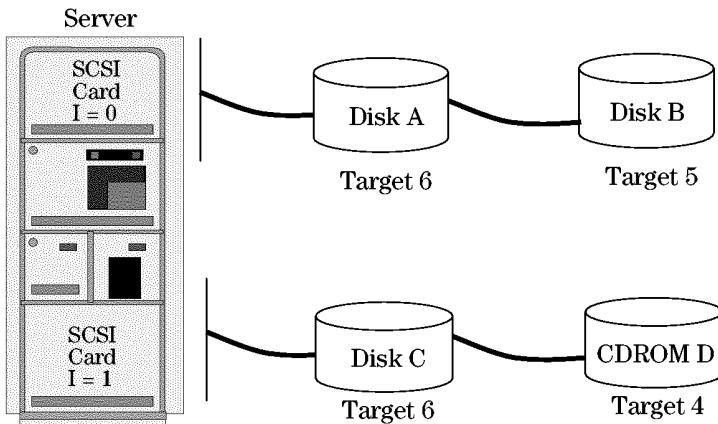
Class	I	H/W Path	Driver	S/W State	H/W Type	Description
ext_bus	0	8/12	c720	CLAIMED	INTERFACE	GSC Fast/Wide SCSI
disk	0	8/12.5.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
disk	1	8/12.6.0	sdisk	CLAIMED	DEVICE	SEAGATE ST32171W
ext_bus	1	8/16/5	c720	CLAIMED	INTERFACE	Built-in SCSI

disk 2 8/16/5.3.0 sdisk CLAIMED DEVICE SEAGATE ST32171W

1. What is the interface card instance number of the disk at 8/12.5.0?
2. What is the target address of the disk at 8/12.5.0?
3. What is the interface card instance number of the disk at 8/16/5.3.0?
4. What is the target address of the disk at 8/16/5.3.0?

7-7. SLIDE: Disk Device File Names

Disk Device File Names



`/dev/dsk/c#t#d#`
`/dev/rdsk/c#t#d#`

a66926

Student Notes

Every disk and CD-ROM has two device files:

- A block device file in `/dev/dsk`
- A character device file in `/dev/rdsk`

The device files within these directories follow the standard naming convention described on the previous slide.

Examples

The device files for disks "A" and "B" on the slide above would be:

<code>/dev/dsk/c0t6d0</code>	Block device file for disk "A"
<code>/dev/rdsk/c0t5d0</code>	Block device file for disk "B"

/dev/rdsk/c0t6d0 Raw device file for disk "A"

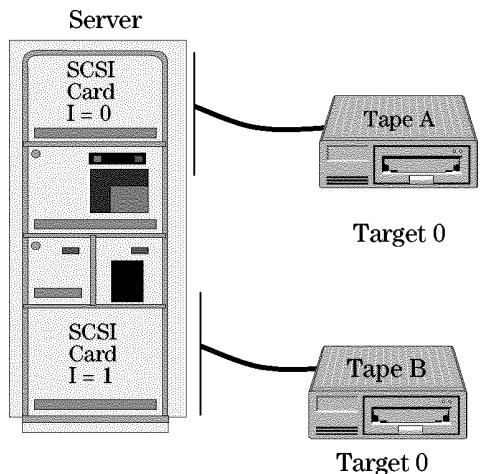
/dev/rdsk/c0t5d0 Raw device file for disk "B"

Questions

1. What would be the device file name for the disk labeled "C" on the slide?
2. What would be the device file name for the CD-ROM labeled "D" on the slide?

7-8. SLIDE: Tape Device File Names

Tape Device File Names



`/dev/rmt/c#t#d#[access options]`

a66927

Student Notes

Tape device files are stored in the `/dev/rmt` directory, and follow the `c#t#d#[options]` naming convention. Unlike disks and CD-ROMSs, however, tape drives often support numerous access options in the `[options]` portion of the device file name.

These options include:

density Density or format used in writing data to tape. This field is designated by the following values:

Table 7-1.

BEST	Highest-capacity density or format will be used, including data compression, if the device supports compression
NOMOD	Maintains the density used for data previously written to the tape. Behavior using this option is dependent on the type of device. This option is only supported on DDS and 8MM drives.
DDS <i>n</i>	Selects one of the known DDS formats; can be used to specify DDS1 or DDS2, as required.
D [<#]	Specifies density as a numeric value to be placed in the SCSI mode select block descriptor.

- C[#] Write data in compressed mode, on tape drives that support data compression. If a number is included, use it to specify a compression algorithm specific to the device. Note, compression is also provided when the density field is set to BEST.
- n No rewind on close. Unless this mode is requested, the tape is automatically rewound upon close.
- b Specifies Berkeley-style tape behavior. When the b is absent, the tape drive follows AT&T-style behavior. The details are described in "Tape Behavioral Characteristics" below.
- w Writes wait for physical completion of the operation before returning status. The default behavior (buffered mode or immediate reporting mode) requires the tape device to buffer the data and return immediately with successful status. For each tape device present, eight device files are automatically created when the system is initialized. Four of these device files utilize the standard naming conventions. These four files contain the density specification "BEST". There are four such files because each of the four different permutations of the "n" and "b" options.

Examples

The following are some device files for tape A:

- | | |
|----------------------|--|
| /dev/rmt/c0t0d0BEST | Use the best density and compression options available |
| /dev/rmt/c0t0d0BESTn | Same as above, but use the "no-rewind" feature |
| /dev/rmt/c0t0d0DDS1 | Use the DDS1 density for compatibility with older drives |

9.x Compatibility

Prior to version 10.x of HP-UX, device files followed an entirely different naming convention:

- | | |
|-------------|---------------------------------|
| /dev/rmt/0m | First tape drive on the system |
| /dev/rmt/1m | Second tape drive on the system |

`/dev/rmt/2m` Third tape drive on the system

/dev/rmt/2mn Third tape drive on the system, "no-rewind" feature enabled

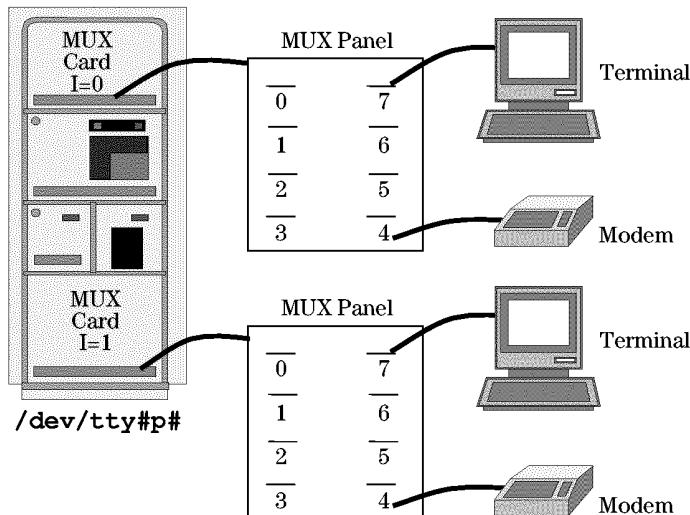
Each device file includes an arbitrary number to distinguish the tape drive from all other tape drives, followed by the letter "m", followed by a series of access options as described earlier in the notes for this slide. For the sake of backwards compatibility, these old style device files still exist. They are simply links to the `/dev/rmt/c0t0d0BEST[options]` device files.

Questions

1. What device file would you use to access tape "B" in the diagram on the slide using the best available density and compression features?
 2. What device file would you use to access tape "B" in the diagram on the slide using the best available density and compression features, and with the auto-rewind feature disabled?

7-9. SLIDE: Terminal and Modem Device File Names

Terminal and Modem Device File Names



a66928

StudentNotes

The naming convention for terminals and modems is slightly different from the naming convention for disks, CD-ROMs, and tape drives.

- Terminals and modems don't have a dedicated subdirectory; their device files are kept directly in the `/dev` directory.
- Terminals and modems don't follow the `c#t#d#` naming convention

Terminal Device Files

Terminal device file names identify both the interface card and MUX port number to which the terminal is attached. If the terminal is attached to a serial port rather than a MUX panel, use port#0 in the device file name.

Examples.

`/dev/tty0p7` Device file for the terminal on the first MUX, port #7

`/dev/ttys1p7`

Device file for the terminal on the second MUX, port #7

Modem Device Files

A fully functional modem requires three device files.

Examples

Device files for the modem on the first MUX, port #4

`/dev/cua0p4``/dev/cul0p4``/dev/ttysd0p4`

Device files for the modem on the second MUX, port #4

`/dev/cua1p4``/dev/cul1p4``/dev/ttysd1p4`

Pseudo Terminals

Pseudo terminals are used by applications that provide terminal emulation capabilities, such as `hpterm`, `xterm`, `telnet`, etc. The pseudo terminal driver provides support for a device-pair termed a pseudo terminal. A pseudo terminal is a pair of character devices, a master device and a slave device.

The device files for pseudo terminals are found in the following places:

slave `/dev/tty xx`. These are links to files in the `/dev/pty` directory
`/dev/pty/tty xx`

master `/dev/pty xx`. These are links to files in the `/dev/ptym` directory
`/dev/ptym/pty xx`

streams- based pseudo slave `/dev/pts/n`. This is used by the `dtterm` terminal emulator

streams- based master `/dev/ptymx`. This is used by the `dtterm` terminal emulator

By default, 60 pseudo terminals of each type are created. To increase this number you must reconfigure the kernel and run `insf` to create the device files.

7-10. SLIDE: How Device Files are Created

How Device Files are Created

- Autoconfiguration
- SAM
- `insf`
- `mksf`
- `mknod`

a59815

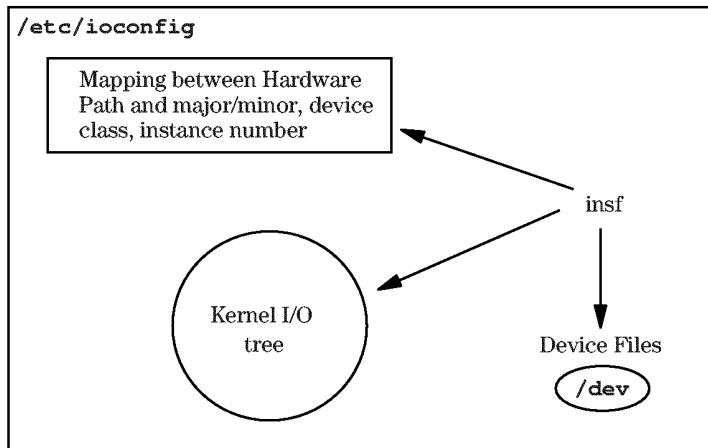
Student Notes

In most cases you do not need to create device files. When the HP-UX operating system is first installed, the `insf` command creates device files for all devices found by the system during its hardware probe. Then each time the system is rebooted, `insf` creates device files for any new devices that have been connected to the system. Hence, most device files you use will have a device file automatically created for them at boot time.

In some special cases you will need to manually create or modify device files. SAM is the recommended method for creating device files. Caution should be used when creating device files with the manual commands.

7-11. SLIDE: Autoconfiguration

Autoconfiguration



a59818

Student Notes

What Is Autoconfiguration?

At system boot, the kernel performs several system initialization tasks, including probing all hardware installed on the system. During the hardware probe, the kernel identifies all devices — buses, channel adapters, device adapters, and external devices — that can be autoconfigured. The kernel binds (matches) an appropriate driver to each device detected at a specific hardware address. This only happens for autoconfigurable devices.

After completing system initialization tasks, including hardware probing, the kernel invokes the `init` command. The `init` process reads the `/etc/inittab` file and invokes several system startup commands listed in the file, including `/sbin/ioinitrc`. The `/sbin/ioinitrc` command usually starts `ioinit`, which does several things.

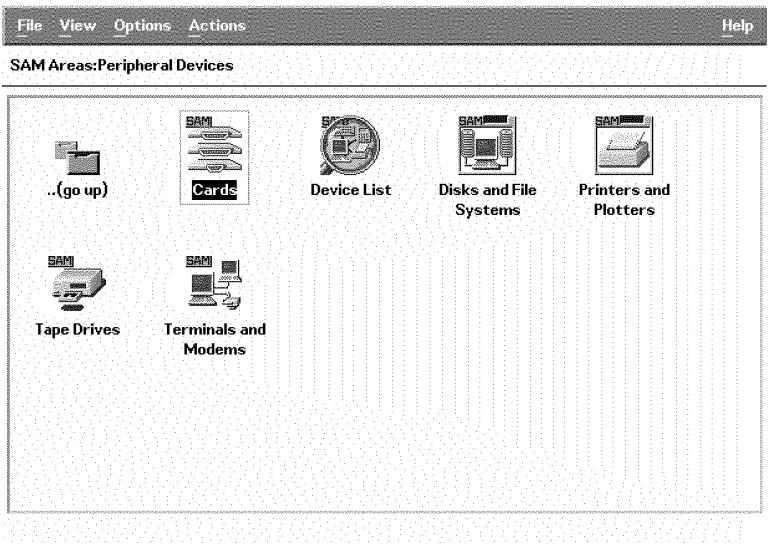
First, `ioinit` reads the contents of the `/etc/ioconfig` file and transfers the device mapping information found there to the kernel data structures, `io_tree`. Next, `ioinit` executes `insf`.

insf will create device files for new devices. It will also update **/etc/ioconfig** and the kernel tree.

All Hewlett-Packard peripheral devices supported by HP-UX Release 10.20 are automatically configurable. Device files are automatically created during the reboot process for devices or I/O cards.

7-12. SLIDE: Creating Device Files with SAM

Creating Device Files with SAM



a55816

Student Notes

In order to create device files in SAM choose Peripheral Devices from the SAM functional area launcher.

Choose the type of device you wish to create device files for. If you are creating additional device files for an existing device be sure to highlight the device first, before going to the Actions menu to add the device.

7-13. TEXT PAGE: Creating a Device File with `mksf`

The `mksf` command is used to create a device file *if the device is already known to the system.*

The syntax for the `mksf` command is shown on the slide. Options to `mksf` are:

<code>-d</code>	selects a particular device by its driver name
<code>-I</code>	selects <i>device instance</i>
<code>-C</code>	matches devices that belong to a given class, like <i>disks</i>
<code>-H</code>	matches a device at a given Hardware Path
<code>-D</code>	override the default device installation directory and install special file in <i>directory</i> . Note that <i>directory</i> must exist.
<i>other driver opts</i>	depend on the driver name
<i>special file</i>	selects an alternate device file name (default: naming conventions). The selected alternative name must be an absolute filename. Relative path names will be used to create files and subdirectories below the <code>/dev</code> directory.

Before creating a customized device file for an existing device, the card instance number can be found using the `ioscan` command.

Options for each driver vary widely. Options that are meaningful to one device driver are meaningless to another. When using this command, use the man pages for an explanation of options.

Examples:

```
# mksf -d tape2 -I 0 -b DDS1
```

creates a character device file `/dev/rmt/c0t0d0DDS1` for the tape drive at instance 0. This device file will write uncompressed format tapes. This is useful if you will be sending the tapes to another system that does not have a tape drive that supports hardware compression.

```
crw-rw-rw- 1 root other 212 0x000400 Feb 22 14:59 1hn
```

```
# mksf -d mux2 -I 0 -p 5 -c -i -a 2
```

creates a device file for a dial-in terminal with CCITT (European) protocol on port 5 of the first MUX. The device file created will be named `/dev/cua0p5`.

```
# mksf -C printer -I 2 /dev/printer
```

creates a device file named `/dev/printer` and maps it to the line printer with instance #2.

7-14. TEXT PAGE: Creating Device Files with `insf`

The `insf` command is used to create a device file *if the device has not been assigned yet*. It creates the device file and also obtains a card instance number for the device.

The syntax for the `insf` command is shown on the slide. Options to `insf` are:

- d selects particular devices by driver name
- C matches devices that belong to a given class, like disks
- H matches a device at a given Hardware Path
- I selects *card instance*
- e create/re-installs device files for existing devices
- D override the default device installation directory and install special file in *directory*. Note that *directory* must exist.

The -d, -H, and -C options are used to select devices with a specified driver, device class, or hardware path address. Use the `lsdev` command to determine drivers and classes in kernel (/stand/vmunix). Use the `ioscan` command to list the hardware paths in the kernel.

The `insf` command can not create device files for existing devices unless you explicitly say to recreate device files using the -e option. This might be required if the device files are accidentally deleted.

```
#insf -e -C printer
```

Device files can be made for all devices on your system. In addition, device files can be made for just one particular device type (driver name) or just an individual device within a device type.

You cannot specify special device options with `insf`. If you have some device that requires special options, you need to use `mksf` after running `insf`.

Examples

- To create device files for all devices of class `tty`, you would use:

```
insf -C tty
```

- To create device files for the device at address 4.2.0:

```
insf -H 4.2.0 -e
```

- To create a total of 100 pseudo ttys, you would use:

```
insf -e -n 100 -d ptym
```

7-15. LAB: Device Files

Directions

Part I: Viewing and Interpreting Device Files

1. PART I

Use **ioscan** to find the names of the device files for all of your disk class devices.

2. Use **ioscan** to find the names of the device files for your system's LAN card(s).
3. You should have a LAN card device file named **/dev/lan0**. Execute the command that lists the characteristics of this device file. What kernel driver is associated with this device file? What hardware path is associated with this device file?
4. How does **lssf** know which kernel driver is associated with each device file?
5. Choose one of the disk device files in your **/dev/dsk** directory. Execute the command that lists the characteristics of this device file. What kernel driver is associated with this device file? What is the hardware path of the disk associated with this device file?
6. (Optional) – If you have a tape drive on your system, run **lssf /dev/rmt/***. Which device file(s) access your tape drive with the no-rewind feature?

Part II: Creating Device Files with SAM

You should have at least two available serial ports on your machine. In this portion of the lab, you will create device files to support a modem on one of your serial ports, and a hard-wired ASCII terminal on the other. SAM provides a simple interface for creating device files for both of these devices.

1. PART II

Go to **SAM** --> **Peripheral Devices** --> **Terminals and Modems**. Are there any modems or terminals currently?

2. First, configure a modem on your first available serial port.

1. Choose Actions -> Add Modem.
2. Choose a serial port/MUX interface hardware path.
3. Click Port Number to choose an available MUX port (Choose 0 on workstations).
4. Choose any baud rate (Doesn't really matter since we don't have a real modem.).
5. Choose to support both incoming and outgoing calls.
6. Click OK.

What device files does SAM create for you?

3. Now configure a hard-wired terminal on another available serial port.

1. Choose Actions -> Add Terminal.
2. Choose a serial port/MUX hardware path.
3. Click Port Number to choose an available MUX port.
4. Choose "H" as the baud rate.
5. Click OK.

What device files does SAM create for you?

4. (Optional) If you have a tape drive on your machine, use SAM to create a non-standard device file for it. Specify a DDS1 density, no compression, Berkeley style semantics. Let the other options default.

This DDS1 device file may be useful if you create tapes that may need to be read on older tape drives that don't support the high density format used by "BEST" device files. What device file does SAM create?

Part III: Creating Device Files with `insf`

Though SAM must be used when you want to create serial-device device files or non-standard device files, most device files can be created automatically with `insf`. In this exercise you will have a chance to try using `insf`.

1. PART III

Remove the block device file for one of your disks.

2. Do an `ioscan -func disk` to ensure the device file is gone.

3. Recreate the device file with `insf -evC disk`. The `e` option ensures that the device file is recreated even if the disk already exists in `/etc/ioconfig`. `v` stands for verbose, and `C` disk just recreates disk class devices.

4. Check to see that the device file was recreated.

7-16. REVIEW: Check Your Understanding

Directions

Write the answers to the following questions.

1. What is a device file?
2. What is the difference between a block and a character device file?
3. Why do you need both block and character device files for disks?
4. What are major and minor numbers?
5. What do the `insf` and `mksf` commands do?
6. What is the difference between the `11` and the `1ssf` commands related to device files?
7. What does the `lsdev` command do?

Module 8 — Configuring Disk Devices

Objectives

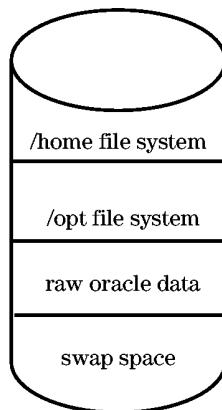
Upon completion of this module, you will be able to do the following:

- Describe the reasons for disk partitioning.
- Partition a disk using the whole disk layout approach.
- Describe the features and benefits of LVM.
- Define the terms **volume group**, **logical volume**, and **physical volume**, and explain how they relate to each other.
- Create physical volumes, volume groups and logical volumes from the command line.

8-1. SLIDE: Disk Partitioning

Disk Partitioning

- Each HP-UX disk can have one or more partitions.
- Each partition can be used for
 - a file system
 - swap space
 - raw data
 - boot area



a66929

Student Notes

Disk space is organized into **partitions**. A partition is nothing more than a portion of disk space allocated for a particular purpose. A partition may span one disk, multiple disks, or a portion of a disk. Each partition may contain one of the following:

- A file system (space allocated for files and directories).
- A swap area (space used by the kernel to supplement physical memory).
- Raw data (data accessed directly by an application, such as a database).
- A boot area (space containing utilities used during the boot process).

HP-UX offers two approaches for creating and managing disk partitions:

- The whole disk approach
- Logical Volume Manager (LVM)

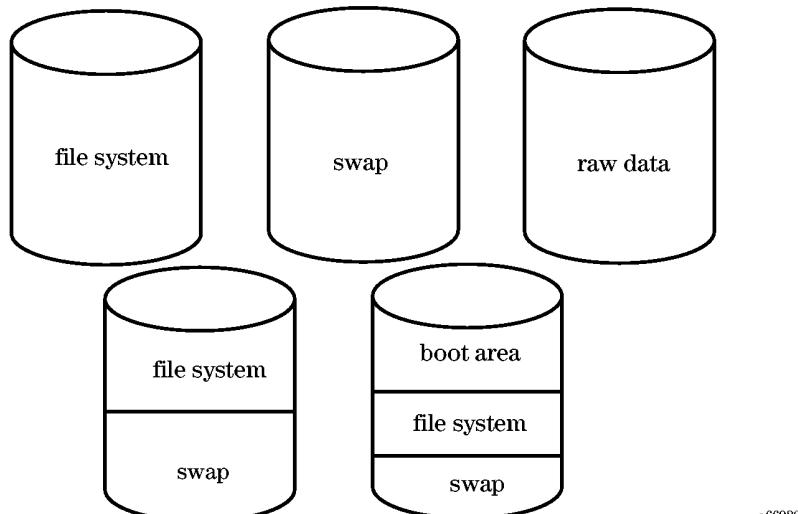
Some of the disks on your system may be configured using the whole disk layout approach, while others may be configured using LVM. Both techniques may be used concurrently on the same system, but not on the same disk.

Both approaches have advantages and disadvantages. Though the whole disk layout approach is easier to configure, LVM offers more flexibility. This chapter will discuss both disk partitioning techniques.

8-2. SLIDE: Whole Disk Partitioning

Whole-Disk Partitioning

There are five ways to partition a disk using the whole disk approach.



Student Notes

Using the whole disk approach, a disk may be configured five different ways:

- The disk can be dedicated entirely for use by a single file system.
- The disk can be dedicated entirely for use as swap.
- The disk can be dedicated entirely for use as a raw partition.
- A portion of the disk can contain a file system, with any remaining space used as swap.
- A disk can be configured as a boot disk, containing the root file system, a swap area, and a 2 MB special boot area containing utilities used during the boot process.

SAM is the easiest way to configure a disk using the whole disk approach:

SAM --> Disks and File Systems

```
--> Disk Devices  
--> Actions --> Add a Disk --> Not Using LVM
```

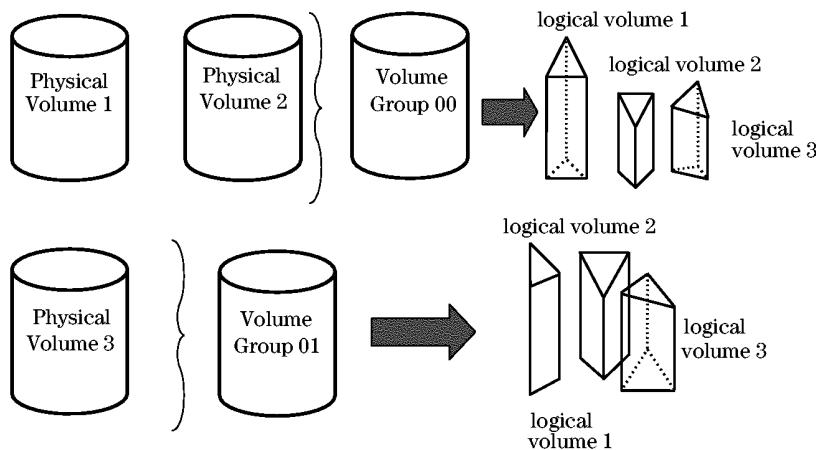
Though the whole disk approach is easy to use, it has several limitations:

- A file system cannot span multiple disks.
- There can only be one file system partition per disk.
- It is difficult to extend a file system if more space is needed.

For these reasons, many administrators choose to use the Logical Volume Manager to manage disk space instead of the whole disk approach.

8-3. SLIDE: LVM Disk Partitioning

LVM Disk Partitioning



a66931

Student Notes

Logical Volume Manager (LVM) enables you to pool space from several disks (known as "Physical Volumes" in LVM) to form a "Volume Group". You can then subdivide the space in the volume group into "Logical Volumes" (the LVM equivalent of a partition). The Logical Volume Manager (LVM) overcomes the limitations of the whole disk layout scheme by making it possible to

- Create logical volumes that span multiple disks
- Create multiple logical volumes on a single disk
- Extend and reduce logical volumes as necessary

Physical Volumes

A disk managed by LVM is known as a physical volume. Several special data structures must be created on a disk before it can be used by LVM. Once these data structures have been created, the disk is considered to be a physical volume, and may be added to a volume group.

Volume Groups

A volume group is a group of one or more physical volumes. The physical volumes in a volume group form a pool of disk space which may be allocated to one or more logical volumes. Volume groups usually follow the naming convention:

- `/dev/vg00`
- `/dev/vg01`
- `/dev/vg02` etc.

However, you may use any naming convention you wish. `vg00` is a special volume group known as the "root volume group" which typically contains the default boot disk and the majority of the HP-UX operating system. You may have other volume groups on your system for applications, and other user and application data.

Logical Volumes

Disk space from a volume group may be allocated to one or more logical volumes. A logical volume is analogous to a partition, and may contain a file system, swap area, or raw partition.

Logical volumes can

- Encompass all, or any portion of, the space on a physical volume.
- Span multiple LVM physical volumes.
- Be resized, or even moved to a different disk if the need arises.

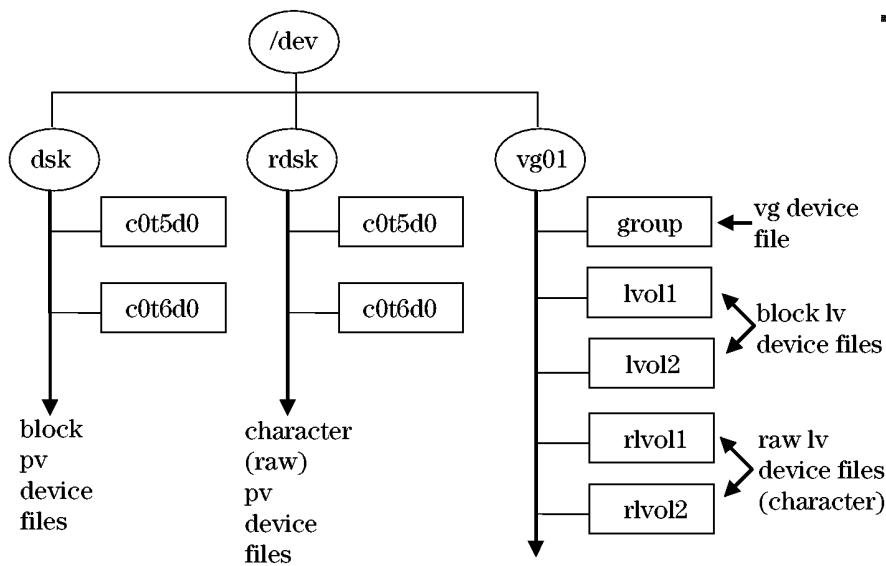
By default, logical volumes within a volume group are assigned names as follows:

- `/dev/vg01/lvol1`
- `/dev/vg01/lvol2`
- `/dev/vg01/lvol3` etc.

However, you may use any naming convention you wish.

8-4. SLIDE: LVM Device Files

LVM Device Files



a66932

Student Notes

Physical volumes, volume groups, and logical volumes are all referenced via device files, much as disk devices are referenced via device files.

Physical Volume Device Files

You can reference a physical volume using the physical volume's associated disk device files. Since disks may be referenced in either block or character mode, each physical volume has both a block and a character device file.

Examples

```
/dev/dsk/c0t5d0
/dev/rdsk/c0t5d0
```

```
# block device file for the disk at SCSI address 5
# raw device file for the disk at SCSI address 5
```

Volume Group Device Files

Volume groups are referenced via device files, too. Each volume group has a subdirectory under `/dev` containing a *group* device file for the volume group itself, as well as the device files for all of the logical volumes within the volume group. The name of the volume group's subdirectory determines the volume group's name.

Examples

```
/dev/vg01                                # directory containing device files associated with vg01
/dev/vg01/group                            # device file for vg01 volume group
```

Logical Volume Device Files

Logical volume device files are stored in the directory of the volume group to which they belong. Each logical volume has two device files: one is used when accessing the logical volume in character mode, the other is used when accessing the logical volume in block mode.

Examples

```
/dev/vg01/lvol1                           # block device file for logical volume "lvol1" in vg01
/dev/vg01/r1lvol1                          # raw device file for logical volume "lvol1" in vg01
```

LVM Major and Minor Numbers

Like all other device files, every logical volume and volume group device file must have both a major and a minor number.

All LVM device files have major number 64, the major number associated with the LVM kernel driver.

The first two digits of the minor number identify which volume group the device file is associated with. The last two digits identify the logical volume associated with the device file.

Name: `/dev/vg01/lvol2`

Major #: 64

Minor #: 0x010002

Example

- The major number for this device file, like all LVM device files, is 64.
- The first two digits in the minor number (01) indicate that this is in `vg01`.
- The last two digits in the minor number indicate that this is `lvol2`.

Questions

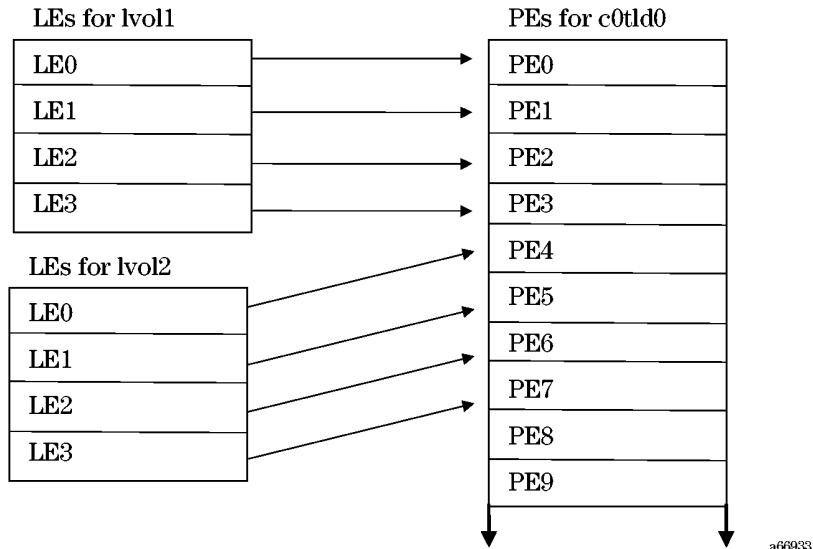
If `vg02` has three logical volumes created using the default naming convention

1. What directory would contain the logical volumes' device files?

2. What would be the name of the volume group's device file?
3. What would be the name of the first logical volume's raw device file?
4. Overall, how many device files should you find in /dev/vg02?
5. What would be the minor number of the third logical volume's device file?

8-5. SLIDE: LVM Extents

LVM Extents



Student Notes

We need to consider one more concept before looking at the practical side of configuring LVM.

The smallest allocatable unit of space in LVM is known as an "Extent." A physical volume is broken into "Physical Extents" (PEs), which are made available for allocation when the physical volume is added to a volume group.

A logical volume consists of a series of sequentially numbered "Logical Extents" (LEs). Each logical extent is nothing more than a pointer to a physical extent on disk. Larger logical volumes have more logical extents, and smaller logical volumes have fewer logical extents. In order to make a logical volume larger, LVM needs only allocate some additional extents.

The PE and LE sizes are consistent throughout a volume group, and may be set when the volume group is initially created. The default extent size is 4 MB.

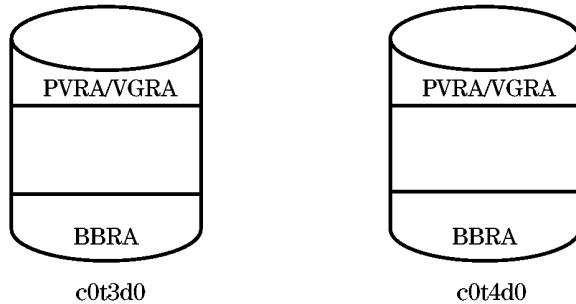
The volume group shown on the slide, vg01, has two 16 MB logical volumes. Since the default extent size is 4 MB, each logical volume has four logical extents. Each logical extent is a pointer to a 4 MB physical extent on the disk.

Questions

1. Assuming a volume group uses the default extent size, how many logical extents would you expect to find in an 88 MB logical volume?
2. Assuming a volume group uses the default extent size, how many physical extents would you expect to find on a 400 MB physical volume?

8-6. SLIDE: Creating Physical Volumes

Creating Physical Volumes



```
pvcreate /dev/rdsk/c0t3d0  
pvcreate /dev/rdsk/c0t4d0
```

a66935

Student Notes

Before you can use the space on a disk for logical volumes, you must designate the disk as an LVM physical volume. Once the disk has been configured as a physical volume, you can add the disk to a volume group and begin allocating space from the disk to logical volumes.

Preparing a Physical Volume

The disk you want to use as a physical volume must be connected to your system and powered on. Use the `ioscan` command to find the device file name for the new disk.

```
# ioscan -func disk
```

You may wish to run the `mediainit` utility if your disk has been used previously, or if you don't trust the integrity of the disk. `mediainit` initializes a disk by formatting the media, writing and reading test patterns to verify media integrity, then sparing any defective blocks found. `mediainit` destroys all existing user data on the disk:

```
# mediainit /dev/rdsk/c0t3d0
# mediainit /dev/rdsk/c0t4d0
```

Next, execute the **pvcreate** command to create the LVM data structures needed by LVM to begin using the disk as a physical volume. If the disk was previously part of another volume group, you may need to use the **-f** option on **pvcreate**. To overwrite any existing LVM or file system structures on the disk:

```
# pvcreate -f /dev/rdsk/c0t3d0
# pvcreate -f /dev/rdsk/c0t4d0
```

Your disk is now ready to be added to a new or existing volume group.

LVM Data Structures

LVM stores information in data structures at the beginning of the Physical Volume.

- **Physical Volume Reserved Area (PVRA)** contains LVM information specific to that Physical Volume. It is created by the **pvcreate(1M)**.
- **Volume Group Reserved Area (VGRA)** contains LVM information specific to the entire Volume Group. A carbon copy of the VGRA is found on each Physical Volume in the Volume Group. Within the VGRA is the Volume Group Status Area (VGSA) which contains quorum information for the Volume Group, and the Volume Group Descriptor Area (VGDA) which contains information the device driver needs to configure the volume group for LVM. The VGRA is created by **vgcreate(1M)**.
- User Data Area contains file systems, virtual memory (swap), or user applications. When a volume group is created, the user data area is divided into fixed-size physical extents, which map to logical extents. The map of Logical Extents is contained in the VGRA.
- **Bad Block Relocation Area (BBRA)**, contains information specific to the bad block recovery mechanism.
- LVM boot disks contain additional data structures required by the boot process.

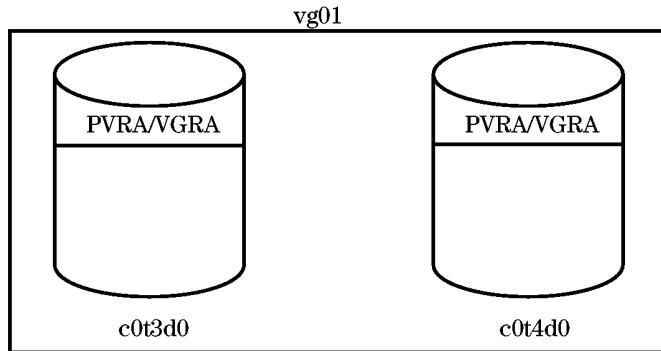
LVM Overhead

The data structures that are used by LVM consume some overhead from the disk space. This overhead is set at a fixed boundary for bootable LVM disks (2912 Kb), and may vary in size for non-bootable LVM disks (typically 400 Kb).

Overhead required on non-bootable disks depends on the parameters used on the **vgcreate(1M)** command or used in SAM. If you set a small extent size or create many physical volumes, your LVM data structures will be larger.

8-7. SLIDE: Creating Volume Groups

Creating Volume Groups



```
Create: mkdir /dev/vg01  
        mknod /dev/vg01/group c 64 0x010000  
        vgcreate vg01 /dev/dsk/c0t3d0 /dev/dsk/c0t4d0  
Check: vgdisplay -v vg01  
       pvdisplay -v /dev/dsk/c0t3d0  
       pvdisplay -v /dev/dsk/c0t4d0
```

a66936

Student Notes

After the disks have been designated as LVM physical volumes you can create volume groups. Once the volume groups are created, you can then create the logical volumes in them.

Step 1

Create a directory for the volume group. The naming convention is `/dev/vg nn` , where nn is the volume group number. Use the next number in sequence on your system. If your system is pre-installed, you already have a volume group `/dev/vg00`.

For example, to create volume group 01, which would be the second volume group on your system, you would type:

```
mkdir /dev/vg01
```

Step 2

The group special file, or control file, provides the means by which LVM kernel and LVM commands communicate within the volume group you create.

Create the control file named **group** in the directory **/dev/vg01**. Use the **mknod(1M)** command. The **group** file is a *character* device file. The major number is always 64. The minor number is hexadecimal, always ends in 0000, and has the following form:

```
0xhh0000
```

where *hh* is the hexadecimal representation of the volume group number.

For example, to create a **group** file for a volume group 01, you would type:

```
mknod /dev/vg01/group c 64 0x010000
```

Step 3

Now you can create the volume group and specify the physical volumes it will contain. You use the **vgcreate(1M)** command. You can assign several volumes to a group at one time.

```
vgcreate /dev/vg01 /dev/dsk/c0t3d0 /dev/dsk/c0t4d0
```

Note that you are using the *block* device file to create the volume group.

The options on **vgcreate** include

-e max_physical_extents	Sets the maximum number of physical extents per physical volume in a volume group (default is 1016).
-l max_logical_vols	Sets the maximum number of logical volumes allowed in a volume group (default 255).
-p max_physical_vols	Sets the maximum number of LVM disks (physical volumes) allowed in a volume group (default 16).
-s physical_extent_size	Sets the size, in megabytes, for each physical extent in a volume group (default 4).

Step 4

You can verify that you have created the volume group using the **vgdisplay** and **pvdisplay** commands:

```
# vgdisplay -v vg01
--- Volume groups ---
VG Name           /dev/vg01
VG Write Access  read/write
VG Status         available
Max LV           255
Open LV          0
Max PV           16
Cur PV           2
Act PV           2
Max PE per PV   1016
VGDA             4
PE Size (Mbytes) 4
Total PE         500
Alloc PE         0
Free PE          500
Total PVG        0
Total Spare PVs 0
Total Spare PVs in use 0

--- Physical volumes ---
PV Name           /dev/dsk/c0t3d0
PV Status         available
Total PE          250
Free PE           250

PV Name           /dev/dsk/c0t4d0
PV Status         available
Total PE          250
Free PE           250

# pvdisplay -v /dev/dsk/c0t3d0
--- Physical volumes ---
PV Name           /dev/dsk/c0t3d0
VG Name           /dev/vg01
PV Status         available
Allocatable       yes
VGDA             2
Cur LV           0
PE Size (Mbytes) 4

Total PE          250
Free PE           250
Allocated PE      0
Stale PE          0
IO Timeout        default

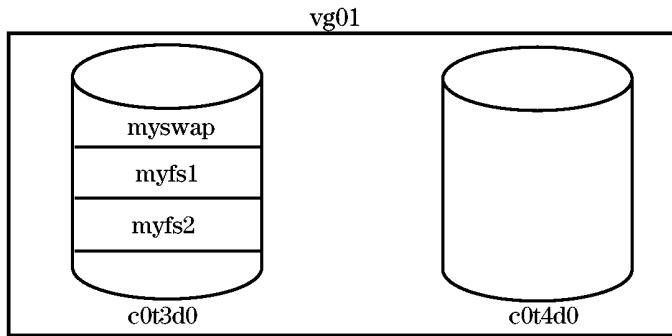
# pvdisplay /dev/dsk/c0t4d0
--- Physical volumes ---
PV Name           /dev/dsk/c0t4d0
VG Name           /dev/vg01
PV Status         available
Allocatable       yes
```

Configuring Disk Devices

VGDA	2
Cur LV	0
PE Size (Mbytes)	4
Total PE	250
Free PE	250
Allocated PE	0
Stale PE	0
IO Timeout	default

8-8. SLIDE: Creating Logical Volumes

Creating Logical Volumes



```
Create: lvcreate -L 16 -n myswap vg01
        lvcreate -L 16 -n myfs1 vg01
        lvcreate -L 16 -n myfs2 vg01
Check: vgdisplay -v vg01
       lvdiskdisplay -v /dev/vg01/myswap
```

a66937

Student Notes

Create logical volumes in volume groups using the `lvcreate(1M)` command.

You can allocate disk space for file systems, swap, or raw data in either megabytes or LVM extents.

When `lvcreate` creates the logical volume, it creates the block and character device files and places them in the directory `/dev/vgn`. It is created without size unless you specify a size when you use the `lvcreate` command. You can create a logical volume using the default characteristics, and change them later.

Options

`-L logical_volume_size`

The size of the logical volume in megabytes. The size specified will be rounded up to the nearest whole logical extent. The default is zero.

-l logical_extents_number	The number of logical extents in the logical volume. The default is zero.
-n name	A custom name you want to assign to the logical volume. The default name follows the naming convention.

Examples

To create a logical volume with the default characteristics, in the volume group /dev/vg01:

```
lvcreate /dev/vg01
```

Only the name will be reserved; neither physical nor logical extents will be reserved. You will need later to extend the logical volume with the command `lvextend(1M)`.

To create a logical volume of 10 logical extents in size:

```
lvcreate -l 10 /dev/vg01
```

To create a logical volume with a size of 100 Mbyte:

```
lvcreate -L 100 /dev/vg01
```

To create a logical volume with a non-standard name.

```
lvcreate -L 16 -n myswap vg01
```

Viewing Your Logical Volumes

You can use two commands to view information about your logical volumes:

```
# vgdisplay -v vg01          # to determine which LVs are in vg01  
# lvdisplay -v /dev/vg01/myswap # view details about the "myswap" LV
```

Full output from the `lvdisplay` command is shown below:

```
# lvdisplay -v /dev/vg01/myswap
--- Logical volumes ---
LV Name                  /dev/vg01/myswap
VG Name                  /dev/vg01
LV Permission             read/write
LV Status                available/syncd
Mirror copies            0
Consistency Recovery     MWC
Schedule                 parallel
LV Size (Mbytes)         16
Current LE               4
Allocated PE              4
Stripes                  0
Stripe Size (Kbytes)    0
Bad block                on
Allocation               strict
IO Timeout (Seconds)    default

--- Distribution of logical volume ---
PV Name          LE on PV  PE on PV
/dev/dsk/c0t3d0      4        4

--- Logical extents ---
LE   PV1           PE1  Status 1
0000 /dev/dsk/c0t3d0 0000 current
0001 /dev/dsk/c0t3d0 0001 current
0002 /dev/dsk/c0t3d0 0002 current
0003 /dev/dsk/c0t3d0 0003 current

# lvdisplay -v /dev/vg01/myfs1
--- Logical volumes ---
LV Name                  /dev/vg01/myfs1
VG Name                  /dev/vg01
LV Permission             read/write
LV Status                available/syncd
Mirror copies            0
Consistency Recovery     MWC
Schedule                 parallel
LV Size (Mbytes)         16
Current LE               4
Allocated PE              4
Stripes                  0
Stripe Size (Kbytes)    0
Bad block                on
Allocation               strict
IO Timeout (Seconds)    default

--- Distribution of logical volume ---
PV Name          LE on PV  PE on PV
/dev/dsk/c0t3d0      4        4

--- Logical extents ---
LE   PV1           PE1  Status 1
0000 /dev/dsk/c0t3d0 0004 current
```

Configuring Disk Devices

```
0001 /dev/dsk/c0t3d0      0005 current
0002 /dev/dsk/c0t3d0      0006 current
0003 /dev/dsk/c0t3d0      0007 current

# lvdisplay -v /dev/vg01/myfs2
--- Logical volumes ---
LV Name                  /dev/vg01/myfs2
VG Name                  /dev/vg01
LV Permission             read/write
LV Status                 available/syncd
Mirror copies              0
Consistency Recovery       MWC
Schedule                  parallel
LV Size (Mbytes)          16
Current LE                 4
Allocated PE                4
Stripes                      0
Stripe Size (Kbytes)        0
Bad block                   on
Allocation                  strict
IO Timeout (Seconds)       default

--- Distribution of logical volume ---
PV Name      LE on PV  PE on PV
/dev/dsk/c0t3d0    4        4

--- Logical extents ---
LE  PV1      PE1  Status 1
0000 /dev/dsk/c0t3d0  0008 current
0001 /dev/dsk/c0t3d0  0009 current
0002 /dev/dsk/c0t3d0  0010 current
0003 /dev/dsk/c0t3d0  0011 current
```

8-9. SLIDE: What's Next?

What's Next

Using Partitions or Logical Volumes to:

- create and mount file systems
- add swap area(s)

Manage Logical Volumes to:

- extend, reduce and remove a VG
- extend, reduce and remove a LV

a66938

Student Notes

After creating partitions or logical volumes you will want to use them for file systems or swap.

We have already mentioned that the LVM System is very flexible. You can manage it in many ways, as indicated on the slide.

8-10. LAB: Logical Volume Manager

Directions

In this exercise you will have an opportunity to create several physical volumes, volume groups, and logical volumes. Except where noted, do all of the exercises from the command line. You will have a chance to create some logical volumes via SAM in a later chapter.

Part I: Choosing a Disk for Use in a Volume Group

1. PART I

How many disks do you have on your system? Determine each disk's hardware path and device file names.

2. Use **vgdisplay** to determine which of the disks on your system are already members of active volume groups. You should have at least one disk that isn't currently in a volume group. Note the free disk's device filename; you will create a new volume group using your free disk in the next part of the lab exercise.
3. Before adding a disk to a volume group, you may want to check the size of the disk. This is accomplished via the **diskinfo** command. How large is your spare disk?

```
# diskinfo /dev/rdsck/c0t5d0
```

Part II: Creating a Physical Volume, Volume Group, and Logical Volumes

1. PART II

Configure your free disk as an LVM physical volume.

2. Can you **pvdisplay** your disk at this point? Try it.
3. Create a new **vg01** volume group using your newly created physical volume.
4. Use **vgdisplay** and **pvdisplay** to check the status of your new physical volume and volume group. How many physical volumes are in the volume group at this point? How many logical volumes are in the volume group at this point? What is the extent size?
5. Create two 24-MB logical volumes in your new volume group. Name the first logical volume **cad** and the second **cam**.
6. Use **vgdisplay** and **lvdisplay** to ensure that your new logical volumes were actually created.
7. Do an **ll** of the **/dev/vg01** directory. What is the name of the volume group device file for your new volume group? Each of your logical volumes should have two device files. Why?

8. Use SAM to remove your `vg01` volume group in preparation for the next portion of this lab. Volume groups can be managed in SAM from the Volume Groups object list:

SAM -> Disks and File Systems -> Volume Groups

9. Try a `vgdisplay` back at the command line to ensure the volume group is gone. Did SAM remove the volume group device files, too?

Part III: (Optional) LVM "What-Ifs"

1. PART III

What happens if you `pvcreate` a disk that is already in use by another active volume group? Try it. See what happens if you `pvcreate` your boot disk in `vg00`. Did this work? Try again using the `-f` "force" option on `pvcreate`. What is the result?

2. Re-`pvcreate` the spare disk you used in the previous part of the lab. (This `pvcreate` *should* work.)

3. For the sake of variety, create a new volume group called `vg02` using the spare disk you just `pvcreated`.

4. Now that you have a volume group, try creating a few logical volumes. Create a logical volume called **test1** in **vg02**. This time, though, don't specify a size for your logical volume. Based on the result of this experiment, what is the default logical volume size?

```
# lvcreate -n test1 vg02  
# vgdisplay -v vg02
```

5. What happens if you don't specify a logical volume name when you **lvcreate**? Try it. Create two new logical volumes of size 12 MB and 16 MB, leaving off the **-n** option in both cases. What names did LVM assign to your new logical volumes? Why?

6. See what happens if you attempt to create an 11 MB logical volume in **vg02** called **test2**. Watch the output from **lvcreate** carefully. What size is your new logical volume? Explain.

7. At some point in your UNIX career, you will almost certainly accidentally use **-1** instead of **-L** on **lvcreate**. What appears to be the difference between these two options? Try it and find out.

```
# lvcreate -1 12 -n test3 vg02  
# lvcreate -L 12 -n test4 vg02  
# vgdisplay -v vg02 | more
```

8. Once again, use SAM to remove the volume group you just created.

SAM -> Disks and File Systems -> Volume Groups

Part IV: Creating One Last Volume Group

1. PART IV

In preparation for the discussion of file systems in the following chapters, create a volume group `vg01` with your spare disk. Create three 12-MB logical volumes in `vg01` called `data`, `app`, and `tables`.

8-11. REVIEW: Check Your Understanding

Directions

Write the answers to the following questions.

1. List two benefits of using LVM.
2. Differentiate between a volume group and a logical volume.
3. What are the two reserved areas on a non-bootable disk?
4. What command do you use to initialize a disk as an LVM physical disk?
5. What are the steps for creating a volume group?
6. Record the commands you use to perform these tasks:
 - a. Make a disk an LVM disk.
 - b. Create a volume group.
 - c. Create a logical volume.

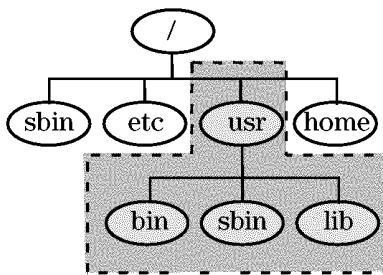
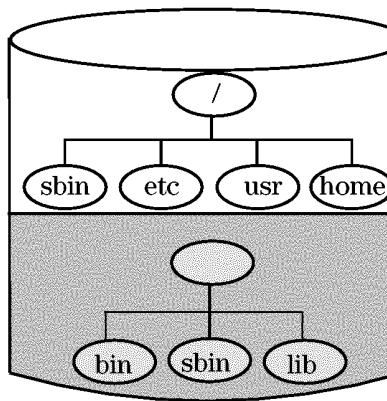
Module 9 — File System Concepts

Objectives

Upon completion of this module, you will be able to do the following:

- List the file system types available in HP-UX.
- Describe the difference between **user data** and **metadata**.
- Describe the structure of an HFS file system.
- Describe the structure of a JFS file system.
- Compare the process used to update HFS metadata versus JFS metadata.
- Define the terms: **superblock** and **inode**.
- Define the terms: **block**, **fragment**, and **extent**.
- Compare **hard** and **soft** links.
- Create hard and soft links.

9-1. SLIDE: What Is a File System?



a669121

Student Notes

What Is a File System?

A UNIX file system is a collection of files and directories stored and managed together as a unit. Each file system is stored in a separate logical volume or whole disk partition. A typical HP-UX machine usually has multiple file systems. The following are just a few of the file systems you may have on your machine:

- The files under /usr are usually stored in one file system.
- The files under /var are usually stored in another file system.
- The files under /tmp are usually stored in yet another file system.

The / (root) file system is a special file system that incorporates the files under several directories including /etc, /dev, /sbin, and possibly others.

The Benefits of Configuring Multiple File Systems

Although all of your files and directories could be stored as part of the root file system, separating subtrees of the file hierarchy into several distinct file systems offers several advantages:

- The administrator can allocate a fixed amount of disk space to each file system to ensure that no single file system is allowed to monopolize an entire disk. The administrator might, for instance, allocate 100 MB to the `/tmp` file system. This ensures that temporary files under `/tmp` can use at most 100 MB of disk space on your system; remaining disk space could be preserved for other file systems.
- Each file system may be tuned independently. There are a number of parameters associated with each file system that can significantly affect system performance. It may be beneficial to optimize some file systems for storage of large files, while others are optimized for storage of smaller files.
- File system maintenance tasks may be performed on one file system, while other file systems remain accessible to your users.

Mounting File Systems

Each file system contains its own **root directory**. The root directory has no name of its own. At boot time, the root file system (typically the file system in `/dev/vg00/lvol3`) is mounted at location `/`. Mounting is the association of a *name* in the file hierarchy with a *device*. For example, if your root disk is on the device represented by `/dev/dsk/c0t6d0`, the name `/` is associated with that device.

Once the root file system has been mounted, other file systems can be mounted. The other file systems are mounted on directories from previously mounted file systems. These directories are known as **mount points**. As long as the file system is mounted the mount point directory's name is associated with the file system's device. If the file system is unmounted, the connection is broken. The system maintains a table, known as a **mount table** that maps names to device files. It is only necessary to keep the mount points in this map. The location of other files can be derived from there.

In the example shown on the slide, there is a disk that has been divided into two logical volumes. Each logical volume has its own device file. A file system has been created in each logical volume. The file system in the first logical volume is the root file system. At boot time it is mounted at `/`. In other words, the name `/` is associated with the first logical volume. In the root file system there is an empty directory named `usr`. The second file system is mounted on `/usr`. In other words, the name `/usr` is associated with the second logical volume, as long as it is mounted. If the second file system were unmounted the name `/usr` would again refer to the directory in the root file system.

The entire file hierarchy is built in this manner. The system administrator determines which portions of the hierarchy should be broken into separate file systems.

Viewing the Mounted File Systems

Two commands allow you to view a list of your currently mounted file systems:

```
# mount -v  # reports which file systems are mounted where  
# bdf      # also reports file system sizes, and other info
```

9-2. SLIDE: File System Types

File System Types

- HP-UX supports several file system types:

HFS	High Performance File System
JFS	Journalized File System
NFS	Network File System
CDFS	CD-ROM File System
- HP-UX file commands work on all HP-UX file system types.

a669122

Student Notes

HP-UX supports several different file system types. The notes below briefly describe some of the features of the most common types of file systems.

High-Performance File System (HFS)

HFS represents HP-UX's standard implementation of the UNIX File System (UFS). HFS file systems reside on mass storage devices, usually hard disk drives. Prior to HP-UX Release 10.01 this was HP's only disk-based file system. HP's long term strategy is for JFS to become the default and for HFS to continue to exist for compatibility.

Journalized File Systems (JFS)

The HP-UX Journalized File System, also known as Veritas File System (VxFS), is an extent-based journaling file system which offers fast file system recovery and on-line features such as on-line backup, on-line resizing and on-line reorganization. An intent log contains

recent file system data structure updates. After a failure the system checks the intent log and performs the required rollback or roll forward.

There are two JFS products, base and online. The **base JFS** file system has the fast recovery feature and is included in all 10.01 and later systems. **Online JFS**, also referred to as **Advanced VxFS** is an optional product that adds extensions to JFS. It offers these additional capabilities:

- online defragmentation and reorganization
- online expansion and contraction of file system size
- online backup

Network File Systems (NFS)

NFS allows many systems to share the same files by using a client/server approach. Since access techniques are transparent, remote file access appears similar to local file access. Both JFS and HFS file systems can be shared with other systems using NFS. Network File System (NFS) provides transparent access to files from anywhere on the network. An NFS server makes a directory available to other hosts on the network by exporting the directory. An NFS client provides access to the NFS server's directory by mounting the directory. To users on the NFS client, the directory looks like part of the local file system.

CD-ROM File Systems (CDFS)

CD-ROM is an acronym for compact disk read-only memory. The information on the CD is virtually permanent; you can read data from a CD, but you cannot write to one. CD-ROMs come in a variety of formats.

Mixing and Matching File Systems

All file systems on your machine may be the same type, but more typically, a machine's file system hierarchy is comprised of several different file system types. The kernel, for example, must reside in an "HFS" file system, so `/stand` is always HFS. However, since JFS file systems offer more flexibility and greater reliability, other file systems on your machine may be JFS. Fortunately for your users, the same file system navigation commands (such as `cd`, `cp`, and `mv`) work across all file system types. You can determine what types of file systems you have on your system with two commands:

```
# mount -v          # what types of file systems are currently mounted?  
# fstyp /dev/vg00/lvol1 # what type of file system is in /dev/vg00/lvol1?
```

9-3. SLIDE: What's in a File System?

What's in a File System?

User data = actual data contained in files

Metadata = file system structural information

- Superblock
- Inodes
- Directories

a669123

Student Notes

Disk space allocated to a file system, regardless of the file system type, is subdivided into multiple file system blocks. The blocks in a file system may be used for two different purposes.

Some of the blocks in a file system store the actual data contained in users' files. These data blocks account for the majority of the blocks in most file systems.

However, some blocks in every file system store the file system's metadata. A file system's metadata describes the structure of the file system. Some of the metadata structures that are common to most file system types are described below.

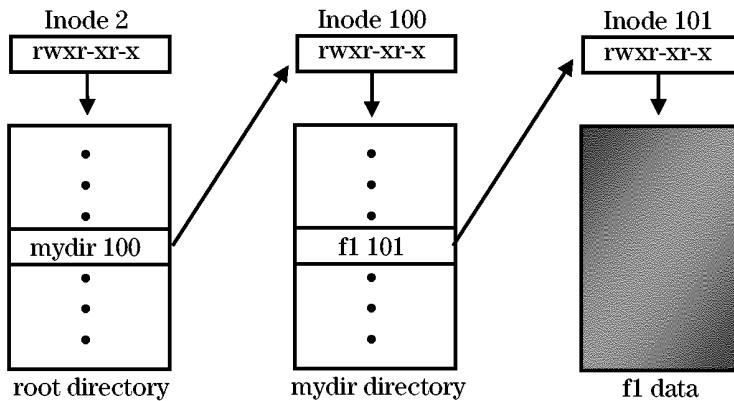
Superblock Every file system has a superblock that contains general information about the file system. The superblock identifies the file system type, size, and status, and contains pointers to all of the other file system metadata structures. Since the superblock contains such critical information, HP-UX maintains multiple redundant copies of the superblock in every file system.

Inodes	Every file has an associated inode containing the attributes of the file. The inode identifies the file's type, permissions, owner, group, and size. A file's inode also contains pointers to the data blocks associated with the file. Each inode is identified by a unique inode number within the file system.
Directories	Users and applications generally reference files by name, not by inode number. The purpose of a directory is to make a connection between file names and their associated inode numbers.

The output and error messages from the HP-UX file system management utilities frequently refer to these metadata structures. A conceptual understanding of these structures and terms will contribute greatly to your success as a system administrator. The remainder of this module examines HFS and JFS metadata structures in somewhat greater detail.

9-4. SLIDE: Accessing a File System

Accessing a File System



```
cat /mydir/f1
```

a669124

Student Notes

The previous slide introduced several file system components: the superblock, directories, inodes, and data blocks. How are these structures used when users access a file or directory? Consider the example on the slide, which shows how a user might access a file called `/mydir/f1`.

First, HP-UX must find the `/` directory. The root directory of any file system is always found at inode number 2. HP-UX checks the permissions recorded in inode number 2 to ensure that the user has access to the `/` directory. The inode also contains pointers to the data blocks for the `/` directory.

A directory's data blocks contain a directory entry for each file or subdirectory within the directory. Each directory entry contains the name of a file or subdirectory and the inode associated with that file or subdirectory. HP-UX searches for the `mydir` directory entry in `/`.

The `mydir` directory entry identifies the inode associated with `mydir`. In the example on the slide, `mydir`'s inode number is 100. HP-UX checks inode number 100 to determine if the user should be granted access to `mydir`.

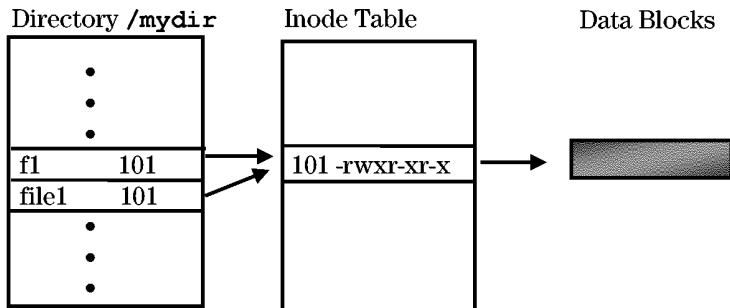
Next, HP-UX follows the pointers in inode 100 to the data blocks for `mydir`. These data blocks should contain a directory entry for file `f1`.

The `f1` directory entry identifies the inode associated with `f1`. In the example on the slide, `f1`'s attributes are recorded in inode number 101. HP-UX checks inode number 101 to ensure that the user has access to the file, then follows the pointers in the inode to `f1`'s data blocks.

Having found the data blocks for `/mydir/f1`, HP-UX displays the contents of those data blocks on `stdout`.

9-5. SLIDE: HP-UX Hard Links

HP-UX Hard Links



```
ln /mydir/f1 /mydir/file1
```

a669125

Student Notes

Although most inodes are associated with exactly one directory entry, hard links make it possible to associate multiple directory entries with a single inode. This, in effect, allows your users to reference a single file via several different file names.

The example on the slide shows a file `/mydir/f1` that has been hard linked to `/mydir/file1`. Both names now reference the same inode, and thus share the same permissions, owner, and time stamp.

Since both file names reference the same inode, they also both ultimately reference the same data blocks. Changes made to `f1` will be reflected in `file1`, and vice versa. `f1` and `file1` are essentially the same file! Oftentimes it is useful to associate multiple file names with a single file in this manner.

A hard link may be created with the `ln` command. The first argument identifies the file name of the existing file, and the second identifies the name of the new link:

```
# ln /mydir/f1 /mydir/file1
```

Creates a link to f1

```
# ll /mydir
```

Shows the number of links to each file

Creating a hard link creates a new directory entry for the new link, and increments the link count field in the inode. The second field in the output from the ll command shows the number of links to each file.

Hard links are oftentimes used to associate multiple file names with a single device file. For instance, some administrators do the following:

```
# ln /dev/rmt/c0t0d0BEST /dev/tape
```

After creating this link, users can access the c0t0d0 tape drive using the intuitive name **/dev/tape**, rather than the cryptic default name **/dev/rmt/c0t0d0BEST**. Be aware of two hard link limitations:

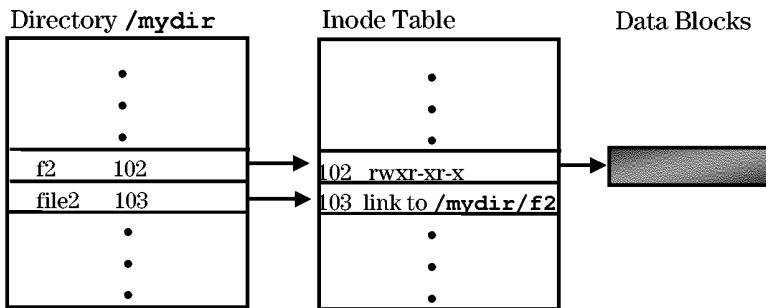
- Hard links cannot cross file system boundaries.
- Hard links cannot link directories.

Questions

1. Why isn't it possible to link across file system boundaries?
2. How would **/mydir/file1** be affected if you were to remove **/mydir/f1**?

9-6. SLIDE: HP-UX Symbolic Links

HP-UX Symbolic Links



```
ln -s /mydir/f2 /mydir/file2
```

a669126

Student Notes

Symbolic links, like hard links, make it possible to associate multiple file names with a single file. Unlike hard links, however, symbolic links:

- can cross file system boundaries
- can link directories

In the example on the slide, /mydir/file2 is a symbolic link to /mydir/f2. f2 and file2 have distinct directory entries and inodes. However, as shown on the slide, /mydir/file2 is nothing more than a pointer to /mydir/f2! Accessing /mydir/file2 yields the same data one would see when accessing /mydir/f2.

Symbolic links are particularly useful when you must move files from one file system to another, but still wish to be able to use the file's original path name. At version 9 of HP-UX, system executables were stored in the /bin directory. At HP-UX 10, many operating system executables were moved to /usr/bin. However, a symbolic link exists from /bin to /usr/bin

so users and applications can still use the version 9 path names. This is just one situation where symbolic links are commonly used.

Use the `ln` command with `-s` to create a symbolic link. The first argument identifies the existing file that you wish to link to. Additional arguments specify the path names of the symbolic links you wish to create to the existing file.

```
# ln -s /mydir/f2 /mydir/file2
```

The `ll` command identifies symbolic links with an `l` in the first character position. Also, the file name field in the `ll` output identifies the file to which a symbolic link leads.

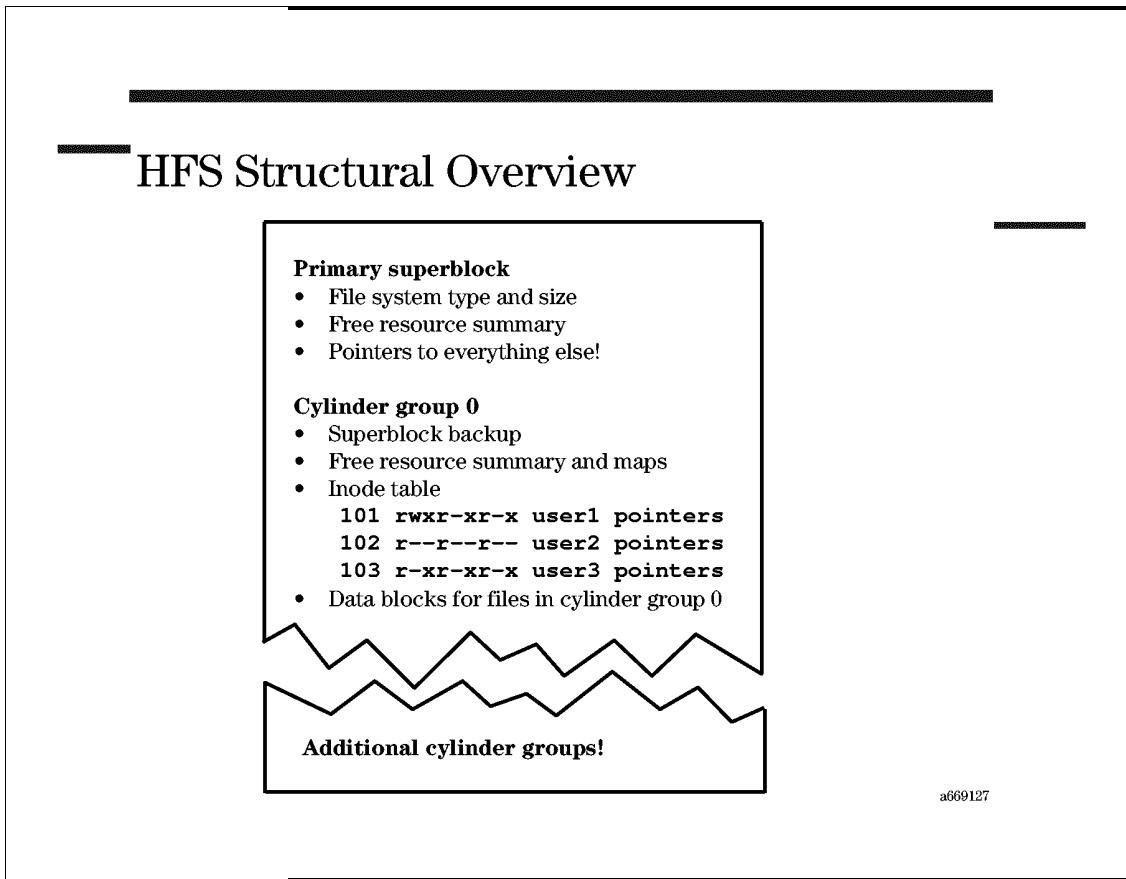
NOTE:

In the example above, removing `/mydir/f2` will not automatically remove the `/mydir/file2` link! After removing `/mydir/f2`, accessing the link will result in an error message.

Question

1. Why is it possible to create symbolic links, but not hard links, across file system boundaries?

9-7. SLIDE: HFS Structural Overview



Student Notes

The last few slides introduced some of the structures found in an HP-UX file system and how those structures are used to find and access a file. Although this procedure is fairly consistent across file system types, there are some important differences between HFS and JFS file systems. The remainder of this module highlights some of the significant differences between HFS and JFS file systems.

The HFS Superblock

The first 8 KB of every HFS file system contains the file system's primary superblock. Recall that the superblock contains general information about the file system's size, usage, status, and configuration. Perhaps most importantly, the superblock contains pointers to all the other HFS metadata structures in the file system.

Because the superblock is so critical, HFS maintains multiple redundant superblock copies. The backup HFS superblock locations are recorded in the `/var/adm/sbtab` file. A damaged HFS superblock may be repaired using the information in these backup superblocks.

HFS Cylinder Groups

Remaining space in an HFS file system is divided among one or more cylinder groups. Larger HFS file systems have more cylinder groups, and smaller file systems have fewer cylinder groups.

Each HFS cylinder group contains a redundant superblock, a portion of the file system's inodes, and a portion of the file system's data blocks.

HFS Inodes

Every file and directory must have an inode. The inode identifies the file's permissions, owner, group, and other attributes.

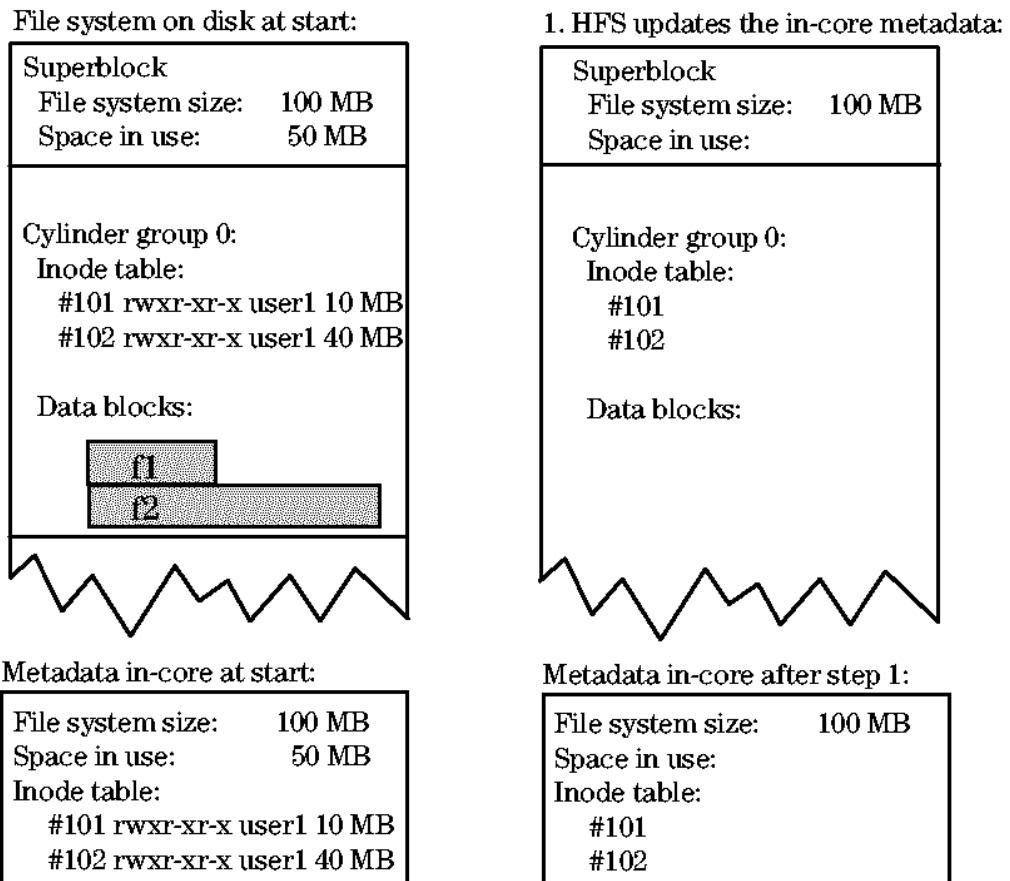
When you create an HFS file system, you must specify how many inodes you wish to create. The inodes are stored in the new file system's inode table. In an HFS file system, it is possible to run out of inodes in the inode table. Without a free inode, you can't create new files and directories, even if there are free data blocks remaining! You can add additional inodes in one of two ways:

- Allocate additional disk space to the file system. Extending the file system also adds additional inodes in the inode table.
- Backup your data, rebuild the file system with more inodes, and restore your data from tape.

9-8. CHALK TALK: HFS File System Updates

This exercise reinforces your understanding of the HFS metadata structures and the implications these structures have for you as an administrator. Your instructor will walk you through the process used by HFS to remove file f1 from the HFS file system.

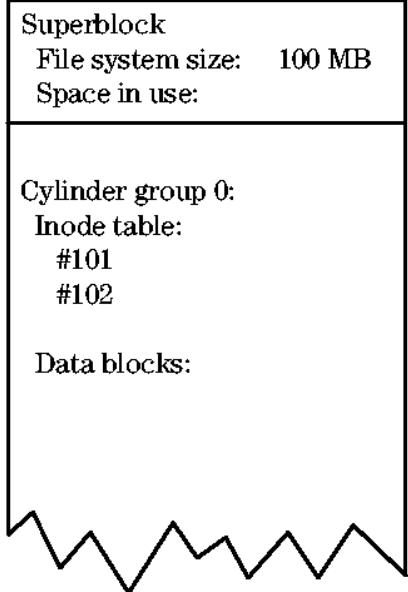
As your instructor discusses each step in the exercise, pencil in the changes to the metadata structures and consider the following question: "What impact would a system crash have on the file system metadata at this point?"



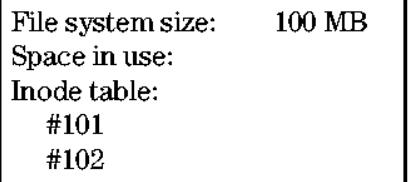
a69715

Figure 9-1.

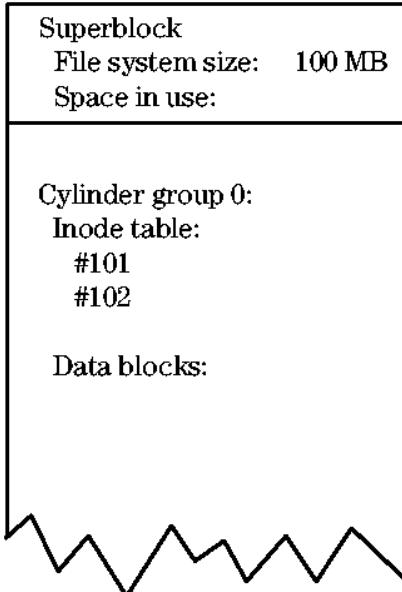
2. HFS de-allocates f1's inode and data blocks



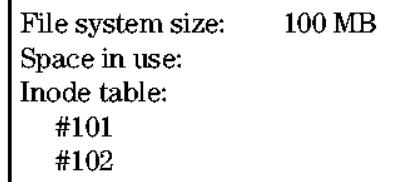
Metadata in-core after step 2:



3. HFS updates the superblock



Metadata in-core after step 3:



a69712

Figure 9-2.

Questions

1. A system crash can cause serious problems in an HFS file system. Why?
2. Conceptually, what must be done to every HFS file system after a system crash?

9-9. SLIDE: HFS Blocks

HFS Blocks

What is an HFS block?

- HFS always reads a *block* of data at a time
- Allowed block sizes: 4 KB, 8 KB, 16 KB, 32 KB, 64 KB
- Blocks are not necessarily contiguous
- Block size is set at file system creation

Assuming 8-KB blocks,

- How many accesses are required for an 8-KB read?
- How many accesses are required for a 32-KB read?
- How many bytes are read if a 1-KB read is requested?

a6975

Student Notes

Our discussion of HFS structures thus far has concentrated on HFS metadata. But how does HFS access and allocate data blocks in the file system?

As you are accessing your files and directories, HFS always reads and writes a block of data at a time. Sequentially accessing a large file may require accessing multiple blocks, which may not be contiguous on disk.

The default HFS block size is 8 KB. However, at file system creation, you may choose to specify a larger or smaller block size. Valid block sizes range from 4 KB to 64 KB, in powers of two. After initially creating a file system, it is not possible to change the block size.

Questions

1. Answer the three questions asked at the bottom of the slide.
2. If you plan to do long sequential reads and writes to a file system would you choose a larger, or a smaller block size?

9-10. SLIDE: HFS Fragments

HFS Fragments

What is an HFS fragment?

- Smallest unit of space HFS can allocate to a file
- Allowed fragment sizes: entire block, 1/2 block, 1/4 block, 1/8 block
- Fragment size is set at file system creation

Example:

- How would HFS allocate space for three files of size 8 K, 4 K, and 3 K if:

Block size = 8 K

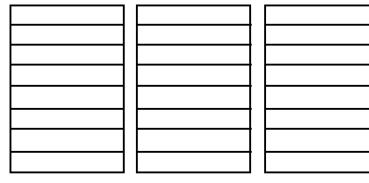
Fragment size = 8 K



8K fragment

Block size = 8 K

Fragment size = 1 K



8 1K fragments 8 1K fragments 8 1K fragments

a669131

Student Notes

Each HFS data block is further subdivided into fragments. The file system fragment size defines the smallest unit of space that can be allocated to a file. A file system's fragment size is defined at file system creation, and cannot be changed. Valid fragment sizes include an entire block, a half block, a quarter block, or an eighth of a block.

When allocating blocks and fragments for a file in HFS:

- Fragments will only be allocated at the end of a file. (In other words, HFS allocates full blocks to a file, unless the remainder of the file requires only a portion of a block.)
- A block may contain fragments from more than one file.
- Multiple fragments can be used for a file, but only at the end of the file. Also, all of the fragments must be contiguous within a single block.
- HFS tries to allocate all of a file's blocks in the same cylinder group.

The slide shows several blocks from two file systems. Given the block and fragment sizes shown on the slide, how might HFS allocate space for three files of sizes 8 KB, 4 KB, and 3 KB in these two file systems? In the graphic on the slide, indicate which fragments would be used for which files.

Questions

1. Based on the example shown on the slide, what advantage does a smaller fragment size offer?
2. The file system must maintain a map indicating which fragments are already allocated, and which fragments are available. Given this fact, what might be the disadvantage of a smaller fragment size?
3. When would you choose a larger fragment size? When would you choose a smaller fragment size?

9-11. SLIDE: HFS Implications

HFS Implications

HFS Advantages

- Easy to manage
- Fast and efficient
- The *only* option for the file system containing the kernel

HFS Disadvantages

- Slow and unpredictable crash recovery
- Must be unmounted to extend
- Impossible to reduce

a669132

Student Notes

The past few slides have discussed some of the structures that underlie an HFS file system. The structure and design of HFS leads to both some advantages and disadvantages that you should consider as you decide whether your file systems should be HFS or JFS.

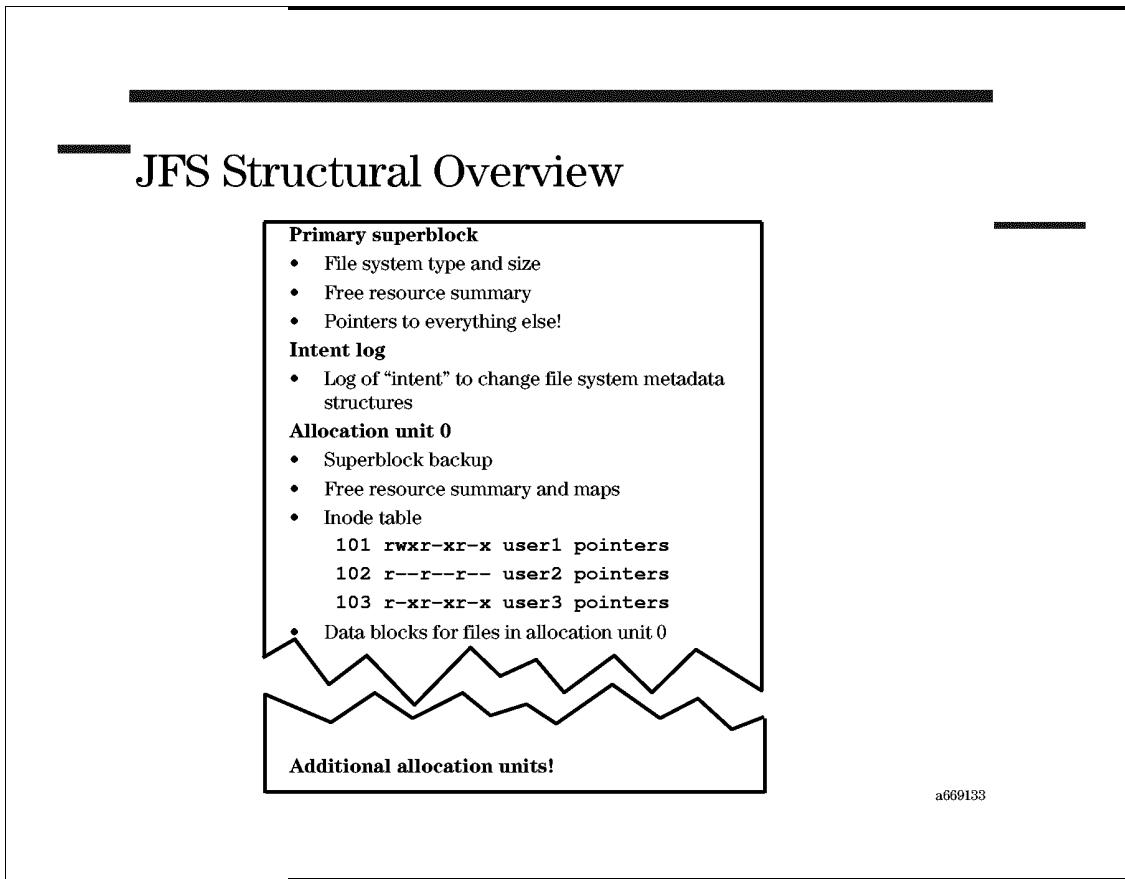
HFS advantages

- The commands required to create and manage an HFS file system are relatively simple. SAM also provides an intuitive interface for creating and managing HFS file systems.
- HFS offers fast and efficient performance, assuming the file system block size, fragment size, and other parameters are configured properly.
- The file system containing the kernel (typically `/stand`) *must* be HFS. Therefore, every HP-UX machine must have at least one HFS file system.

HFS disadvantages

- A system crash or improper shutdown wreaks havoc on an HFS file system. Recovery is slow and unpredictable; in some cases, it may be necessary to recreate the file system and restore all data from tape.
- An HFS file system may only be extended while unmounted. In a high availability environment, this may not be a practical option.
- It is not possible to reduce an HFS file system.

9-12. SLIDE: JFS Structural Overview



Student Notes

The Journal File System (also known as the Veritas or VxFS file system) was designed to address many of the limitations of HFS. This slide introduces some of the structures underlying JFS, and the next few slides explore the JFS structures in more detail.

JFS Superblocks

Like HFS, JFS has a superblock at the beginning of the file system that describes the file system's size, status, and configuration. The information contained in the superblock is critical! If the primary superblock is corrupted, it may be reconstructed using redundant superblock backups that JFS automatically maintains. Unlike HFS, JFS can find its redundant superblocks automatically; thus, JFS backup superblocks are not listed in `/var/adm/sbtab`.

The JFS Intent Log

A key advantage of JFS is that all metadata changes are recorded in an intent log. This logging mechanism ensures the integrity of the file system, and allows the file system to be recovered quickly in the event of a system crash.

A record of all metadata updates for a given transaction is written with one disk access. This ensures that all the metadata updates are *atomic*. In other words the file system will reflect either a before or after image of an operation, but never an intermediate image. The actual metadata updates to the superblock, inode, maps, and directory can be delayed until a `sync` is issued. Once all the metadata updates associated with a transaction have been completed, a *done* record is written to the intent log.

If the system were to crash, the file system can quickly be recovered by applying all changes in the JFS Intent Log. Since only entire transactions are logged, there is no risk of a file change only being partially updated, only some metadata updates related to the transaction being logged, and other metadata updates related to the same transaction not being logged). The logging of only entire transactions prevents the file system from being out-of-sync due to some transaction occurring in the middle of the crash. Either the whole transaction is lost or the whole transaction is logged to the intent log. This allows the JFS intent log to be used in a recovery situation.

After a system crash, the file system metadata can be returned to a consistent state by simply replaying the transactions in the intent log. This can be completed within just 10–20 seconds!

JFS Allocation Units

Disk space allocated to a JFS file system is subdivided into one or more 32 MB allocation units. Each allocation unit contains a portion of the inodes for the file system, as well as a portion of the data blocks. Larger JFS file systems have more allocation units, and smaller JFS file systems have fewer allocation units.

JFS Inodes

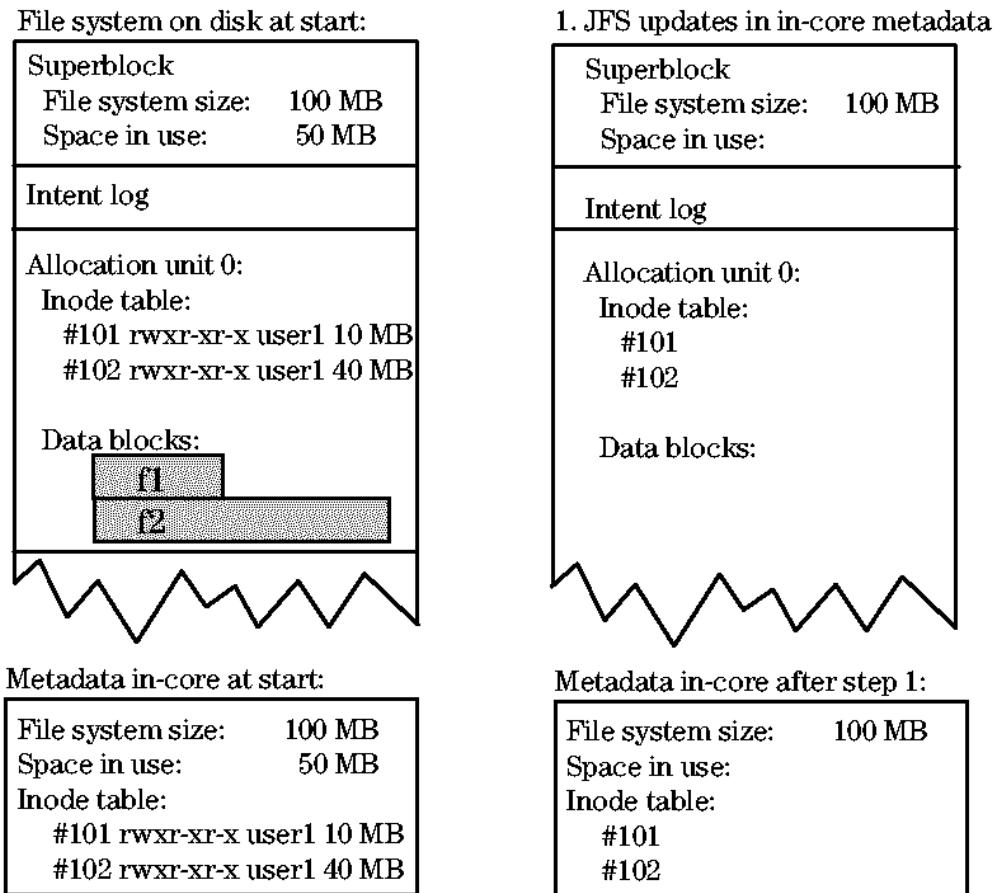
Like HFS, JFS stores file and directory attributes in inodes. Unlike HFS, however, JFS creates inodes dynamically. JFS creates several inodes initially, then may create additional inodes as needed. As long as there are free data blocks in the file system, JFS can create more inodes.

Questions

1. What metadata structures does JFS have in common with HFS?
2. What metadata structures and features distinguish JFS from HFS?

9-13. CHALK TALK: JFS File System Updates

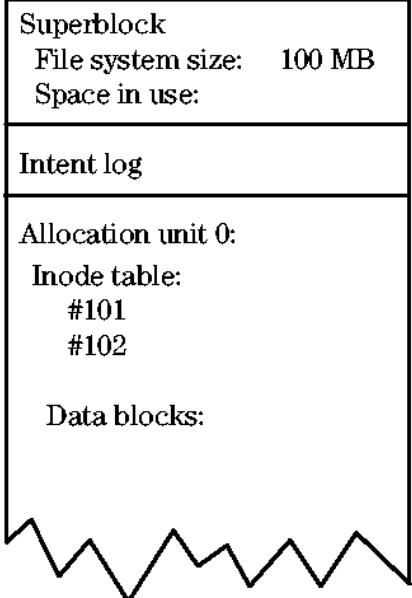
This exercise should reinforce your understanding of the JFS metadata structures. Your instructor will walk you through the steps JFS must perform when a user removes file f1 from the JFS file system below. As your instructor discusses each step in the exercise, pencil in the changes to the metadata structures and consider the following question: "What impact would a system crash have on the file system metadata at this point?"



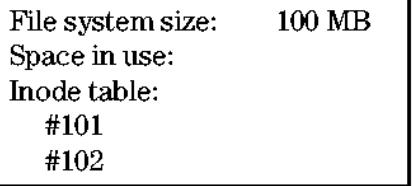
a69713

Figure 9-3.

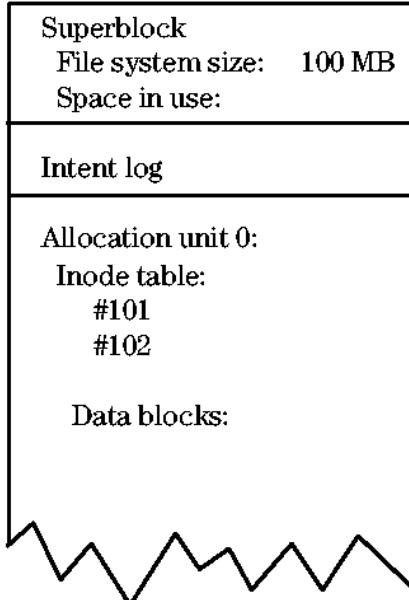
2. JFS creates an intent log entry



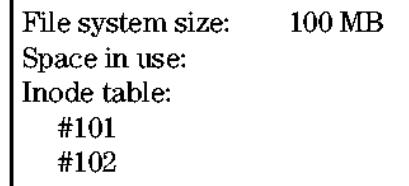
Metadata in-core after step 2:



3. JFS de-allocates f1's inode and data blocks



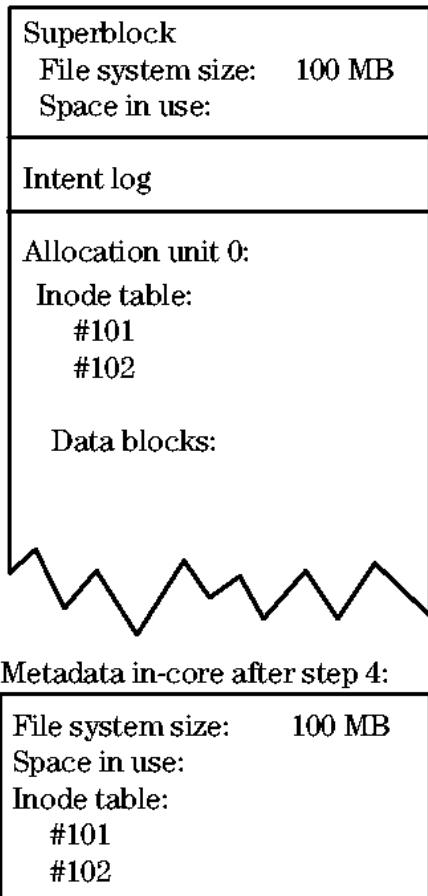
Metadata in-core after step 3:



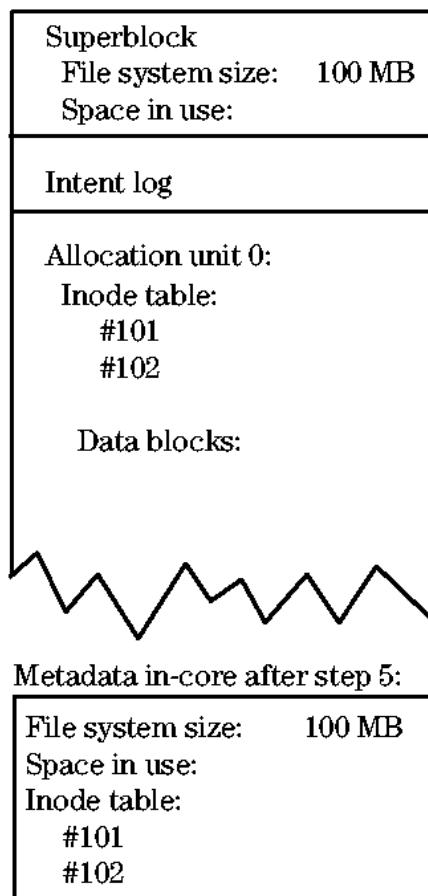
a669135

Figure 9-4.

4. JFS updates the superblock



5. JFS marks the intent log done



a69714

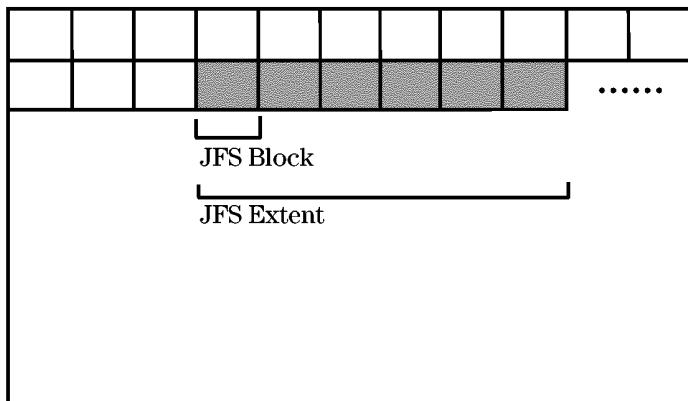
Figure 9-5.

Question:

1. After a system crash, what must JFS do to restore metadata consistency?

9-14. SLIDE: JFS Blocks and Extents

JFS Blocks and Extents



a669136

Student Notes

In JFS, a block represents the smallest unit of disk space that can be allocated to a file. Although you can change the block size when creating a JFS file system, the default 1-KB block size is recommended in most situations.

When you create a file, JFS allocates an extent to the file. An extent is a contiguous group of blocks in the file system. As the file grows, JFS tries to increase the size of the extent if contiguous space is available. If contiguous space is not available, JFS uses another extent elsewhere in the file system. In order to optimize performance, JFS tries to allocate the next extent in the same allocation unit if possible.

Since a file's blocks are arranged contiguously on disk in a JFS file system, JFS can read large amounts of data into memory with a single I/O operation.

9-15. SLIDE: JFS Implications

JFS Implications

JFS Advantages

- Fast, reliable crash recovery
- Online resizing
- Online backups

JFS Disadvantages

- Fragmentation issues
- “Online” functionality not included with HP-UX
- Kernel can’t be in a JFS file system

a669137

Student Notes

You have seen that JFS structures and the procedure JFS uses to update those structures are somewhat different than the HFS structures and procedures discussed earlier in the module. These differences give JFS a number of advantages over HFS, which are described below. To be fair, several disadvantages are discussed as well.

Advantages

- The JFS intent log guarantees that JFS file systems can be up and running within a matter of seconds after an improper shutdown.
- The JFS `fsadm` utility makes it possible to extend a JFS file system even while the file system is still mounted. This is an important feature in 24x7 shops. In some cases, `fsadm` can also be used to reduce a JFS file system.
- JFS also provides a file system snapshot capability, which makes it possible to backup a file system even while files and directories are being modified.

Disadvantages

- JFS is an extent-based file system. In order to achieve optimum performance, the data blocks in your files and directories should be contiguous. In dynamic file systems, this may require regular execution of the fsadm defragmentation utility.
- HP offers two JFS products. The base JFS product comes standard with HP-UX and includes the utilities necessary to create a JFS file system, as well as the fast crash recovery feature. However, online resizing, online backups, and the defragmentation utility are only included in the add-on online JFS product, which you must purchase from HP.
- The kernel can't reside in a JFS file system. Thus, though the majority of your file systems may be JFS, `/stand` will always be HFS.

9-16. REVIEW: Check Your Understanding

Directions

Write the answers to the following questions.

1. List three types of file systems available on an HP-UX system.
2. List three types of file system metadata.
3. List three pieces of information in a superblock.
4. List some of the information found in an inode.
5. Describe what is created when the link command is executed.
6. What is the purpose of the JFS intent log?

Module 10 — File System Creation

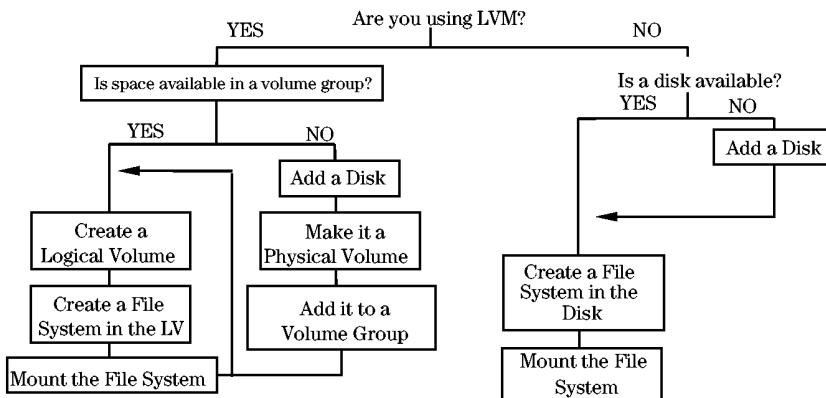
Objectives

Upon completion of this module, the student will be able to:

- Create a file system from the command line and SAM.
- Mount or unmount a file system from the command line.
- Automatically mount a file system via `/etc/fstab`.

10-1. SLIDE: Overview of File System Creation

Overview of File System Creation



a538275

Student Notes

You can expand your file hierarchy's overall capacity by creating new file systems and attaching (mounting) them to the existing file system tree. A file system can be created in either a logical volume or a whole disk. You can use either SAM or a sequence of HP-UX commands to create a file system.

You don't necessarily have to add a new disk drive to create a new file system. If you have free extents in any of your volume groups, you can simply create another logical volume, and then create the file system.

There are many reasons why you may want to add a new file system. Some reasons include

- You anticipate that your current file system will soon reach maximum capacity.
- Your current file system has already reached maximum capacity.
- You wish to physically separate portions of a file system for a particular reason, such as different uses or different groups of users.

- You want to create a file system with unique characteristics of ownership and/or mirroring.

The steps to create a new file system are shown on the slide. We will talk about each step in detail on the following pages.

10-2. SLIDE: Creating a New File System

Creating a New File System

1. Create the new file system

```
newfs -F vxfs /dev/vg01/rmyfs1
```

2. Create a mount point directory for the file system

```
mkdir /myfs1
```

3. Mount the new file system

```
mount /dev/vg01/myfs1 /myfs1
```

4. Add the file system to the `/etc/fstab` file

a6976

Student Notes

First, you must create a superblock, inode table, and other file system metadata structures for the new file system. The `newfs(1m)` command is the utility of choice for creating both VxFS and HFS file systems. The example on the slide creates a VxFS file system in logical volume `/dev/vg01/rmyfs1`. The next slide presents more `newfs` examples.

Once the file system metadata structures have been created with `newfs`, you must create a mount point directory for the file system, and mount it.

Finally, if you wish to make a file system permanently available, it should be added to the `/etc/fstab` file. Every time the system boots, the `/etc/fstab` file is consulted to determine which file systems should be automatically mounted. Thus, if a new file system is not added to `/etc/fstab`, it will not be automatically mounted after the next system boot.

After creating and mounting a new file system, you may want to issue the `bdf(1m)` command or `mount(1m)` to ensure that your file system successfully mounted.

10-3. SLIDE: The newfs Command

The newfs Command

Examples

1. `newfs -F vxfs /dev/vg01/rmyfs1`
2. `newfs -F vxfs -o largefiles /dev/vg01/rmyfs1`
3. `newfs -F hfs /dev/vg01/rmyfs2`
4. `newfs -F hfs -o largefiles /dev/vg01/rmyfs2`
5. `newfs -F hfs -b 4096 -f 2048 -m 5 -i 4096 /dev/vg01/rmyfs2`

a66940

Student Notes

After a logical volume has been set aside for use as a file system, you must create a superblock, inode table, and other metadata structures before you can create files and directories in the file system. The `newfs` command is the utility of choice for creating these structures.

Slide Examples

- The first example on the slide simply creates a VxFS file system in a logical volume called `/dev/vg01/rmyfs1`. The `-F vxfs` option specifies that a VxFS file system should be created. Also note that the `newfs` command requires a raw device file as an argument. The logical volume or disk specified will be overwritten, and any existing data in that logical volume or disk will be destroyed.
- The second example is much like the first. However, it includes the `largefiles` option. HP-UX version 10.20 introduced support for large files up to 128 GB in size. Without the `-o largefiles` option, however, the largest file size allowed in a file system is 2 GB. The table below summarizes large file support for various versions of HP-UX.

Table 10-1.

HP-UX Version	Max File System Size	Max File Size
9.x-10.01	4 GB	2 GB
10.10	128 GB	2 GB
10.20	128 GB	2 GB (without -o largefiles) 128 GB (with -o largefiles)

The third example creates an HFS file system in the `/dev/vg01/rmyfs2` logical volume.

The fourth example creates an HFS file system with large file support in `/dev/vg01/rmyfs2`.

The last example creates an HFS file system in `/dev/vg01/rmyfs2`. However, unlike the previous HFS examples, this example explicitly sets several file system parameters. The `-b 4096` sets the block size to 4 KB, `-f 2048` sets the fragment size to 2 KB, `-m 5` changes the file system "min free value" to 5% rather than the default 10%, and `-i 4096` specifies that one inode should be created for every 4 KB of space in the file system. These parameters only apply to HFS file systems, and can be set at file system creation to improve performance. For more information, see the detailed description in the notes that follow.

newfs Options Common to HFS and VxFS

The `newfs` command has many options. Some apply to both file system types, some only apply to HFS, and some apply only to JFS. The following list is a description of some of the options that apply to both file system types.

- F hfs|vxfs** Defines the desired file system type. If "-F" is not specified, the default file system type is determined from the `/etc/default/fs` file.
- o largefiles** Determines whether or not the file system will allow "large files" over 2 GB in size. "nolargefiles" is the default.
- s size** Specifies the total size of the file system in blocks. By default, `newfs` uses all the space available on the specified logical volume or disk.
- v** Verbose. List all actions performed by `newfs`.

newfs Options Specific to HFS

Some `newfs` options are usually only used when creating HFS file systems:

- L/-S** Early versions of HP-UX only supported "short" filenames up to 14 characters in length. You can still enforce this limit if you wish using the "-S" option. By default, however, file systems allow "long" filenames up to 256 characters in length.
- b block-size** Specifies the file system's HFS block size in bytes. The block size defaults to 8KB.

-f <i>frag-size</i>	Specifies the file system's fragment size in bytes. The fragment size usually defaults to 1KB.
-m <i>min-free</i>	This indicates the percentage of space in the file system reserved for use by root. If the amount of free space in the file system falls below this percentage, the super-user is the only one who can write to the file system. This defaults to 10%. Consider increasing this value in very large file systems.

newfs Options Specific to VxFS

There are several JFS-specific options on **newfs**. However, with the exception of **-o largefiles** as described previously, it is recommended that you take the default options when creating a JFS file system. See the **newfs_vxfs(1m)** and **mkfs_vxfs(1m)** man pages for details.

newfs Option Specific to the Whole-disk Approach

Most administrators today choose to partition disks using LVM. On workstations, however, you may prefer the simplicity of the whole-disk approach.

If you wish to use an entire disk for a file system use a **newfs** command similar to one of the following:

```
# newfs -F hfs /dev/rdsck/c0t2d0 # creates an HFS on disk c0t2d0
# newfs -F vxfs /dev/rdsck/c0t2d0 # creates a JFS on disk c0t2d0
```

The "-R" option reserves space at the end of the disk for use as swap space:

```
# newfs -F hfs -R 200 /dev/rdsck/c0t2d0 # HFS, with 200MB reserved for swap
# newfs -F vxfs -R 200 /dev/rdsck/c0t2d0 # VxFS, with 200MB reserved for swap
```

You may also create a boot disk using the whole disk approach; see the **newfs_hfs(1m)** man page for more information.

Output from newfs

After creating the new file system, **newfs** displays several lines of information about the new file system:

```
# newfs -F vxfs /dev/vg01/rmyfs1
      version 3 layout
      16384 sectors, 16384 blocks of size 1024, log size 1024 blocks
      unlimited inodes, 16384 data blocks, 15288 free data blocks
      last allocation unit has 16384 data blocks
      first allocation unit starts at block 0
      overhead per allocation unit is 0 blocks

# newfs -F hfs /dev/vg01/rmyfs2
mkfs (hfs): Warning - 94 sector(s) in the last cylinder are not allocated.
```

```
mkfs (hfs): / dev/vg01/rmyfs2 - 16384 sectors in 107 cylinders of 7 tracks, 22
sectors
16.8Mb in 7 cyl groups (16 c/g, 2.52Mb/g, 384 i/g)
Super block backups (for fsck -b) at:
    16,    2504,    4992,    7480,    9968,   12456,   14944
```

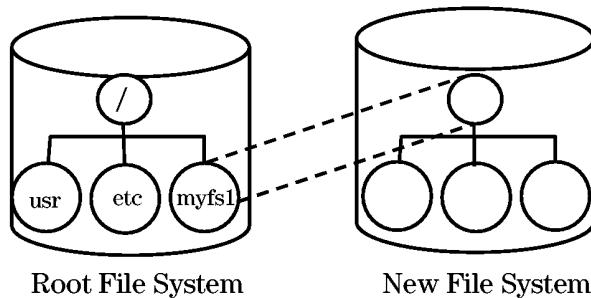
NOTE:

There are several man pages for the **newfs** command.

- **newfs(1m)** describes **newfs** options common to all file systems.
 - **newfs_hfs(1m)** describes HFS-only options
 - **newfs_vxfs(1m)** describes VxFS-only options.
-

10-4. SLIDE: Mounting the New File System

Mounting the New File System



```
mkdir /myfs1
mount /dev/vg01/myfs1 /myfs1
mount -v
```

a66941

Student Notes

Mounting a File System

HP-UX cannot use a file system unless it is mounted. After a file system has been created on a logical volume or device, it must be incorporated into the system's file hierarchy by creating a mount point and mounting the file system. Example:

```
# mkdir /myfs1          # create a mount point
# mount /dev/vg01/myfs1 /myfs1    # mount a file system on the mount point
```

NOTE:

The **mount** command requires a block device file, while **newfs** requires a character device file.

Mounting a file system in this manner logically associates the root directory of the new file system with the mount point directory. Accessing the mount point directory actually references the file system mounted on that directory.

Guidelines for Choosing a Mount Point

Though mount points can be created in any directory, most file systems are mounted on mount points immediately under the / directory. /usr, /tmp, and /home are just a few examples of mount point directories you may have on your system.

Also, file systems should only be mounted on empty directories. If a file system is mounted on a directory that already contains files and directories, those files and directories will be hidden until the file system is unmounted!

Finally, note that it is not possible to mount a file system on a directory that is already in use by another user or process. Trying to mount a file system on a directory that is already in use results in a "device busy" message.

Viewing Mounted File Systems

Two commands list your currently mounted file systems.

The **mount -v** command displays a verbose listing showing which file systems were mounted where and when.

```
# mount -v
/dev/vg00/lvol3 on / type vxfs log on Fri Jun 26 15:04:49 1998
/dev/vg00/lvol1 on /stand type hfs defaults on Fri Jun 26 15:04:51 1998
/dev/vg00/lvol8 on /var type vxfs delaylog on Fri Jun 26 15:04:57 1998
/dev/vg00/lvol7 on /usr type vxfs delaylog on Fri Jun 26 15:04:57 1998
/dev/vg00/lvol6 on /tmp type vxfs delaylog on Fri Jun 26 15:04:57 1998
/dev/vg00/lvol5 on /opt type vxfs delaylog on Fri Jun 26 15:04:58 1998
/dev/vg00/lvol4 on /home type vxfs delaylog on Fri Jun 26 15:04:58 1998
/dev/vg01/myfs1 on /myfs1 type vxfs delaylog on Fri Jun 26 15:55:10 1998
/dev/vg01/myfs2 on /myfs2 type hfs defaults on Fri Jun 26 15:18:55 1998
```

The **bdf** command also displays the amount of space in use and available in each mounted file system.

```
# bdf
# bdf
Filesystem      kbytes   used   avail %used Mounted on
/dev/vg00/lvol3     86016   23302   58855   28% /
/dev/vg00/lvol1     67733   20068   40891   33% /stand
/dev/vg00/lvol8     69632   14138   52387   21% /var
/dev/vg00/lvol7    397312  334233   59179   85% /usr
/dev/vg00/lvol6     49152    1326   44902    3% /tmp
/dev/vg00/lvol5    294912  261523   31352   89% /opt
/dev/vg00/lvol4     24576   19333    4978   80% /home
/dev/vg01/myfs1     15893       9   14294    0% /myfs1
```

```
/dev/vg01/myfs2      16384    1109    14328    7% /myfs2
```

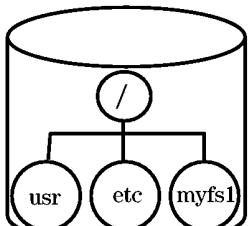
NOTE:

There are several man pages for the **mount** command.

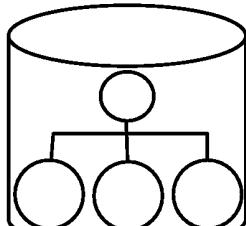
- **mount(1m)** describes **newfs** options common to all file systems.
 - **mount_hfs(1m)** describes HFS-only options.
 - **mount_vxfs(1m)** describes VxFS-only options.
-

10-5. SLIDE: The **umount** Command

The **umount** command



Root File System



Unmounted File System

```
umount /dev/vg01/myfs1  
or  
umount /myfs1
```

a66942

Student Notes

Now that you know how to mount a new file system, you should also be aware of how to logically disassociate, or unmount, the new file system from the root file system. The command used to unmount the file system is **umount**.

NOTE:

The command is **umount**, *not* "unmount". The command uses the block device file or mount-point directory.

umount options include:

- a umount "all" currently mounted file systems.
- F *FStype* specify a file system type
- v report the output with the *FStype* displayed

Instead of using the **umount -a** command, you can use the **umountall(1M)** command.

A file system cannot be unmounted if any files are open or if any user's current working directory is a directory in that file system. You can use the **fuser** command to identify which processes are using a file or file structure. You can specify either the device file or the mount point (when using the mount point also add the **-c** option):

```
# fuser -u /dev/vg01/myfs1
```

This lists process IDs and login names of processes using **/dev/vg01/lvol1**.

```
# fuser -u /etc/passwd
```

This lists process IDs and login names of processes that have the **passwd** file open.

```
# fuser -uc /opt
```

This lists process IDs and login names of processes that have open files in the **/opt** file system.

```
# fuser -ku /dev/vg01/myfs1
```

This terminates all processes that are preventing logical volume **lvol1** of volume group **vg01** from being unmounted, listing the process ID and login name of each as it is killed.

Always umount all mounted file systems *before* bringing the system down or you may cause corruption to the file systems. The **umount -a** command will unmount all the mounted file systems. The **shutdown** script unmounts all file systems before bringing the system down.

You cannot **umount** the root file system.

You cannot **umount** a file system that has file system swap enabled without rebooting the system.

10-6. SLIDE: Automatically Mounting File Systems

Automatically Mounting File Systems

- Place an entry in the **/etc/fstab** file
- File systems will be mounted when the system is booted, or you can use **mount -a** or **mountall**
- You can mount file systems by absolute directory names.

Sample **/etc/fstab**:

```
/dev/vg00/lvol3 /      vxfs delaylog 0 1
/dev/vg00/lvol1 /stand hfs defaults 0 1
/dev/vg00/lvol4 /home  vxfs delaylog 0 2
/dev/vg00/lvol5 /opt   vxfs delaylog 0 2
/dev/vg00/lvol6 /tmp   vxfs delaylog 0 2
/dev/vg00/lvol7 /usr   vxfs delaylog 0 2
/dev/vg00/lvol8 /var   vxfs delaylog 0 2
/dev/vg01/myfs1 /myfs1 vxfs delaylog 0 2
/dev/vg01/myfs2 /myfs2 hfs defaults 0 2
```

a66943

Student Notes

All file systems are unmounted during system shutdown. Any file systems that you wish to mount automatically after the next system reboot should be added to the **/etc/fstab** file. During the boot process, the **/sbin/init.d/localmount** script executes the **mount -a** command, which automatically mounts file systems listed in **/etc/fstab**. This configuration file is not automatically maintained by the system; it should be manually edited when file systems are created and removed.

After adding a file system to **/etc/fstab**, you needn't enter the full form of the **mount** command when mounting the new file system. Look at the following examples:

```
# mount -a                      # mount all file systems in /etc/fstab
# mount /myfs2                   # mount /myfs2 - no need to name the logical volume
```

```
# mount /dev/vg01/myfs2    # mount /dev/vg01/myfs2 - no need to name the mt. point
```

Syntax for /etc/fstab

Fields in the /etc/fstab file are

<i>block</i>	the block device file that corresponds to the mounted file system
<i>directory</i>	the directory to which mount mounts the device
<i>type</i>	the file system type. Types include:
	<ul style="list-style-type: none"> • cdfs - local CD-ROM file system. • hfs - high-performance (McKusick) file system. • nfs - network or remote file system. • vxfs - journaled file system. • swap - the device file name is made available as a piece of swap space by the swapon command. • swapfs - the file system which <i>directory</i> resides in is made available as swap space by the swapon command. • lofs - the file system is a loopback file system. • ignore - marks unused sections (on multi-file system disks).
<i>options</i>	a comma-delimited list of options used by mount (1M) and swap (1M) . Examples are: <ul style="list-style-type: none"> • defaults – Sets options rw, suid and noquota. When used, this must be the only option specified. You may not specify additional options along with defaults. • rw (default) – read/write • ro - read only • suid (default) – set user-id allowed • nosuid – no set user-id allowed • quota – enables checking of disk quota on this file system • noquota (default) - no quota checking on this file system
<i>backup-frequency</i>	reserved for possible use by future backup utilities.
<i>pass-number</i>	used by the fsck command to determine the order in which file system checks are done.
<i>comment</i>	a comment field (must be preceded by a #)

NOTE:

A more detailed description of the /etc/fstab file syntax can be found in the **fstab(4)** man page. Also take a look at the man pages for **mount**: **mount (1m)**, **mount_hfs (1m)**, and **mount_vxfs (1m)**.

10-7. SLIDE: CD-ROM File Systems (CDFS)

CD-ROM File Systems (CDFS)

- Allows mass distribution and easy retrieval of large amounts of information.
- You can read data from a CD, but you cannot write to it.
- To use CDFS volumes:
 1. Configure the appropriate driver into the kernel.
 2. Create the device files (if necessary).
 3. Mount the CDFS volume with the **mount** command.
- CDFS Examples:
 1. LaserRom Manual
 2. Application CD

a6977

Student Notes

CD-ROMs are becoming a popular media choice since they can store a large volume of data inexpensively. They are read-only, however. CDs are prepared and mastered using a special publishing process and can not be changed. Today, most HP-UX applications, utilities, and documentation are available in electronic format on CDs.

Using a CD-ROM containing a CDFS file system requires several steps. (CD-ROMs are available for HP-IB and SCSI interfaces.) To start, make sure that the CD-ROM drive is properly connected and configured, and that the proper driver is configured in your kernel. Then, shut down the computer, connect the CD-ROM drive, and power back up again.

Assuming the proper device driver has been installed, the system should create the necessary device files during the boot process.

After booting, use any mount point directory name (this example uses **/cdrom**).

Do the following:

```
# ioscan -funC disk          # find the block device file for your CD-ROM
# mkdir /cdrom               # create a mount point directory
# mount /dev/dsk/c1t3d0 /cdrom # mount the CD
```

CDFS file systems can be included in the **/etc/fstab** file for automatic mounting at boot much like any other file system (note that the **ro** in the mount options field stands for read-only):

```
/dev/dsk/c1t3d0 /cdrom cdfs ro 0 0 # local CD-ROM drive
```

Once mounted, CD-ROM file systems can be accessed using the same HP-UX file system commands used to navigate an HFS or JFS file system.

Converting File Names

CD-ROMs come in several formats. ISO9600 is the standard typically used for PC CD-ROMs. Another commonly used standard is High Sierra format. ISO9600 and High Sierra file names are all uppercase 8.3;1 format, so they look like FOO.EXE;1. This format may be inconvenient to work with in an HP-UX environment. Remember the shell interprets a ";" as a command separator. You can convert these file names by using the **-o cdcase** option on the **mount** command. This option suppresses the display of version numbers and shows and matches file names as lower case.

```
mount -F cdfs -o cdcase /dev/dsk/c0t2d0 /cdrom
```

The Portable File System (PFS) provides other conversion options as well as the capability to export to remote PFS clients. Refer to the **pfs(4)** and **pfs_mount(1M)** man pages for more details.

10-8. LAB: Creating File Systems

Directions

Record the commands used to complete the tasks below, and answer all of the questions.

Part I: Preliminary Steps

1. PART I

The exercises in this lab assume that you already have the following three logical volumes:

```
/dev/vg01/data  
/dev/vg01/app  
/dev/vg01/tables
```

If you already have these logical volumes, you can skip ahead to Part II of the lab. Otherwise, create the **vg01** volume group and all three of the logical volumes listed above. Make each of the logical volumes 12 MB.

```
# pvcreate /dev/rdsk/c0t5d0  
# mkdir /dev/vg01  
# mknod /dev/vg01/group c 64 0x010000  
# vgcreate vg01 /dev/dsk/c0t5d0  
# lvcreate -L 12 -n data vg01  
# lvcreate -L 12 -n app vg01  
# lvcreate -L 12 -n tables vg01  
# vgdisplay -v vg01
```

Part II: Creating File Systems from the Command Line

1. PART II

Create an HFS file system in the **data** logical volume. Create a JFS file system in the **app** logical volume. Don't mount your file systems yet.

2. Do a **mount -v**. Why don't your new file systems appear at this point?

3. Create a mount point for each of your new file systems. Use `/data` as the mount point for the file system in the `data` logical volume, and `/app` as the mount point for the `app` logical volume. Again, don't mount your file systems, yet.
4. Add your new file systems to the `/etc/fstab` file so they will be mounted automatically at each system boot. Again, don't mount your file systems, yet.
5. Go ahead and mount your file systems now by doing a `mount -a`. Watch the resulting messages carefully. You should see several error messages indicating that `/dev/vg00/lvol1` and several other file systems are already mounted. Do the `mount -a` output messages offer any indication that your new file systems were successfully mounted?
6. Execute `mount -a` a second time and note the output messages again. Why did `mount -a` mention your new file systems in its output this time, but not when you did a `mount -a` in the previous exercise?
7. Use `mount -v` to see what file systems are now mounted. Did your new file systems mount properly? What other information can you glean from the `mount -v` output about your mounted file systems? List three fields presented in the `mount -v` output.

Part III: Experimenting with mount and umount

1. PART III

The exercises in this section will be more meaningful if you have some real files in the `/data` and `/app` file systems. Towards that end, create a few files in these file systems using the following commands:

```
# cd /data; touch d1 d2 d3  
# cd /app; touch a1 a2 a3  
# ls /data /app  
# cd /
```

NOTE:

You may notice a `lost+found` directory in your new file systems. `newfs` creates this directory for you automatically. This special directory serves as a home for files that are victims of file system corruption. This directory will be discussed in some detail in a later chapter.

2. Can you access files in a file system that is unmounted? Try an experiment to find out: Unmount `/data`, then do an `ls` of `/data`. Does the mount point still exist? Can you access `/data/d1` and `/data/d2`? Why or why not? Do whatever is necessary to regain access to `/data/d1` and `/data/d2`.
3. Can you unmount a file system that is still in use? Try an experiment to find out: Open a second window on your screen. In this second window, `cd` to the `/data` file system. Back in your original window try unmounting the `/data` file system. What message do you get? Why?
4. Before you can unmount a file system, you will have to kill all processes accessing the file system. HP-UX provides a command to solve this very problem. Try the following command:

```
# fuser -u /dev/vg01/data
```

Each entry in the **fuser** output lists the PID of a process accessing the file system, a single letter code indicating how the process is using the file system ("c" indicates that a user has changed to a directory in the file system), and the name of the user that owns the offending process. **fuser** can also kill all of the offending processes:

```
# fuser -ku /dev/vg01/data
```

The effect of this command should be pretty dramatic. What happens? Now try unmounting **/data**.

5. What happens if you mount a file system on a directory that already contains files? Try an experiment to find out:

While the **/dev/vg01/data** file system is still unmounted, touch a few files under the **/data** mount point:

```
# touch /data/junk1 /data/junk2
# ls /data
```

Now mount the **/dev/vg01/data** file system and list the contents of **/data** again:

```
# mount /data
# ls /data
```

Do you still see **junk1** and **junk2**? Unmount **/data** again and check to see if **junk1** and **junk2** still exist. Why are **junk1** and **junk2** hidden while the data file system is mounted? Explain.

NOTE: File systems should always be mounted on empty mount point directories.

6. Remove the **junk** files you created in the previous exercise, and remount all of your file systems again.

7. One last experiment: Can you unmount all of your file systems? Try a **umount -a** and explain the result.

8. Remount all of your file systems before continuing.

Part IV: Experimenting with newfs

1. PART IV

HP-UX supports two different types of disk-based file systems: HFS and JFS. In all of the examples you have tried so far, you have specified your desired file system type via the **-F** option. What happens if you forget the **-F**? Try it: do a **newfs** on the **/dev/vg01/tables** logical volume, but leave off the **-F** option. Look at the **newfs** output carefully. How does the system determine your default file system type? What is your default file system type?

2. The **newfs** command is a powerful tool for creating file systems, but it must also be treated with some respect. What happens if you accidentally **newfs** a logical volume or disk that already contains a file system? Try an experiment to find out: What happens if you attempt to create a new **hfs** file system on the data logical volume with the following command:

```
# newfs -F hfs /dev/vg01/rdata
```

3. What happens if you try to **newfs** a logical volume that contains an unmounted file system? Try it.

```
# umount /data
# newfs -F hfs /dev/vg01/rdata
# mount /data
# ls /data
```

Did you encounter any problems? What happened to the **d1**, **d2**, **d3** files you created earlier? Always use **newfs** with care.

Part V: Using SAM to Create Logical Volumes and File Systems

1. PART V

So far, you have created several file systems from the command line. File systems can be created and managed within SAM as well. Open SAM, then do the following:

1. Select: **SAM --> Disks and file systems --> Logical volumes**
2. Select: **Actions --> Create**
3. Click: **Select a Volume Group** (Choose **vg01**)
4. Click: **Define New Logical Volumes**
5. Fill in the blanks in the dialog box that appears:

LV name: app2
LV size (MB): 12
Usage: File System
Mount Dir: /app2

6. Click: **Modify FS defaults**.
7. Fill in the blanks in the dialog box that appears:

- a. Create a Journaled file system.
- b. Choose **Now AND Every system boot** options under **When to mount**.
- c. For now, take the defaults for the rest of the screen, and click **OK**.
8. You will be returned to the **Define new logical volumes** screen.
9. Click **Add** to add your new logical volume to the list of new LVs to create.
10. Click **OK** to go back to the **Create new logical volumes** step menu.
11. Click **OK** to create the new logical volume.

2. Do a **mount -v** from the command line. Is the new file system mounted?

3. Look at **/etc/fstab**. Did SAM ensure that the new file system will be mounted with every system boot?

Part VI: Cleaning Up

1. PART VI

Before moving on to the next chapter, use SAM to remove the file systems and logical volumes you created during this lab exercise:

1. Select: **SAM --> Disks and file systems --> Logical volumes**
2. While holding down **CTRL**, select each of the **vg01** LVs. *Do not remove any of the logical volumes in vg00.*
3. Select: **Actions --> Remove**
4. When asked for confirmation, answer **Yes**.

2. What did SAM do on your behalf?

Are the file systems unmounted?

Are the file systems still listed in **/etc/fstab**?

If you do a **vgdisplay**, do the logical volumes still appear?

Do the mount points still exist?

3. You should also remove volume group **vg01** before continuing on to the next chapter.
Again, use SAM:

1. Select: **SAM --> Disks and file systems --> Volume groups**
2. Select **vg01**
3. Select: **Actions --> Remove**

Module 11 — File System Repair

Objectives

Upon completion of this module, you will be able to do the following:

- Describe how HFS/JFS handle file system updates.
- Describe how `sync` prevents file system corruption.
- List three causes of file system corruption.
- Check and repair an HFS file system with `fsck`.
- Check and repair a JFS file system with `fsck`.

11-1. SLIDE: File System Maintenance

File System Maintenance

- Routine Maintenance
 - Check file system integrity
 - Employ regular backup procedures
 - Monitor disk usage

a53862

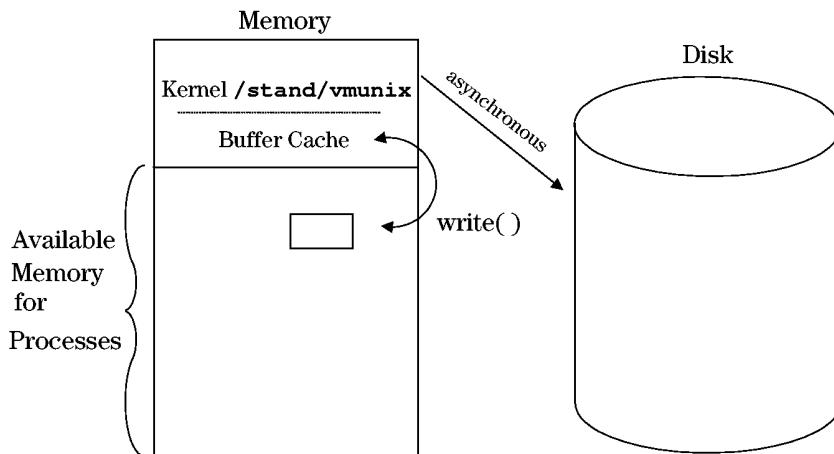
Student Notes

One of the principal responsibilities of a system administrator is preservation of the user's data. Since the data storage structure utilized by HP-UX is the file system, it's imperative that the storage environment of the file system be checked regularly for possible problems. The integrity of the file system must never be compromised. There are many things the administrator can do to help maintain the integrity of the file systems.

In this module we will concentrate on checking and maintaining file system integrity using the `fsck` utility. Before we look at `fsck`, let's review how file system updates occur.

11-2. SLIDE: File System Updates

File System Updates



a66945

Student Notes

When a file system is mounted, its superblock is copied into RAM. When this occurs the file system flag is set to "dirty". All updates to the superblock are first made to the core copy. When a `sync` system call or command is issued, the disk copy is updated. When a file system is unmounted, all in-core structures are flushed out to disk and the file system flag is set to "clean".

All changes to metadata are made first to the in-memory copy and are then written to disk. Some metadata changes are written to disk immediately, others will be written to disk when a `sync` is issued.

Most file system transactions result in several pieces of metadata being updated. For example the command `rm myfile` will result in the following changes:

1. The directory entry for `myfile` must be cleared.
2. The inode that was being used to describe `myfile` must be freed.
3. The maps that keep track of free data blocks and free inodes must be updated.
4. The free count of data blocks and inodes in the superblock must be updated.

Unfortunately, all the metadata is not stored on disk contiguously, so it will take several writes to complete this transaction. If the system were to crash part way through there would be inconsistencies in the metadata. For example, if the directory entry for `myfile` was cleared, but the inode was not yet freed, the result would be an inode with a link count of 1, but with no directory entries pointing to it. This is inconsistent metadata.

The Buffer Cache

When user data needs to be written, the actual write does not occur immediately. The data is initially copied to an in-memory buffer called the buffer cache. This is a much faster operation than if the system had to perform an actual write to the disk. The data, along with the inode information, is written to the disk sometime later, usually when the buffer cache fills up and the system needs to clear some buffer space. If the system is halted without writing the buffer to disk, the file system could become corrupted. You should never allow work to continue if you suspect that the file system is corrupted.

Some of the benefits and advantages of having a buffer cache are

- The use of the buffer cache allows for uniform disk access, because the kernel does not need to know the reason for the I/O. The kernel just always writes buffers to the disk, not parts of buffers or real numbers. So system design is simpler, from a disk I/O standpoint.
- By using a buffer scheme, programs are more easily ported to other UNIX systems. Disk I/O may be different on different UNIX machines, but the programs don't have to know that. They simply write to buffers, without having to worry about how the disk is set up.
- Using a buffer cache reduces the amount of disk traffic, thereby increasing overall system throughput and decreasing response time. In other words the system runs faster.
- Re-use of data files in the buffer cache can also speed up a system.

JFS Intent Log

With JFS, all metadata updates required to complete a transaction are grouped together. The updates are first made to the in-core metadata. An intent log transaction record is created in memory, and then written to the JFS intent log. A transaction is not committed until the intent log record has been written. This prevents a transaction from being partially completed.

11-3. SLIDE: Flushing Buffer Cache

Flushing the Buffer Cache

sync (1m)

- Writes buffer contents to disk
- Keeps the file system current
- Is normally invoked on a regular basis by the **syncer** daemon

syncer (1m)

- **syncer** is started automatically at system boot.
- The syntax of the **syncer** program is:

```
syncer [ seconds ]
```

Question: What might prevent **syncer** from properly flushing the buffer cache?

a66946

Student Notes

As we have seen, data is written to an in-core buffer cache before it is written to disk. A physical write from the buffer to disk is delayed until:

- The system needs the buffer for another operation.
- The last byte of the block is modified.
- The file system is unmounted.
- The **sync** command is executed.
- The system is shut down or rebooted.

syncer (1m)

syncer is normally started from the **/sbin/init.d/syncer** script at system startup, and is responsible for automatically flushing the buffer cache periodically. You should not execute the

syncer command directly; it should be executed at system boot time via
`/sbin/init.d-syncer`.

sync (1m)

sync executes the system call **sync (2)**, which flushes all previously unwritten system buffers, including modified superblocks, modified inodes, and delayed block I/O, out to disk. This ensures that all file modifications are properly written to disk before performing a critical operation such as system shutdown. You can manually execute the **sync** command at any time, though a **sync** is automatically performed at regular intervals by the syncer daemon.

11-4. SLIDE: Introducing **fsck**

Introducing **fsck**

Why run **fsck**?

- Checks file system metadata consistency
- Repairs metadata corruption as needed

When should **fsck** be run?

- Runs automatically after improper shutdown
- Run manually whenever corruption is suspected

a66947

Student Notes

When an HP-UX system is shutdown improperly, pending file system changes may be lost or incomplete.

The **fsck (1m)** utility runs automatically after a system crash or improper shutdown to verify the structural integrity of your file systems. The utility attempts to correct any identified inconsistencies.

fsck runs automatically after an improper shutdown, but also may be run manually if you suspect file system corruption.

11-5. SLIDE: Running **fsck**

Running **fsck**

Example: Running **fsck** on **/dev/vg01/myfs2**

1. **mount -v**
2. **umount /dev/vg01/myfs2**
3. **fsck -F hfs /dev/vg01/rmyfs2**
4. **mount /dev/vg01/myfs2**
5. Restore any corrupted files:
 - Did **fsck** remove any files?
 - Did **fsck** reconnect any files?

a66948

Student Notes

Several steps are required to run **fsck**.

1. **mount -v**

Start by issuing the **mount -v** command to determine which file systems are mounted where. Also note which file systems are HFS and which are JFS. You will need to know the file system type when you run **fsck**.

2. **umount /dev/vg01/myfs1**
umount /dev/vg01/myfs2

fsck should be run on quiescent file systems; unmount the file system before proceeding.

3. **fsck -F vxfs /dev/vg01/rmyfs1**
fsck -F hfs /dev/vg01/rmyfs2

Run **fsck**. You can run **fsck** on both HFS and JFS file systems. Just be sure to specify the proper file system type. For optimal performance, specify the raw device file of the logical volume or disk containing the file system to check.

When run on a JFS file system, **fsck** simply replays the intent log, and completes any pending transactions. See the detailed explanation of JFS options in the notes below for more information. When run on an HFS file system, **fsck** examines the specified file system a number of times, examining a different feature of the file system with each pass. When **fsck** identifies a file system inconsistency, it reports the problem, and asks if corrective action should be taken.

If the administrator answers "yes," **fsck** attempts to fix the problem. If the administrator answers "no," **fsck** ignores the inconsistency and continues. There are very few occasions when a "no" response is appropriate.

4. `mount /dev/vg01/myfs1`
`mount /dev/vg01/myfs2`

Once **fsck** completes, remount the file system(s).

5. Restore any corrupted files.

In order to fix file system corruption, **fsck** may have to remove one or more files. Watch for "REMOVE" messages in the **fsck** output, and be sure to restore the affected files from tape.

fsck may also RECONNECT orphaned files. If you see any RECONNECT messages, check the file system's **lost+found** directory. The next slide discusses the **lost+found** directory in greater detail.

Some General **fsck** Options

There are several common options that may be used on **fsck** when checking either HFS or JFS file systems:

- n Assume a "no" response to all questions from **fsck**; **fsck** checks the file system but doesn't correct inconsistencies. You can use this option to assess the state of a file system, but then be sure to run **fsck** again to actually fix any identified problems.
- y Assume a "yes" response to all questions. Beware that data may be removed as a result of a "yes" answer. Consider using this option after running **fsck** with "n" to assess the state of your file systems.

Some Special HFS **fsck** Options

The following **fsck** options only apply to HFS file systems:

- b *block#* This option tells **fsck** to use the superblock at the specified block# as the superblock for the file system check instead of the default superblock. This is useful if the primary superblock becomes lost or corrupted. All backup superblock locations are written to the **/var/adm/sstab** file. If you can't access this file, try block #16, which always contains the first alternate

superblock. If you seem to be getting a lot of strange errors from **fsck**, try using this **-b** option.

-f

By default, **fsck** gives a warning message and requests confirmation when run against a mounted file system. **-f** forces **fsck** to run, even if the specified file system is mounted. This option should only be used in single-user mode. After running **fsck**, reboot immediately with the **-n** option. By default, the reboot and shutdown commands flush buffer cache, which may overwrite the corrections made by **fsck**. **-n** reboots without flushing buffer cache.

Some Special JFS **fsck** Options

Because of the JFS intent log mechanism, JFS file system metadata should not be corrupted by an improper shutdown. After an improper shutdown, **fsck** need only complete any pending intent log transactions to bring a JFS file system to a consistent state. This is known as an intent log replay. **fsck** requires minutes or even hours to repair an HFS file system, but can complete a JFS intent log replay in a matter of seconds.

If you wish, you can force **fsck** to do a full check of all the metadata structures in a JFS file system with the **-o full** option.

The **-o nolog** option prevents an intent log replay. This may be useful if the intent log area is physically damaged.

Sample **fsck** Output

```
# mount -v
/dev/vg01/myfs1 on /myfs1 type vxfs delaylog on Fri Jun 26 15:55:10 1998
/dev/vg01/myfs2 on /myfs2 type hfs defaults on Fri Jun 26 15:18:55 1998

# umount /myfs1
# umount /myfs2

# fsck -F vxfs /dev/vg01/rmyfs1
file system is clean - log replay is not required

# fsck -F hfs /dev/vg01/rmyfs2
** /dev/vg01/rmyfs2
** Last Mounted on /myfs2
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
UNALLOCATED I=397
OWNER=root MODE=0
SIZE=0 MTIME=Dec 31 18:00 1969
NAME=/data1/last

REMOVE? y
UNALLOCATED I=397
OWNER=root MODE=0
SIZE=0 MTIME=Dec 31 18:00 1969
NAME=/data1/lastb
REMOVE? y
```

```
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
FREE INODE COUNT WRONG IN SUPERBLK
FIX? y

** Phase 5 - Check Cyl groups
20 BLK(S) MISSING
BAD CYLINDER GROUPS
FIX? y

** Phase 6 - Salvage Cylinder Groups
110 files, 0 icont, 13987 used, 33842 free (50 frags, 4224 blocks)

***** FILE SYSTEM WAS MODIFIED *****

# mount /myfs1
# mount /myfs2
```

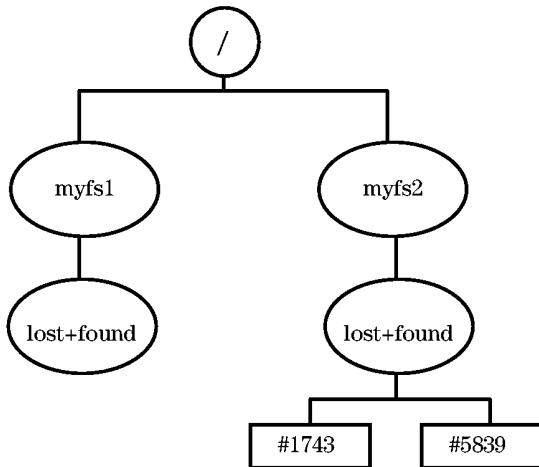
NOTE:

There are several **fsck** man pages:

- **fsck(1m)** describes general fsck features
 - **fsck_hfs(1m)** describes HFS-specific fsck features.
 - **fsck_vxfs(1m)** describes JFS-specific fsck features.
-

11-6. SLIDE: Checking `lost+found`

Checking `lost+found`



- Exists in every file system
- Orphaned files copied to this directory by `fsck`
- Should be checked after every `fsck`

a66949

Student Notes

Every file system should have a `lost+found` directory at the root of its file system. The `lost+found` directory is created by `newfs`. However, you should verify that the directory exists before using `fsck` to check the file system. If `lost+found` does not exist, you can rebuild it with the `/usr/sbin/mklost+found` command.

`fsck` places any problem files or directories in the `lost+found` directory. After `fsck` completes, you should examine the contents of the directory. The files that are placed there should be moved back to their original directories. The name assigned is the inode number, so it may be difficult (and sometimes impossible) to determine where the files actually belong, but you should try to find owners. Run the `file` command on a file. If the file contains text, look at its contents to try and determine the owner. If the file contains executable code check to see if it has an SCCS identification string. If it does, the `what` command will list SCCS identification information. If the file does not have an SCCS identification string, use the `strings` command to print the literal strings from the file. These strings may help identify the owner. Do not execute an executable file found in the `lost+found` directory to try and figure out what it is. It may be the program that corrupted the disk.

Examples

```
# cd /myfs2/lost+found  
# ll \#1743  
# file \#1743  
# strings \#1743  
# mv \#1743 new_file_name
```

11-7. LAB: **fsck**

Directions

In the first portion of the lab, you will have an opportunity to run **fsck** on a number of corrupted file systems via an **fsck** simulation program. You may run the simulator by typing: **/labs/simfsck**. The **simfsck** simulator contains five file systems:

```
/dev/vg00/lvol1  HFS      /
/dev/vg01/app     HFS      /app
/dev/vg01/cad     HFS      /cad
/dev/vg02/db       HFS      /db
/dev/vg02/data    VXFS     /data
```

The simulator recognizes all of the UNIX commands necessary to run **fsck** on the above file systems. Note that the simulator only recognizes the most common options on each of these commands.

```
bdf
cat
cd
file
fsck
ls
ll
mount
pwd
strings
umount
```

Run the simulations suggested below. Record the commands you use, and answer all the questions. After gaining some experience with **fsck** in the simulator, the last few exercises ask you to run **fsck** on some "real" file systems as well.

Part I: Running **fsck** on an HFS File System

1. PART I

Once in the simulator, choose simulation #1 from the menu. Simulation #1 asks you to run **fsck** on the **/app** file system.

2. You will need to know the name of the logical volume containing the /app file system when you run **fsck**, so start by viewing a list of the currently mounted file systems.

3. File systems must be unmounted before running **fsck**. Unmount the /app file system.

4. Now run **fsck** on the raw logical volume containing /app. Be sure to specify the file system type. **fsck** will check the file system and ask which problems should be corrected.

Although one could conceivably answer "no" in response to **fsck**'s questions, you should generally answer "yes" – otherwise you will be left with a still-corrupted file system. Take note of the problems identified by **fsck**, but go ahead and answer "yes" to all of the prompts.

5. Once **fsck** terminates, remount the file system.

6. After mounting the file system, you should check the **lost+found** directory to see if **fsck** identified any orphaned files. Are there any orphaned files in this case?

7. Did the **fsck** prompts indicate that **fsck** "REMOVE'd any files from /app? If you have a tape backup of the file system, you should restore these files at this point. In the case of our simulator, however, we don't have a tape backup so any removed files are lost forever.

8. Go back to the main simulator by typing: "menu".

Part II: Another Corrupted HFS File System

1. PART II

Choose simulation #2, which asks you to run **fsck** on the **/db** file system. Enter choice: 2

2. Following the same procedure used in the previous exercise, **fsck** the **/db** file system and fix all the corruption identified.
3. Are there any files in **/db/lost+found** that need attention? If there are any files in **lost+found**, what can you do to find the file's owner and filename?
4. Did **fsck** "REMOVE" any files this time?
5. Return to the simulator menu by typing "menu".

Sim: menu

Part III: Running `fsck` on a JFS File System

1. PART III

Run simulation #3, which asks you to run `fsck` on the `/data` file system. Enter choice: 3

2. What happens if you specify the wrong file system type to `fsck`? Unmount the data file system, and try running `fsck -F hfs /dev/vg02/rdata`. What happens? Despite the worrisome message `fsck` offers here, the superblock really isn't damaged. Can you guess why you get this message?
3. Try running `fsck` again with the `-F vxfs` option. How does running `fsck` on a `vxfs` file system differ from running `fsck` on an `hfs` file system?
4. After running `fsck`, remount the file system and return to the main simulator menu.

Part IV: Another JFS File System

1. PART IV

Run simulation #4, which asks you to run a "full" `fsck` on the `/data` file system. Menu choice: 4

2. You saw in the previous simulation that, by default, **fsck** simply replays the JFS intent log. After a system crash, an intent log replay is all that is required. An intent log replay simply scans the intent log completes any pending transactions; it does not check the consistency of your superblock, inodes, and allocation units.

If you suspect more serious file system corruption in a JFS file system, you can perform a full check of the file system. Try the following:

```
Sim: mount -v
Sim: umount /data
Sim: fsck -F vxfs -o full /dev/vg02/rdata
Sim: mount /data
Sim: ll /data/lost+found
```

3. How did this **fsck** differ from the **fsck** in the previous JFS simulation?

Part V: Automatically Running **fsck at Boot**

1. PART V

In all of the examples tried thus far, you have manually run **fsck** to check and repair your file systems. However, the system automatically runs **fsck** for you each time you boot. Simulation #5 will show you console messages that appear while booting an E35 server. Run the simulation and watch the console messages that follow. Enter choice: 5

2. Did the system **fsck** all of your file systems?

3. Every file system superblock contains a "file system clean flag". When a file system is mounted, it is marked "dirty." Properly unmounting the file system with `umount` toggles the flag back to a "clean" state. The `fsclean` utility that runs during the boot process only `fsck`'s dirty file systems. Why might a file system be left in a dirty state?
4. Did `fsck` fix any problems in your "dirty" file system?
5. Do a `mount -v` to ensure that all of your file systems are properly mounted after the boot.

Part VI: Another Boot Simulation

1. PART VI

Try running simulation #6, another boot simulation. Start the boot simulation, and watch the console messages carefully. Enter choice: 6

2. In this simulation, `fsck` again automatically identified and corrected several minor problems in the `/app` file system during the boot process. However, `fsck` also identified a problem that requires removal of a file. When running automatically during the boot process, `fsck` will never make any repairs that cause loss of data. When `fsck` identifies problems that may require removal of data, you will be prompted to run `fsck` manually. Which file system requires a manual `fsck` this time?

3. Run `fsck` on `/app` and fix any identified problems.

4. When you complete the `fsck`, allow the boot process to complete. Return to the main menu, then exit out of the simulator.

Part VII: Running `fsck` on "Real" File Systems

1. PART VII

Now that you have had an opportunity to run `fsck` in the simulator a few times, try running `fsck` on some real file systems. Do a `mount -v` to determine which file systems are mounted where on your system.

2. You should have an HFS file system mounted on `/stand`, and a JFS file system mounted on `/home`. Unmount both of these file systems, and run the appropriate `fsck` commands to check them both. Does `fsck` identify any problems?

3. If you haven't already remounted your file systems, do so now.

Module 12 — File System Management

Objectives

Upon completion of this module, you will be able to do the following:

- Monitor space available in the file system with `bdf` and `du`.
- Clean up the file systems by purging unused files and core files.
- Clean up the `/var` file system by trimming log files.
- Extend a volume group from the command line.
- Extend a logical volume from the command line.
- Extend a file system from the command line.

12-1. SLIDE: Monitoring Disk Usage

Monitoring Disk Usage

Monitor available file system space with **bdf**:

```
# bdf
Filesystem      kbytes    used  avail   %used Mounted on
/dev/vg00/lvol5 294912  261523 31352    89%   /opt
/dev/vg00/lvol4  24576   19333  4978    80%   /home
/dev/vg01/myfs1  16384   1174   14331    8%    /myfs1
/dev/vg01/myfs2  15893   14006  297     98%   /myfs2
```

Determine space used by directory subtrees with **du**:

```
# du -sk /myfs2/*
844      /myfs2/data1
1327     /myfs2/data2
1073     /myfs2/data3
10757    /myfs2/data4
4        /myfs2/lost+found
```

a66950

Student Notes

The system administrator is responsible for monitoring the amount of free disk space on the system. The easiest way to do this is with the **bdf** command. The fields have the following meaning:

Filesystem	Block device file of the file system
kbytes	The number of kilobytes of total disk space on the file system
used	The number of kilobytes of disk space used by existing files
avail	The number of kilobytes of available disk space on the file system
capacity	The percentage of disk space used by files
Mounted on	Directory to which the indicated file system is mounted

The **-i** option adds three columns to the output that give information relating to the availability of inodes in the file system.

iused Number of inodes currently in use on the file system

ifree Number of free inodes on the file system

iused Percentage of inodes used on the file system

When you want to see more details, for example how much space is used beneath a directory, you can do this with the **du** command. By default, **du** shows the amount of space in blocks of 512 bytes. It is recursive, meaning that it starts at the current directory (or *file* specified) and reports on all files and directories from that point on down.

The main options are:

-k Report output in kilobytes.

-s Print only the grand total of disk usage for each of the specified *directory* operands.

For further information refer to the **du(1)** manual entry.

12-2. SLIDE: Routine Management

Routine Management

- Trim log files that grow without bound:
- Remove core files
- Remove large, old files
- Extend a file system
 - into existing free space
 - onto a new disk in the volume group

a66951

Student Notes

Disk space is often at a premium. The System Administrator should monitor disk free space regularly, and take steps to prevent a situation of running out of disk space. There are some proactive measures that the System Administrator can take, including monitoring files that continuously grow, removing core files, trimming log files, and removing or archiving large files that have not been used in a long time.

Trimming Log Files

The `/var` file system is often among the first to cause "file system full" messages. `/var` contains system log files and spool files which can quickly fill the file system if they aren't monitored carefully.

SAM has an intuitive interface for interactively trimming log files to reclaim disk space:

`sam` → Routine Tasks → System Log Files, select a log file

The SAM interface lets you easily trim selected system log files to a recommended size, a certain number of lines, a percentage of the current size, or size zero. Though SAM initially only lists HP-UX system log files, other application log files may be added to SAM's list as well.

Although the SAM interface provides a convenient mechanism for interactively trimming log files, you may want to schedule a job to automatically trim log files on a regular basis. The examples below truncate the **wtmp** and **btmp** log files to zero length from the command line:

```
# > /var/adm/btmp
# > /var/adm/wtmp
```

These commands can be scheduled for automatic periodic execution via the **cron** daemon.

NOTE: Never empty a log file with:

```
# rm logfile
# touch logfile
```

Doing so may not set the proper permissions for the logfile.

Removing Core Files

A "core" file is occasionally created when a process terminates abnormally as a result of a serious error condition or the QUIT signal. A core file resulting from either of these situations contains a core image of the terminated process. Programmers may use this to determine what the process was doing at the time of termination.

Core files can be quite large, and should be removed from the system when no longer needed. You can easily find and remove core files on the system with the **find** command:

```
# find / -name core -exec l1 {} \;      # list all core files
# find / -name core -exec rm {} \;      # remove all core files
```

You may wish to schedule the **find** command to execute periodically via cron. You can also use SAM to interactively find and remove core files:

sam -> Routine Tasks -> Selective File Removal

Large, Old Files

Often, users create large files, then forget to remove the files when they are no longer needed. You should periodically search for large files on your system that haven't been recently accessed, and determine if the files are still needed. Removing or archiving large, unneeded files may reclaim a significant amount of disk space.

The **find** command can be used for this purpose. The example below finds and lists all files in **/tmp** that are over 1000 characters in length, and haven't been accessed in at least 30 days:

```
# find /tmp -atime +30 -size +1000c -exec ll -ud {} \\;
```

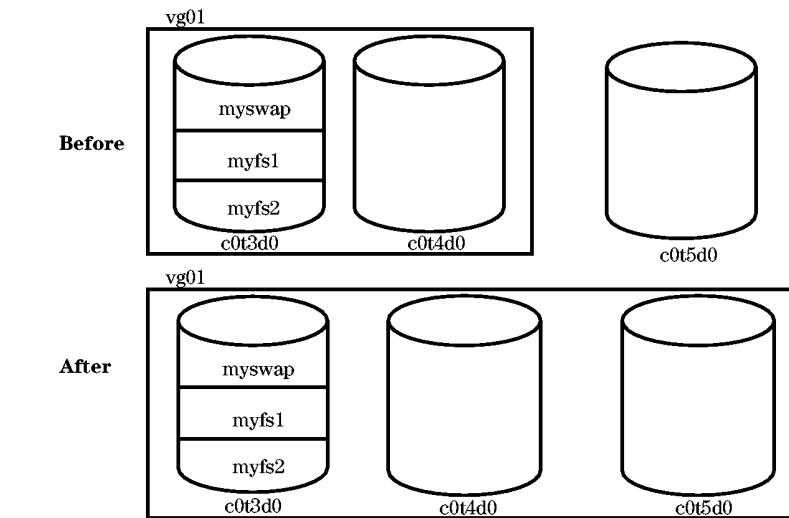
Print the resulting list of files, then contact the file owners to determine which files may be removed.

Extending a File System

If a file system reaches 100% of capacity, you may need to add additional space. If the file system's volume group already has some free physical extents, you may extend the file system to take advantage of those extents. If all of the physical extents in the volume group have already been allocated to other logical volumes, you may need to add a disk to the volume group.

12-3. SLIDE: Extending a Volume Group

Extending a Volume Group



```
pvcreate /dev/rdsk/c0t5d0
vgextend vg01 /dev/dsk/c0t5d0
vgdisplay -v
```

a66952

Student Notes

In order to extend a file system, it may first be necessary to add a disk to the volume group containing that file system's logical volume.

Adding a Disk to a Volume Group

Adding a disk to a volume group is a two-step process. First, you must **pvcreate** the new disk to create the necessary LVM data structures. After creating the necessary LVM data structures on the disk with **pvcreate**, you can add the disk to an existing volume group with **vgextend**. The example on the slide adds disk **c0t5d0** to volume group **vg01**:

```
# pvcreate /dev/rdsk/c0t5d0
Physical volume "/dev/rdsk/c0t5d0" has been successfully created.

# vgextend vg01 /dev/dsk/c0t5d0
Volume group "vg01" has been successfully extended.
Volume Group configuration for /dev/vg01 has been saved in
```

```
/etc/lvmconf/vg01.conf
```

The **vgextend** command accepts multiple physical volumes as arguments, if you need to add multiple disks to the volume group.

Checking the Volume Group Configuration

You can check to ensure that the disk was successfully added to the volume group with the **pvdisplay** and **vgdisplay** commands. Check to make sure the new physical volume is included in the **vgdisplay -v** physical volume list, and note the "VG Name" field in the **pvdisplay** output.

```
# vgdisplay -v vg01
--- Volume groups ---
VG Name                  /dev/vg01
VG Write Access          read/write
VG Status                available
Max LV                  255
Cur LV                  3
Open LV                 3
Max PV                  16
Cur PV                  3
Act PV                  3
Max PE per PV           1016
VGDA                     6
PE Size (Mbytes)        4
Total PE                1011
Alloc PE                16
Free PE                 995
Total PVG               0
Total Spare PVs          0
Total Spare PVs in use  0
--- Logical volumes ---
... LIST OF vg01's LOGICAL VOLUMES ...
--- Physical volumes ---
PV Name                  /dev/dsk/c0t3d0

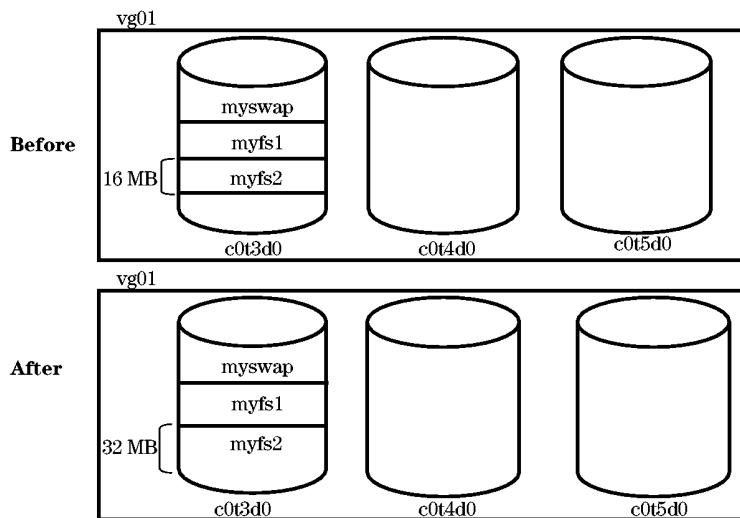
PV Status                available
Total PE                 250
Free PE                  234
PV Name                  /dev/dsk/c0t4d0
PV Status                available
Total PE                 250
Free PE                  250
PV Name                  /dev/dsk/c0t5d0
PV Status                available
Total PE                 511
Free PE                  511

# pvdisplay /dev/dsk/c0t5d0
--- Physical volumes ---
PV Name                  /dev/dsk/c0t3d0
VG Name                  /dev/vg01
```

PV Status	available
Allocatable	yes
VGDA	2
Cur LV	3
PE Size (Mbytes)	4
Total PE	511
Free PE	511
Allocated PE	0
Stale PE	0
IO Timeout	default

12-4. SLIDE: Extending a Logical Volume

Extending a Logical Volume



```
lvextend -L 32 /dev/vg01/myfs2 /dev/dsk/c0t3d0  
lvdisplay -v /dev/vg01/myfs2
```

a66953

Student Notes

After adding a disk to a volume group, you can allocate the physical extents from the new disk to logical volumes within the volume group. Extending a logical volume requires just one command: `lvextend`.

The example on the slide extends the `/dev/vg01/myfs2` logical volume from 16 MB to 32 MB. The `/dev/dsk/c0t3d0` argument on the end of `lvextend` forces LVM to allocate the new physical extents from disk `c0t3d0`. If you don't specify where LVM should allocate the new extents, LVM simply uses the first available extents in the volume group. After extending the logical volume, it's a good idea to do an `lvdisplay` to see the result.

```
# lvextend -L 32 /dev/vg01/myfs2 /dev/dsk/c0t3d0
Logical volume "/dev/vg01/myfs2" has been successfully extended.
Volume Group configuration for /dev/vg01 has been saved in
/etc/lvmconf/vg01.conf

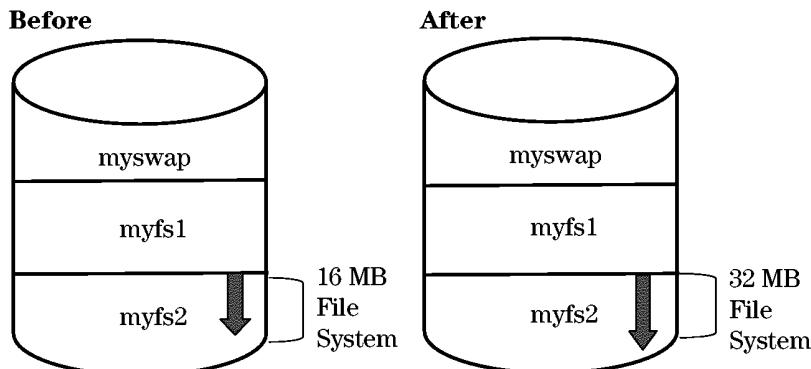
# lvdisplay -v /dev/vg01/myfs2
--- Logical volumes ---
LV Name           /dev/vg01/myfs2
VG Name           /dev/vg01
LV Permission     read/write
LV Status         available/syncd
Mirror copies    0
Consistency Recovery MWC
Schedule          parallel
LV Size (Mbytes) 32
Current LE        8
Allocated PE      8
Stripes           0
Stripe Size (Kbytes) 0
Bad block         on
Allocation        strict
IO Timeout (Seconds) default
--- Distribution of logical volume ---
PV Name           LE on PV  PE on PV
/dev/dsk/c0t3d0   8
                 8
--- Logical extents ---
LE   PV1           PE1  Status 1
0000 /dev/dsk/c0t3d0 0008 current
0001 /dev/dsk/c0t3d0 0009 current
0002 /dev/dsk/c0t3d0 0010 current
0003 /dev/dsk/c0t3d0 0011 current
0004 /dev/dsk/c0t3d0 0012 current
0005 /dev/dsk/c0t3d0 0013 current
0006 /dev/dsk/c0t3d0 0014 current
0007 /dev/dsk/c0t3d0 0015 current
```

NOTE: A logical volume can span multiple physical volumes, but it cannot span multiple volume groups. A logical volume can only be extended to other disks in the logical volume's volume group.

NOTE: Simply lvextend'ing a logical volume does not extend the file system within a logical volume. Turn to the next slide to learn how to make the new space in the logical volume available to the file system.

12-5. SLIDE: Extending a File System

Extending a File System



```
# umount /myfs2
# extendfs /dev/vg01/rmyfs2
# mount /myfs2
# bdf
```

a6978

Student Notes

Simply extending a logical volume does not make the new space available for use by the file system in the logical volume. The new space in the logical volume won't be used until the file system's superblock, and other metadata structures have been notified that new space is available.

Extending an HFS File System

In order to extend an HFS file system, you must temporarily unmount the file system. After unmounting the file system, use `extendfs` to make the additional extents available for use by the file system, then remount and check your file system with `bdf`.

Example: (Assume that `/dev/vg01/myfs2` has been lvextended to 32 MB)

```
# umount /myfs2
# extendfs -F hfs /dev/vg01/rmyfs2
# mount /myfs2
# bdf /myfs2
```

The entire `/dev/vg01/myfs2` logical volume is now available for use by the `/myfs2` file system.

Extending a JFS File System (With On-line JFS)

If you don't have the On-line JFS product, a JFS file system must be extended just like an HFS file system; simply change the file system type flag following the `-F`.

Example: (Assume that `/dev/vg01/myfs1` has been lvextended to 32 MB.)

```
# umount /myfs1
# extendfs -F vxfs /dev/vg01/rmyfs1
# mount /myfs1
# bdf /myfs1
```

The entire `/dev/vg01/myfs1` logical volume is now available for use by the `/myfs2` file system.

Extending a JFS File System (With On-line JFS)

With the On-line JFS product, you can extend a file system without unmounting. This proves invaluable in High Availability shops that can't afford downtime.

Example: (Assume that `/dev/vg01/myfs1` has been lvextended to 32 MB.)

```
# fsadm -F vxfs -b 32768 /myfs1
# bdf /myfs1
```

Note that the `fsadm` command requires you to specify the new file system size in "blocks". JFS blocks are typically 1K. To compute the desired file system size in blocks, simply multiply the logical volume size in megabytes by 1024 KB/MB.

12-6. TEXT PAGE: Summary of LVM Commands

LVM provides numerous commands to configure and manipulate logical volumes. This course has only covered the most common commands. A complete list of LVM commands is provided below for your reference:

extendfs	Extend an off-line file system
lvchange	Changes the characteristics of a logical volume
lvcreate	Creates a logical volume in a volume group
lvdisplay	Displays information about logical volumes
lvextend	Increases the number of physical extents allocated to a logical volume
lvolnboot	Prepares a logical volume to be a root, swap or dump volume
lvmerge	Merges previously mirrored volumes into one logical mirrored volume
lvreduce	Decreases the number of physical extents allocated to a logical volume
lvremove	Removes one or more logical volumes from a volume group
lvrmboot	Removes a logical volume link to root, swap or dump volume
lvsplit	Splits a mirrored logical volume into two logical volumes
lvsync	Synchronizes logical volume mirrors that are stale in one or more logical volume
pvchange	Changes the characteristics of a physical volume in a volume group
pvcreate	Creates a physical volume that can be used as part of a volume group
pvdisplay	Displays information about one or more physical volumes within a volume group
pvmove	Moves allocated physical extents from one physical volume to another
vgcfgbackup	Saves LVM configuration for volume group
vgcfgrestore	Restores LVM configuration onto the volume group
vgchange	Sets the status of a volume group to on or off
vgcreate	Creates a volume group
vgdisplay	Displays information about volume groups
vgextend	Extends a volume group by adding physical volumes to it

vgexport	Exports a volume group from a system
vgimport	Import a volume group onto the system
vgscan	Scans the system's physical volumes for volume groups
vgreduce	Reduces a volume group by removing one or more physical volumes from it
vgremove	Removes the definition of one or more volume groups from the system
vgsync	Synchronizes logical volume mirrors that are stale in one or more volume groups

12-7. LAB: File System Management

Directions

Perform the tasks suggested below. Record the commands you use and the answers to all of the questions.

1. Occasionally, the `/tmp` file system fills up, causing problems on a system. Using SAM or the command line, list all of the files in `/tmp` that haven't been accessed within the last 2 days.

2. `/var` is another file system which sometimes reaches 100%. Often this is a result of log files that haven't been properly trimmed. Trim the following log files back to size 0:

```
/var/adm/btmp  
/var/adm/wtmp
```

3. The questions that follow give you an opportunity to fix a full file system. You should have a script on your system called `/labs/fillfs.sh`. The `fillfs.sh` script will fill one of your file systems to capacity. Run `fillfs.sh`. As the script executes, you may see a number of file system full messages scroll across your screen. Don't worry – yet!

4. Which file system appears to be full? How many kbytes were allocated for this file system? What percent of the space in that file system is in use?

5. What happens at this point if a user tries to copy a file to the full file system? Is anything recorded in `syslog.log`? Copy a large file (e.g., `/stand/vmunix`) to `/home` to find out.

6. List two possible solutions to this full file system problem.
7. Are there any core files in the problem file system that could be removed? If so, remove them. What commands did you use to find the core files?
8. Which directory under `/home` is taking the most space? Which command can you use to find out? Mail a message to the culprit asking him or her to purge some files.
9. You could wait for your users to purge some old files, but in many cases, you will eventually need to add some additional space to the full file system. `pvccreate` your free disk and add it to `vg00`. Use `vgdisplay` to ensure that the new disk was successfully added to the volume group.
10. Double the size of the logical volume containing `/home`. Which disk contains the new extents? What command did you use to find out?
11. Using `bdf`, check to see if the `/home` file system contained in the logical volume you just extended increased in size. Does extending a logical volume automatically extend the file system within the logical volume?

12. Execute the commands necessary to extend the `/home` file system to take advantage of the additional space in the logical volume.
13. Unmount the file system and try doing `extendfs` on the logical volume containing `/home` again. Explain the resulting message.
14. Before moving on to the next chapter, remove all the "bigfiles" from user5's home directory.

12-8. REVIEW: Check Your Understanding

1. List and define two commands to monitor the free disk space on the system

2. What are some different solutions to recover space on your file system?

Module 13 — System Backup

Objectives

Upon completion of this module, you will be able to do the following

- Explain why backups are necessary.
- Create a graph file to determine which files are included in the system backup.
- Perform a full backup with `fbackup`.
- Perform an incremental backup with `fbackup`.
- Perform a backup across the network with `fbackup`.
- Create a system recovery tape with `make_recovery`.
- List steps needed to document the system configuration to ensure a smooth recovery in case of a system crash.

13-1. SLIDE: Why Back Up?

Why Back Up?

How Much Data Can You Afford To Lose?

Data is sometimes lost by:

- File system corruption
- Accidental removal of files
- Hardware failures
- System crash

Regular backups:

- Minimize data loss
- Keep users happy
- Provide stability and order

a669141

Student Notes

One of the principal responsibilities of a system administrator is preserving the data stored on the system. Unfortunately data is sometimes lost. A piece of hardware may fail, a file may be accidentally removed or overwritten, a command may go astray, or the system may crash. The user community has a reasonable expectation that the administrator has planned and implemented regular backup procedures to minimize data loss.

To further minimize the chance of data loss, all backup media should be stored at a location geographically distant from the system's disk drives. Picture the scenario of a multi-year project under development and a system administrator who has dutifully maintained system backups since the project's inception. However, for ease of retrieval, the backup media are stored in the same room as the computer system. If a fire, flood or other disaster destroyed the computer room, all the work done on the project would be lost.

Consequently, the safest course of action is to store the backup media in a secure environment separate from the computer equipment. In some cases this may mean the next room, a data vault on the premises, or even a data vault at another site.

The exact backup procedure employed is determined by a number of factors. A heavily used system, both in terms of number of users and amount of activity, may require some form of backup to be conducted daily. A lightly used system may only need to be backed up weekly or bimonthly. Media used for backups may prove to be expensive. If complete system backups are performed daily, the costs in terms of materials and personnel may grow to be significant. Consequently, a weekly or bimonthly backup could result in minimal costs added to the maintenance of the system.

In this module, we'll look at different backup strategies. Each has its advantages and disadvantages. A backup strategy should be implemented as soon as users begin to work on the system.

13-2. SLIDE: What Do You Back Up?

What Do You Back Up?

- Backup the entire file system (full backup)
- Backup part of the file system
 - Files that have changed since the last backup (incremental or delta backups)
 - A subtree of the file system
 - Application data
 - Users' files
- Backup the database configuration
- Backup the LVM configuration

a66955

Student Notes

Before we discuss how to back up the system, we must first discuss what to back up, and when. Doing a full back up of all files and directories under the / directory is perhaps the easiest solution.

On larger systems, however, daily full backups may not be practical:

- A full backup may take too long. It may not be possible to complete a full backup overnight before users return the next morning.
- A full backup may consume multiple tapes, and require an operator to swap tapes during the backup. Both of these requirements can be expensive.
- A full backup may not even be necessary. Some file systems, such as /tmp and /cdrom, don't need to be included in the backup. You can significantly decrease the size and duration of your backups by carefully selecting the directories to include.

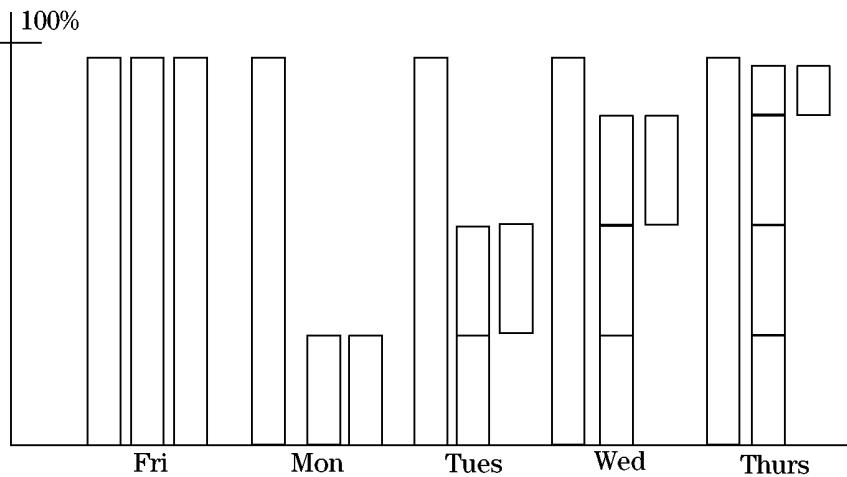
For all of these reasons, many administrators choose to back up only a portion of the file system on a daily basis. The list on the slide suggests several possible approaches.

In addition to the files and directories in your file systems, you may need to follow special backup procedures for your database configuration and partitions.

Finally, if an LVM disk has to be replaced, you will need to have a backup of the LVM configuration. Anytime you add, remove, or extend a logical volume, the `vgcfgbackup` command automatically writes a backup of the LVM configuration out to the `/etc/lvmconf` directory. Make sure `/etc/lvmconf` is included in your regular tape backups.

13-3. SLIDE: How Often Do You Back Up?

How Often Do You Back Up?



a66956

Student Notes

A question closely related to determining which data to back up is determining how often to back it up. The critical question is, "how much data can you afford to lose?" Evaluate the applications running on your system and the needs of your users to determine how critical the data on your system is to them. This will give you a guideline as to how often to back up the various files on your system. Consider the following things when determining how frequently to back up a particular file (or type of file):

- How often do the contents of the file change?
- How critical is the file's contents?

How often you make backups depends on how much data you can afford to lose. If you can afford to lose a month of data, then you need only back up the system once each month. If you can only afford to lose 6 hours of data, then you must back up every 6 hours. However, backing up every 6 hours can become prohibitive, and other possibilities (such as redundant systems) must be considered. For most applications, full backups once each week and partial backups each night are sufficient.

You should create a backup schedule for your system that describes how often you will perform full backups and incremental backups of the various files on your system. It is best to back up your system when there are few or no users logged in. Ideally, you should change your system's run-level to the system administration state (single-user mode), before initiating the backup procedure. This will ensure that you are the only one logged in.

Backup levels are a way of specifying varying degrees of incremental backup. For example, suppose you wanted to set up the following backup schedule:

- On the first day of the month, back up an entire set of selected files.
- Every Friday, back up all files in the selected set that have changed since the first of the month.
- Every day except Friday, back up all of the files in the selected set that have changed since the last Friday or first of the month, whichever is most recent.

There are three levels associated with the above schedule (the once per month level, the once per week level, and the once per day level). The once per month level is a full backup. The other two are incremental backups. The problem is how to distinguish between the two types of incremental backups. This is accomplished with backup levels. You can define up to ten backup levels (0 – 9). The backup strategy you choose should be determined by the level of activity on your system and the capacity of your media.

Recovery Example Using Three Backup Levels

To implement the earlier example of monthly, weekly, and daily backups, use the following backup levels:

level 0	full monthly backup
level 1	weekly backup on Friday
level 2	daily backup, except Friday

This table illustrates the level numbers for implementing this example.

Table 13-1.

Date of the month:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	1	
Day of the month:	Su	M	T	W	Th	Fr	Sa	Su	M	T	W	Th	F	Sa	Su	...		
backup level	0	2	2	2	2	1	2	2	2	2	2	2	2	1	2	2	...	0

If your data became corrupt on Thursday the 12th, then on Friday the 13th you would use the following sequence to restore your system to its Wednesday the 11th state:

- Restore the monthly full backup from Sunday the 1st.
- Restore the weekly incremental backup from Friday the 6th.

- Restore the incremental backup from Wednesday the 11th.

Recovery Example Using Two Backup Levels.

The following example illustrates a weekly full backup and daily incremental backup, two backup levels. When implementing your backup strategy using SAM, only two levels of backups are supported. The figure illustrates the level numbers supported by SAM:

Table 13-2.

Date of the month:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...	1
Day of the month:	Su	M	T	W	Th	Fr	Sa	Su	M	T	W	Th	F	Sa	Su	...	
backup level	0	1	1	1	1	1	1	0	1	1	1	1	1	1	0	...	0

If your data became corrupt on Thursday the 12th, then on Friday the 13th you would use the following sequence to restore your system to its Wednesday the 11th state:

- Restore the full backup from Sunday the 8th.
- Restore the incremental backup from Wednesday the 11th.

13-4. TEXT PAGE: System Backup Worksheet

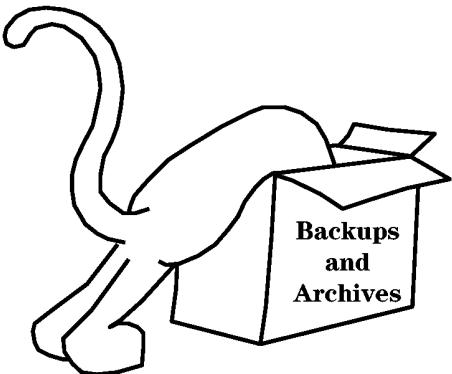
System Backup Worksheet

Table 13-3.

Type of Data	Loc. of Data	Import. of Data	Size	Media	Frequency (daily, weekly, hourly)	Level (full, incr.)	Sched. Time	Cmnd.
Operating System	vg00	low	150 mb					
lvm config	all disks	med	bytes					
config files	vg00	med	bytes					
Applic'n								
app1	disk2	low	300 mb					
Databases								
db1	vg02	v,high	300 mb					
db2	disk3	high	1.5 gb					
User File Systems								
/home	vg03	high	100mb					

13-5. SLIDE: How do you Perform the Backup?

How Do You Perform the Backup?



- **fbackup/frecover** are the most flexible standard HP-UX backup utilities.
- Other utilities are available as well.

a66957

Student Notes

There are many commands and utilities you can use to save files to backup media. Some are more flexible than others. Some are limited to only certain types of file systems. The chart on the following page compares the standard HP-UX backup utilities.

Table 13-4.

Task	Backup Utility				
	fbackup/ frecover	cpio	tar	dump/ restore (hfs only)	vxdump/ vxrestore (2) (vxfs only)
Supported on other UNIX platforms	no	yes	yes	yes	yes
Do an incremental backup	Has a powerful multilevel backup	Use find to locate new or modified files	Use the -u option to add any new or modified files to the end of the archive.	Possible on a single file system only	
Efficient use of tape	Medium	Low	High	High	High
Backup/restore large files (>2GB)	Possible	Not possible	Not possible	Not possible	Not possible
Backup/restore across a network	Possible	Possible	Possible	Possible	Possible
Find a particular file	Relatively easy; use frecover	Moderate; Wildcards are allowed, but searches the entire tape	Relatively difficult; Wildcards not allowed; searches the entire tape.	Relatively easy; interactive commands available	Relatively easy; interactive commands available
Multiple, independent backups on a single tape	Not possible (fbackup rewinds the tape)	Use mt with no-rewind device to position the tape, then use cpio	Use mt with no-rewind device to position the tape, then use tar	Use mt with no-rewind device to position the tape, then use dump	Use mt with no-rewind device to position the tape, then use vxdump
Verify backup	Use the -xNv options	Not possible	Not possible	Not possible	Not possible

Table 13-5.

Task	Backup Utility				
	fbackup/ recover	cpio	tar	dump/restore (hfs only)	vxdump/ vxrestore (vxfs only)
List files as they are backed up or restored	Possible; use the -v option on the command	Possible; use the -v option on the command	Possible; use the -v option on the command	Possible (on a restore only)	Possible (on a restore only)
Do a backup based on selected criteria (such as group)	Not possible	Possible; use find	Not possible	Not possible	Not possible
Cross disk or file system boundaries	Use fbackup -n to cross NFS boundaries	Possible; use find	Possible	Not Possible	Not Possible
Restore absolute path names to relative location	Relative to the current directory; use -x	Limited; can specify path/ name on each file with cpio -ir	Not possible	Relative to the current directory; use restore -r	Relative to the current directory; use vxrestore -r
Interactively decide on files to restore	Not possible	Can specify path or name on each file with cpio -ir	"yes" or "no" answer possible using tar-w	In interactive mode can specify which files	In interactive mode can specify which files
Use wildcards when restoring	Not possible	Possible	Not possible	Only in interactive mode	Only in interactive mode
Ease of selecting files for backup from numerous directories	High	Medium	Low	Not possible	Not possible
Backup a snapshot file system	Not possible; (fbackup gives error msgs. when backing up read-only files	Possible	Possible	Possible	Possible
Backup/restore extent attributes	Possible	Not possible	Not possible	Not possible	Possible

The utilities covered in the chart all backup files. The following utility performs a byte for byte backup:

dd

The **dd** command is useful in some limited situations, but is technically *not* a backup command. The **dd** command is a general purpose physical file copy utility. This is different from all the above utilities in that **dd** copies no file names or file attributes to a backup media; it simply copies everything, bit for bit. Thus no selective restore is possible. For these reasons, **dd** is *not* recommended for regular system backups. It is generally used for two purposes:

- Make a duplicate copy of a disk quickly. This assumes a destination disk the same size or larger than the source.
- Read or translate a foreign 9-track magnetic tape. For example, **dd** has the capability for reading backup media with user-defined record sizes, ASCII or EBCDIC translation, byte switching, and other useful options.

NOTE:

Since there are many commands available, and many options to each of these commands, it is strongly recommended that you write the command you use to create your backup on the label of your backup media.

13-6. SLIDE: Using fbackup

Using fbackup

Backing up a single Directory or File

Examples

```
fbackup -f /dev/rmt/0m -i /home -I index.home  
fbackup -f /dev/rmt/0m -i . -I index.pwd
```

Full and Incremental Backups

Examples

```
cd /var/adm/fbackupfiles  
vi graph  
  i /  
  e /cdrom  
fbackup -f /dev/rmt/0m -u0g graph -I index.full  
fbackup -f /dev/rmt/0m -u1g graph -I index.incremental
```

a66958

Student Notes

The **fbackup** command is the primary tool for creating both full and incremental backups. The **fbackup** command is quite flexible and allows each system administrator to develop a backup strategy that best suits the needs of the installation.

fbackup options

- f device** the device to which output will be sent
- [**-0-9**] backup level — default is 0
- [**-u**] update */var/adm/fbackupfiles/dates* file (only if used with **-g**)
- [**-i path**] include path (file or directory) in the backup
- [**-e path**] exclude path (file or directory) from the backup

- [-g *graph*] file that contains a list of files and directories to be included or excluded from the backup
- [-I *path*] write an index to file *path*

The **fbackup** command does *not*, by default, write to the standard output and the **-f *device*** part of the command is not optional. The *device* can be a regular file or a device (special) file. You can specify **-** as the *device* to have **fbackup** write to the standard output. For most systems, a magnetic tape drive is used as the backup device. In this case, use the appropriate device special file for the raw (character) device in the **/dev/rmt** directory.

Backup Levels

The [0-9] option allows the user to define and use what are referred to as backup levels. Recall that an incremental backup strategy only makes backup copies of files that have changed. The key question to answer is "changed since *when*"? Certainly, any file on the system can be considered to have changed if the file's time stamp is compared to the beginning of time. In other words, an incremental backup that copies files that have been modified more recently than the beginning of time is actually a full backup!

- Level 0 is predefined to mean "the beginning of time" by the **fbackup** command. If invoked without a backup level option, **fbackup** performs a level 0 backup that results in a full backup.
- Levels 1 – 9 are used for incremental backups. Each time **fbackup** is invoked with a backup level option, **fbackup** makes copies of files that have changed since the last time a backup was made at a *lower* level. For example, suppose **fbackup** is run at level 0 on Monday, level 1 on Tuesday, and level 2 on Wednesday. A full backup would be made on Monday, an incremental backup of files that had changed since Monday would be made on Tuesday, and an incremental backup of files that had changed since Tuesday would be made on Wednesday.

It is up to the system administrator to assign meaning to each backup level. For example, a backup strategy that calls for full backups on Monday and incremental backups dating back to the most recent full backup on Tuesday, Wednesday, Thursday, and Friday could be implemented by making a level 0 backup on Monday and level 5 backups on Tuesday, Wednesday, Thursday, and Friday. The reason for selecting level 5 for the incremental backups is to allow for some flexibility in the backup scheme should changes be required.

Graph Files

HP-UX file system trees can get very large. Multiple physical disks can be mounted under a single file system to produce a logical file system tree that is enormous. For this reason, it is desirable to have a mechanism to specify only parts of the file system to be backed up. It may also be the case that part of a file system is entirely static and may require only occasional backup (for example, monthly or semiannually). Simply put, you may wish to specifically include or exclude parts of the file system during backup.

Inclusion and exclusion is accomplished either through the use of graph files (-g option), or through the inclusion of parameters on the command line (-i and -e options). A graph file is a file that contains ASCII text. Each line in the file contains a directory path that is to be

included or excluded. Lines that begin with an " i" indicate a directory path that should be included. Lines that begin with an "e" indicate files that will be excluded.

```
i /
e /cdrom
```

The combination of backup levels and graph files provide significant flexibility. Recall that **fbackup** uses backup levels as a mechanism to identify files that have changed since the most recent backup that was made at a lower backup level. When invoked with a backup level *and* a graph file, **fbackup** makes a backup of files that changed since the most recent backup of that graph at a lower level.

Backup Levels and Graph Files

As described above, **fbackup** is capable of determining when a backup of a specific level was last made. This implies that there must be a data file used by **fbackup** to retain the necessary information. Such a data file does exist and is readable ASCII text. The file is **/var/adm/fbackupfiles/dates**.

The file **/var/adm/fbackupfiles/dates** contains information about when the last backup at each backup level was performed. The **dates** file contains:

- the graph file used for the backup
- the level of the backup
- the date of the backup
- the start and end time for the backup

This information is used by **fbackup** to determine which files defined in the graph file are included in the backup. The **fbackup** command uses the following search sequence on the **dates** file to determine the base backup on which to build an incremental backup:

- matching graph file
- next lowest level number
- most recent date

If no lower level is found, a full backup at the specified level is performed. If there are duplicates of a lower level found, the most recent is used as the base for the incremental backup.

Creating an Index File

The [-I *path*] option creates an index of files included in the backup. The index is written to the file specified in the *path* argument. Beware that a full system backup may generate a very large index file.

Slide Examples

The following example writes a backup of all files and directories under `/home` to the tape in `/dev/rmt/0m`. `fbackup` also writes an index of the files included in the backup to `index.home`.

```
# fbackup -f /dev/rmt/0m -i /home -I index.home
```

The next example creates a backup of all files and subdirectories under the present working directory.

```
# fbackup -f /dev/rmt/0m -i . -I index.pwd
```

The next couple of commands lay the groundwork for running full and incremental backups by creating a graph file in `/var/adm/fbackupfiles`. The graph file includes the `/` directory, but excludes `/cdrom`.

```
# cd /var/adm/fbackupfiles  
# vi graph  
i /  
e /cdrom
```

After creating a graph file, you can do a full, level 0 backup of the files specified in the graph using the next command on the slide. `-u` ensures that the `dates` log file is updated with the results of the backup, and `-I` creates an index of the files included in the backup.

```
# fbackup -f /dev/rmt/0m -u0g graph -I index.full
```

The last example does an incremental backup of the graph file, and again updates the "dates" file and creates an index.

```
# fbackup -f /dev/rmt/0m -u1g graph -I index.incremental
```

Many other examples and options are described in the `fbackup(1m)` man page.

13-7. SLIDE: Using **frecover**

Using **frecover**

Restoring Files and Directories

Examples

```
frecover -f /dev/rmt/0m -rv  
frecover -f /dev/rmt/0m -i /home/user1 -xv  
frecover -f /dev/rmt/0m -i /home/user2 -xv
```

Extracting an Index

Example

```
frecover -f /dev/rmt/0m -I index
```

a66959

Student Notes

The real reason for employing a backup strategy is to enable the recovery of lost files. As mentioned earlier, lost files can be the result of inadvertent removal by a user, or a file system disaster. The **frecover** command is the partner to the **fbackup** command. It is designed to retrieve files from backups that were created with **fbackup**. Like **fbackup**, **frecover** is flexible and has many options that modify its default mode of operation. Only a few key options are explained here. The **frecover(1M)** manual page contains a full description of all the options.

There are three basic modes of operation for **frecover**:

- | | |
|-------------------------|--|
| frecover -r | Recover everything that is on a backup volume. |
| frecover -x | Extract certain files from a backup volume. Files to be extracted must be specified with the -i option. |
| frecover -I path | Read the index from the backup volume and write it to <i>path</i> . This retrieves a table of contents. |

Unlike **fbackup**, **frecover** does have a default for input. The default is **/dev/rmt/0m**. If a different input source (device) is to be used, a **-f device** option may be specified on the command line. As with the **fbackup** command, **-f -** can be used to specify that the standard input should be used.

frecover -r and **frecover -x** have some options in common. Some of the most commonly used options are described below:

- v** Verbose. List all files and directories as they are restored.
- h** Used to recover (or extract) only directories, and not the files contained in them.
- o** Used to force **frecover** to overwrite a newer file with an older one. Normally, **frecover** will not overwrite an existing disk file with an older version of the file.
- F** Causes **frecover** to strip all the leading directories from the path names of files being recovered. If **/usr/bin/vi** and **/bin/sh** were on the backup and were recovered using the **-F** option while in **/home/root**, the resulting files would be **/home/root/vi** and **/home/root/sh**.
- x** Makes all recovered files relative to the current working directory. Suppose the current working directory was **/home/root** and the file **/usr/bin/vi** were being recovered. With the **-x** option, the file would be deposited in **/home/root/usr/bin/vi**. This option can be very useful when you are unsure about the directory and files that might result from an **frecover** session.
- N** Prevent **frecover** from actually recovering any files onto disk, but read the backup as if it was, in fact, recovering the data from the backup, producing the same output that it would on a normal recovery. This option is useful for verifying backup media contents.

An option that is unique to the **frecover -x** mode is **-g graph**. This allows you to use a graph file in the same way as with **fbackup**. The file format for the *graph* file is the same. Lines that begin with an **i** indicate a path that is to be included in the recovery. Lines that begin with an **e** indicate a path that is to be excluded from the recovery. This is useful for partial recoveries.

Slide Examples

The first example restores all files from the tape in **/dev/rmt/0m**. The **-v** verbose option lists each file and directory as it is restored.

```
# frecover -f /dev/rmt/0m -rv
```

The second and third examples extract **/home/user1** and **/home/user2** respectively, from the **/dev/rmt/0m** archive:

```
# frecover -f /dev/rmt/0m -i /home/user1 -xv
# frecover -f /dev/rmt/0m -i /home/user2 -xv
```

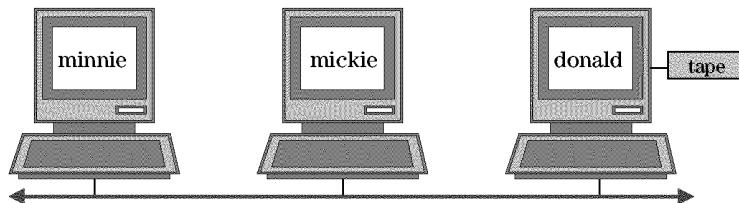
The final example on the slide creates an index of the files and directories in the **/dev/rmt/0m** archive. The index is written to a file called "index". Note that this index file

may not accurately represent the true contents of the tape. Since **fbackup** writes the index before writing the backup itself, the index on the tape lists the files that **fbackup** intended to backup, not the files actually included in the backup. The index created by the **-I** option on **fbackup** (as the tape is created), more accurately reflects the true contents of the backup tape.

```
# frecover -f /dev/rmt/0m -I index
```

13-8. SLIDE: Network Backup and Recovery

Network Backup and Recovery



```

donald# vi ~root/.rhosts
      mickie
      minnie

mickie# fbackup -f donald:/dev/rmt/0m -u0g graph -I index
minnie# fbackup -f donald:/dev/rmt/0m -u0g graph -I index

mickie# frecover -f donald:/dev/rmt/0m -rv
minnie# frecover -f donald:/dev/rmt/0m -rv

```

a66960

Student Notes

The **fbackup** and **frecover** examples discussed so far have all used a local tape drive. If your system doesn't have a local tape drive, you may need to write your backups to a tape drive connected to another host on the network.

Running **fbackup** across the Network

The example on the slide shows a network with three hosts. Only one of the three has a local tape drive. Below is a procedure that might be followed by mickie and minnie to write backups to donald's tape drive.

In order for mickie and minnie to be able to write to donald's tape drive, donald's administrator must grant mickie and minnie (the hosts), access to his system by creating a file called **~root/.rhosts**. This file should contain a list of hosts that need access to donald's tape drive. In this example, donald's **.rhosts** file would contain the following:

```
donald# vi ~root/.rhosts
      mickie
```

minnie

NOTE:

Creating the `~root/.rhosts` file not only allows other hosts to write to your tape drive; it also allows "root" on the specified hosts to log into your machine. This can be done with the `rlogin` command, and does not require a password. Therefore, only add a host to your `~root/.rhosts` file if you know and trust the host's system administrator. Unfortunately, this is the only way to allow a host to access a remote tape drive with `fbackup/frecover`.

After donald has created a `~root/.rhosts` file, mickie and minnie can each create graph files and start the backup as shown below:

```
mickie# fbackup -f donald:/dev/rmt/0m -u0g graph -I index  
minnie# fbackup -f donald:/dev/rmt/0m -u0g graph -I index
```

In the examples above, mickie and minnie's graph and index files reside locally on mickie and minnie respectively, but both backups are written to donald's `/dev/rmt/0m` tape drive. Note that the backups cannot occur in parallel; only one host at a time may write to donald's tape drive.

Running `frecover` across the Network

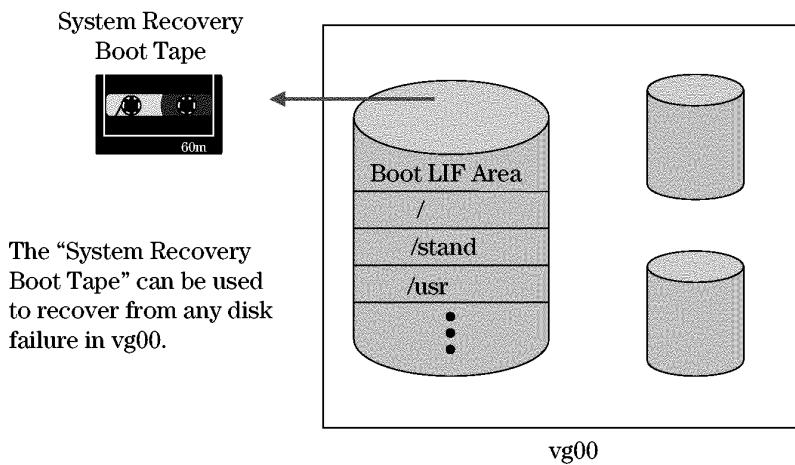
`frecover` makes it possible to recover files across the network, too. Files could be restored to mickie and minnie as shown below:

```
mickie# frecover -f donald:/dev/rmt/0m -rv  
minnie# frecover -f donald:/dev/rmt/0m -rv
```

The `-x` and `-I` options on `frecover` may be used when accessing remote tape drives as well.

13-9. SLIDE: Being Prepared with `make_recovery`

Being Prepared with `make_recovery`



a66961

Student Notes

`frecover` is adequate for recovering *some* files and directories. However, if the root disk becomes corrupted, the system will be unbootable, and it will no longer be possible to boot the kernel, let alone run `frecover` to restore files and directories.

The `make_recovery` command is designed precisely for this situation. `make_recovery` creates a bootable tape image of selected, critical files and directories in the root volume group. If any disk in the root volume group becomes corrupt, the administrator can recover a minimal system using the recovery tape.

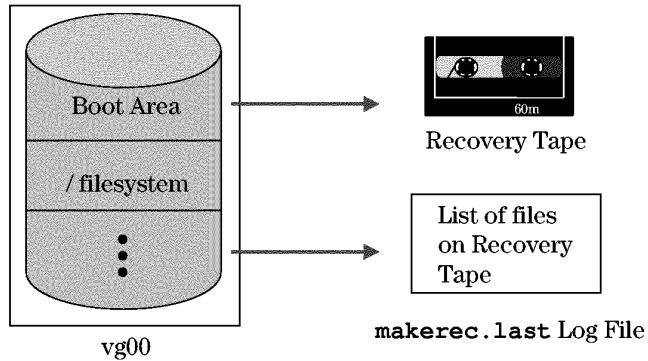
Booting from the recovery tape will

1. rebuild a boot area on the root disk
2. recreate the logical volumes and file systems on the disks in `vg00`
3. restore selected, critical files and directories to the root disk

The critical files restored by **make_recovery** make it possible to run **frecover** to restore other files and directories from the most recent tape backup.

13-10. SLIDE: Creating a Recovery Tape

Creating a Recovery Tape



Create a recovery tape and log file.
make_recovery -C

a66962

Student Notes

Step 1—Install the Ignite-UX Product

make_recovery is part of the Ignite-UX product which is not included in a standard load of HP-UX. Ignite-UX is included on the applications CD with HP-UX 11.x, but for version 10.x, it must be downloaded from the HP software web site. The URL for the web site is:

<http://www.software.hp.com>

Step 2—Run **make_recovery**

Run **make_recovery** to create the recovery tape using the options shown on the slide. By default, the recovery tape will include an archive of the following critical directories:

- /stand
- /sbin
- /dev
- /etc

and selected critical files from

- **/usr**
- **/opt**
- **/var**

All non-OS related files in these directories are excluded from the recovery tape archive. It may take half an hour or more to create the tape. Also, **make_recovery** requires at least 32 MB of space in the **/var** file system while creating the tape.

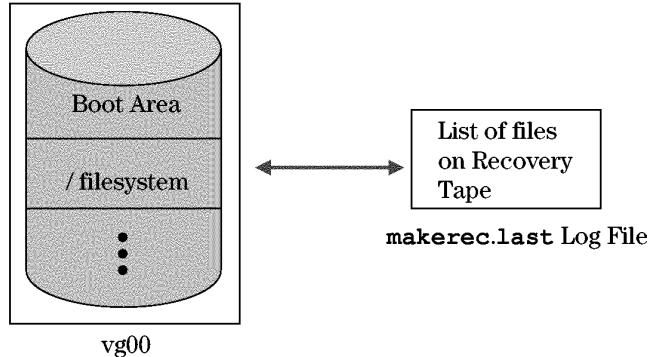
The **-C** option shown on the slide creates a log file of the names, timestamps, and checksum values of all files included on the recovery tape. The **check_recovery** command can use this file at a later date to determine if the files on the recovery tape are outdated. If you choose to run **make_recovery** without **-C**, there is no way to determine if the tape is out of date. The pathname for the file created by **make_recovery -C** is:

/var/opt/ignite/recovery/makerec.last

See the **make_recovery** and **check_recovery** man pages for additional options.

13-11. SLIDE: Updating the Recovery Tape

Updating the Recovery Tape



Compare current OS files to recovery tape:

```
# check_recovery
```

a66963

Student Notes

The system recovery tape needs to be kept current to reflect new hardware, software and patches.

The `check_recovery` command compares the current core OS files to the `/var/opt/ignite/recovery/makerec.last` system recovery status file (created when the last `make_recovery` was invoked) to determine if a new system recovery tape needs to be created. Only core OS and user core OS files are validated. `check_recovery` displays all discrepancies found and, based on these, the system administrator can determine whether a new system recovery tape needs to be created.

`check_recovery` detects the following discrepancies:

- Additions: A file existing on the current system and not listed in the system recovery status file is a file added to the system since the last system recovery tape was created.
- Deletions: A file not existing on the current system and listed in the system recovery status file is a file deleted from the system since the last system recovery tape was created.

- **Modifications:** A file existing on the current system but with a different “last modification date” is validated by its checksum. If the file’s checksum is different from the value in the system recovery status file, then the file has been modified.

NOTE:

make_recovery must have been previously executed with the **-C** option to create the system recovery status file.

Example

check_recovery has no options. Check the current system to determine whether a new system recovery tape needs to be created.

```
check_recovery
```

13-12. TEXT PAGE: System Recovery Checklist

To ensure that you are able to recover from a crashed disk, we recommend that you keep each item on this checklist at hand. With help from your HP Support Representative and this list, you stand a good chance of timely recovery.

- Recent recovery tape created with `make_recovery`
- Tape backup of `/etc/lvmconf`
- Printed `vgdisplay -v` output for each Volume Group
- Printed `lvdisplay -v` output for each Logical Volume
- Printed output of `lvolboot -v`
- Printed `/etc/fstab`
- Printed copy of output from a recent `bdf(1m)`
- Printed output of `swapinfo(1m)`
- Printed `pvdisplay -v` for each physical volume.
- Printed `ioscan -fun`
- Printed `ioscan -kf`
- Printed information from any software packages (database packages) that use logical volumes for raw data storage. This is not information that HP-UX keeps track of, but information that is used to configure software packages.

The last element in the list is probably the most overlooked and the hardest to reconstruct if there is some sort of problem on your system. Many of your major database packages support the use of raw logical volumes and you must have a list of the logical volumes that are used by your database package. Many database packages have an option in their administration menu that displays the raw disk devices that it uses. Print out this information.

13-13. LAB: Backup and Recovery

Directions

Perform the tasks suggested in the questions that follow. Record the commands you use, and answer all the questions.

Since your class machine may not have a tape drive, perform the backups to a series of fake tape files called `tape1`, `tape2`, etc. Note that on production machines you would more commonly use a true tape device file (e.g. `/dev/rmt/0m`).

Part I: Preliminary Steps

1. PART I

If don't already have a `/var/adm/fbackupfiles` directory on your machine, create it. By default, this is the directory where `fbackup` maintains the "dates" log file of backups completed on your system. Make this directory your present working directory.

2. Run the `/labs/corp1.sh` script, which will create a few directories that you can practice backing up. Note the directories that are created by the script.

Part II: Full and Incremental Backups

1. PART II

Create a graph file that includes everything in the `/corp` directory except `/corp/dept3`.

2. Perform a full backup of `/corp` using the graph file you just created. Write the backup to a file called `tape1`, write an index of the files included in the backup to "index1", and use the `-u` option to ensure that the "dates" file records the time stamp of your backup.

3. What was recorded in the "dates" file as a result of your **fbackup**? Were all of the directories under **/corp** backed up? Look at the "index1" file that was created, and explain what you see.
4. Run **/labs/corp2.sh**. This script should create a few new files under the **/corp** directory. Note the changes.
5. Do a level 1 backup of **/corp** using your graph file. Write the backup to **tape2**, create an index of your backup in **index2**, and use the **-u** option to ensure that the **dates** file is updated again.
6. What was recorded in your **dates** file as a result of your backup? Were all of the files and directories under **/corp** backed up? Look at the index file for this backup and explain what you see.
7. Run the **/labs/corp3.sh** script. Note the newly created files.
8. Run another level 1 backup. Write the backup to **tape3**. Ensure that the **dates** file is updated and create an index file called **index3**.

9. What changed in your "dates" file as a result of your backup? Explain. Which files and directories under /corp backed up? Explain.

10. Without running another corp script, try running a level 2 backup of /corp using your graph file. Again, ensure that fbackup updates the "dates" file and creates an index. What files and directories are included in this backup? Explain.

Part III: Restoring Files with frecover

1. PART III

Run the /labs/corp4.sh script, and note the effect this script has on your /corp directory.

2. You did a total of four backups in the previous part of this lab. Is it necessary to restore all four tapes to fully recover your system? Explain.

3. Do whatever is necessary to fully restore /corp.

```
# frecover -f tape1 -rv          # restore the most recent level 0  
# frecover -f tape3 -rv          # restore the most recent level 1  
# frecover -f tape4 -rv          # restore the most recent level 2  
# find /corp                      # ensure that the recovery worked.
```

4. Occasionally, users accidentally remove a file and expect the administrator to be able to restore the missing file from tape. Remove `/corp/dept1/reporta`. How can you restore this single file? Do it.

5. Will `frecover` overwrite newer data in a file with older data from a tape archive? Try it and find out.

```
# vi /corp/dept1/reporta          # Make a change to reporta  
# ll /corp/dept1/reporta         # Note the time stamp  
  
# frecover -f tape3 -i /corp/  
dept1/reporta -xv  
  
# cat /corp/dept1/reporta        # Were your changes overwritten by frecover?
```

Part IV: (Optional) Network Backup and Recovery

If you administer multiple hosts on a network, you may wish to backup all of your machines on a central backup server. This technique may, in fact, be the only possible backup alternative for nodes that don't have local tape drives.

In the exercises that follow, you will work with a team of classmates to perform a network file backup and recovery. If there are a number of tape drives in your classroom, your instructor will assign you to work with several classmates for this exercise. One of the hosts in your group should have a tape drive; this machine will be called the "backup server" for the purposes of this exercise. The tapeless hosts in your group will be "clients".

1. PART IV

Ensure that your server has granted your client(s) root access in the `~root/.rhosts` file. Also make sure there is a writable tape in the server's tape drive.

2. On the client, `cd` to the `/var/adm/fbackupfiles` directory.
3. From one of your clients, initiate a backup of the directories listed in the graph file you created for `corp` earlier in the lab. Do the backup to the tape device file `/dev/rmt/0m` on the server. Also create an index in "index5".
4. On which machine was the index file created? Which machine's dates file was updated as a result of the backup?
5. Remove the client's `/corp` directory structure. Then see if you can restore it from your tape.
6. If time permits, do a backup from the other clients as well.

Module 14 — Scheduling cron Jobs

Objectives

Upon completion of this module, you will be able to do the following:

- Submit, list, and remove time-scheduled jobs with `cron`.
- Schedule full and incremental backups to run automatically.
- Grant non-root users access to `cron`.

14-1. SLIDE: The cron Daemon

The cron Daemon

- Executes commands at specified dates and times
- Automates routine tasks
- Examples:
 - Trimming system log files
 - Performing system backups
 - Generating weekly reports



a66964

Student Notes

System administrators often need to run backups and other processes on a daily, weekly, or monthly basis. Although such processes can be started manually, it is often desirable to schedule them to run automatically. HP-UX provides the **cron** deamon for just this purpose.

The **cron** daemon is started during the system boot process and executes time scheduled jobs submitted by the system administrator and other users.

You can check to see if the **cron** daemon is running with the **ps** command:

```
# ps -ef | grep cron    # is cron running?  
# cron                  # start the cron daemon, if it isn't currently running
```

Regular users, as well as **root**, can utilize **cron** for repetitive execution of programs. Jobs are submitted to **cron** with the **crontab** command. **root** controls who can use **crontab** through the **/var/adm/cron/cron.allow** file. Users are permitted to use the **crontab** command if their names appear in the **cron.allow** file. If **cron.allow** does not exist, then

/var/adm/cron/cron.deny is checked to determine if the user should be denied access. If both exist, cron.allow takes precedence. If neither file exists, only root is allowed to submit a job. An empty cron.deny file allows all to use crontab.

Table 14-1. Who Can Use cron?

*.allow	*.deny	Who Can Use?
—	—	superuser
exists	ignored	everybody in *.allow
—	exists	everybody who is not in *.deny
empty	ignored	superuser
—	empty	everybody

14-2. SLIDE: cronfile

cronfile

- **cronfile** contains one line for each schedule job.
- Example:

```
# min    hour   date   month   day     command
# 0-59  0-23  1-31  1-12   0-6    must redirect output!
      *        *       *       *       *
      0        *       1,15   *       *
      0        6       *       *       1-5   /usr/bin/who | /usr/bin/lp
```

a6979

Student Notes

cron jobs are defined in a **cron** file. **cron** jobs are submitted to the **cron** daemon in **cron** files. Processes to be submitted to the **cron** daemon should be listed in the **cron** file, one per line.

The entries in **cronfile** must be in a specific format to be interpreted successfully by **cron**. Each entry in the file is a line containing six fields, separated by white space or tabs. The first five fields contain integers that represent the date and time a command is to be executed. They are shown on the slide. Each of these fields may contain an asterisk, which represents all legal values, or a list of entries, separated by commas. Each entry can be either a number or two numbers separated by a dash, which specifies a range.

The last field is a string that is executed by the shell at the specified times. A percent character in this field (unless escaped by \) is translated as a new-line character.

NOTE:

You must redirect the standard output and standard error of your commands. If you do not, any output generated will be mailed to you.

Always use full path names for your commands and file names, since **cron** uses only the standard environment of **/usr/bin/sh** and does not know about your environment, for example, your current directory, variables and PATH.

Questions

What values would you include in the **cron** file time fields if you wanted to execute a command:

1. every day at 6:30 p.m?
2. every weekday at 6:30 p.m?
3. every Monday, Wednesday, and Friday at 6:30 p.m?
4. every 10 minutes, every hour, every day?

14-3. SLIDE: Managing cronfile with crontab

Managing cronfile with crontab

- Edit and submit a **cronfile**:
`crontab -e`
- View your **cronfile**:
`crontab -l`
- Remove all your scheduled **cron** jobs:
`crontab -r`

a66966

Student Notes

Each user who is authorized to issue jobs to cron has *one* file in the directory, `/var/spool/cron/crontabs`. The name of this file is the user's login name. Once the user's **cronfile** is activated with the **crontab** command, any future use of this command causes the **crontab** file in `/var/spool/cron/crontabs` to be replaced.

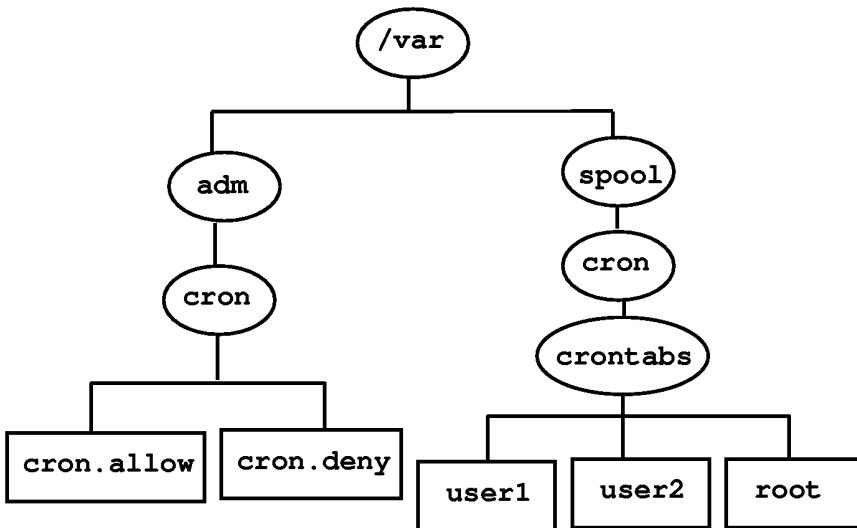
The **crontab** command can be used to create, modify or remove scheduled **cron** jobs:

- crontab -e** Automatically copies your currently scheduled **cron** jobs to a temporary **cron** file, then launches vi so that you can make changes. You can add, remove, and delete lines in this temporary **cron** file, then save your changes so the **cron** daemon recognizes your changes.
- crontab -r** Removes all of your currently scheduled **cron** jobs.
- crontab -l** Lists all of your currently scheduled **cron** jobs.

By default, **crontab** opens the user's own **cron** file. However, the system administrator can modify any user's **cron** file by specifying the user's username as an argument with the **crontab** command.

14-4. SLIDE: What Happens When a Job is Scheduled?

What Happens When a Job is Scheduled?



a66967

Student Notes

By default, only the system administrator can submit jobs to the `cron` daemon. If you wish to allow other users access to the `cron` daemon, add their names to the `/var/adm/cron/cron.allow`. Alternatively, you can define which users *cannot* use the `cron` daemon in `/var/adm/cron/cron.deny`.

Each user authorized to schedule `cron` jobs has a `cron` file in the `/var/spool/cron/crontabs` directory. The `cron` deamon consults these files to determine which jobs should be executed when.

When you use the `crontab` command to create and submit a `cron` file to the `cron` daemon, `crontab` automatically copies your `cron` file to the `/var/spool/cron/crontabs` directory. The copy of your `cron` file is named with your username.

When the scheduled time arrives, `cron` starts your job and logs a record to the `/var/adm/cron/log` file. The `cron` log grows without bound, and should be checked and emptied periodically.

NOTE:

Although the files in `/var/spool/cron/crontabs` can be viewed with `ls` and `cat`, they should never be directly edited or removed. Doing so can leave the cron daemon in an undefined state. Always use the `crontab` command when making changes to your scheduled cron jobs.

14-5. LAB: cron

In the exercises in Part I, you will schedule a simple **cron** job that repeatedly displays the output from the date command in the system console window. To ensure that you see these messages, open a console window by typing: # dtterm -C -name console &

Any output sent to /dev/console now appears in this window.

Part I: Managing Scheduled cron Jobs

1. PART I

cron should start automatically during the boot process. Check to ensure that the cron daemon is running on your machine.

2. Create and submit a **cron** job to display the current time and date to the console every minute.

3. Add another **cron** job that sends the output from the who command to the console every 10 minutes.

4. List your scheduled **cron** jobs. Are they both there?

5. Edit your **cron** file again. See what happens if you precede the /usr/bin/date line with a # sign. Save your changes. What effect does the # sign have?

6. Why would commenting a line out of the `cron` file be preferable to simply removing the line?

7. How can you remove ALL of your scheduled cron jobs? Do it.

Part II: (Optional) Scheduling Backups with cron

Your goal in this part of the exercise is to schedule full and incremental backups of your home file system to occur automatically via `cron`.

1. PART II

Create a `graph` file that includes `/home`, but nothing else. Name your graph file `/home.graph`.

2. Schedule a full backup of `/home.graph` to occur every Sunday night, and an incremental backup to occur every weekday night. Schedule both backups to run at 11 p.m. Redirect the error messages from your backups to a file called `/home.err`.

3. List your cron jobs to ensure they are properly scheduled. Leave your scheduled `cron` jobs in place, and check your `/home.err` file tomorrow morning to see if your `cron` job succeeded.

Module 15 — Managing Swap Space

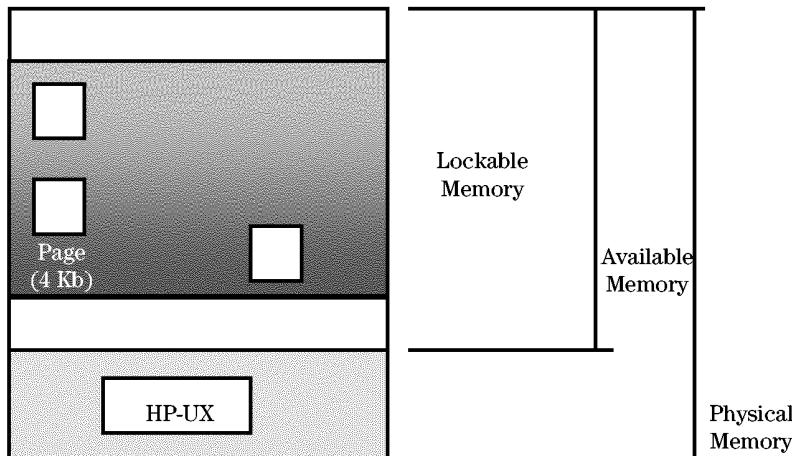
Objectives

Upon completion of this module, you will be able to do the following:

- Explain the concept of **demand paging**.
- Define **physical**, **available**, and **lockable** memory.
- Determine the amount of configured physical, available, and lockable memory.
- Determine the amount of swap space currently configured and in use.
- Configure device swap from the command line.
- Configure file system swap from the command line.
- Disable swap space.
- List considerations for selecting appropriate file system and device swap areas.

15-1. SLIDE: System Memory

System Memory



Student Notes

Physical memory is the random access memory (RAM) installed in your computer. At system startup, the system displays on the system console, the amount of physical memory installed:

Physical: xxxxxxxx Kbytes

Not all physical memory is available to HP-UX processes. Some memory is reserved for kernel code and data structures. The amount of memory remaining is referred to as **available memory**, and is used by the system for **demand paging**. During system startup, the system displays on the system console the amount of available memory:

Available: xxxxxxxx Kbytes

All or part of available memory can be locked by a subsystem or by user processes. **Locked memory cannot be swapped out to disk**. Typically, locked memory holds frequently accessed programs or data structures. By keeping them memory-resident, process performance improves. If most of the available memory is locked the system may deadlock. Some unlockable memory must be available to prevent deadlock.

During system startup, the amount of memory that can possibly be locked is displayed on the console:

```
lockable: xxxxxxxx Kbytes
```

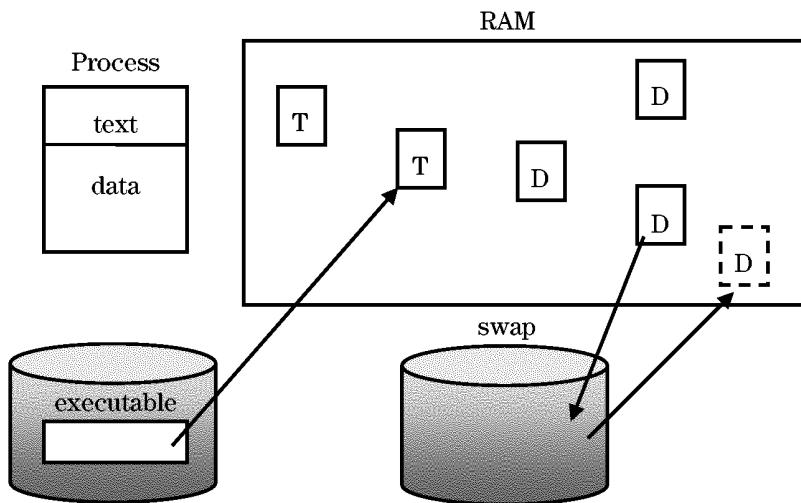
Available memory minus the memory locked by subsystems or user processes is the memory that is actually usable for virtual memory demand paging. The system parameter, **unlockable_mem** reserves the amount of memory that cannot be locked.

NOTE:

The **dmesg** command will show you the messages output by your system when you boot up. These messages include the amounts for physical, available, and lockable memory in kilobytes (KB).

15-2. SLIDE: What Is Swap Space?

What is Swap Space?



a669119

Student Notes

Swap space is an area on a high-speed storage device, reserved for use by the virtual memory system for paging processes.

Physical memory is a finite resource on a computer. This means that only so many processes can fit into physical memory at any one moment in time, even though many more processes may actually be ready to run or execute. While executing, a program's pages of data and instructions (text) are copied to and from secondary storage as needed. This is referred to as **demand paging**. Generally, the text portion of a program does not change as the program executes. Therefore, when needed, text is copied into RAM from the file containing the executable. The data pages of an executing program do change. Therefore, if a data page must be removed from RAM to free up space for other pages, it must first be copied to the swap space.

Paging

The kernel always tries to maintain a threshold of free pages in order to keep the system running efficiently. As long as this threshold, referred to as **lotsfree**, is maintained, no paging

occurs. When the number of free pages drops below this threshold, a daemon known as **vhand** is started. The daemon will select pages that have not been recently referenced. If necessary, the page will be copied to the swap area before being put on the free list. This is referred to as a **page out**. A **page fault** occurs when a process tries to access an address that is not currently in memory. The page will then be copied into RAM, either from the swap space or from the executable on disk.

On systems with very demanding memory needs (for example, systems that run many large processes), the paging daemons can become so busy swapping pages in and out that the system spends too much time paging and not enough time running processes. When this happens, system performance degrades rapidly, sometimes to such a degree that nothing seems to be happening. At this point, the system is said to be **thrashing** because it is doing more overhead than productive work.

The Swapper

The term **swap** dates back to early UNIX implementations that managed physical memory resources by moving entire processes between main memory and secondary storage. Most modern virtual memory systems today no longer swap entire processes, because this method causes the system to spend most of its time processing I/O instead of doing real work.

Swapping has been replaced by a **deactivation** scheme, which allows pages to be pushed out over time by a paging mechanism. Paging is a more efficient memory resource management mechanism for virtual memory.

When the system begins thrashing, or when free memory falls below another threshold, known as **minfree**, the swapper becomes active. The swapper deactivates processes, which prevents them from running, and thus reduces the rate at which new pages are accessed. Pages belonging to a process that is deactivated will not be referred to and will become good candidates to be freed by the paging daemon. When the swapper detects that available memory has risen above the minfree threshold and the system is no longer thrashing, it will reactivate the deactivated processes.

Swap Reservation

The swapping subsystem reserves swap space at process creation time, but does not allocate swap space from the disk until pages need to go out to disk. Reserving swap at process creation protects the swapper from running out of swap space.

When the system cannot reserve enough swap space for a new process, it will not allow the processes to be started. Additionally, as running processes try to dynamically acquire more memory, more swap space is reserved. If there is insufficient swap space for the additional reservation, the process will be killed.

While swap reservation prevents the system from running out of swap space, it also prevents processes from being created before all the swap space is actually allocated.

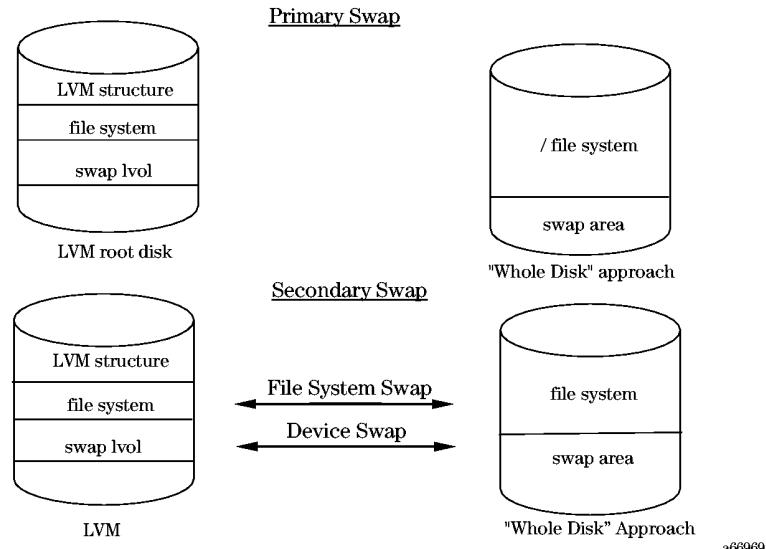
Evaluating Swap Space Needs

Before you install your system, try to get an idea of how much swap space you will need. Most application programs need a minimum amount of swap space to operate properly. This figure is usually contained in the documentation that comes with the application.

System administrators should monitor their swap space usage and make adjustments as necessary. We will see how to do this later in this module.

15-3. SLIDE: Types of Swap Space

Types of Swap Space



Student Notes

Types of Swap Areas

Device swap	A disk area or logical volume that is used exclusively for swap.
File system swap	File system swap, unlike device swap, is a file system that not only supports files and their data structures, but also has space available for swapping.
Primary swap	A special type of device swap which is available at boot. By default, primary swap is located on the same disk as the <code>root</code> file system. It is initialized by the kernel at boot time.
Secondary swap	Device or file system swap that is used in addition to primary swap. It is usually located on a disk other than the <code>root</code> disk.

Device Swap

Device swap is faster than file system swap. Device swap resides in its own partition.

When using the **whole disk** approach, you can either use an entire disk for swap, or reserve space at the end of the disk after the file system by using the **-R** option of **newfs**. For example, the following command creates a file system on a disk and reserves 200 megabytes (MB) for swap:

```
newfs -R 200 /dev/rdsck/c0t2d0
```

When using LVM you create a separate logical volume for device swap. The following is an example of an **lvcreate** command used to create a 200-MB swap logical volume:

```
lvcreate -L 16 -n myswap /dev/vg01
```

File System Swap

File system swap is a form of secondary swap. It can be configured dynamically. File system swap allows a process to use an existing file system if it needs more than the designated device swap space. File system swap is used only when device swap space is insufficient to meet demand-paging needs. File system swap consumes a variable amount of space because it only uses that portion of a file system that it needs. You can limit file system swap to a fixed size to prevent it from consuming too much space.

When file system swap is enabled in a file system, a directory called **/paging** is created under the root directory of the file system. A file is created for each swapchunk used in the file system. By default, a swapchunk is 2 megabytes.

Note that once a file system has been enabled for file system swap, it isn't possible to unmount that file system until the swap is disabled at the next system reboot.

Primary Swap

Your system must have at least one device swap area available when it boots. This area is known as the primary swap area. By default, primary swap is located on the same disk as the root file system.

If you are using LVM, the location of the primary swap area is stored in the boot data reserved area (**BDRA**) in the LVM structures on disk. The command **lvolboot -s lvol** is used to define primary swap. If you wish to change the primary swap definition, you must first use the command **lvrmbboot -s** to remove the prior definition.

Secondary Swap

In addition to primary swap, other swap can be used. Such swap is referred to as **secondary** swap. If you are using device swap as secondary swap, for better performance, allocate it to reside on a disk other than the root disk. File system swap is always secondary swap.

Secondary swap can be enabled automatically at boot time or it can be dynamically added while the system is running.

15-4. SLIDE: Enabling Swap from the Command Line

Enabling Swap from the Command Line

Examples:

- ① `swapon /dev/vg01/myswap`
- ② `swapon /dev/dsk/c0t2d0`
- ③ `swapon -p 4 -l 4M /myfs2`
- ④ `swapon -a`

Explanations:

- ① Swap on a logical volume
- ② Swap on a whole disk
- ③ Swap on a file system
- ④ Enable all swap entries in `/etc/fstab`

SAM can help!

a66970

Student Notes

Both file system and device swap can be enabled from the command line using the `swapon(1m)` command. The examples on the slide show several common uses of `swapon`:

1. This example enables the `/dev/vg01/myswap` logical volume for use as device swap. The entire logical volume will be claimed as swap, so it will no longer be available for use as a file system. If the logical volume contained a file system in the past, you may need to use the `-f` option to "force" an overwrite of the old file system structures.
2. This example enables device swap on the whole disk `/dev/dsk/c0t2d0`. If the disk contains a file system that was created with `newfs -R 200 /dev/rdsck/c0t2d0`, you can preserve the file system and simply enable swap on the available space reserved at the end of the disk by including the `-e` option on `swapon`. If you wish to overwrite the file system on the disk, use the `-f` force option instead.
3. This example enables the file system mounted on `/myfs2` for use as file system swap. The `-p` option sets the priority for this swap area to 4, and the `-l` ensures that `vhand` can take no more than 4 MB from the file system for use as swap.

NOTE: You cannot unmount a file system while the file system is activated for use as file system swap.

4. All swap areas are automatically deactivated at shutdown. To ensure that a swap area is automatically enabled at the next system boot, it must be added to the `/etc/fstab` file. The syntax for swap entries in `/etc/fstab` is described on the next slide. Issuing `swapon -a` immediately activates all swap entries in `/etc/fstab`.

Enabling Swap Using SAM

To add swap using the System Administration Manager (SAM) choose **Disks and File systems**, from the SAM functional area launcher. Next choose **Swap**. SAM will display your current swap areas. Choose either **Add Device Swap** or **Add File System Swap** from the **Actions** menu. Fill out the menu. SAM will allow you to enable the swap now and at every system boot. Selecting at every system boot causes SAM to update the file `/etc/fstab`. This file is used at boot time to activate swap areas.

Enabling Swap Using `swapon`

Syntax for Adding Device Swap

```
/usr/sbin/swapon [-p priority] [-e | -f] device
```

- priority* Indicates the order in which space is taken from the file systems and devices used for swapping. 0 is the highest and 10 is the lowest (default is 1).
- e** Uses space after the end of the file system on block device for paging. This option cannot be used with the **-f** option. Do not confuse this with swapping to a file system; this option is to be used with the "whole disk" approach, when a disk has both file system and swap space on it (`newfs -R swap`).
- f** Forcibly enables device swapping to a device where a file system exists (*destroys the file system*.)
- device* Block special file which is to be used for paging.

Syntax for Adding File System Swap

```
/usr/sbin/swapon [-m min] [-l limit] [-r reserve] [-p priority] directory
```

- m min** *min* specifies the amount of paging space the paging system will initially take from the file system. *min* can be specified in units of kilobytes (k suffix), megabytes (M suffix), or file system blocks (no suffix).
- l limit** *limit* specifies the maximum space the swap system is allowed to take from the file system. *limit* can be specified in units of kilobytes (k suffix), megabytes (M suffix), or file system blocks (no suffix). (The default is no limit.)

-r <i>reserve</i>	<i>reserve</i> specifies the space, in addition to the space currently occupied by the file system, that is reserved for file system use only, making it unavailable to the paging system. This reserved space is in addition to the minimum free space specified by the administrator when the file system was created.
-p <i>priority</i>	Same as for device swap.
<i>directory</i>	The directory where the file system swapping occurs. The system creates a directory named paging under the root of the file system for swap.

NOTE: You cannot unmount a file system when file system swap is active.

Swap Size Limitations

Several kernel tunable parameters limit the amount of swap that can be made available.

- The default maximum amount of swap space you can configure, for both device swap and file system swap combined, is approximately 512 MB. The tunable system parameter, **maxswapchunks**, controls this maximum. This parameter (default value of 256) limits the number of swap space chunks. The default size of each chunk of swap space is 2 MB. The size of a swapchunk can be modified with the **swchunk** kernel tunable parameter.
- The system parameter **nswapdev** in **/stand/system** sets the maximum number of dynamically configured swap devices at ten (default). The maximum is 25. More than **nswapdev** swap devices would require a kernel regeneration.
- The system parameter, **nswapfs**, determines the maximum number of file systems you can enable for swap. The default is 10 and the maximum is 25.

15-5. SLIDE: Enabling Swap via /etc/fstab

Enabling Swap via /etc/fstab

Swap areas listed in **/etc/fstab** are automatically enabled at system boot.

Examples

```
/dev/vg01/myfs1 /myfs1 vxfs      delaylog    0 2
/dev/vg01/myfs2 /myfs2 hfs       defaults     0 2
① /dev/vg01/myswap   .   swap      defaults     0 0
② /dev/dsk/c0t2d0   .   swap      defaults     0 0
③ .           /myfs2 swapfs  pri=4,lim=4M  0 0
```

Explanations:

- ① Swap on a logical volume
- ② Swap on a whole disk
- ③ Swap on a file system

a66971

Student Notes

In order to ensure that a swap area is enabled at every system boot, define the swap area in the **/etc/fstab** file. During the boot process, the **swapon -a** command enables all **/etc/fstab** swap entries. Fields in the **/etc/fstab** file are described below:

<i>block device</i>	The block special file name.
<i>directory</i>	The name of the root of the mounted file system, if there is one. If <i>type</i> is swapfs , it can be the name of any directory.
<i>type</i>	Can be swap , swapfs , or ignore (other values are available, for mounting file systems). If the <i>type</i> field is swap , <i>directory</i> , <i>backup-frequency</i> , and <i>pass_number</i> are ignored. If the <i>type</i> field is swapfs , <i>block device</i> , <i>backup-frequency</i> , and <i>pass_number</i> are ignored.
<i>options</i>	Options to the swapon command, if the <i>type</i> is swap or swapfs .

<i>min= min</i>	Amount of paging space the paging system will initially take from the file system. Same as swapon -m option.
<i>lim= limit</i>	Maximum space the paging system can take from the file system. Same as swapon -l option.
<i>res= reserve</i>	Space reserved for files in the file system. Same as swapon -r option.
<i>pri= priority</i>	Swap priority. Same as swapon -p option.
<i>end</i>	Use space after end of file system. Same as swapon -e option.
<i>backup frequency</i>	Reserved for future use
<i>pass number</i>	Unused with swap and swapfs (used by the fsck command to determine the order in which file system checks are done).
<i>comment</i>	Optional field that starts with #.

By listing swap devices in **/etc/fstab** you ensure that the swap device is automatically enabled when the system is rebooted. The command **swapon -a** is run as part of the **/sbin/init.d/swap_start** script, which executes when you boot the system to state 1 (in **/sbin/rc1.d** directory). The **-a** option instructs the system to read **/etc/fstab** for swap information, and to activate all swap areas.

NOTE: After making changes to **/etc/fstab** swap entries, run **swapon -a**. This forces the system to reread the **fstab** file and enable any newly-defined swap entries, as well as identify possible syntax errors.

15-6. SLIDE: Monitoring Swap Space Usage

Monitoring Swap Space Usage

Examples:

- ① **swapinfo**
- ② **swapinfo -f**
- ③ **swapinfo -d**
- ④ **swapinfo -tm**

Explanations:

- ① Report usage of all swap areas
- ② Only list file system swap areas
- ③ Only list device swap areas
- ④ Report values in megabytes, with a total line

a66972

Student Notes

After configuring one or more swap areas on your system, monitor the use of those swap areas over time using **swapinfo (1m)**. **swapinfo** lists the configured swap areas and reports what percentage of each is currently in use. If you are running low on swap space, you may need to configure an additional swap area to ensure that your users are able to run the applications they need.

Selected **swapinfo** options:

- t Add a totals line with a type of *tot*.
- m Display information in megabytes instead of kilobytes.
- d Show information about device swap areas only.
- f Show information about file system swap areas only.
- q Quiet mode. Print only a total Kb AVAIL.

Sample swapinfo output:

```
# swapinfo -t
```

	Kb	Kb	Kb	PCT	START /	Kb		
TYPE	AVAIL	USED	FREE	USED	LIMIT	RESERVE	PRI	NAME
dev	204800	0	204800	0%	0	-	1	/dev/vg00/lvol2
dev	16384	0	16384	0%	0	-	1	/dev/vg01/myswap
dev	1025730	0	303104	0%	0	-	1	/dev/dsk/c0t2d0
localfs	4096	0	4096	0%	4096	0	4	/myfs2/paging
reserve	-	81264	-81264					
memory	93564	55308	38256	59%				
total	1344574	136572	485376	10%	-	0	-	

Explanation of output:**TYPE****dev** Device swap.**localfs** File system paging space on a file system residing on a local disk.**network** File system paging space on a file system residing on another machine. This file system will have been mounted on the local machine via NFS.**reserve** Paging space on reserve. This is the amount of paging space that may be needed by processes that are currently running, but that has not yet been allocated from one of the above paging areas.**memory** Memory paging area (also known as pseudo-swap). This line appears only if memory paging is enabled.**Mb AVAIL** The total available space from the paging area including any paging space already in use. For file system paging areas the value is not necessarily constant. It is the current space allocated for paging (even if not currently used), plus the free blocks available on the file system to ordinary users, minus RESERVE (but never less than zero).**Mb Used** The current number of 1-Mbyte blocks used for paging in the paging area. For the memory paging area, this count also includes memory used for other purposes and thus unavailable for paging.**Mb FREE** The amount of space that can be used for future paging.**PCT USED** The percentage of capacity in use, based on Mb USED divided by Mb AVAIL; 100% if Kb AVAIL is zero.**START/ LIMIT** For device paging areas, START is the block address on the mass storage device of the start of the paging area. The value is normally 0 for devices

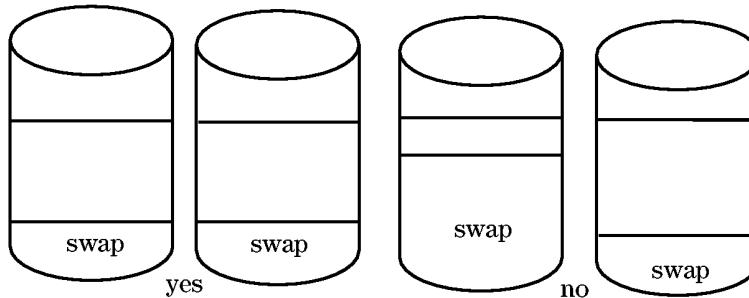
dedicated to paging. For file system paging areas, LIMIT is the maximum number of Mbs that will be used for paging, the same as the limit value given to **swapon**.

RESERVE	For device paging areas, this value is always -. For file system paging areas, this value is the number of Mbs reserved for file system use by ordinary users, the same as the reserve value given to swapon .
PRI	The same as the priority value given to swapon . This value indicates the order in which space is taken from the devices and file systems used for paging. Space is taken from areas with lower priority values first. <i>priority</i> can have a value between 0 and 10.
NAME	For device paging areas, the block special file name whose major and minor numbers match the device's ID. For file system swap areas, NAME is the name of a directory on the file system in which the paging files are stored.

15-7. SLIDE: Guidelines for Selecting Device Swap Areas

Guidelines for Selecting Device Swap Areas

- Two swap areas on different physical disks are better than one single swap area
- Only one swap partition (logical volume or reserved space) per disk
- Device swap areas should be of similar size
- Consider the speed of the disks



a53878

Student Notes

For device swap, you must allocate logical volumes or portions of a whole disk. The size of space reserved on a whole disk is specified on the `newfs` command (-R option). You can create logical volumes in the size you require.

The guidelines for selecting swap areas are given above. Most of the recommendations are for performance reasons. You can set up swap space any way you like, but your system will run slower if the above rules are not applied.

From the point of view of performance, two swap areas on different disks are better than one swap area with the equivalent amount of space. Also for performance reasons, multiple swap sections on the same disk should *not* be used.

Multiple swap areas are used in an interleaved fashion. **Interleaved swapping** means that space from one swap device is used, and then space from another space device. If the swap areas are on different disks, the swap areas can be written to concurrently in order to avoid disk head contention for multiple writes to the same disk.

Don't create a separate device swap area on the disk containing the primary swap area, because this causes excessive head movement on that disk and slows the system down.

If you are using LVM, set up multiple device swap areas in logical volumes that are on different disks (physical volumes).

Device swap areas should be of similar sizes for best performance. Otherwise, when all space in the smaller device swap area is used, the larger swap area is all that is available and interleaving is no longer possible.

How Swap Space Is Prioritized

All swap devices and file systems enabled for swap are assigned a priority ranging in value from 0 to 10. The priority number determines which swap areas will be used first. The priority is set by swap activation with the `-p` option of `swapon`. The default priority number is 1. Pages are sent out to your configured swap areas according to the following rules:

- Start at the highest priority swap device or file system. The lower the number, the higher the priority; that is, space is taken from a system with a zero priority before it is taken from a system with a one priority.
- If multiple devices have the same priority, swap space is allocated from the devices in a round-robin fashion. Thus, to interleave swap requests among a number of devices, the devices should be assigned the same priority. Similarly, if multiple file systems have the same priority, requests for swap are interleaved among the file systems.
- If a device and a file system have the same swap priority, all the swap space from the device is allocated before any file-system swap space. Thus, the device at priority 1 will be filled before the swap is allocated from the file system at priority 1.

It is recommended that you assign the same swapping priority to most swap devices, unless a device is significantly slower than the rest. Assigning equal priorities limits disk head movement, which improves swapping performance.

15-8. SLIDE: Guidelines for Selecting File System Swap Areas

Guidelines for Selecting File System Swap Areas

- Avoid using busy file systems such as the root file system
- Use **bdf** to check file system space availability
- Set priorities appropriately
 - Faster devices over slower devices
 - Infrequently-used file systems over busier file systems
- It is preferable to enable swap on file systems that are located on separate disks or, in the case of LVM, separate physical volumes.

a53879

Student Notes

When you need more swap space and you have no disk space available for additional device swap, you can dynamically add file system swap to your system.

These guidelines are provided to help the system administrator decide which file system sections should be used for swap space. Once again, most of these rules stem from performance issues.

When you add swap areas, you can assign a priority to each swap area. Priorities range from 0 (the highest) to 10 (the lowest). The system uses the swap areas with higher priority before using the lower priority swap areas. If you assign the same priority to two different swap areas, the system will use each of them on an alternating basis.

In general, it is best to assign highest priorities to the swap areas that afford the fastest performance. Therefore, give device swap areas priority over file system swap areas; give faster devices priority over slower devices; give lower use file systems priority over higher use file systems. Swap devices of the same type, (e.g. all device swap on disks of approximately the same speed) should all be assigned the same priority to take advantage of interleaved swap.

Enabling two file systems on the same disk (or, physical volume), can result in excessive head movement and will result in slower system performance.

NOTE:

Device swap offers better performance than file system swap. If given a choice, use device swap.

15-9. LAB: Swap

Directions

Perform the following tasks. Write the commands you use, and the answers to any questions that are asked.

1. How much memory does your training system have? How much is lockable, available, and physical?
2. Create a 48M logical volume and dynamically add it as swap.
3. Create a file system in a 20M logical volume. Mount the file system at **data**. Dynamically enable it as file system swap, limiting the amount of space that the paging system can take to 10M, and reserving 4M for files. Display swap space usage.
4. How do you make sure that swap will be enabled each time the system is rebooted?
5. Create a 12M logical volume using SAM. Use SAM to add this logical volume as swap. Enable the swap area now and at every boot. Examine **/etc/fstab**. What entry did SAM add?
6. Can you unmount a file system that has file system swap enabled? Try it.

7. You should have discovered in the previous question that once swap is enabled in a file system, it is impossible to unmount that file system. Is this a problem? Explain.

8. Try to view the **man** page for **swapoff**. What happens?

NOTE: To disable swap, edit the **/etc/fstab** file; comment out or delete the lines for the swap areas you wish to disable; and reboot. When the system reboots, the commented lines in **fstab** will be ignored. There is no way to disable swap without rebooting.

9. Remove the **/etc/fstab** entries for the swap areas you created in this lab, then reboot your machine by typing:

```
# vi/etc/fstab  
# cd /  
# shutdown -ry 0
```

Module 16 — Printer Management

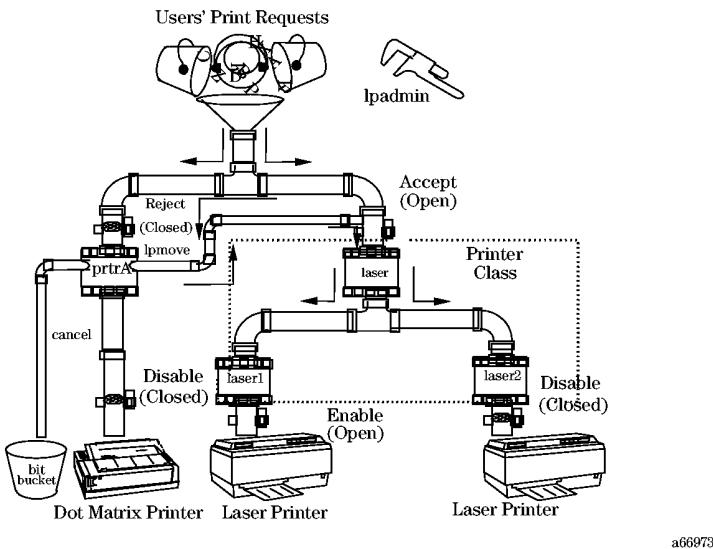
Objectives

Upon completion of this module, you will be able to do the following:

- Distinguish between local, remote and network printers.
- Add and remove local, remote, and network printers using System Administration Manager (SAM).
- Start and stop the lp spooler from the command line.
- Check lp spooler status from the command line.
- Manage print queues from the command line.
- Troubleshoot problems with the lp spooler.

16-1. SLIDE: Overview of the Spooling System

Overview of the Spooling System



Student Notes

On a multiuser system, access to the printers requires careful management and control. Since users can send print requests possibly to the same printer at the same time, we must have a way to:

- Make sure each file is printed separately.
- Determine which file will be printed first.

The **lp spooler** is a collection of utilities and commands that controls the print requests of users. The lp spooler stores print requests in the spool directory until a printer is available. When a printer is available, print requests are processed one at a time, in the background.

If the lp spooler system is installed, any user can

- submit a job to be printed
- obtain the status of all printers or any single printer

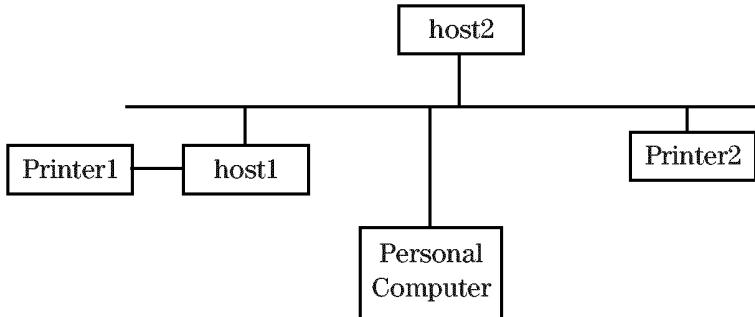
- cancel any print job
- declare printers to be in and out of service

Furthermore, the system administrator can add, modify, and remove printers in the spooling system as necessary.

16-2. SLIDE: Types of Printers

Types of Printers

- Local
- Remote
- Network



a538103

Student Notes

There are three categories of printers.

- local
- remote
- network

Local Printers

A **local printer** is connected directly to the system on which you are printing. In the example shown on the slide, printer1 is a local printer on host1. It may be connected to a parallel port or a serial port. There will be a device file for the printer on the local system. An example of a device file for a parallel printer is `/dev/c1t0d0_1p`.

Remote Printers

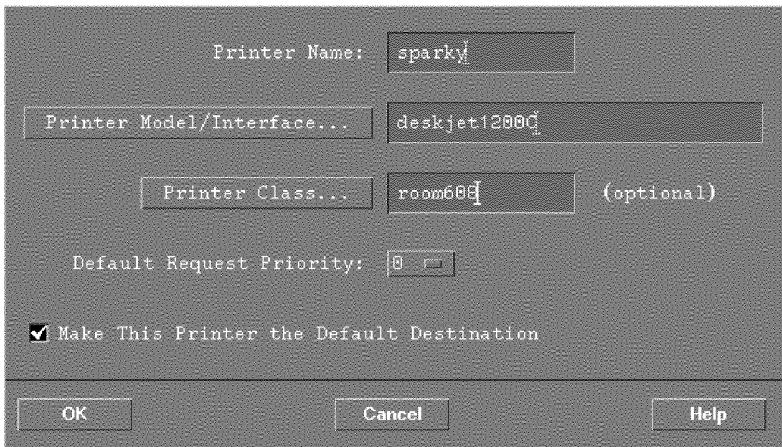
A **remote printer** is configured on a system other than the one you are logged in on when you make your print request. In the example shown on the slide, `printer1` can be set up as a remote printer on `host2`. A remote printer must first be defined as a local or network printer on some other host.

Network Printers

A **network printer** is a printer that has its own connection to the local area network (LAN). This is accomplished by installing a networking card, such as a JetDirect card. The printer can then be shared by all the systems on the LAN, which can include systems running other operating systems.

16-3. SLIDE: Adding Local Printers Using SAM

Adding Local Printers Using SAM



a538104

Student Notes

When adding a local printer you must be able to answer the following questions:

- What is the name you are giving to this printer?
- What is the appropriate model script?
- Do you want to set a default printer priority?
- Do you want to specify a printer class?
- Will this be the system default printer?

In addition you must be sure the printer driver is in the kernel. You can then add the local printer either using SAM or the `lpadmin` command. If you are using `lpadmin` you must also know the device file name of the printer.

To add a local printer using SAM, select **Printers and Plotters** from the functional area launcher. Select **lp Spooler** from the sub menu. Select **Printers and Plotters** from the next sub menu, then select **Add Local Printer/Plotter** from the **Action** menu. SAM will present a list of types of local printers from which to choose. The list includes Parallel and Serial. If the print driver is not in the kernel SAM will prompt you to reconfigure the kernel first before proceeding. SAM will scan the hardware for devices of the appropriate type. For example if you choose Parallel as the type of printer, SAM will display all the parallel ports. Highlight the device you wish to configure as a printer and press **OK**. SAM will then display a menu to add the printer.

For each local printer you wish to add you must specify the following:

Printer Name	This is the name being assigned for use whenever this particular printer or plotter is accessed. This is the name that accompanies the -d option with the lp command. It can be any unique combination of up to 14 alphanumeric characters and underscores.
Printer Model/ Interface	A printer model is a model of an interface script that is shipped with the print spooler software. Each printer has an interface program that tells the print spooler how to access the special features of the printer. The interface models can be used without modification, or you can create customized interface programs or modify copies of the model interfaces. The interface model scripts are in the directory /usr/lib/lp/model . To see a list of the printer models available on the system, select the Printer Model/Interface button. If your printer does not match any of these, use the dumb model. When you display the list of model scripts you will notice that there are two types of interface scripts available for many printers. One interface script is the standard HP model script, the other is for use with SharedPrint/UX. SharedPrint/UX enhances the basic capabilities of the lp spooling system, using a client-server implementation. SharedPrint is not activated by default. It requires either Network Computing System (NCS) or HP DCE/9000. One of these must be configured in order to use SharedPrint. We will not be covering SharedPrint/UX configuration in this course. For more information refer to: <i>SharedPrint/UX User and Administrator's Guide</i> , Part Number B1171-90124.
Printer Class	A class is a group of related printers and plotters. When print requests are sent to a class, they are serviced by the first available printer or plotter that is a member of that class. A class name can be any unique combination of up to 14 alphanumeric characters and underscores. To create a new class, simply enter a new name.
Default Request Priority	This defines the default priority of print requests submitted to this printer. Any job submitted to the printer uses this priority unless the priority is explicitly set to a different value by a -p option in the lp command used to send the request to the spooler. Priorities range from 0 to 7 with 0 being the lowest priority and 7 being the highest.
Default Destination	All print requests sent by the lp command without a -d option are sent to the default destination, unless the LPDEST variable is set in the user's shell environment.

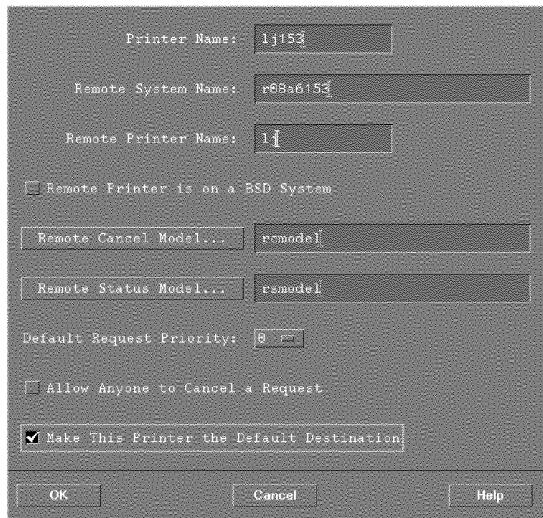
When you have finished filling out the menu press OK. SAM will give you the option of adding the printer to the CDE front panel. SAM will temporarily shut down the lp spooler to add the printer and then restart the lp spooler.

CAUTION:

Anytime a printer is added to the lp spooler, SAM stops and restarts the scheduler. This causes all currently printing jobs scheduled by your host to restart at page 1.

16-4. SLIDE: Adding a Remote Printer Using SAM

Adding a Remote Printer Using SAM



a66974

Student Notes

A remote printer is a printer that has already been configured on another system. To configure a remote printer you must be able to answer the following questions:

- What is the name you are giving to this printer?
- What is the remote system name?
- What is the remote printer name?
- Will this be the system default printer?
- Will you allow anyone to cancel a request, or only `root`?
- Is the remote printer on a BSD system?

To add a remote printer select Add Remote Printer/Plotter from the Actions menu of the Printers/Plotters area. SAM will present you with a menu. You must specify the following:

Printer Name	This is the name assigned for use whenever this particular printer or plotter is accessed. It is the name that accompanies the -d option with the lp command. It can be any unique combination of up to 14 alphanumeric characters and underscores. This name does not have to match the name used on the system to which the printer is connected.
Remote System Name	This is the Internet host name of the system that print requests for this remote printer are sent to. This hostname must be able to be resolved either in the /etc/hosts file or by using a naming service such as DNS or NIS.
Remote Printer Name	This is the name the remote system uses to access the printer.
BSD System?	If your print requests are intended to go to (or pass through) a Berkeley Software Distribution (BSD) system, you must check this box. Berkeley systems use a different spooler.
Remote Cancel Model	The system uses an interface program to cancel print requests on a remote system. For remote printers, use the cancel model script rmodel . This script is found in the /usr/lib/lp/cmodel directory.
Remote Status Model	The system uses an interface program to obtain the status of print requests on a remote system. For remote printers, use the status model script smodel . This script is found in the directory /usr/lib/lp/smodel .
Default Request Priority	This defines the default priority of print requests submitted to this printer. Any job submitted to the printer uses this priority unless the priority is explicitly set to a different value by a -p option in the lp command used to send the request to the spooler. Priorities range from 0 to 7 with 0 being the lowest priority and 7 being the highest.
Allow Anyone to Cancel?	Toggle to allow users to cancel other users' print requests. The default (toggle not set) is that a user can cancel only his or her own print requests. This option is not available for local printers.
Default Destination?	All print requests sent by the lp command in which no -d option was specified, are sent to the default destination, unless the LPDEST variable is set in the user's shell environment.

Notice, you do not specify a model interface script. The model interface script name **rmodel** is used for remote printers. This script will issue the **rlp** command to send LP line printer requests to a remote system.

Tasks Performed on the Remote System

If you are having trouble printing on the remote system, make sure that the remote system is configured properly:

- Edit the file **/etc/services** and, if needed, uncomment the line beginning with the word **#printer** by removing the #.
- Ensure that no systems are disallowed access via **/var/adm/inetd.sec**.

- **rlpdaemon** must be running on the remote system so that it can accept remote print requests. Edit the file **/etc/inetd.conf** and, if needed, uncomment the line beginning with the word **#printer**. Then invoke the command **inetd -c**.
- Ensure the **/etc/hosts** files on both systems are configured.

16-5. SLIDE: Adding a Network-based Printer

Adding a Network-based Printer

```
*****
*           CONFIGURATION          *
*       HP JetAdmin Utility for Unix   *
*****  
  
Printer Network Interface  
1) Create printer configuration in BOOT/TFTP  
   database  
2) Remove printer configuration from BOOT/TFTP  
  
Spooler:  
3) Add printer to local spooler  
4) Delete printer from local spooler  
5) Modify existing spooler queue(s)  
  
?) Help      q) Quit
```

a66975

Student Notes

A network printer is one that is equipped with its own controller and direct network connection so that it can receive print requests directly from the network (LAN) without having to be connected to a separate computer system. Network printers are configured using software that must be obtained from the vendor. HP's network printer software is called JetAdmin, which is included in the HPNP product on your HP-UX application.

When you add a network printer using SAM, SAM will run the command `/opt/hpnp/admin/jetadmin` for you. You can also run this command from the command line. In either case you begin a JetAdmin dialog.

In order to print to a network printer, it must be able to boot properly and it must be added to the lp spooler. The first step may have already been done on another system in the network. When a network printer is powered on it must retrieve networking information and files from a specified host on the network. On a UNIX machine this is done by configuring the printer in the `/etc/bootptab` file. It is not necessary for each host to be able to supply this information. When you run JetAdmin you can choose whether or not to "Create the printer configuration in BOOTP/TFTP database". This step is only necessary if the network printer has not been

configured on any other host. In order to perform this task you will need to supply the following information:

- The LAN hardware address of the printer. This is obtained from the printer with the JetDirect card installed.
- The hardware address of the printer. This is assigned by your network administrator.
- The IP address of the printer. This is shown on your printer's test page output.

Once the network printer has been configured to boot from some system, it must be added to the local spooler. To do this you will need to supply the following information:

- The printer's IP address or hostname. This can be obtained from the network administrator.
- The `lp` destination name. You assign this name. It is used whenever this particular printer is accessed. This is the name that accompanies the `-d` option with the `lp` command.
- The model script.
- What printer class (if any) to which you will assign this printer.

16-6. TEXT PAGE: Configuring a Network Printer Using JetAdmin

This page demonstrates how to run JetAdmin to configure a network printer. The example sets this system up to be able to act as a boot server for the network printer as well as adding the printer to the lp spooler.

```
*****  
*          CONFIGURATION          *  
*      HP JetAdmin Utility for Unix  *  
*****
```

Printer Network Interface:

- 1) Create printer configuration in BOOTP/TFTP database
- 2) Remove printer configuration from BOOTP/TFTP

Spooler:

- 3) Add printer to local spooler
- 4) Delete printer from local spooler
- 5) Modify existing spooler queue(s)

?) Help q) Quit

Please enter selection: 1

You will be asked a series of questions. After all of the questions have been answered, the responses are used to create an /etc/bootptab entry, and an optional configuration file. This configuration file is retrieved by the network printer with TFTP after it receives the BOOTP response.

These responses apply to all questions:

```
"q"      - returns you to the next higher level menu  
"??"     - prints help text  
"return"  - skips optional parameters or selects the default value
```

Enter the printer's LAN hardware address: 0800090edeb1

Enter the network printer name/IP name (q - quit): room9prt

Enter IP address: 15.24.186.179

Add room9prt and 15.24.186.179 to /etc/hosts? (y/n/q, default=y) :y

Printer name and IP address have been added to /etc/hosts.
 If your /etc/hosts file is updated automatically from a master source, add the name and IP address to your master source after the configuration is complete.

Following are optional parameters you may set for JetDirect. Select any non-zero numbers to make the changes. The settings are used to create a BOOTP/TFTP database when '0' is selected. To abort the operation, press 'q'

Other optional parameters:

-
- 1) Set printer location
- 2) Set printer contact
- 3) Set subnetmask
- 4) Set gateway
- 5) Set syslog
- 6) Change idle timeout
- 7) Create access list (up to 10 names). (Default: all allowed).
- 8) Other SNMP parameters:
 (GET/SET community name, trap and community name, authentication trap)

Select an item for change, or '0' to configure (q - quit): 1

Enter the printer location (q - quit): Building 37

Following are optional parameters you may set for JetDirect. Select any non-zero numbers to make the changes. The settings are used to create a BOOTP/TFTP database when '0' is selected. To abort the operation, press q

Other optional parameters:

-
- 1) Set printer location
- 2) Set printer contact
- 3) Set subnetmask
- 4) Set gateway
- 5) Set syslog
- 6) Change idle timeout
- 7) Create access list (up to 10 names). (Default: all allowed).
- 8) Other SNMP parameters:
 (GET/SET community name, trap and community name, authentication trap)

Select an item for change, or '0' to configure (q - quit): 0

Completed creating BOOTP/TFTP configuration database for room9prt.

Tftp service is also used to boot up JetDirect. Make sure /var/adm/inetd.sec allows JetDirect's IP to access tftp service on this node.

Please wait...

Testing BOOTP with 080009000000....:

RESULT: Passed BOOTP test 1 with 080009000000.

.....

BOOTP/TFTP has been verified functional.

Configuration data is now in place. The next test is to ping the printer for the IP name you just assigned it. To continue the test, you MUST do the following so that the printer can configure itself with the configuration data:

Power cycle the printer.

Wait until the printer finishes the self test.

(Note: It may take 20 sec to 1 min for a token ring HP

JetDirect interface to finish the configuration.)

Press the return key to continue the test.

If you are not ready for the next test (for example, the IP name has not taken affect in your DNS server), press q to return to the configuration menu now.

Do you want to send test file(s) to this printer (y/n, default=n)?
n

* CONFIGURATION *
* HP JetAdmin Utility for Unix *

Printer Network Interface:

- 1) Create printer configuration in BOOTP/TFTP database
- 2) Remove printer configuration from BOOTP/TFTP

Spooler:

- 3) Add printer to local spooler
- 4) Delete printer from local spooler
- 5) Modify existing spooler queue(s)

?) Help q) Quit

Please enter selection: 3

Enter the network printer name/IP name

(default=room9prt, q - quit):

Following is a list of suggested parameter values for this queue. You may change any settings by selecting the corresponding non-zero numbers. The values will be used to configure this queue when '0' is selected.

To abort the operation, press 'q'.

Configurable Parameters:	Current Settings
-----	-----
1) Lp destination (queue) name:	[room9prt]
2) Model Script:	[net_lj4x]
3) Status Log	[(No Log)]
4) Default Printing Language	[AUTO]
5) Queue Class	[(Not assigned)]
6) Job Recovery	[ON]
7) True End-of-Job	[OFF]
8) JobMonitor	[ON]
9) Default Queue	[NO]
10) Banner Page	[ON]
11) Add queue to the HP VUE front panel	[NO]

Select an item for change, or '0' to configure (q - quit): 1

Following is a list of suggested parameter values for this queue. You may change any settings by selecting the corresponding non-zero numbers. The values will be used to configure this queue when '0' is selected.
To abort the operation, press 'q'.

Configurable Parameters:	Current Settings
-----	-----
<u>lj609</u>	
1) Lp destination (queue) name:	[lj609]
2) Model Script:	[net_lj4x]
3) Status Log	[(No Log)]
4) Default Printing Language	[AUTO]
5) Queue Class	[(Not assigned)]
6) Job Recovery	[ON]
7) True End-of-Job	[OFF]
8) JobMonitor	[ON]
9) Default Queue	[NO]
10) Banner Page	[ON]
11) Add queue to the HP VUE front panel	[NO]

Select an item for change, or '0' to configure (q - quit): 0

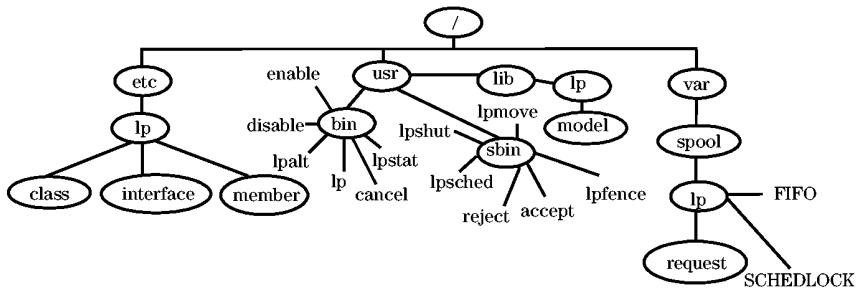
Ready to configure lj609.

OK to continue? (y/n/q, default=y) y

Finished adding "lj609" to the spooler.

16-7. SLIDE: Directory Overview

Directory Overview



a538107

Student Notes

The following is a road map of the directories in the spooling system.

- | | |
|--------------------------|---|
| /var/spool/lp | lp spooler system parent directory. All information about the setup and printing queues is located here. |
| /etc/lp/class | Printer classes directory. This contains the files that define which printers belong to which printer classes. |
| /usr/lib/lp/model | System-supplied interface programs. This directory contains the model shell scripts designed to submit requests to various printer models. The model scripts are responsible for generating banner pages and setting default fonts and other options. When adding a printer to the spooling system you will select a model script from this directory. The model script is copied into the /etc/lp/interface directory and renamed to match the name of the printer. |

/etc/lp/interface	Interface programs in use on your system. This has shell scripts from /usr/lib/lp/model that may be modified for particular printers. For example, if you do not want a printer to produce the banner page, modify the interface program.
/var/spool/lp/request	Destination queues. This is where all lp requests are queued. It usually contains a subdirectory for each printer configured on the system.
/usr/bin	Contains user-executable commands, including lp spooler commands that general users can execute.
/usr/sbin	Contains administrator commands, including the lp commands that only root or the lp user can execute.
/etc/lp/member	Lists all configured printers, one file per printer. The contents of each printer's member file is the assigned device file.
/var/spool/lp/SCHEDLOCK	This file is created when the lp scheduler is started with the lpsched command. It ensures that only one copy of the scheduler can run. The SCHEDLOCK file is removed when the scheduler is properly shut down with the lpshut command. If the scheduler is not properly shut down, you must remove the SCHEDLOCK file manually before restarting the scheduler. It is done automatically at boot time.
/var/spool/lp/FIFO	This file is used internally by the lp scheduler. Like SCHEDLOCK, it is created automatically by the scheduler and removed when the scheduler is stopped.
/usr/lib/lp/fonts	Contains fonts for LaserJet printers.

16-8. SLIDE: What Happens when a File Is Submitted with lp

What Happens when a File Is Submitted with lp

```
$lp -dLJ filename  
request id is LJ-####
```

Print request submitted
Spooler assigns unique request
ID and queues request

filename is linked or
copied to spool directory
/var/spool/lp/request/LJ

Printer becomes available

lpsched invokes the interface program
/etc/lp/interface/LJ

File is printed

a66976

Student Notes

When the *lp* command is invoked, a print destination is determined and a print request is submitted to the spooling system. A unique request ID number is assigned to each print request.

The print destination (in our example, LJ) is a logical name known to the spooling system and related to a spool directory. Usually, the destination is the name of a particular printer. However, several printers can be combined in a class of printers, so a class may also be a destination. If the destination of a print request is the name of a printer, the request will go to that printer's queue only; however, if the destination is a class of printers, the request will be spooled on to that class's queue and will be printed by the first printer that is available within that class. This helps avoid backups in a particular queue.

All print requests are queued in their destination spool directory strictly in priority order, that is the job with the highest priority will print first. If two jobs have the same priority they will be printed in FIFO (first-in-first-out) order.

Before a print request is actually printed, an interface script is invoked by the spooling system. An **interface script** is the link between the scheduler and the device (that is, its device file). It takes arguments from the scheduler and sends header and configuration information to the device. Each device has its own interface script which can be modified by the administrator.

Why Modify Interface Scripts?

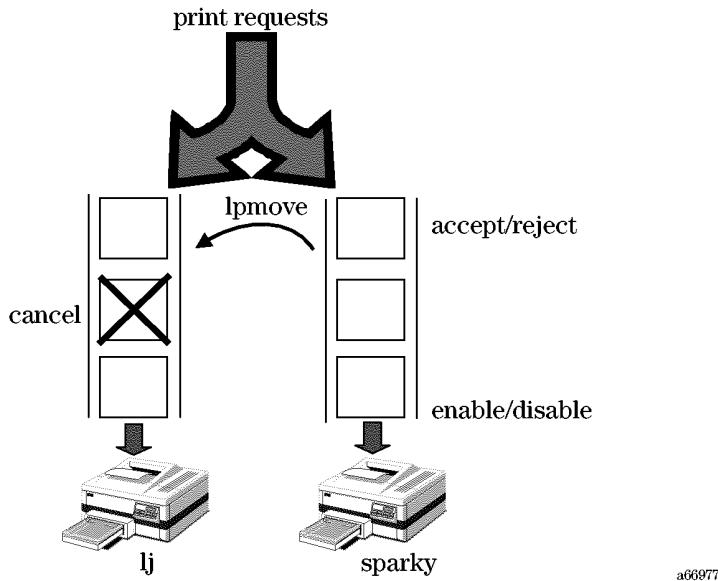
You may want to modify the speed of the serial line to which the printer is attached. The default speed is 9600 bits per second (BAUD), but many printers are able to work at much higher speed, thus decreasing the time to print a file.

Another example would be if you generally don't want to waste paper with the banner page. You can comment out the do_banner function in the interface scripts.

You must be a bit more familiar with shell scripts to modify the interface scripts, but for these two examples the changes are easy. Search for "9600" or "do_banner" in the file and do the modifications.

16-9. SLIDE: Managing Print Queues

Managing Print Queues



a66977

Student Notes

When a print request is made it is put on a print queue. The queue may be for a specific printer or for a class of printers.

Accept/Reject

The **lp** command cannot place a request on a queue unless the destination is accepting requests. When a destination is **rejecting requests**, any attempt by the **lp** command to queue a request to the destination will fail. Destinations are configured to accept or reject requests either with SAM or the **accept** and **reject** commands.

Example:

```
# reject -r "Sparky is down. Use the destination lj" sparky
```

```
# accept sparky
```

Enable/Disable

While **accept** and **reject** control what can be placed on the queue, **enable** and **disable** control whether the printer will process requests from the queue. The **enable** command activates the named printers, enabling them to print requests taken by **lp**. **disable** deactivates the named printers, disabling them from printing requests taken by **lp**. By default, any requests that are currently printing on the designated printers are reprinted in their entirety either on the same printer or on another member of the same class. This can be useful if a print job has begun printing on the wrong type of paper. Printers can also be enabled and disabled using SAM. Any user can enable and disable printers.

Examples:

```
# disable sparky
# enable sparky
```

Moving Request

Requests can be moved from one queue to another using either SAM, **lpmove(1m)** or **lpalt(1)**. You can either move all the requests in the queue or move individual requests. Before using **lpmove**, you must first shut down the scheduler. be aware that this restarts all currently printing requests.

Examples

# lpshut	<i>Must stop the schedules before lpmoveing requests</i>
# lpmove sparky 1j	<i>Move all requests from sparky to 1j</i>
# lpmove sparky-123 1j	<i>Move only one request to 1j</i>
# lpsched	<i>Restart the scheduler</i>
# lpalt sparky-123 -d 1j	<i>Move only one request. This can be used by an ordinary user</i>

Cancel

The **cancel** command cancels requests that were made with the **lp** command, even if they are currently printing. At least one ID or printer must be specified. Print requests can also be cancelled using SAM.

Examples:

# cancel sparky-123	<i>Cancel request sparky-123</i>
# cancel sparky	<i>Cancel current request printing on sparky</i>

By default any user can cancel any other user's requests.

Display Status

The **lpstat** command displays the output of the line printer spooling system. It displays the status of the user's print requests when used with no options.

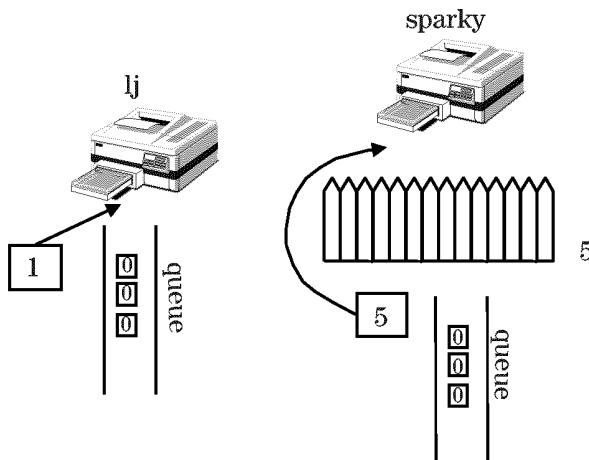
There are many more options which display various areas of the lp spooler. The **-t** option displays all status information.

Example:

```
# lpstat -t
scheduler is running
system default destination: lj
device for lj: /dev/null
    remote to: lj_id on hpsfecid
device for sparky: /dev/c1t0d0_lp
sparky accepting requests since Dec 19 08:18
lj accepting requests since Dec 19 08:54
printer sparky disabled since Dec 19 10:47 -
    reason unknown
    fence priority : 0
printer lj is idle. enabled since Dec 19 08:54
    fence priority : 0
sparky-18          root           priority 0  Dec 19 10:47
    hosts                         37410 bytes
printer queue for lj
no entries
```

16-10. SLIDE: Priorities and Fences

Priorities and Fences



a66978

Student Notes

Priorities are set both for printers and print jobs. The printer priority is set during printer setup, either with SAM or with the `lpadmin` command. The printer priority can be changed later. If a priority is not specified when issuing the `lp (1)` command, the request will be sent with the printer's default priority. Priorities range from 0 to 7, with 0 being the lowest priority.

All print requests are queued in their destination spool directory strictly in priority order, that is, the job with the highest priority prints first. If two jobs have the same priority they are printed in FIFO (first-in-first-out) order.

In the example shown on the slide, the default priority for `lj` is 0. All `lp` requests made without the `-p priority` option are queued with priority 0. The request made by issuing the command: `lp -p1 myfile` will jump to the front of the queue. Any user can use this option.

Fence Priority

A printer's fence level defines the minimum required priority for the spooled file to be printed and is configured with SAM or the **lpfence** command. A print request sent with a priority lower than the printer's fence level will sit in the printer's queue until the print job's priority is raised or the fence level is lowered.

In the example shown below, the printer **sparky** has the fence level set at 5. The requests in the queue with priority 0 will not print, even if the printer is not busy. A new request made with a priority of 5 meets the fence requirement and will print.

Examples:

# lpfence sparky 5	<i>Set the fence priority to 5</i>
# lp -p5 -dsparky myfile	<i>Jump the fence with this request</i>
# lpalt sparky-123 -p5	<i>Allow one request to print</i>
# lpfence sparky 0	<i>Lower the fence and allow other requests to print</i>

The **lpalt** command can be issued by ordinary users. **lpfence** is restricted.

16-11. SLIDE: Troubleshooting the Spooler

Troubleshooting the Spooler

Potential Spooler Problems:

- Spooler won't start
- Scheduler won't stop
- Paper jam/printer out of paper
- Runaway printout
- Printer won't print

a538111

Student Notes

Scheduler

If the scheduler won't start up when you execute the `lpshed` command, check to see if the `SCEDLOCK` file exists in the `/var/spool/lp` directory. If it exists, remove it with this command:

```
rm -f /var/spool/lp/SCEDLOCK
```

Then try again to start the scheduler with `lpshed`.

If the scheduler won't stop using `lpshut`:

```
# kill -15 lpsched_pid
```

Paper Jam/Paper Out

When a paper jam occurs or the paper runs out, you may wish to either print the output again from the beginning, or restart printing from where it stopped.

Restart from the Beginning

1. Put printer off-line.
2. **disable** the printer.
3. Clear the jam or load paper.
4. Put printer on line.
5. **enable** the printer.

Printing begins from the beginning.

Restart from Stopping Point

1. Clear the jam or load paper.
2. Put printer on line.
3. You may need to **enable** the printer.

Runaway Printout

1. Determine the request-id of the output.

```
lpstat -u
```

2. Cancel the request.

```
cancel
```

```
request-id
```

Printer Won't Print

1. Move its print requests to another printer.

<code>lpshut</code>	<i>shut down the lp scheduler</i>
<code>lpmove sourceprinter destination printer</code>	<i>move the print jobs</i>
<code>lpsched</code>	<i>restart the lp scheduler</i>

2. Ask Questions.

— Do other printers print?

- Has this printer printed before?
- Were there any error messages?
- Can other users print on this printer?

3. Take action based on the answers to questions.

If the problem affects only one printer, check physical connections (power, cables, etc) Make sure the printer is on line. Use `lpstat -s` to make sure the printer is correctly defined in the spooling system. Is the correct device file associated with the printer? Try

```
sleep 200 < /dev/tty np n &
```

This opens the port.

```
stty 9600 CS8 -istrip -parenb opost onlcr < /dev/tty np n
```

This configures the printer.

```
cat /etc/motd > /dev/tty np n
```

If the problem affects all the printers on the system, make sure the scheduler is running

```
lpstat -r
```

If other users can print to this printer, check the printer's priority fence. If your print request has a lower priority than its printer's priority fence, it will not print. Also, make sure the user `lp` has permission to access the file you are trying to print.

SAM's Help with Spooling Problems

SAM has the ability to save and restore the actual spooling system configuration. This can be very helpful if someone (e.g. `root`? ...) has removed a job by deleting the jobs file in `/var/spool/lp/request`. Since the spooler state is kept in a few more (binary) files, you won't be able to restore normal spooling operation except by going through every file and directory and delete file contents and directory files. This is most time consuming.

You should always save the current spooler state with SAM after adding or deleting a printer.

16-12. LAB: Hands-On Adding Printers

Directions

Perform the following tasks. Write the commands you use, and the answers to any questions that are asked.

1. If you have a printer available, use SAM to configure the printer. Add the printer to the class named class1.

If no printer is available use your terminal (or a window) as your printer. To find the device file associated with your terminal (or window) issue the command:

```
tty
```

This will return the device file name. Change the permissions on the device file so that lp can read and write to the file.

```
chmod 666 devicefile
```

You can now use SAM to configure your printer. You will add it as a local printer with "Nonstandard Device File". Use the **dumb** Model/Interface. Place your printer in the class named class1.

Exit SAM when after you have added your printer.

2. List the status of the lp scheduler. Is the scheduler running? What is your system default destination? Is your printer enabled? Is your printer accepting requests?

3. Try to print to your printer. Issue the command:

```
banner success|lp -d
```

```
your_destination
```

Try to print to the destination class1.

4. Set the fence priority on your printer to a priority higher than your default priority. Issue the `lp` command without the `-p` option. Did the request print? Display the status of the scheduler. Do you see your request on the queue? Raise the priority on your request to make it print.
5. Use SAM to remove your printer.
6. Configure your neighbor's local printer as a remote printer for your machine. Print a copy of `/etc/passwd` on your new remote printer and see what happens.
7. If the equipment is available in the classroom, configure a network printer.

16-13. REVIEW: Check Your Understanding

Directions

Write the answers to the following questions.

1. What functions does the spooling system provide, and why are they required?
2. Which of these functions are available to the administrator and which to normal users?
3. What is the difference between a device, a printer and a destination?
4. Is it possible, to install two printers for one device?
5. How would you cancel your own print request?
6. How would you cancel a print request owned by someone else?
7. What is an interface program?

8. If you have stopped the scheduler (with `lpshut`), does printing continue?
9. If you have stopped the scheduler (with `lpshut`), can you still use the `lp` command to add print requests to the queues?
10. How can you tell other users that a printer is "broken"?
11. How can you move print requests from the "printer1" queue to the "printer2" queue?

Module 17 — Shutdown and Reboot

Objectives

Upon completion of this module, you will be able to do the following:

- Describe the difference between single and multiuser mode.
- Properly shutdown the system using shutdown and reboot.
- Describe how the PDC, ISL, and HP-UX utilities load the kernel in memory.
- Boot from the primary boot device.
- Boot from an alternate boot device.
- Boot from an alternate kernel.
- Boot to single-user mode to perform system maintenance.
- Describe what occurs once the kernel is loaded in memory.

17-1. SLIDE: HP-UX Operation States

HP-UX Operation States

Multi-user mode

- Normal machine state
- Users can log in
- File systems mounted
- Most services plus daemons available

Single-user mode

- Required for some admin tasks
- Only root login allowed
- Non-critical file systems unmounted
- Non-critical daemons shut down

Halt state

- Nothing running

a66979

Student Notes

An HP-UX system will always be in one of several operation states. The system state determines what services and functionality is available. Some of the system administrative tasks described in this course require you to change your current operation state. This chapter will present an overview of the functionality available in each state.

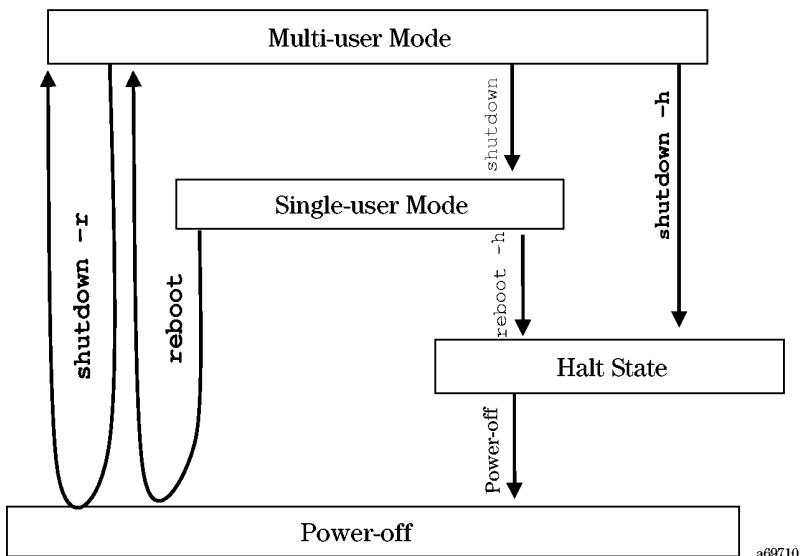
Most of the time, your system will be in **multiuser mode**. In multiuser mode, users can log in; file systems are mounted; and most system services and daemons are available.

There are occasions, however, when you may need to take your system down to **single-user** mode to perform administrative tasks. In single-user mode, users cannot log in, non-critical file systems are unmounted, and non-critical system daemons are shut down. Moving to single-user mode brings your system to a quiet state, from which you can do system backups, run `fsck`, or perform other system administrative tasks without interference from user processes.

Bringing your system to a **halt state** unmounts all file systems, and kills *all* processes. It is necessary to bring the system to a halt state before powering off an HP-UX box to move the system to a new location or install new interface cards.

17-2. SLIDE: Changing State with Shutdown and Reboot

Changing State with Shutdown and Reboot



a69710

Student Notes

CAUTION: Use the **shutdown** and **reboot** commands to properly move between single-user mode, multiuser mode, and the halt state as shown in the diagram on the slide.

The **shutdown** Command Details

Executing the **shutdown** command stops system activities in an orderly and consistent manner. The **shutdown** command performs the following tasks:

- Prompts the administrator for a broadcast message to send to all users.
- Broadcasts the warning message to all user terminal sessions.
- Grants a 60 second (by default) grace period for users to log out.

- Kills all user logins.
- Shuts down all non-critical processes.
- Unmounts all non-critical file systems.

Depending on the option specified, shutdown will either leave the system in single-user mode (no options), the halt state (if **-h** was specified), or initiate a reboot (if **-r** was specified).

Common shutdown options:

```
# shutdown -hy 600      # shutdown to a halt state in 600 seconds.  
#           # "-y" (yes) option prevents shutdown from  
#           # requesting confirmation before proceeding.  
  
# shutdown -ry 600      # reboot in 600 seconds without requesting confirmation.  
  
# shutdown -ry 0       # reboot immediately without requesting confirmation.
```

Reboot Command Details

The reboot command uses "kill -9" to kill running processes, which takes the system down quickly, but can cause problems for applications and file systems. The shutdown command shuts down applications and processes more gracefully, and thus is the preferred method for halting or rebooting the system from multi-user mode. Reboot may be used if:

- The system is already in single-user mode.
- You need to bring the system down very quickly.

Common reboot options: (For a complete list of options, see the man page for **reboot (1m)**)

```
# reboot -h          # shutdown to a halt state (only use this from single-user  
# mode).  
# reboot            # reboot
```

Rebooting or Shutting down a V-Class Server

A V-Class server consists of two systems: A Teststation and a V-Class node. The **shutdown** command needs to be run from the sppconsole window on the teststation. To reboot or shut down your V-Class server, perform the following steps:

1. Select the sppconsole window on the teststation.

test station console—message output	sppconsole—complex console
Message window <ul style="list-style-type: none"> • ccmd daemon status approx. 60 seconds after power • Failures and hard errors • Prompt = <i>hostname: path</i> 	Console window <ul style="list-style-type: none"> • Power-on self-test status • HP mode (Boot menu) • Prompt = Command: • Prompt = Console login:
tcsh	tcsh
sppuser = tsh shell <ul style="list-style-type: none"> • Commands and scripts executing on the teststation • Prompt = <i>hostname</i> 	sppuser = tsh shell <ul style="list-style-type: none"> • Commands and scripts executing on the teststation • Prompt = <i>hostname</i>

a64920

Figure 17-6.

2. Log in as root.
3. Change to the root directory. Enter:

```
# cd /
```

4. Shut down the system using the **shutdown** command. Enter:

```
# shutdown
```

Progress messages detailing system shutdown activities print to your terminal. Upon reaching run-level 0, the system:

- Restarts in single-user mode
- Displays the root prompt

5. Bring the system to a complete stop with the **reboot** command. Enter:

```
# reboot -h
```

CAUTION:

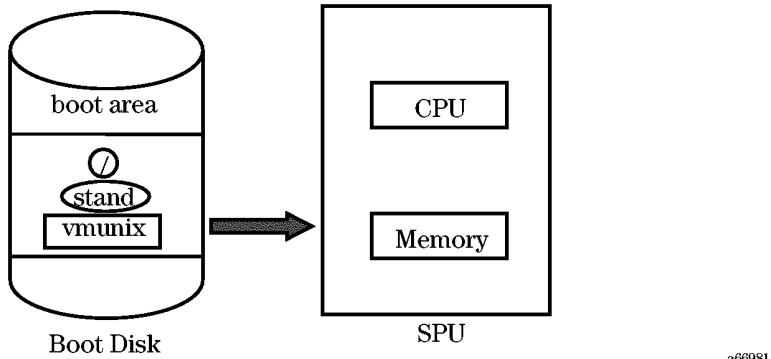
Turn the power off to the system only after the words **CPU halted** have been displayed in the sppconsole window.

17-3. SLIDE: System Boot Introduction

System Boot Introduction

Goal

- Load kernel in memory.
- Bring system up to a fully functional state.



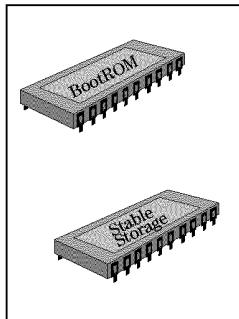
Student Notes

After shutting down your system, you will at some point need to boot it back up again. The goal of the system boot process is to load the kernel executable from the boot disk into memory and start all the daemons and services necessary to bring the system back up to a fully functional state.

17-4. SLIDE: System Boot Players

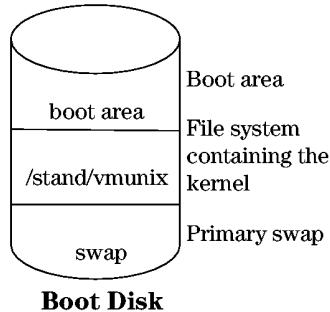
System Boot Players

SPU



BootROM

Stable storage



a66982

Student Notes

There are several players involved in the system boot process.

BOOTROM:

Every HP 9000 SPU has a BootROM chip containing the "Processor Dependent Code" firmware executable. The PDC executable is loaded in memory in the early stages of the boot process to do a hardware self-test, and identify the system boot disk.

Stable Storage:

The PDC consults "Stable Storage" to determine which disk to boot from. Stable storage is a non-volatile area of memory that contains the hardware paths of the primary boot disk, an alternate boot source, and the system console. The contents of stable storage may be modified by the administrator.

Boot Area:

Every system requires at least one boot disk. Each boot disk has a 2-MB "Boot Area" containing the utilities needed to find and load the kernel. Files in the boot area are stored in a

special "Logical Interchange Format" that is not directly viewable with `ls`, `cat` or other regular UNIX commands. The primary files of interest in the LIF area are the Initial System Loader (ISL), the AUTO file, and the HPUX utility. These will be discussed in more detail on the next slide.

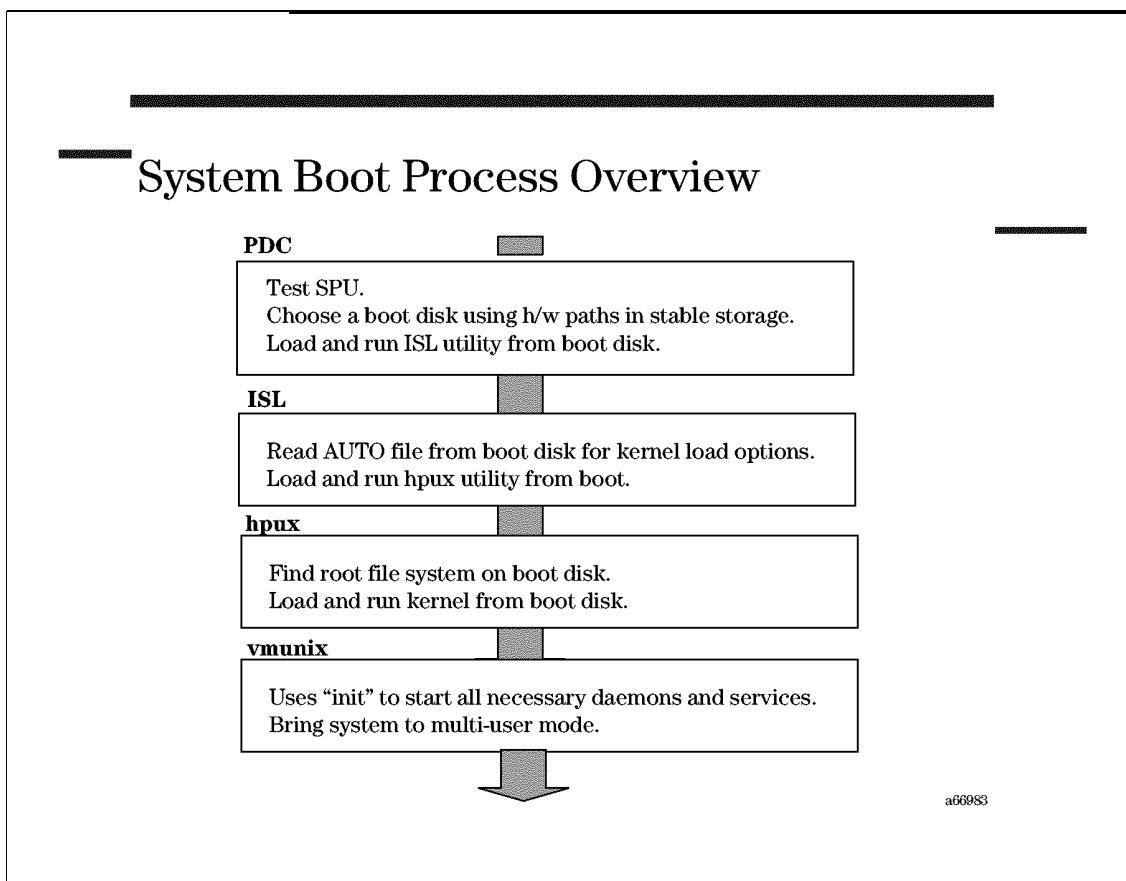
/stand/vmunix:

The boot disk must also have a file system containing a kernel to boot. The kernel is typically stored in the `/stand` HFS file system, with filename `/stand/vmunix`. The boot disk also typically contains the "/" file file system, which may be either HFS or JFS.

Primary Swap:

Finally, the boot disk contains a primary swap area that is enabled and used early in the boot process.

17-5. SLIDE: System Boot Process Overview



Student Notes

Several steps are required to bring an HP-UX system to a fully functional state.

The PDC Chooses a Boot Disk

The processor dependent code (PDC) is the first player in the boot process after the boot process is initiated. The PDC does a self-test on the SPU, then initializes the system console, and checks Stable Storage to determine which disk to boot from. Finally, the PDC loads the ISL utility from the LIF area on the chosen boot disk.

ISL Chooses a Kernel to Boot

The ISL consults the AUTO file to determine the pathname of the default kernel, and any options that should be passed to the hpx kernel loader. Finally, the ISL loads and runs the hpx utility from the LIF area on the boot disk.

HP-UX Loads the Kernel

HP-UX uses the options and kernel pathname provided by the ISL to find and load the kernel. If the ISL called hpx without any options or arguments, HP-UX loads the default kernel `/stand/vmunix`.

Kernel Brings the System to a Fully Functional State

The kernel then scans the hardware, mounts the root file system, and starts the init daemon. The init daemon starts the deamons and services necessary to bring the system up to multiuser mode.

Booting the V-Class Server

A V-Class server consists of two systems (test station and the V-Class node). Booting the V-Class server involves both systems and their hardware and software:

- Test station
 - boots the V-Class server
 - monitors the V-Class server for hardware errors
 - debugs a hung system
 - runs HP-UX
- V-Class node
 - hosts OpenBoot PROM (OBP) software
 - runs HP-UX

Due to variations in their architecture from other HP 9000 systems, V-Class systems go through a slightly different boot sequence. Once a V-Class machine is powered on, firmware controls the system until HP-UX starts running.

HP 9000 V-Class systems go through the following general sequence during the boot process:

1. **Power-on Self-Test (POST)** executes.
2. Special firmware routines called **OpenBoot PROM (OBP)**:
 - a. Probe all devices attached to the system.
 - b. Load **SPP Processor Dependent Code** (SPP-PDC) into memory and run it.
 - c. Start the HP-UX loader, which uses SPP-PDC to set up CPUs, memory, and I/O devices in a way that HP-UX understands.
3. If autoboot is enabled, HP-UX is then loaded into memory and started.

4. HP-UX goes through its initialization process and begins normal operation, ready for you to log in.

17-6. SLIDE: Autoboot versus Manual Boot

Autoboot versus Manual Boot

Autoboot

- System boots without admin intervention
- Uses default boot disk kernel
- Normal mode of operation

What if . . .

- The primary boot disk is damaged?
- The default kernel is unbootable?
- The root password has been lost?

A manual boot may be required!

a66984

Student Notes

Usually, the boot procedure described on the previous slide occurs without intervention from the System Administrator. The system boots from the primary boot disk using the default kernel. This is known as an **autoboot**.

However, in some cases you may wish to override the autoboot process:

- To boot from an alternate boot disk if the primary is corrupt
- To boot from an alternate kernel if `/stand/vmunix` is corrupt
- To boot to single-user mode to perform administrative tasks

The remainder of this chapter discusses the procedure used to manually boot your HP-UX system, and the functionality available at each step of the process.

17-7. SLIDE: Initiating the Boot Sequence

Initiating the Boot Sequence

From a power-off state

- Hit the power switch!

From a running system

- Multiuser mode: `shutdown -r`
- Single-user mode: `reboot`

a66985

Student Notes

The HP 9000 boot sequence may be initiated several different ways.

- From a power-off state, simply hit the power switch.
- From a halt-state, power cycle.
- From single-user mode, type `reboot`.
- From multiuser mode, type `shutdown`.

The system will proceed with an autoboot without any further interaction.

If you wish to do an attended manual boot from an alternate disk or kernel, start hitting the escape key immediately to interrupt the autoboot sequence.

17-8. SLIDE: Interacting with the PDC/BootRom

Interacting with the PDC/BootROM

Interrupting the autoboot

Power

Escape

```
menu choice: a          # some models go straight to BOOT_ADMIN
```

Useful PDC commands

```
BOOT_ADMIN> help          # view help menu
BOOT_ADMIN> search         # list all SCSI devices
BOOT_ADMIN> search ipl     # list all possible boot devices
BOOT_ADMIN> path           # list all boot paths in stable storage
BOOT_ADMIN> path pri scsi.6.0 # set scsi.6.0 as primary boot path
BOOT_ADMIN> path alt scsi.0.0 # set scsi.0.0 as alternate boot path
BOOT_ADMIN> boot alt       # boot from the alternate boot path
BOOT_ADMIN> boot pri       # boot from the primary boot path
BOOT_ADMIN> boot pri isl   # boot from the primary to isl prompt
```

a66986

Student Notes

Why Interact with the PDC?

After the boot process is initiated, the Processor Dependent Code chooses a boot disk, and continues the boot procedure without any further interaction with the administrator. In some cases, however, interaction with the PDC may be required:

- to boot from the primary boot disk, but from an alternate kernel
- to boot from a backup boot disk if the primary is corrupt
- to boot from install media to (re)install the OS

Getting to the PDC

To interact with the PDC, hit the escape key after initiating the boot process. The messages that appear on the screens that follow will vary somewhat from model to model. Some models

place you directly at a **BOOT_ADMIN>** prompt, from which you can issue the PDC commands shown on the slide. Other models place you at an initial menu similar to the one shown below.

Device Selection	Device Path	Device Type	
P0	scsi.6.0	HP	2213A
P1	scsi.5.0	HP	2213A
P2	scsi.3.0	HP	HP35480A
P3	scsi.2.0	HP	S6300.650A
b)	Boot from specified device		
s)	Search for bootable devices		
a)	Enter Boot Administration mode		
x)	Exit and continue boot sequence		
?)	Help		

Select from menu:b p0 ipl

At the **Select from menu:** prompt, you can boot from one of the devices listed by typing b p0 ipl as shown in the sample screen capture. Alternately, you can go to a **BOOT_ADMIN** prompt and issue any of the commands shown on the slide by typing **a** at the **Select from menu:** prompt.

Executing Commands at the PDC

The slide lists the commands most commonly used at the PDC. Others are shown in the table below. Note however, that not all commands are available on all models.

Table 17-1. Boot Administration Commands

Command	Action
auto	Display state of Autoboot/Autosearch flags
autosearch	Set state of Autosearch flag
autoboot	Set state of Autoboot flag
boot	Boot from primary/alternate path or specified device
date	Read/Set the Real-Time Clock
diagnostic	Show (on)/Conceal (off) diagnostic messages during boot
exit	Return to previous menu
help <i>item</i>	Display Help information for <i>item</i>
info	Display boot/revision information
information	Display boot/revision information
lan_addr	Display LAN station address
lanaddress	Display LAN station address
monitor	Display/Set monitor type
path	Display/Modify path information
reset	Reset the system
search	Search for boot device
secure	Display/set secure boot mode
show	Display the results of the previous search

V-Class Manual Boot

On HP 9000 V-Class systems, if the **autosearch** and **autoboot** flags are not enabled, or if a user presses a key during the 10-second delay, the HP-UX loader stops the boot process and displays the "HP mode boot menu." It then prompts you with

Command: and waits for your input.

For a normal boot, enter

Command: **boot**. At this point, the boot process will continue unattended.

17-9. SLIDE: Interacting with the ISL

Interacting with the ISL

Getting to the ISL

Power

Escape

```
menu choice: a          # some models go straight to BOOT_ADMIN
BOOT_admin> boot pri isl      # use "isl"
Interact with IPL? Yes      # some models go straight to ISL
```

Useful ISL commands

```
ISL> help          # list available commands
ISL> hpxx          # boot from default kernel
ISL> hpxx -is       # boot to single-user mode
ISL> hpxx ls        # list contents of /stand
ISL> hpxx -is /stand/vmunix.prev # boot using an alternate kernel
ISL> hpxx show autofile    # show the LIF AUTO file contents
```

a66987

Student Notes

Why Interact with the ISL?

After the PDC identifies a boot disk, the initial system loader (ISL or IPL) is responsible for identifying a kernel to load, and for calling the HP-UX kernel loader. Usually, the ISL reads the AUTO file and proceeds without any interaction with the administrator.

Interaction may be required, however, if

- the default kernel (/stand/vmunix) is unbootable
- you wish to boot to single-user mode rather than multiuser mode
- you forgot the root password (Booting to single-user mode logs you in as `root` at the console without prompting for a password.)

Executing Commands at the ISL

The most commonly-used ISL commands are shown on the slide.

```
ISL> help
```

This simply lists the utilities available from the ISL.

```
ISL> hpx
```

Without any options or arguments, the HP-UX utility loads `/stand/vmunix` and boots the system to multiuser mode. This is the default behavior during an unattended boot.

```
ISL> hpx -is
```

The `-is` option boots the system to single-user, rather than multiuser mode using the default kernel. This proves especially useful when an autoboot fails as a result of errors in `/etc/passwd`, `/etc/fstab`, or other critical configuration files.

```
ISL> hpx ls
```

If your default kernel is corrupt, use this command to search for a backup kernel in `/stand`. Look for filenames in `/stand` that begin with "vmunix".

```
ISL> hpx -is /stand/vmunix.prev
```

If your default kernel is corrupt, use this command to boot from a backup kernel. This may be necessary after reconfiguring the kernel, if the combination of kernel parameters you set prove to be incompatible.

```
ISL> hpx show autofile
```

During an autoboot, the ISL reads the LIF AUTO file on the boot disk to determine which options and arguments should be passed to the HPUX kernel loader. This command displays the contents of the auto file.

```
ISL> hpx set autofile "hpx /stand/vmunix.custom"
```

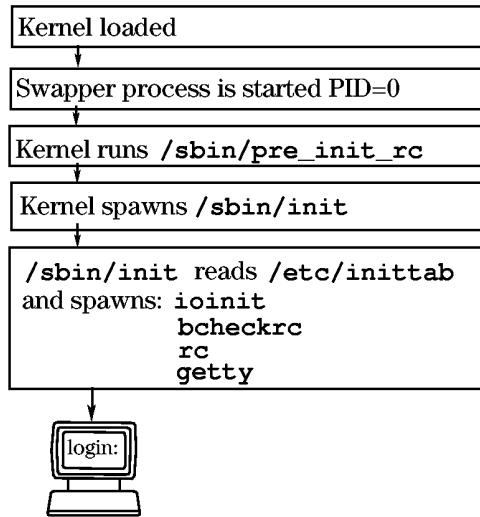
The `hpx` command can also be used to change the contents of the LIF AUTO file to modify the default kernel used during an autoboot.

NOTE:

On a V-Class system the step of transferring to the ISL is bypassed. You specify the `boot` command and the name of the alternate kernel (as an example) right at the PDC prompt. The same arguments are used for the `boot` command on the V-Class systems as are used with the `hpx` command on other systems.

17-10. SLIDE: What Happens after the Kernel Is Loaded?

What Happens after the Kernel Is Loaded?



a66988

Student Notes

After the kernel is loaded in memory, several steps are still required to bring the system to a fully functional state.

- The kernel starts the swapper daemon.
- The kernel calls `/sbin/pre_init_rc` to `fsck` the root file system.
- The kernel calls `/sbin/init`.
- The init process oversees the remainder of the boot process:
 - calls `/sbin/ioinit` to scan the hardware and build the kernel `iotree`
 - calls `/sbin/bcheckrc` to check file systems listed in `/etc/fstab`
 - calls `/sbin/rc` to start additional services (lp, cron, CDE etc.)

- calls `/usr/sbin/getty` to display login prompts on hardwired terminals

17-11. SLIDE: Run Levels

Run Levels and `init`

Init starts services in stages (run levels)

Possible run levels	= 0, s, 1, 2, 3, 4, 5, 6
Lower run levels	= few services available
Higher run levels	= more services available

Simplified example:

Run level:	Services available:
3	syncer, spooler, CDE
2	syncer, spooler
1	syncer
0	

a66989

Student Notes

After the kernel is loaded in memory, the `init` daemon is responsible for bringing the system to a fully functional state. How does `init` ensure that the required services and deamons are started in an orderly fashion?

`Init` starts system services in stages known as **run levels**. A run level is a system state in which a specific set of processes is allowed to run. The run level your system is at determines what functionality and services are available.

- More services are available at higher run levels.
- Fewer services are available at lower run levels.

Your system comes with several predefined run-levels: run-levels 0, 1, 2, 3, 4 and run-levels s and S.

Run-level 0	is reserved for system shutdown. When running in run-level 0, the system performs the normal shutdown procedure, thereby stopping all processes and halting the system.
Run-level s	is a special run-level reserved for system administration tasks. It is also referred to as single-user run-level meaning it is reserved for a single user, typically, the system administrator. For example, shutting down the system (<code>/sbin/shutdown</code>) brings you to run-level s.
Run-level S	is similar to run-level s. In run-level s only the physical system console has access to the operating system, whereas in run level S (uppercase S) the capabilities of the system console are switched to the terminal where you are logged in, thus making it the virtual system console.
Run-level 1	Similar to single-user, but file systems are mounted and the syncer is running. This run level can also be used to perform system administration tasks.
Run-level 2	Multiuser state. This run level allows all users to access the system.
Run-level 3	For HP CDE users, HP CDE is active at this run level. Beginning with HP-UX Release 10.20, CDE is the default user desktop environment. Also at run-level 3, NFS file systems can be exported; this capability is called Networked Multi-user state.
Run-level 4	For HP VUE users. In this mode, HP VUE is active.
Run levels 5 and 6	currently aren't defined by HP, but may be defined by the system administrator.

17-12. SLIDE: Changing Run Levels with init

Changing Run Levels with init

Viewing the current run level

```
# who -r
. Run-level 3 Dec 1 12:13 3 0 S
    ↑   ↑   ↑   ↑
    |   |   |   |
    |   |   |   previous run level
    |   |   |   # times at this level since boot
    |   |   |   current run level
    |   |   |   when this level was entered
    |   |   |   current run level
```

Changing run levels

```
# init 4      # moves up to run level 4
# init 2      # moves down to run level 2
# init 3      # moves back up to run level 3
```

a66990

Student Notes

You can determine your current and previous run level with the `who -r` command. You may also change your system run level with the `init` command as shown on the slide. Although it is possible to change run levels with the `init` command, practically speaking, this is rarely done.

Try the sample `init` commands shown on the slide, then answer the following questions:

1. What happens when you move up to run level 4? Do any additional services appear to start?
2. What happens when you move from run level 4 to run level 2? Do any services disappear?
3. How does changing run levels affect your users?
4. When is it useful to change run levels?

17-13. SLIDE: Configuring init via /etc/inittab

Sample inittab file

```
### change the default run level here
init:3:initdefault:

### pre-configured lines required for boot - don't change!
iomin:sysinit:/sbin/iomintrc          # scan h/w and create dev files
muxi:sysinit:/sbin/dasetup            # initialize console mux
stty:sysinit:/sbin/stty 9600          # set console baud rate, etc.
brcl:bootwait:/sbin/bcheckrc         # run fsck on file systems
cpri:bootwait:/sbin/cat /etc/copyright # display copyright
sqnc:wait:/sbin/rc                   # start add'l daemons and svc's

### these lines display login prompts
cons:123456:respawn:/usr/sbin/getty console # login prompt on console
t0p1:234:respawn:/usr/sbin/getty -h tty0p1 H # login prompt for tty0p1
t0p2:234:respawn:/usr/sbin/getty -h tty0p2 H # login prompt for tty0p2
```

(Note: some arguments are truncated from commands to fit on slide.)

a69711

Student Notes

We have seen that the run level of an HP-UX system is controlled by **init**. The actions of **init** are, in turn, controlled by a configuration file called **/etc/inittab**.

Fields in inittab

Each line in the inittab file contains several fields. Consider the following example:

cons:123456:respawn:/usr/sbin/getty console console
(a) (b) (c) (d)

Label: A one-to-four character unique label for the entry.

Run-level: Defines run levels at which the inittab entry will be processed. If the field is blank, the entry is valid for all run levels.

Action: Defines what action **init** should take on the **inittab** entry. Some of the more common actions are described below. Other actions are defined on the **inittab(4)** man page.

- initdefault:** Defines the default system run level.
- boot,** **bootwait,** **sysinit:** Execute only during the boot process.
- wait:** Don't proceed until this process dies.
- respawn:** Monitor and restart the process if it dies.

The **sqnc /etc/inittab** Entry

In the past, much more of the system startup process was configured in the **/etc/inittab** file. These days, most system services are started and stopped by the **/sbin/rc** daemon, which is called by **init** each time you change the system run level.

/sbin/rc calls a series of startup and shutdown scripts in the **/sbin/rc.0** through **/sbin/rc.6** directories to start system services such as cron, syncer, lp, and all the others that appear in the checklist that scrolls across your screen during the system shutdown and boot.

Changing **/etc/inittab**

The entries in **inittab** are rarely changed manually. Many lines in **inittab** are pre-configured, and are required to ensure a successful system boot. Only a few of the lines are commonly changed by the administrator.

If you need to change the default system run level, you may need to change the **initdefault** entry.

Every modem and hard-wired terminal requires a **uucgetty** or **getty** entry in the **inittab** file to generate a login prompt. These lines are added to **inittab** for you automatically by SAM when you configure a new terminal or modem device.

Other entries can be added to the **inittab** file, but rarely are. If you make a change to **inittab**, type **init q** to force the **init** deamon to scan the file for changes.

Questions

Look at the sample **inittab** file on the slide and answer the following questions.

1. What is the default run level on this system?
2. At which run levels would you see a console login on this system?
3. Why don't you see a console login at run level 0?
4. When would you see a login prompt on the ASCII terminals on this system?

17-14. LAB: Shutting Down and Rebooting Your System

Part I: System Shutdown

1. PART I

If you are at a graphics terminal, it is good practice to shutdown X-windows before shutting down your system. Otherwise, console messages generated during shutdown are often garbled by the windows on your screen.

To shutdown X-windows, log out. Back at the CDE login screen, click on the [Options] button and choose "Command line login". Immediately hit **Return** and log in as root. (If you wait too long before logging in, X-windows restarts and you will have to try again.)

2. Currently, your system should be in multiuser mode. Note which file systems are mounted, and how many processes are running.

3. Shut down to single-user mode. Then check to see what processes and mounted file systems remain. What differences do you see between single and multiuser modes?

4. From single-user mode, take your machine down to the halt state. What can you do in the halt state? Why might it be necessary to take your system down to the halt state?

```
# reboot -h
```

5. Power-off your machine before continuing on to the next part of the lab.

Part II: Booting the System

1. PART II

Power-on your machine, but immediately begin hitting **[ESC]** to interrupt the autoboot process. Proceed to the boot admin menu.

2. Search for possible boot devices on your system. How many SCSI devices does your system have? If a system isn't booting properly, this is one way of determining if all your SCSI devices are properly connected and powered on.
3. Now search for disks that contain an IPL that you can boot from. How many of your disks appear to be bootable?
4. Display your primary and alternate boot paths. Which disk is defined as your primary boot device?
5. Boot to the ISL from your primary boot disk. *Be sure to specify that you want to interact with the ISL/IPL.*
6. At the ISL prompt, get a list of valid ISL commands.

7. Interacting with the ISL may be useful if your primary kernel is corrupted and you have to boot from a backup kernel. Do you currently have a backup kernel in your `/stand` directory? (Kernel filenames usually begin with `vmunix`.)

8. By default, the system boots to multiuser mode using `/stand/vmunix`. Boot to single-user mode from the default kernel.

9. Did the system prompt you for a password when you were brought to single-user mode? When might this be helpful?

Part III: Moving from Single-User to Multiuser Mode

1. PART III

Single-user mode is useful for some system administration tasks, but your users won't be able to log in until you bring the system to multiuser mode. During an autoboot, the system automatically boots to the default run level, which is usually defined as run-level 3. What is your default system run-level?

2. In the steps that follow, we will bring your system up to multiuser mode one run-level at a time to see what happens at each step. To start, use `init` to bring your system to run level 1. Watch the console messages carefully.

3. The **init** daemon calls the **/sbin/rc** program, which is responsible for starting most of your system services and daemons. **/sbin/rc** generates a checklist of services needed at each run level. Based on the checklist on your console, put a "1" beside any services in the table below that **/sbin/rc** started while bringing your system to run level 1.

Table 17-2.

Level Service started	
	Mount file systems
	Set the system hostname
	Enable auxilliary (secondary) swap
	Start the syncer daemon to periodically flush buffer cache
	Start the "internet services deamon" that provides telnet, FTP access
	Start the mail daemon
	Start the LP "print spooler" daemon
	Start the clock (cron) daemon
	Start CDE

4. Now bring your machine to run level 2 and note the additional services that start. Update the table shown in question 3.
5. Now bring your machine to run level 3 and note the additional services that start. Update the table in question 3. Your system should now be in a fully functional state.
6. Based on the table you completed above, could you telnet to a machine that is in single-user mode?

7. Based on the table you completed above, can you print a file while in single-user mode?

 8. Reboot your system again, but this time let it boot unattended using the default boot disk and kernel.

Module 18 — Reconfiguring the Kernel

Objectives

Upon completion of this module, you will be able to do the following:

- List three reasons for reconfiguring the kernel.
- Define and contrast static versus dynamic kernel modules.
- Describe the structure of the `/stand` directory.
- Add and remove device drivers via SAM.
- Add and remove a kernel subsystem via SAM.
- Change a configurable kernel parameter via SAM.
- Boot from a backup kernel.

18-1. SLIDE: Why Reconfigure the Kernel?

Why Reconfigure the Kernel?

Reasons for reconfiguring:

- To add or remove device drivers
- To add or remove subsystems
- To modify system parameters

Methods for reconfiguring:

- SAM
- HP-UX commands

a66992

Student Notes

There are several situations in which kernel reconfiguration is necessary. The kernel must be reconfigured to change the following:

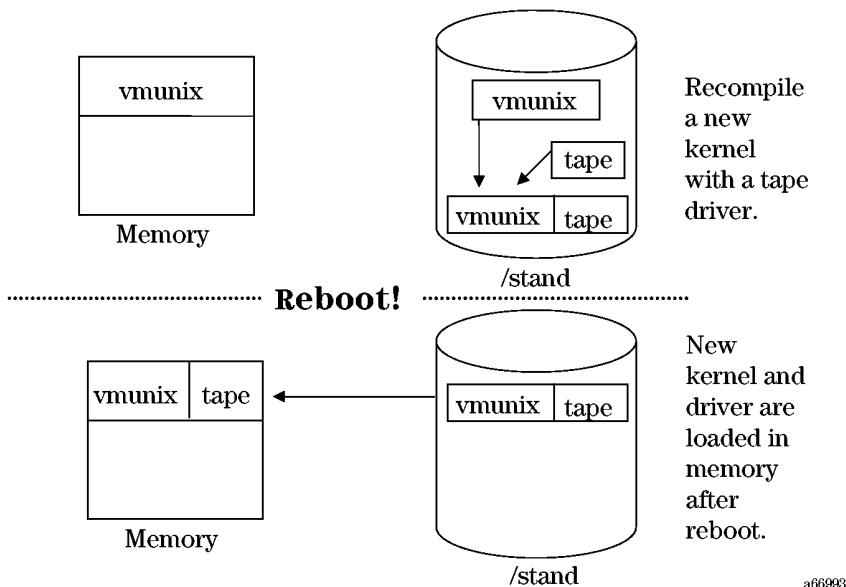
Drivers	The default kernel includes many device drivers. However, if you add a new device type to the system you may need to add the driver to the kernel in order to use the device. Conversely, you may want to remove drivers if you do not need them. If your system is memory constrained you can free up some of the space used by the kernel by reducing its size. One way to reduce the size of the kernel is to eliminate unneeded drivers.
Subsystems	The kernel includes several subsystems including LVM, CD-ROM Support, and LAN Support. If these subsystems are not configured into the kernel you will not be able to use their functionality. If you are not using a subsystem and are memory constrained you may want to remove it from the kernel.
System parameters	System parameters affect the behavior of the system as well as the size of the kernel. The size of many kernel tables are determined by system

parameters. For example, the maximum number of concurrent processes is determined by the size of the process table. The system parameter `nproc` determines the size of the process table. Extreme care should be taken when modifying system parameters. Often times, when installing software, such as databases, the software vendor will advise you to modify system parameters in order for their product to function properly.

You can modify the kernel either by using SAM, the System Administration Manager utility, or by using HP-UX commands. In most cases, the system must be rebooted to load the new kernel and put the modifications in place.

18-2. SLIDE: Static Kernel Modules

Static Kernel Modules



Student Notes

An HP-UX kernel is constructed from multiple kernel modules. Adding additional modules provides additional functionality to your kernel, and removing a module removes functionality.

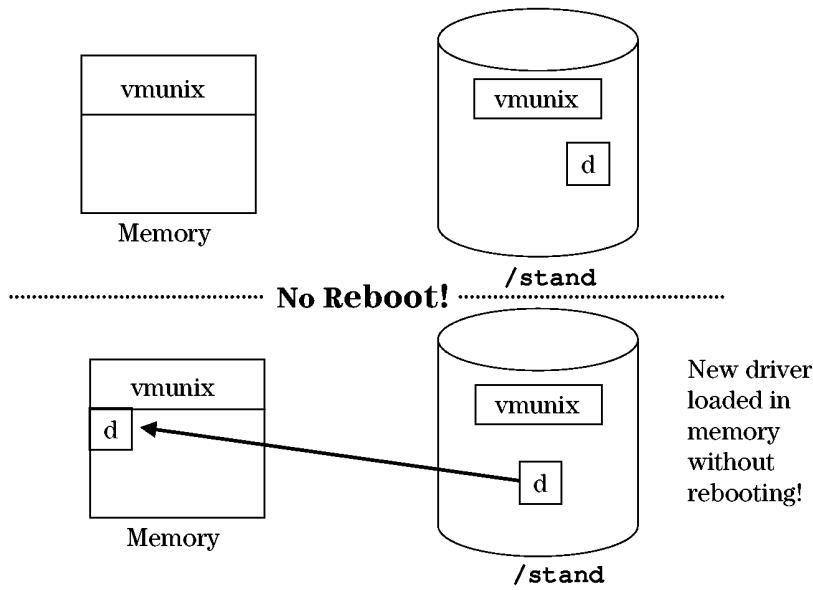
Prior to version 11.x of HP-UX, adding or removing a driver or subsystem module required the administrator to

- Rebuild the entire kernel on disk.
- Then reboot the system to move the new kernel into place.

Thus, the kernel was essentially **static**. Changes could not be made to the running kernel without a system reboot.

18-3. SLIDE: Dynamic Kernel Modules

Dynamic Kernel Modules



Student Notes

Version 11.x of HP-UX introduced the infrastructure necessary to support dynamically loadable kernel modules (DLKMs). DLKM drivers and subsystems can be dynamically loaded into the running kernel as needed, then unloaded when no longer needed. This offers a number of advantages:

High availability: DLKMs don't require a system reboot to add needed drivers and subsystems.

Efficiency: DLKMs that are no longer needed by the kernel can be unloaded.

DLKM is being implemented in phases. Though the DLKM architecture is in place at version 11.0 of HP-UX, all the system drivers and subsystems are still **static** at this point. Subsequent releases will offer DLKM drivers and support online replacement of DLKM modules.

18-4. SLIDE: Using SAM for Kernel Configuration

Using SAM for Kernel Configuration

Kernel Configuration (m/1450kmo)						
						Help
Pending Kernel Based Upon: Current Kernel						
Drivers						1 of 63 selected
Name Current State Pending State Class Type Load Module At Root? Description						
cent14	In	In	Driver	Static	N/A	Parallel
CharDrv	In	In	Driver	Static	N/A	Simple Ch
GSCtoPCI	Out	Out	Driver	Static	N/A	PCI Bus A
asio0	In	In	Driver	Static	N/A	Built-In
asynedsk	Out	Out	Driver	Static	N/A	Asynchronous
audio	In	In	Driver	Static	N/A	Audio Dri
autowx0	Out	Out	Driver	Static	N/A	MO Autouch
beep	In	In	Driver	Static	N/A	Non-HIL B
c720	In	In	Driver	Static	N/A	SCSI Inte
ccid	In	In	Driver	Static	N/A	Unknown C
core	In	In	Driver	Static	N/A	Core IO C
cs89	Out	Out	Driver	Static	N/A	CS89 Disk

a64924

Student Notes

To configure the kernel using SAM, select Kernel Configuration from the SAM functional area launcher. You can then choose from the four areas:

- drivers
- subsystems
- dump devices
- configurable parameters

As you enter each area SAM will always show the *current* values and the *pending* values. In the case of a loadable module, SAM will also show how the parameter or driver is configured (static or loadable). SAM retrieves the current value by querying the current kernel. Initially the pending values will be the same as the current values.

For drivers and subsystems the value shown will be either "In" or "Out." Drivers will also show if the module has been built and configured as a static kernel component (which cannot be removed without rebuilding the kernel and rebooting the system) or as a loadable module (which can be added or removed without rebooting the system). To change the state, highlight

the driver or subsystem and select Add or Remove from the Actions menu. Only the pending state will change at this point.

For configurable parameters the current and the pending values will be displayed. To change the value of a parameter, highlight it and select Modify Configurable Parameter from the Actions menu. Configurable parameters can be specified as an absolute value or as a formula, based on other kernel parameters.

Templates

If you wish to change your kernel by changing it to match another kernel or system file you can do so by using the "Template" feature of kernel reconfiguration.

Select **Templates** from the **Actions** menu of any of the kernel configuration subareas. Select **Load a Template**. Enter the full path name of the template file. A template file can be a kernel or configuration file that normally resides on the current system, a file copied from a remote system, or any other valid kernel or configuration file. To view the template values, use the **View->Columns** menu selection to assign a column position to the template values. Template values will then automatically be displayed in any view or modify dialog boxes. The template values are not automatically applied to the kernel you are building. To apply template values choose **Templates** from the **Actions** menu. Select **Apply Template Values**. SAM will allow you to choose to apply template values to configurable parameters and drivers or subsystems.

Tuned Parameter Sets

You can choose to set your system parameters using predefined Kernel Parameter Sets. Parameter Sets are available for the following:

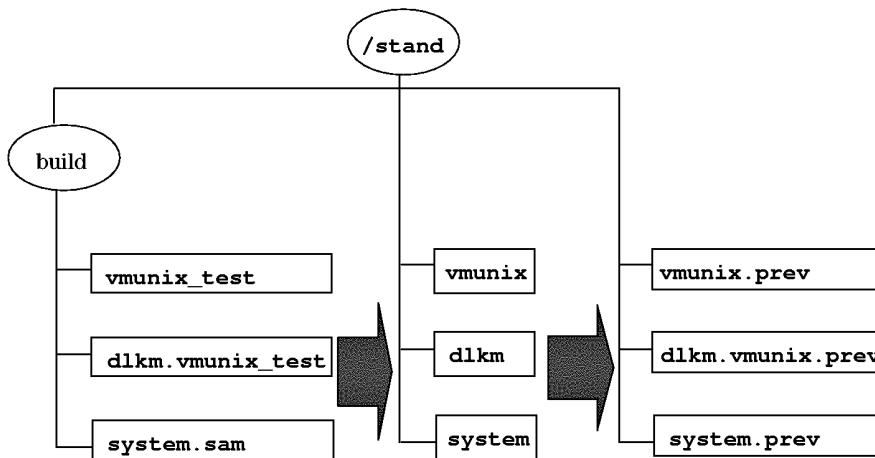
- Single-User Commercial Desktop Workstations
- General OLTP/Database Client System
- General OLTP/Database Monolithic System
- General OLTP/Database Server System
- CAE/ME/EE Engineering Workstation
- V-class Technical Server

Creating the Kernel

When you have completed all the kernel changes you wish to make using SAM, choose the **Process New Kernel** action to build a new static kernel based on the changes you have made.

18-5. SLIDE: Moving the New Kernel into Place

Moving the New Kernel into Place



a66995

Student Notes

After you have made your desired changes to the pending kernel configuration, and have selected **Process New Kernel** from the **SAM Actions** menu, SAM builds a new kernel for you.

The newly-built kernel has three components:

`vmunix_test` The static kernel executable.

`dlkm.vmunix_test` Compiled DLKM modules associated with the new kernel.

`system.SAM` Text file listing the static drivers, subsystems, and parameters incorporated in the new kernel.

SAM initially creates the new kernel in the `/stand/build` directory. At next reboot, however, the system will boot from the kernel in `/stand/vmunix`. Thus, you must move the new kernel and its associated files into place under `/stand` and reboot before your changes take effect.

You should also retain a copy of your existing kernel, in case your new kernel proves to be unbootable. Backup copies of the existing kernel files are typically kept in the **/stand** directory with **.prev** extensions.

SAM will make a backup copy of your existing kernel and immediately move your new kernel into place for you if you select **Move Kernel into Place and Shutdown/Reboot System Now**. If users are currently logged into your system, you may prefer to move the new kernel into place and reboot later. This requires several steps at the command line:

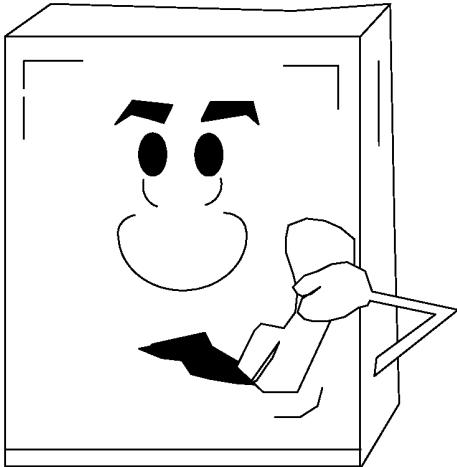
```
# cd /stand  
# cp /stand/system /stand/system.prev  
# cp /stand/build/system.SAM /stand/system  
# kmupdate /stand/build/vmunix_test  
# shutdown -ry 0
```

NOTE:

Do not use **cp** or **mv** to replace the current **vmunix** file or **d1km** directory.
Use the **kmupdate** command instead.

18-6. SLIDE: What If the New Kernel Won't Boot?

What If the New Kernel Won't Boot?



Call /stand/vmunix.prev

a538161

Student Notes

You may have made changes to the kernel using either HP-UX commands or SAM. Although SAM tries to check for invalid and incompatible settings, it is not foolproof. The only way to completely test whether your new kernel is good or not is to attempt to boot from it.

If the system will not boot from your new kernel, or if it boots but does not run acceptably (for whatever reason), you need a way to get back to where you were before you started. That involves booting from the backup copy of the kernel, that you made before beginning the whole procedure.

Booting from the Backup Kernel

If your new kernel doesn't boot, you will have to boot from your backup kernel instead. You DID make a backup kernel, didn't you?

To boot the backup kernel, `/stand/vmunix.prev`, redirect the boot sequence provided by the Boot Console User Interface by pressing the **[ESC]** key until the message,

Terminating selection process is displayed.

The automatic boot sequence has now been halted and you are in fully "attended" or interactive mode.

Ask to boot from the primary boot path and interact with ISL. Type the boot command:

```
ISL> hpx /stand/vmunix prev
```

18-7. TEXT PAGE: Manually Tuning an HP-UX 10.x Kernel

The Cookbook

Prior to version 11.x of HP-UX, the procedure for configuring the kernel was slightly different. A 10.x kernel consists of just two components: a single `/stand/vmunix` static kernel executable, and a system file that defines which drivers and subsystems are included in the kernel.

At 10.x the system file and the associated kernel executable can be configured via SAM or from the command line. The 10.x and 11.x kernel configuration SAM screens are very similar. If you wish to configure your 10.x kernel from the command line, use the following cookbook procedure.

1. Change directory to the build environment:

```
cd /stand/build
```

2. Create a kernel parameters file which you can then edit. Extract the parameters from the currently running system.

```
/usr/lbin/sysadm/system_prep -s system
```

3. Use your favorite editor to edit the kernel parameter file. This is an ASCII text file containing a list of drivers, subsystems, and parameters to define in the new kernel. To remove a driver or subsystem, simply delete the associated line in the system file. To add a driver or subsystem, simply add a line for the desired driver or subsystem to the end of the system file. To change a kernel parameter add a line to the end of the file that lists both the parameter name and value separated by one or more spaces.

```
vi system
```

4. Build a new kernel

```
/usr/sbin/mk_kernel -s ./system
```

This creates `/stand/build/vmunix_test`.

5. Save the old `system` and `vmunix` files in case something goes wrong. You will still have a bootable kernel.

```
mv /stand/system /stand/system.prev  
mv /stand/vmunix /stand/vmunix.prev
```

6. Move the new system and kernel files from the build environment into their proper place, ready to use after the next reboot. Note that the `kmupdate` command is new with version 11.x, so at 10.x the kernel and system files must be moved into place manually.

```
mv /stand/build/system  /stand/system  
mv /stand/build/vmunix_test  /stand/vmunix
```

7. Reboot the system to test the new kernel.

18-8. TEXT PAGE: Some Configurable Parameters

This is a list of some of the system parameters and their descriptions. For details on each of these parameters, refer to SAM online help screens.

Operating System Parameters

Accounting Subsystem

<code>acctresume</code>	Resume accounting due to disk usage.
<code>acctsuspend</code>	Suspend accounting due to disk usage.

Asynchronous I/O Subsystem

<code>aio_listio_max</code>	Max number of AIO operations that can be specified in a <code>lio_list()</code> call.
<code>aio_max_ops</code>	Maximum number of AIO operations that can be queued at any time.
<code>aio_physmem_pct</code>	Maximum number of AIO operations that can be specified in a <code>lio_list()</code> call.
<code>aio_prio_delta_max</code>	Maximum slowdown factor; greatest priority reduction allowed in a <code>aio_reqprio</code> field.

Dump Parameters

<code>alwaysdump</code>	Bit-mask of kernel memory pages included in dumps.
<code>dontdump</code>	Bit-mask of kernel memory pages excluded from dumps.
<code>initmodmax</code>	Maximum number of kernel modules saved by system-crash dump.
<code>modstrmax</code>	Maximum size of the kernel-module savecrash table.

Fiber Channel Subsystem

Number of Tachyon Adapters

<code>num_tachyon_adapters</code>	Number of Fiber-Channel Tachyon adapters in the system if system does not support I/O virtual addressing.
-----------------------------------	---

Maximum Concurrent FCP Requests

<code>max_fcp_reqs</code>	Maximum number of concurrent Fiber-Channel FCP requests that are allowed on any FCP adapter installed in the machine.
---------------------------	---

File System Related Parameters

Configurable File System Buffer Parameters

<code>bufpages</code>	Pages of static buffer cache
<code>dbc_min_pct</code>	Minimum dynamic buffer cache
<code>dbc_max_pct</code>	Maximum dynamic buffer cache
<code>nbuf</code>	Number of buffer headers

Configurable Open or Locked Files Parameters

<code>maxfiles</code>	Soft limit to the number of files a process can have open.
<code>maxfiles_lim</code>	Hard limit for the number of files a process can have open.
<code>nfile</code>	Maximum number of active “open” system calls at any one time in the system.
<code>nflocks</code>	Possible number of file/record locks in the system.
<code>ninode</code>	Maximum number of open inodes which can be in-core.

Configurable Asynchronous Write Parameter

<code>fs_async</code>	enable/disable asynchronous disk writes
-----------------------	---

Configurable VxFS (Journaled) File-System Parameter

<code>vx_ncsize</code>	Memory space reserved for VxFS directory pathname cache.
------------------------	--

—Logical Volume Manager (LVM)

<code>maxvgs</code>	Maximum volume groups on system
<code>no_lvm_disks</code>	No volume groups on system (workstations only)

Memory Swap Subsystem

Configurable Parameters for Memory Paging

<code>maxswapchunks</code>	Maximum number of swap chunks
<code>nswapdev</code>	Maximum number of device swap partitions

<code>nswapfs</code>	Maximum number of file system swap partitions
<code>swapmem_on</code>	Enable pseudo-swap space allocation
<code>swchunk</code>	Size of each swap chunk (in units of 1 KB)

Variable Page Size Parameters

<code>vps_ceiling</code>	Maximum system-selected page size in Kbytes.
<code>vps_chattr_ceiling</code>	Maximum chattr-selected page size in Kbytes.
<code>vps_pagesize</code>	Default user page size in Kbytes.

Process Management Subsystem

Configurable Parameters for Process Management

<code>maxdsiz</code>	Maximum size of the data segment (in bytes) of an executing process.
<code>maxssiz</code>	Maximum size of the stack segment (in bytes) of an executing process.
<code>maxtsiz</code>	Maximum size of the shared text segment (in bytes) of an executing process.
<code>maxuprc</code>	Maximum number of simultaneous processes per user.
<code>nkthread</code>	Maximum number of kernel threads allowed on the system at one time.
<code>nproc</code>	Maximum total number of processes that can exist simultaneously in the system.
<code>timeslice</code>	Time slice allocation between competing processes

Character-Mode I/O Streams Parameters

<code>NSTREVENT</code>	Maximum number of outstanding streams bufcalls that are allowed to exist at any given time on the system.
<code>NSTRPUSH</code>	Maximum number of outstanding streams modules that are allowed to exist in any single stream at any given time on the system.
<code>NSTRSCHED</code>	Maximum number of streams scheduler daemons that are allowed to run at any given time on the system.
<code>STRCTLSZ</code>	Maximum number of control bytes allowed in the control portion of any streams message on the system.

STRMSGSZ	Maximum number of bytes that can be placed in the data portion of any streams message on the system.
nstrpty	System-wide maximum number of streams-based PTYs that are allowed on the system.
stremppipes	Force all pipes to be streams-based.

System V Inter-Process Communication Mechanism

IPC Message Parameters

mesq	Enable/disable IPC messages (Series 700 only).
msgmap	Dimensions the resource map used to allocate. The buffer space for messages.
msgmax	Message maximum size.
msgmnb	Max number of bytes on the message queue.
msgmni	Number of message queue on the system.
msgseg	Number of segments in message queue.
msgssz	Message segment size.
msgtql	Maximum number of total messages on system.

Semaphore Related Parameters

sema	Enable/disable semaphores (workstations only)
semaem	Maximum value-change limit.
semmap	Size of free-semaphore resource map.
semnni	Number of semaphore identifiers system-wide.
semnns	Maximum user-accessible semaphores system-wide.
semnnu	Number of semaphore undo structures.
semume	Maximum semaphore undo entries per process.
semvmx	Semaphore maximum value.

Shared memory related parameters

shmem	Enable/disable shared memory (workstations only).
shmmax	Maximum shared memory segment in bytes.

<code>shmmni</code>	Shared memory maximum number of identifiers.
<code>shmseg</code>	Maximum number of shared memory segments that can be attached to a process at any given time.

VME I/O Subsystem Parameters

<code>vmebpn_public_pages</code>	Number of kernel I/O space pages needed by VME.
<code>vmebpn_sockets</code>	Socket domain <code>SF_VME_LINK</code> is active
<code>vmebpn_tcp_ip</code>	Maximum number of DLPI PPAs.
<code>vmebpn_tcp_ip_mtu</code>	Maximum PPA transmission unit size in Kbytes.
<code>vmebpn_total_jobs</code>	Maximum number of VME ports open concurrently.
<code>vme_io_estimate</code>	Number of 4-Kbyte kernel I/O space pages needed by VME.

Miscellaneous Parameters

<code>clereservedmem</code>	Bytes of system memory to reserve for cluster interconnect.
<code>create_fastlinks</code>	Whether to use fast symbolic links.
<code>default_disk_ir</code>	Immediate reporting for disk I/O.
<code>dst</code>	Whether to convert to daylight savings time.
<code>eqmemsize</code>	Size of equivalently mapped memory pool.
<code>ksi_alloc_max</code>	System-wide limit of queued signals that can be allocated.
<code>max_async_ports</code>	System-wide maximum number of ports to the asynchronous disk I/O driver that processes can have open at any given time.
<code>maxusers</code>	Defines macro MAXUSERS, which determines the size of system tables.
<code>ncallout</code>	Maximum number of timeouts that can be scheduled by the kernel at any one time.
<code>ncdnode</code>	Maximum number of open CD-ROM FS nodes.
<code>ndilbuffers</code>	Maximum number of DIL open device files at any one time.
<code>npty</code>	Number of telnet session device files.
<code>nstrtel</code>	Maximum translation look-aside buffer entries
<code>o_sync_is_o_dsync</code>	Enable or disable translation of O_SYNC to O_DSYNC in <code>open()</code> and <code>fcntl()</code> system calls.
<code>pfail_enabled</code>	Enable power-fail recovery.
<code>public_shlibs</code>	Allow public protection IDs on shared libraries.
<code>rtsched_numpri</code>	Number of real-time scheduling priority levels.
<code>scroll_lines</code>	ITE scroll buffer size.
<code>sendfile_max</code>	Special parameters for web-servers
<code>unlockable_mem</code>	Maximum amount of memory that will always be available for virtual memory and/or system overhead.

18-9. LAB: Kernel Configuration

Part I: Viewing, Adding, and Removing Kernel Drivers with SAM

Introduction

The kernel requires device drivers to communicate with devices on your system. Without the proper driver, the kernel is unable to access a device. As you add new devices and interface cards to your system, you may need to install new drivers. Removing unused drivers may save some space in RAM.

1. PART I

SAM is the easiest tool for managing kernel drivers. Go to the SAM kernel driver screen by selecting: **SAM --> Kernel Configuration --> Drivers**. This should give you a list of all the drivers currently installed on your system. The **Current State** column indicates which of the drivers are actually configured in your kernel. The **Pending State** column indicates which drivers will be included in the next kernel rebuild. Based on this list, can your kernel successfully communicate with a tape drive that requires the stape driver? Can you dynamically load or unload the stape driver without rebooting, or is it a **static** kernel driver that requires a reboot?

2. Select a "static" driver that isn't yet configured in your kernel, and use the **Actions** menu to add the driver to the kernel.

3. What changed on the SAM screen to indicate that the static driver you selected will be included in the next kernel rebuild?

4. Dynamically load the hwgw DLKM driver:

- Select the hwgw driver from SAM's list of available drivers.
- Select **Actions --> Add driver to kernel**.

- Click **Yes** to confirm that you want to proceed.

You should be sitting in the **Kernel Module Attributes** window at this point. DLKM drivers may be statically or dynamically loaded in the kernel. If you set the module type to **Static**, the new module will become available only at the next kernel build and reboot.

If you choose **Loadable**, the driver becomes available immediately. If you choose **Load automatically at boot**, the kernel will load the DLKM automatically during the boot process. Otherwise, the kernel will load only the DLKM as needed.

- Set the **Module Type** to **Loadable**.
- Leave **Load automatically at boot** set to **No**.
- Click **Modify**.
- Click **OK**.
- Click **Yes** in the confirmation box that follows.

5. What changed on the SAM screen to indicate that the module is immediately available for use by devices on your system?

Part II: Adding a Kernel Subsystem

1. PART II

Choose **List** --> **Subsystems**. Is your kernel currently configured to support LVM? Is your kernel currently configured to support a LAN connection? How can you tell?

2. Choose a subsystem that isn't currently configured and add it to your kernel.
3. Is the new subsystem functionality added immediately? How can you tell?

PART III: Tuning a Kernel Parameter

1. PART III

Choose **List --> Configurable Parameters**. What is the maximum number of processes your kernel can support? How many files can be open simultaneously?

2. Double the current value of the **maxusers** parameter. Does this change take effect immediately? How can you tell?

3. After changing the value of **maxusers**, note that the pending value for **nproc** changed as well. Select the **nproc** parameter and select **Actions --> Modify Configurable Parameter**. How does the definition of this parameter differ from the **maxusers** parameter? Why do you think **nproc** is defined in this manner?

Part IV: Rebuilding the Kernel

1. PART IV

Now that you have marked several pending changes to the kernel, choose **Actions --> Process New Kernel**. Before moving the kernel into place, look at the next question.

2. Shortly, SAM will ask if you want to move the new kernel into place and reboot. SAM built your new kernel in the **/stand/build** directory. Which three files or subdirectories in **/stand/build** must be moved into place in **/stand** before rebooting with the new kernel?

3. Allow SAM to move the new kernel into place and reboot.

4. When your system returns, try the following:

```
# lsdev    # Lists currently configured static kernel drivers  
# sysdef  # Lists currently running your kernel's tunable parameters
```

Did your changes take effect?

Part V: Booting from the Backup Kernel

Introduction

Hopefully, your new kernel will boot successfully and bring you up to multiuser mode. In some cases, however, the parameters in your new kernel may conflict and cause the boot to fail. Fortunately, SAM keeps a backup of your previous kernel in `/stand` called `/stand/vmunix.prev`.

1. PART V

Reboot your system to single user mode using the backup kernel.

2. Use `lsdev` and `sysdef` to check your parameter values and driver list again. Which kernel appears to be running? (Note: The `lsdev` and `sysdef` executables are in `/usr/sbin`, so you will need to do a `mount -a` first.)

3. Move your backup kernel and associated files back into place as the default kernel and reboot again.

Module 19 — Managing Software with SD-UX

Objectives

Upon completion of this module, you will be able to:

- Relate SD-UX terms and definitions.
- Install software using SD-UX.
- List software using SD-UX.
- Remove software using SD-UX.

19-1. SLIDE: Introducing SD-UX

Introducing SD-UX

Managing Software On Your Local Host

- Installing Software
- Copying Software
- Removing Unwanted Software
- Listing Software
- Verifying Installations
- Packaging Software
- Configuring Software

a538211

Student Notes

What Is SD-UX?

SD-UX is series of commands used to manage and distribute both operating system software and application software on the local host. These commands are provided as part of the HP-UX operating system and offer many options for flexible software management tasks, such as

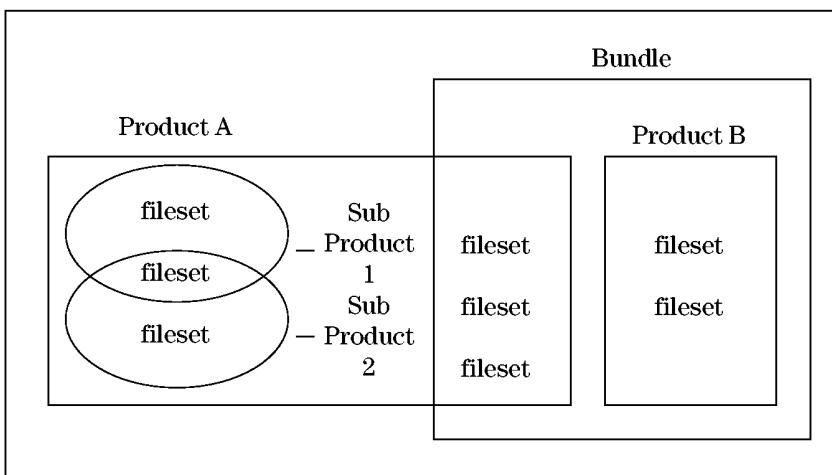
- Installing or updating software on local systems (`swinstall`).
- Building and configuring a network software server, or copying software from a distribution source or media onto a system (`swcopy`).
- Removing software from your system (`swremove`).
- Listing software that is installed on your system or on various media (CD-ROM, tape, etc.) (`swlist`).

- Verifying that software products are compatible with your system before actually installing them (**swverify**) , and reporting if modifications have been made to the installed software (**swverify**).
- Creating software "packages" that make later software installations quicker and easier (**swpackage**).
- Configure, unconfigure, or reconfigure installed software (**swconfig**).

SD-UX Provides GUI, TUI and Command Line Interfaces

All SD-UX commands can be excuted from shell scripts or the command line with a series of options and arguments. The **swinstall**, **swcopy**, **swremove**, and **swlist** utilities also provide intuitive menu-based graphical and terminal interfaces.

19-2. SLIDE: SD-UX Software Structure



a66996

Student Notes

Software in SD-UX is treated as a hierarchy of objects—bundles, products, subproducts and filesets—that make up the applications or utilities you want to manage.

Filesets	Filesets include all the files and control scripts that make up a product. They are the smallest manageable (selectable) SD-UX software object. Filesets can only be part of a single product but they could be included in several different HP-UX bundles, and more than one subproduct.
Subproducts	Subproducts are used to group logically related filesets within a product if the product contains several filesets. The same fileset can be part of more than one subproduct.
Products	Collections of subproducts (optional) and filesets. The SD-UX commands maintain a product focus but still allow you to specify subproducts and filesets. Different versions of software can be defined for different platforms and operating systems, as well as different revisions (releases) of the product

itself. Several different versions could be included on one distribution media or depot.

Bundles Collections of filesets, possibly from several different products, that are "encapsulated" by HP for a specific purpose. Bundles can be stored in software depots and copied, installed, removed, listed, configured and verified as single entities. Bundles, since they are groups of filesets, are *not* necessarily supersets of products.

SD-UX commands refer to this product structure in the form: bundle[.] or product[.]subproduct.[fileset] with periods separating each level.

Examples

Below are some examples of bundles:

HPUXEngCR700	B.10.10	English HP-UX CDE Runtime Environment
HPUXEngRT700	B.10.10	English HP-UX VUE Runtime Environment

An example of a product is:

Networking	B.10.10	HP-UX_10.0_Lanlink_Product
------------	---------	----------------------------

Examples of subproducts are:

Networking.Runtime
Networking.MinimumRuntime

The Runtime subproduct contains all the filesets in the Minimum Runtime as well as some additional filesets.

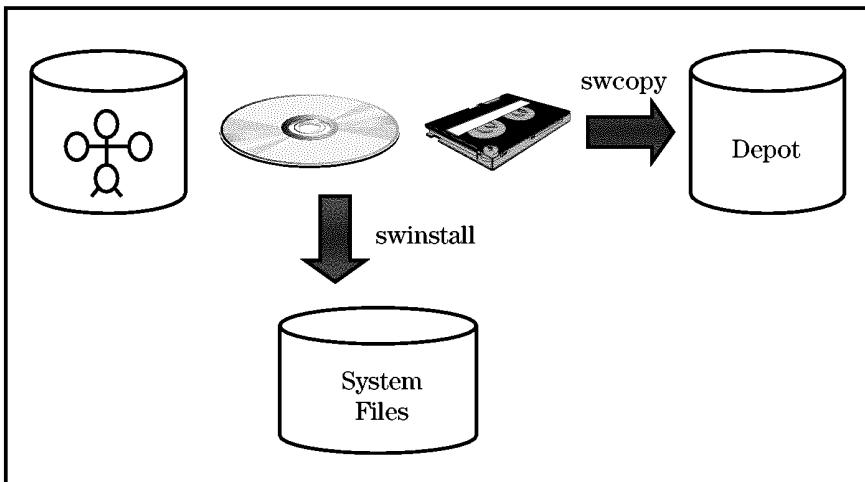
Examples of filesets are:

Networking.LAN-KRN
Networking.LAN-PRG
Networking.LAN-RUN
Networking.SLIP-RUN

These filesets are all part of both bundles, HPUXEngCR700 and HPUXEngRT700. The first three are included in both **Networking.Runtime** and **Networking.MinimumRuntime** while the last one is only part of **Networking.Runtime**.

19-3. SLIDE: SD-UX Software Depots

SD-UX Software Depots



a66997

Student Notes

SD-UX stores bundles, products, and filesets in software “depots.” Software can be copied from depot to depot using the `swcopy` command, or it can be installed to the local files and directories using the `swinstall` command. There are two types of software depots:

Directory Depot	Software in a directory depot is stored under a normal directory on your file system (by default <code>/var/spool/sw</code>). This software is in a hierarchy of subdirectories and filesets organized according to a specific media format. A directory depot can be writable or read-only. When using the SD-UX commands, you refer to a directory depot via its top-most directory. In a CD-ROM depot, this directory would be the CD-ROM’s mount point.
Tape Depot	Software in a tape depot is formatted as a <code>tar</code> archive. Tape depots such as cartridge tapes, DAT and 9-track tape are referred to by the file system path to the tape drive’s device file. A tape depot can only be created by using <code>swpackage</code> and it cannot be verified or modified with SD-UX software management commands. You cannot copy software (using <code>swcopy</code>) directly to a tape; use <code>swpackage</code> for this operation. Software in a tape depot can be

installed directly on the local host, but must first be transferred to a directory depot before it can be "pulled" by other hosts on the network. A tape depot can be accessed by only one command at a time.

A host can contain several depots. For example, a designated software distribution server on your network might contain a depot of word processing software, a depot of CAD software and a spreadsheet software depot, all on the same server.

Network Source Depots

If a depot resides on a system that is connected to a network, then that system can be a network source for software. Other systems on the network can install software products from that server instead of installing them each time from a tape or CD-ROM.

A network source offers these advantages over installing directly from tape or CD-ROM:

- Several users can "pull" software down to their systems (over the network) without having to transport the tapes or disks to each user.
- Installation from a network server is faster than from tape or CD-ROM.
- Many different software products from multiple tapes, CD-ROMs and network servers can be combined into a single depot serving all others on the network.

19-4. SLIDE: SD-UX IPD

The SD-UX IPD

swinstall		swremove	
Accounting	B.10.20		
AdvJournalFS	B.10.20		
AudioSubsystem	B.10.20		
CDE	B.10.20		
DiskQuota	B.10.20		
International	B.10.20		
InternetSrvcs	B.10.20		
JournalFS	B.10.20		
Keyshell	B.10.20		
LVM	B.10.20		

- Maintained by SD-UX utilities
- Records installed software

a66998

Student Notes

In order to intelligently manage software on a host, SD-UX must know what software is currently installed. SD-UX records this information in the “Installed Product Database”.

- When installing a new software bundle, product, or filesset, the new software must be added to the host’s IPD.
- When removing software, the software must be removed from the host’s IPD.
- Listing software is simply a matter of querying the IPD.

The IPD is stored in a directory structure under `/var/adm/sw/products` and is managed by the SD-UX utilities.

NOTE:

Never manually edit the IPD. The IPD is maintained automatically by the SD-UX utilities.

19-5. SLIDE: SD-UX Daemon/Agents

SD-UX Daemon/Agents

Process	Description
swagentd	<p>Listens for requests, then schedules a “swagent” to do the work</p> <p>Initiates communication between the target and the source</p> <p>There must be one swagentd running on the system to use “SD-UX” commands</p>
swagent	<p>Performs software management tasks</p> <p>Started as needed by “swagentd”</p>

a538215

Student Notes

SD-UX uses “software agent” (**swagent**) processes to accomplish software management tasks. When installing software, two swagents are required. One **swagent** must run on the host containing the depot from which the software is being pulled, and one **swagent** process must run on the host on which the software is installed. Other SD-UX utilities require **swagent** processes as well.

swagent processes are started on an as-needed basis by the **swagentd** daemon. The **swagentd** daemon must be running in order to perform any SD-UX software management tasks. The **swagentd** daemon starts automatically at run level two, and should run continuously on the system until shutdown.

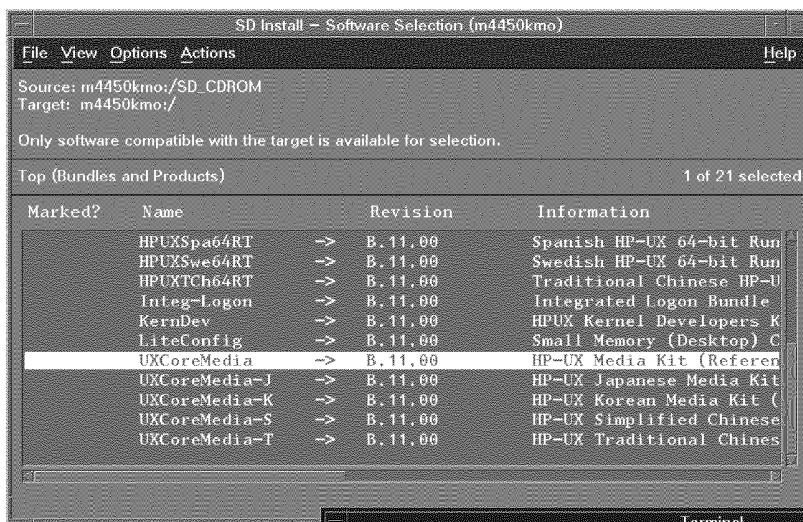
If **swagentd** dies, you can restart it with the commands below.

```
# /sbin/init.d/swagentd start      # start it
# ps -ef | grep swagentd          # check it
```

NOTE:

By default, **swagentd** is not available in single-user mode. If you wish to install software from single-user mode, you will have to manually start the **swagentd** daemon as explained above.

19-6. SLIDE: **swinstall Main Menu**



a64917

Student Notes

To run **swinstall** interactively issue the command:

```
/usr/sbin/swinstall
```

swinstall will be run using either a GUI or a TUI depending on the setting of the DISPLAY variable.

When you invoke `/usr/sbin/swinstall`, you will be prompted for the source host and depot. You will also be given a choice of what type of software will be displayed. By default all top level software is displayed. Once you have specified the source, a window similar to the one shown on the slide appears on your screen. You can skip being prompted for the source by specifying it on the `swinstall` command:

```
# swinstall -s /dev/rmt/0m          # from a local tape depot  
# swinstall -s /var/spool/sw        # from a local directory depot  
# swinstall -s /cdrom              # from a CD mounted on /cdrom  
# swinstall -s depothost:/mnt/mydepot  # from a network depot
```

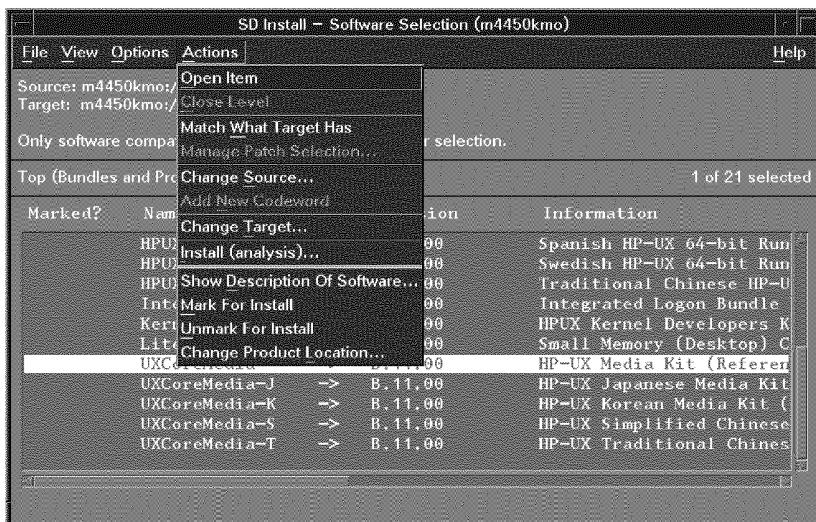
`swinstall` reads information about available software from the specified medium and displays it.

Using SAM to get to Software Distributor

Several SD-UX functions are accessible through **SAM**. Select Software Management from the **SAM** functional area launcher. When you choose Install Software to Local Host **SAM** invokes the `swinstall` command for you.

19-7. SLIDE: Select Software to Install

Selecting Software to Install



a66999

Student Notes

After launching the `swinstall` GUI or TUI, you will be presented with a list of bundles and products that are available from the current depot. You can choose to install:

1. entire bundles
2. selected products
3. selected filesets

Initially, you will only see a list of bundles, and products not contained in other bundles. To view a list of all products on the depot, select

View → Change Software View → Start with Products.

To drill down to subproducts and individual filesets, double-click on any of the listed bundles or products.

Selecting Software to Install

See the steps below for how to select products, bundles, or filesets for installation.

1. Select a product/bundle/fileset with the space bar (TUI) or mouse (GUI).
2. Select **Actions --> Show Description of Software** to view a software description.
3. Select **Actions --> Mark for Install**.
4. To meet dependencies, **swinstall** may automatically select additional filesets.
5. Repeat steps 1-4 to select additional software.
6. Select **Actions --> Install (Analysis)** to install the selected software.

Selecting Software to Update

The procedure described above can also be used to update already installed software to a newer version. Simply select the products to update using **Actions -> Mark for Install**, then choose **Actions -> Install Analysis**.

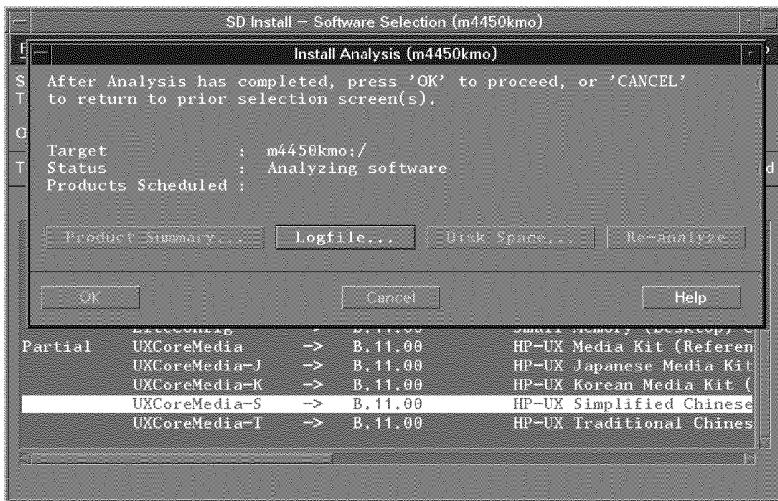
If you wish to update all of the software on your machine, you can use **Actions -> Match what Target Has** to automatically select all of the depot software that matches products and bundles already installed on your machine. Then choose **Actions -> Install Analysis** to start the update.

NOTE:

By default, **swinstall** does not reinstall filesets if the same version already exists on your machine. If you want to reinstall the same version, you can change the install options by selecting **Options -> Change Options**.

19-8. SLIDE: Starting the Update

Start the Update



a6495

Student Notes

Installing and Updating the Software Selections

After marking software to install or update and selecting Actions → Install (analysis), `swinstall` checks available disk space and software dependencies to ensure that the install will be successful. A dialog box appears so you can monitor the analysis process.

After the analysis completes, click on the Logfile and Disk Space buttons to see the results of the analysis. Assuming the analysis is successful, click the OK button to begin the install or update.

Viewing the Install/Update Log

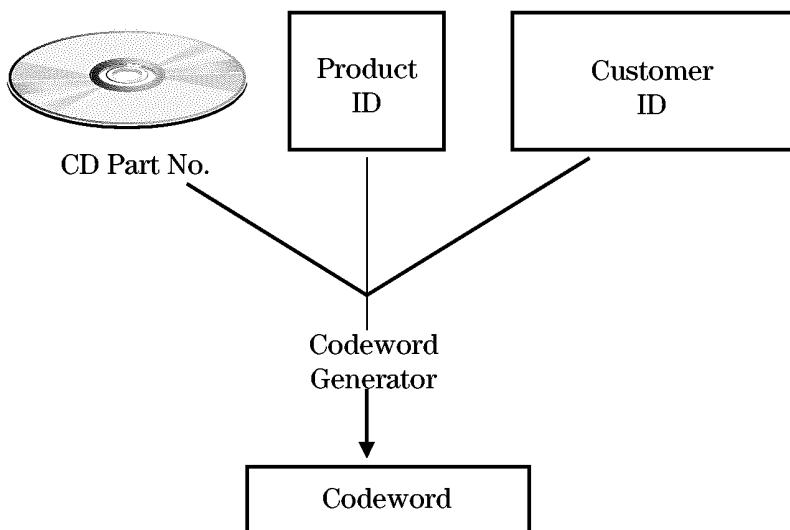
`swinstall` writes all of its actions to the file `/var/adm/sw/swinstall.log`. You should check this file for possible errors, and follow any instructions that are given.

The `/var/adm/sw/swinstall.log` file contains a description of the events and any errors that occurred during the update process. The following items are message labels and their meanings. Search the file for **ERROR**, **WARNING** or **NOTE**. These labels record anything important to know about the update process.

=====	Indicates that a task within swinstall is beginning or has completed.
ERROR	Indicates that the program cannot proceed, or that it needs corrective action. In some cases this impacts swinstall so much that it cannot continue.
WARNING	Usually indicates that the program can continue. However, it does say something went wrong or requires attention (now or later). Read the information attached to the WARNING and perform the tasks noted.
NOTE	Indicates that something out of the ordinary or worth special attention has happened. The message may require no action on your part. In other cases, the NOTE: message will require action. In some cases you must infer the action that is necessary after the update.

19-9. SLIDE: Installing Protected Software

Installing Protected Software



a669100

Most HP software products are shipped to you on CD-ROM as "protected" products. That is, they cannot be installed or copied unless you provide a "codeword" and customer ID. Software that is unlocked by a codeword can only be used on computers for which you have a valid license to use that software. *It is your responsibility to ensure that the codeword and software are used in this manner.* The codeword for a particular software product is found on the CD-ROM certificate which you receive from HP. It shows the codeword and the customer ID for which the codeword is valid. One codeword usually unlocks all the products on a CD-ROM which you have purchased. When an additional HP software product is purchased, an additional codeword will be provided by HP. Just enter the new codeword and customer ID, and they will be merged with any previously entered codewords.

A codeword for a particular customer ID and CD-ROM only needs to be entered once per target system. The codeword and customer ID are stored for future reference in `/var/adm/sw/.codewords`. SD-UX will prompt you for these codewords or numbers prior to the installation of protected software. You can enter or change the numbers via the Graphical User Interface (using `Add New Codeword` from the `Actions` menu) or by using the appropriate default (`-x codeword= xxxx` and `-x customer_id= xxx`) on the command line.

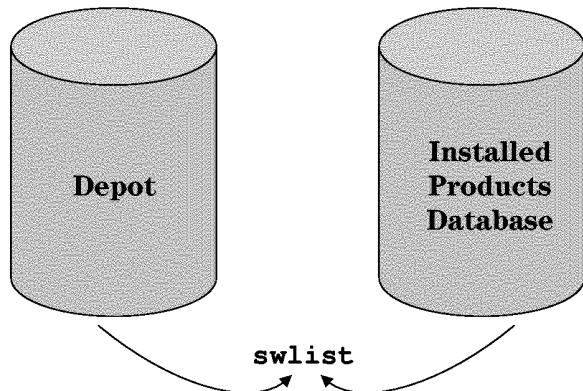
Sample Codeword Certificate

HP Sales Order Number:12345678-90123C
Date:14Feb96
DISC PART#:B3108-31083
CUSTOMER ID: 12345678-90123C
CODEWORD: 1234 5678 9012 3456 7890 1234 5678

PRODUCT NUMBER	PRODUCT DESCRIPTION
-----	-----
B2491A	MirrorDisk/UX
B3701AA	GlancePlus Pak

19-10. SLIDE: Listing Software

Listing Software



- List installed software
- List software in a depot

a669143

Student Notes

The **swlist** utility creates customizable listings of software products that are installed on your local host or placed in depots for later distribution.

With **swlist** you can

- Specify the "level" (bundles, products, subproducts, filesets or files) to show in your list.
- Show the product structure of software selections.
- Show software attributes, such as size, revision, and vendor
- Display the depots on a specified host.

Examples

```
swlist
```

Lists the software installed on your local system

```
swlist -d @ /mydepot
```

Lists the software in the depot **/mydepot**

```
swlist -l depot @ r08a6153
```

Lists the depots on the host r08a6153

```
swlist -l file LVM
```

Lists all the files that are part of the LVM product

Beginning with Release 11.00 **swlist** can also be run interactively.

Examples of **swlist** Running Interactively

Interactively list software installed on your local system

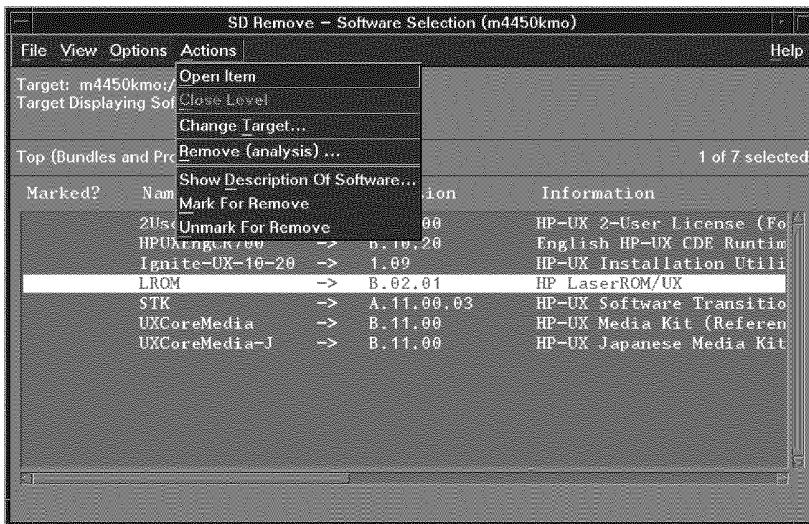
```
swlist -i
```

Interactively list software in the depot at location **/var/depot**

```
swlist -i -d @ /var/depot
```

19-11. SLIDE: Removing Software

Removing Software



a6497

Student Notes

SD-UX also provides a mechanism for safely removing software from your system. This may be useful if

- The software is no longer needed on the system.
- The disk space occupied by the software is needed for some other purpose.

The `/usr/sbin/swremove` command can be used to remove software. `swremove` uses a GUI/TUI interface much like `swinstall`. The steps are below:

1. Run `/usr/sbin/swremove`
2. Select the product/bundle/fileset to remove with the mouse or space bar.
3. Select Actions → Mark for Remove
4. Select Actions → Remove (analysis)

After checking dependencies, **swremove** removes the appropriate files and directories, and removes the product's entry from the IPD. After the remove is complete, you should check the log to ensure that the remove was successful.

19-12. SLIDE: SD-UX Command Summary

SD-UX Command Summary

SD-UX Command	Purpose
<code>swinstall</code>	Install software
<code>swremove</code>	Remove software
<code>swpackage</code>	Package software into a depot
<code>swcopy</code>	Copy software from one depot to another
<code>swlist</code>	List software in a depot or installed on a system
<code>swreg</code>	Make a depot visible to other systems
<code>swverify</code>	Verifies the integrity of installed software and depot software
<code>swconfig</code>	Configures and unconfigures installed software
<code>swacl</code>	Change access to SD-UX software objects
<code>swagentd</code>	Distribution daemon

a669101

Student Notes

As you can see, the Software Distributor is very powerful software. We have looked at how to install, list, and remove software. The table above shows some of the other functions of the software distributor.

19-13. LAB: Hands-On, Using the Software Distributor

Directions

The instructor will provide the name of a depot to use in the labs.

1. Verify that the `swagentd` is running.
 2. List the products installed on your system.
 3. Remove the Keyshell product. When removal has completed check the log file. If the product was successfully removed, the file `/usr/bin/keysh` should be gone.
 4. Reinstall the Keyshell product. When installation has completed, check the log file. If the product was successfully installed, the file `/usr/bin/keysh` should be back.

Module 20 — Patch Management (SD-UX)

Objectives

Upon completion of this module, you will be able to do the following:

- List five sources for obtaining HP-UX patches.
- Retrieve and install patches from the patch database.
- Retrieve and install patches from a tape or CD patch depot.
- List currently installed patches.
- Remove unneeded patches.

20-1. SLIDE: Why Install Patches?

Why Install Patches?

- New functionality
- New hardware support
- Bug fixes

a669103

Student Notes

As a system administrator you will likely be responsible for installing patches for the HP-UX operating system and other utilities. Patches may be installed for several different reasons:

- New functionality: HP is constantly incorporating new functionality in the HP-UX operating system. In order to make this functionality available to the installed customer base without requiring a full operating system update, HP sometimes provides these new features in the form of a patch.

Example: A recent 10.20 patch provided a new version of `sendmail` that included anti-spamming functionality.

- New hardware support: HP-UX may require a patch to support new types of interface cards and devices.

Example: A recent 10.20 patch provided support for UltraSCSI interface cards.

- General Release Patches: HP regularly releases patches to fix bugs identified in the operating system and other applications.

20-2. SLIDE: Patch Naming Conventions

Patch Naming Conventions

Patch Name Format is: **PHxx_yyyy**

where:

PH = Patch HP-UX.

xx = area patched:

CO - general HP-UX commands.

KL - kernel patches.

NE - network specific patches.

SS - all other subsystems: X11, Starbase, etc.

yyyy = a unique number

Example: **PHSS_4014** - an HP-UX subsystem patch name.

a669138

Student Notes

Browsing a list of patches can be overwhelming—there are literally hundreds of patches available for HP-UX. The patch naming convention ensures that the administrator can identify a patch's general functionality by simply reading the patch name. The slide above explains the naming convention.

Examples:

PHCO_6349	<i>SAM patch</i>
PHKL_6681	<i>Patch for process management</i>
PHNE_6372	<i>LAN products patch</i>
PHSS_6433	<i>MC/ServiceGuard patch</i>

20-3. SLIDE: Obtaining Patches

Obtaining Patches

- HP Electronic Support Center (ESC)
 - individual patches
 - custom patch manager
- The Extension Software (XSW) CD-ROM
- HP Enterprise Response Centers
 - custom patch tapes

a669104

Student Notes

Patches can be obtained from HP in several different ways.

HP Electronic Support Center Patch Database

Patches can be obtained from the HP Electronic Support Center (ESC) web site at

```
http://us-support.external.hp.com      # (Americas and Asia-Pacific)
http://europe-support.external.hp.com  # (Europe, Africa, and the Middle East)
```

The ESC web site has a searchable database containing engineering notes, software status and release bulletins, security bulletins, and other technical information.

You can also

- Browse and search a patch database.

- Download individual patches as needed via file transfer protocol (FTP).

Each patch is packaged as a **shar** archive containing a text file description of the patch and the patch itself in the SD-UX depot format.

HP Electronic Support Center Custom Patch Manager

The "Custom Patch Manager" tool is an intelligent patching agent available on the ESC web site that helps you make informed patching decisions, and guides you through patch analysis, selection, and installation.

Custom Patch Manager makes it easy for you to identify and install only the patches that apply to your system, and reduces common patching errors. Now you can create your own patch bundles, as well as benefit from patch bundles developed by HP's experts.

Suggested patches	An automated inventory of existing software and patches generates a list of suggested patches, so you no longer have to maintain a manual inventory of all your systems. Once you receive the suggested patch list, you can accept the entire patch list or shorten it by applying user-defined search and filter criteria.
User-defined Search and Filter Criteria	Powerful, user-defined searches and filters give you control over patch selection. You can select patches on a variety of criteria, such as patch criticality, keywords, file sets, and patch categories.
Automated patch conflict analysis	Patches are automatically evaluated for potential conflicts such as unsatisfied dependencies and obsolete patches. Obsolete patches are flagged and patch dependencies identified so unexpected problems can be avoided or quickly resolved.
Patch Information Online	Patch descriptions such as defect symptoms, affected file sets, and patch size are easily viewed on-line to help you determine whether or not the patch is relevant to your specific situation.
Single-Update Packaging	Selected patches are packaged into a "single-update" archive requiring at most one reboot which reduces the down time for your system.
Web-based technology	Access through a web browser such as Mosaic or Netscape lets you take advantage of the most powerful and easy-to-use technology available.
Security	A unique user identification number and password lets you take advantage of world wide web access while securing your system from unauthorized usage.
Prerequisites	Taking advantage of Custom Patch Manager is easy. All you need is a current HP-UX Series 700 or 800 HP Personalized System Support agreement, Internet access with Web browser capabilities, and FTP access so you can download patches from HP's server.

HP Extension Software Release CD-ROM

HP distributes an Extension Software CD-ROM on a bi-monthly basis which contains a tested collection of the latest HP-UX OS patches. The CD contains both general release patches, as well as patches to support new hardware and system functionality. The administrator can select appropriate patches and install them easily via `swinstall`.

HP Enterprise Response Center Patch Tapes

Customers with support contracts can call the HP Enterprise Response Center and request a custom patch tape, tailored by Response Center engineers to solve a specific problem. Response Center tapes use the standard SD-UX depot format.

20-4. SLIDE: Retrieving Patches from the Web Patch Database

Retrieving Patches from the Web Patch Database

PHKL_6681.text

PHKL_6681.depot

PHKL_6681 shar archive

- 1 Back up system
- 2 Download patch from the ESC web site to /tmp
- 3 cd /tmp
- 4 sh PHKL_6681
- 5 more PHKL_6681.text
- 6 swinstall -s PHKL_6681.depot

a669105

Student Notes

If you need to install a few patches to solve a specific bug on your system, you may choose to find and download the necessary patch from the ESC patch database on the web.

The following steps are required to obtain a patch from the ESC web site:

1. Before you install a patch, always do a full system backup.
2. Next, go to the ESC web site and follow the links to the patch database. The URL for the ESC is

```
http://us-support.external.hp.com      # (Americas and Asia-Pacific  
http://europe-support.external.hp.com  # (Europe, Africa, & the Middle East)
```

Browse the database until you find an appropriate patch, then click to FTP the patch to your local machine. Copy or move the patch to the /tmp directory.

3. Change directory (`cd`) to the `/tmp` directory.
4. Patches downloaded from the web site are distributed in a `shar` archive format. Strip off the `shar` archive wrapper by typing:

```
# sh PHKL_6681 # (This example uses patch PHKL_6681)
```

This should create two files in `/tmp`: a `.text` file describing the patch, and a `.depot` file containing the patch itself in the SD-UX depot format.

5. Read the `.text` file that describes the patch. Are there any dependencies or conflicts? Will installing this patch require a reboot? What is the purpose of the patch? Note that the `text` file even includes instructions for installing the patch. A complete description of the `.text` file fields follows this procedure.
6. `swinstall` the patch. The `-s` option shown on the slide identifies the patch depot as the source for the `swinstall` session. The command shown on the slide will launch the interactive GUI/TUI for `swinstall`. Alternately, follow the instructions in the patch's `.text` file to immediately install the patch.

The `.text` File

The text file will include the following information:

Patch name	The name of the patch.
Patch description	A one-line description that describes the cumulative patch.
Creation date	The date that the patch was created.
Post date	The date that the patch was released for general distribution.
Hardware platforms - OS releases	The hardware platforms and HP-UX OS releases on which this patch can be installed.
Products	This field lists the product name and all product revisions to which this patch applies if it is a patch for an optional (that is, non-core operating system) product. If the patch is for the core operating system, the value in this field is N/A.
File sets	This is a list of all the file sets with their respective revision, architecture, and vendor attribute that will be updated by this patch.
Automatic reboot?	Either Yes or No depending on whether or not this patch will cause a system reboot after the installation succeeded.
Status	Either General Release or Special Release depending on the support status of the patch. <ul style="list-style-type: none"> • With General Release, the patch should be installed on all systems meeting the OS, product, and dependency requirements.

	<ul style="list-style-type: none"> With Special Release, the site-specific patches for installation at one specific customer or set of customers, and other non-General Release patches. 																						
Critical	<p>Yes No followed by text. A patch is considered "critical" if it fixes a critical problem or supersedes a patch fixing a critical problem. A problem is considered critical if at least one of the following conditions is true:</p> <ul style="list-style-type: none"> Causes the system (OS/kernel) to fail, hang, crash or panic. Causes a major application to fail and thus impact severely the system's operation. Causes data loss or corruption. 																						
Category tags	<p>Identifies the type of patch. Possible values can be one or combinations of the following:</p> <table> <tbody> <tr> <td><code>critical</code></td><td>Fixes one or more of the critical conditions depicted above under "Critical".</td></tr> <tr> <td><code>panic</code></td><td>Fixes a system panic.</td></tr> <tr> <td><code>halts_system</code></td><td>Fixes a problem that leads to a system halt or hang.</td></tr> <tr> <td><code>corruption</code></td><td>Fixes a data corruption problem.</td></tr> <tr> <td><code>memory_leak</code></td><td>Fixes a memory loss problem that may lead to severe performance degradation and/or system halt.</td></tr> <tr> <td><code>defect_repair</code></td><td>Fixes a software defect.</td></tr> <tr> <td><code>hardware_enablement</code></td><td>Provides support for new hardware.</td></tr> <tr> <td><code>enhancement</code></td><td>Provides new functionality.</td></tr> <tr> <td><code>general_release</code></td><td>Should be installed on any system with the appropriate software installed.</td></tr> <tr> <td><code>special_release</code></td><td>Should only be installed under specific conditions.</td></tr> <tr> <td><code>trial_patch</code></td><td>Preliminary patch. Should be replaced by the <code>general_release</code> or <code>special_release</code> version as it becomes available.</td></tr> </tbody> </table>	<code>critical</code>	Fixes one or more of the critical conditions depicted above under "Critical".	<code>panic</code>	Fixes a system panic.	<code>halts_system</code>	Fixes a problem that leads to a system halt or hang.	<code>corruption</code>	Fixes a data corruption problem.	<code>memory_leak</code>	Fixes a memory loss problem that may lead to severe performance degradation and/or system halt.	<code>defect_repair</code>	Fixes a software defect.	<code>hardware_enablement</code>	Provides support for new hardware.	<code>enhancement</code>	Provides new functionality.	<code>general_release</code>	Should be installed on any system with the appropriate software installed.	<code>special_release</code>	Should only be installed under specific conditions.	<code>trial_patch</code>	Preliminary patch. Should be replaced by the <code>general_release</code> or <code>special_release</code> version as it becomes available.
<code>critical</code>	Fixes one or more of the critical conditions depicted above under "Critical".																						
<code>panic</code>	Fixes a system panic.																						
<code>halts_system</code>	Fixes a problem that leads to a system halt or hang.																						
<code>corruption</code>	Fixes a data corruption problem.																						
<code>memory_leak</code>	Fixes a memory loss problem that may lead to severe performance degradation and/or system halt.																						
<code>defect_repair</code>	Fixes a software defect.																						
<code>hardware_enablement</code>	Provides support for new hardware.																						
<code>enhancement</code>	Provides new functionality.																						
<code>general_release</code>	Should be installed on any system with the appropriate software installed.																						
<code>special_release</code>	Should only be installed under specific conditions.																						
<code>trial_patch</code>	Preliminary patch. Should be replaced by the <code>general_release</code> or <code>special_release</code> version as it becomes available.																						
Path name	<p>The path name is the patch's storage location on the HP Electronic Support Center systems. The only valid location for 11.x patches is <code>/hp-ux_patches/s700_800/11.x/PHxx_yyyy</code>. See also under "Hardware Platforms - OS Releases" above.</p>																						
Symptoms	<p>The external symptoms of the problem, specifically, what a user would experience.</p>																						
Defect description	<p>A detailed description of the defect that specifically addresses the explicit conditions which caused the problem (if known),</p>																						

and how to reproduce the problem (if known). Also can include methods to verify if the patch needs to be installed.

SR

All Service Request (SR) numbers addressed by this patch and all its predecessors. An SR is a formal request from a customer to have a defect resolved or a feature added to HP software.

Patch files

The full, installed path names of all files in this patch. If the patch replaces an object module in a library, the full path of the library is listed, with the object module following in parentheses. For example, if a patch replaces the object module `vers.o` in the library `/usr/conf/lib/libhp-ux.a`, the path listed would be

`/usr/conf/lib/libhp-ux.a(vers.o)`. Each entry is listed. A new copy of the `/sbin/mkboot` command is available in the patch. This new version applies to both 32-bit and 64-bit platforms, so it will be applied to both platforms.

The `/usr/share/man/man1m.Z/mkboot.1m` man page will be installed on both 32-bit and 64-bit platforms, if the file set `OS-Core.KERN-ENG-A-MAN` is installed on the system.

what(1) output

The `what(1)` output for each file or library object file listed in the Patch Files field. The `what` string is a means of identifying the software version, and thereby verifying that the patch is installed. Usually the `what` string references the name of the patch that introduced it. For cumulative patches, this may be one of the superseded patch names.

As with the Patch Files information the `what(1)` output is listed with the corresponding base file set.

An sample `what(1)` output is:

```
OS-Core.CORE-SHLIBS,fr=B.11.00,fa=HP-UX_B.11.00_32/64,v=HP:  
/usr/lib/libc.1:  
PATCH-11.00:PHCO_13284 libc.1 Nov 19 1997 16:35:42
```

Patch conflicts

All known patch conflicts, both on a file basis as well as on a behavioral basis.

Patch dependencies

All patches that must be installed to ensure proper operation of this patch.

Hardware dependencies

Specific system models to which this patch is limited.

Other dependencies

Any non-patch and non-hardware requirements that may exist.

Supersedes

The list of all patches superseded by this patch.

Equivalent patches

All known comparable patches for other hardware platforms and OS releases, not including this patch.

Patch package size

The SD-UX depot size in Kbytes.

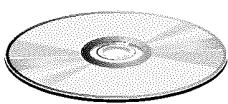
Installation instructions

The standard installation instructions common to all patches.

Special installation
instructions

Any special instructions not included in those mentioned above.

20-5. SLIDE: Retrieving Patches from Tape or CD



S/W Extension
CD

1. Do a full backup.
2. `mkdir /patchCD`
3. `mount /dev/dsk/cxtxdx /patchCD`
4. `swinstall -s /patchCD`



Depot Format
Patch Tape

1. Do a full backup.
2. `swinstall -s /dev/rmt/0m`

a669106

Student Notes

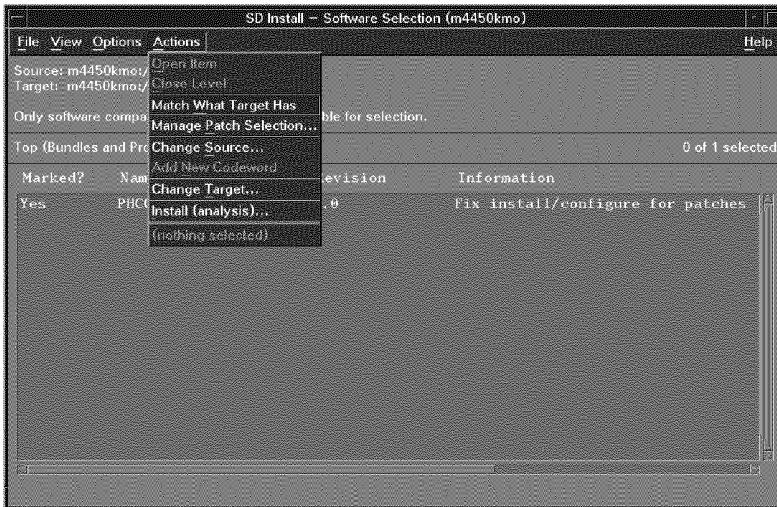
If you are working with the HP Response Center to solve an issue on your system, your Response Center Engineer may send a patch tape containing all the necessary patches to fix your problem. Downloading multiple patches, one at a time, from the patch database can be time consuming. A patch tape is often a good choice if you have many patches to install.

If you wish, you can subscribe to the Extension Software CD service. On a bi-monthly basis you will receive a CD containing the most recent HP-UX Operating System patches. (Application patches are not included on the CD.) This is a convenient way to obtain the most recent collection of proven patches. CDs are automatically mounted by the `swinstall` command.

Response Center tapes and the Extension Software CD use the SD-UX format, and thus are easily installed with `swinstall`.

20-6. SLIDE: Installing Patches with **swinstall**

Installing Patches with **swinstall**



a669107

Student Notes

Patches can be obtained from a software extension CD, a patch tape from the Response Center, or a depot extracted from a **shar** archive that was downloaded from the web. In all cases, however, patches are stored in an SD-UX depot format and can be installed with **swinstall**.

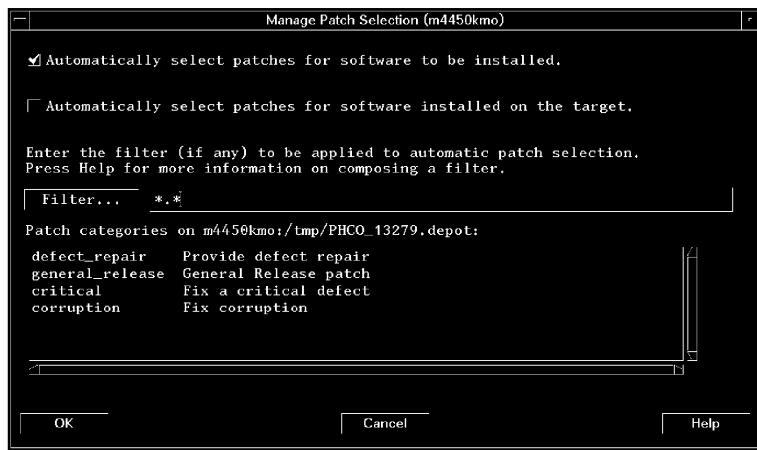
A single patch depot, however, can contain dozens of patches, which may or may not be appropriate for your system. How do you select the right patches to install? Installing all the patches in a depot can consume excessive disk space. SD-UX provides several techniques for selecting appropriate patches.

Selecting Patches via Actions --> Match What Target Has (10.x)

At HP-UX 10.x, HP recommends choosing the **Match What Target Has** option from the **Actions** menu to select patches from a patch depot. This option automatically selects all patches in the depot that match software already installed on your system. Patches for products that aren't currently installed are not marked for install.

Selecting Patches via Actions --> Manage Patch Selection (11.x)

HP-UX version 11.x introduced a new **swinstall** dialog box for managing patches.



a6499

Automatically select patches for software to be installed ensures that as new products are marked for install, the patches associated with the new product are automatically marked as well. This feature is enabled by default. **Automatically select patches for software installed on the target** compares the already installed software against the patches available in the current depot. Any applicable patches are automatically marked for install.

Selecting Patches via Patch Filtering (11.x)

Selecting all the patches that match software already installed on your system may select more patches than you truly need. Within the **Manage Patch Selection** dialog box of **swinstall**, you can specify additional patch selection criteria via a filter. Valid patch selection criteria are listed on the table.

Patch Criteria

<code>*.*</code>	<code>*.*, general_release</code>
<code>*.*,c=critical</code>	<code>*.*, special_release</code>
<code>*.*,c=panic</code>	<code>*.*,c=trial_patch</code>
<code>*.*,c=halts_system</code>	<code>*.*, c=corruption</code>
<code>*.*,c=memory_leak</code>	<code>*.*,c=defect_repair</code>
<code>*.*,c=hardware_enablement</code>	<code>*.*,c=enhancement</code>
<code>*.*,c=hand_certified</code>	<code>*.*,c=test</code>

Starting the Install

After selecting the appropriate patches in swinstall, select **Actions --> Install Analysis** to start the install.

Installing Patches from the Command Line

If you wish, you can bypass the GUI/TUI interface and install patches directly from the command line. The 11.x example below installs all the critical patches from the named depot:

```
# swinstall -s depot_name \
-x auto_reboot=true \
-x patch_match_target=true \
-x patch_filter='*.*,c=critical'
```

The next example installs all applicable patches on a 10.x machine.

```
# swinstall -s depot_name \
-x auto_reboot=true \
-x match_target=true
```

20-7. SLIDE: Listing Patches

Listing Patches

11.00

```
swlist -l patch '*.*, c=patch'
```

10.x

```
swlist -l product PH*
```

a669108

Student Notes

To determine if a patch has already been installed, you may want to list the patches already installed on your system. This is easily accomplished with **swlist**:

```
swlist -l patch '*.*,c=patch'
```

The following is sample output:

```
# Initializing...
# Contacting target "serve11"...
#
# Target:  serve11:/
#
# PHCO_12772.ADMN-ENG-A-MAN      HPAutoRaid Utilities Manual Pages    applied
# PHCO_12772.ARRAY-MGMT          HPAutoRaid Utilitirt.C-INC    applied
# PHKL_13052.CORE-KRN           OS-Core.CORE-KRN    applied
# PHKL_13052.CORE2-KRN          OS-Core.CORE2-KRN   applied
```

The patch_state can be

- | | |
|-----------|--|
| applied | The patch is currently installed. |
| committed | The patch's rollback files have been removed or were never created, since the flag <code>-x patch_save_files=false</code> was set during installation. |

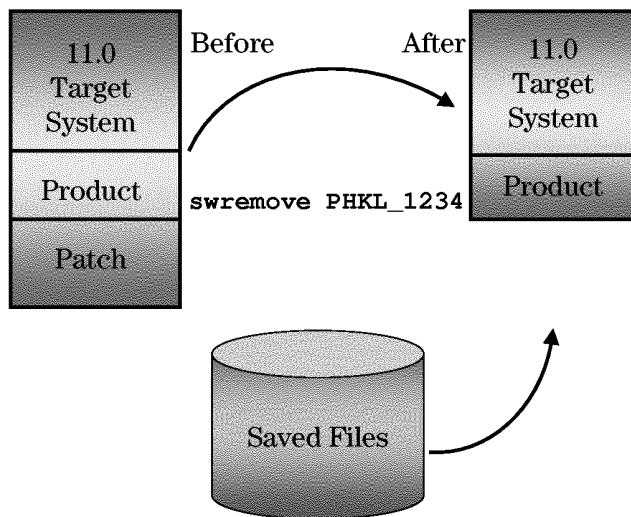
HP-UX Release 10.x

In HP-UX Release 10.x, Software Distributor cannot identify patches. To see what patches are installed, list the products that begin with *PH* using the command:

```
swlist -l product PH*
```

20-8. SLIDE: Removing Patches

Removing Patches



a669109

Student Notes

Occasionally, a new patch may conflict with existing patches or products on the system. For this reason, SD-UX makes it possible to remove a patch.

Removing Patches at HP-UX 11.x

Patches, when installed, will save the original patched files to a special holding directory under `/var/adm/sw/save`. The backup copies of the original files kept in this directory make it possible to restore files to their pre-patched state. The command below rolls back patch `PHKL_1234`:

```
# swremove PHKL_1234
```

The only problem with the rollback capability is the disk space that it consumes. The `/var` directory can quickly reach 100% of capacity if you are installing many patches. Once you are convinced that a particular patch is working properly, you may want to recover the disk space consumed by the backup copies of the original files in `/var/adm/sw/save` with the `swmodify` command:

```
# swmodify -x patch_commit=true PHKL_1234
```

Removing these files commits you to retaining the patch on the system.

Release 11.0

Patch removal and rollback is performed in accordance with the following rules:

- Removal of the base product file set of a patch file set, using `swremove`, removes all patches to that file set. For example, if `swinstall` overwrites a particular base product file set, all patches for that base are removed.
- Files saved for rollback are also removed when their corresponding base product file set is removed.
- An installed patch that has been superseded may not be rolled back unless the superseding patch is also rolled back.
- Removal of a patch automatically causes the rolled-back files to be restored, *unless*
 - the base product file set is also removed.
 - `patch_commit` option was set to *true*.
 - `patch_save_files` was set to *false*.

Release 10.0

In HP-UX Release 10.x, prepatched files are saved in the `/var/adm/sw/patch` directory. There is no software distributor command to commit patches. The `cleanup` command manages space in this directory. This tool is contained in patch PHCO_5400 and can be obtained from the Response Center.

If you do not wish to save prepatched files in HP-UX Release 10.x, touch the `/var/adm/sw/patch/PATCH_NOSAVE` file. Use extreme caution; this will make the patches impossible to remove.

20-9. LAB: Patch Management

Part I: Preliminary Steps

1. PART I

Run the `swinstall` utility to install the EchoApp utility on your machine. EchoApp should be available in a local directory depot on your machine called `/labs/depots/echoapp.dep`.

2. List the products installed on your host. Is EchoApp listed? Try running `echoapp` to see that it works: # `/opt/echoapp/bin/echoapp`

Part II: Installing a Patch from a shar Archive

1. PART II

Oftentimes administrators download and install patches from HP's SupportLine web site. For this lab exercise, that won't be necessary. The patch you will install, PHSS_01111, is already in your `/labs` directory. To start, copy the patch to the `/tmp` directory.

2. `cd` to the `/tmp` directory and unpack the patch's `shar` archive. Read the `.text` file. What problem does this patch fix?
3. Follow the directions in the patch's `.text` file to install the patch. Since we may need to remove this patch eventually, do NOT use the `patch save_files=false` option on `swinstall`. Try running `echoapp` again. Did the patch seem to work?

4. `swinstall` keeps copies of all files that have been replaced by patches in the `/var/adm/sw/save` directory. Use the `find` command to take a look at the PHSS_01111 files under this directory. Which two files were replaced by the PHSS_01111 patch?

5. Now run `swremove` to remove the patch. Again look at the files under `/var/adm/sw/save`. What changed?

6. Now remove the EchoApp product as well.

Part III: Installing and Removing Patches with Applications

1. PART III

Let's try installing EchoApp again, but this time install the product from `/labs/depots/echoapp+patch.depot`. Note that this depot contains both the EchoApp product and the PHSS_01111 patch.

2. Run `echoapp`. Based on this experiment, what is the advantage of having products and their associated patches in the same depot? (Note: This feature is new at 11.x.)

3. Now remove the EchoApp product. Afterwards, use `swlist` to see if PHSS_01111 is still installed on your system. When you remove a product, what happens to the patches associated with that product?

4. Confirm your conclusion in the previous question by checking the `/var/adm/sw/save` directory.

Part IV: (Optional) Committing Patches

1. PART IV

Install EchoApp and the PHSS_01111 patch again from the echoapp+patch.depots. Use the "du" command to check the amount of space occupied by PHSS_01111 in the /var/adm/sw/save directory.

2. As you install patches on your system, the `/var/adm/sw/save` directory can consume a significant amount of disk space. You can save some space in `/var/adm/sw/save` by committing a patch. Try committing the PHSS_01111 patch. What effect does the `swmodify` command have on the `/var/adm/sw/save` directory? Use `du` to find out.

```
# swmodify -x patch_commit=true 'PHSS_01111.*'  
# du -sk /var/adm/sw/save/PHSS_01111
```

3. Try to remove the patch. What appears to be the downside of committing a patch?

4. Can you still remove the product associated with the patch? Try it. Then `swlist` for both the EchoApp product and the patch.

Module 21 — Connecting to a Network

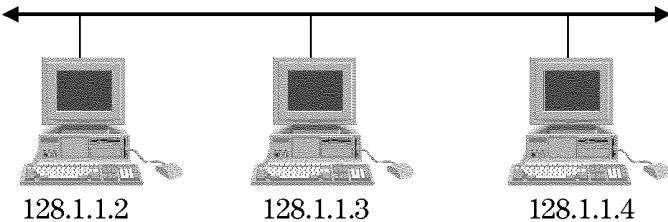
Objectives

Upon completion of this module, you will be able to do the following:

- Physically connect an hpx machine to an existing LAN
- Configure a hostname using `set_parms` or SAM.
- Configure an IP address using `set_parms` or SAM.
- Configure a default route using `set_parms` or SAM.
- Configure a host as a DNS client using `set_parms` or SAM.
- Test connectivity with `ping` and `nslookup` or SAM.

21-1. SLIDE: Setting an IP Address and Subnet Mask

Setting an IP Address and Subnet Mask



IP addresses

- Assigned to each network node
 - Must be unique
 - Define via SAM or set_parms

a669110

Student Notes

What Is an IP address?

Each machine on a local area network (LAN), whether it is a laser printer, workstation, or server, is assigned a unique Internet protocol (IP) address. Every packet of data sent across the network contains a destination IP address which determines the recipient of that packet of data, and the route taken to reach that destination.

An IP address consists of four dot (.) separated integers in the range 0-255.

IP Address Examples

- 125.67.12.43
- 15.34.67.9
- 212.1.1.34

Your network administrator or ISP should assign you an IP address, or multiple IP addresses if you have multiple LAN interface cards in your machine.

What Is a Subnet Mask?

Many companies these days have large networks that are divided into "subnetworks". Dividing a larger network into smaller "subnets" can greatly improve performance and reliability on large LANs. If your company has a subnetted network, your network manager will assign you a "subnet mask". The subnet mask, in conjunction with your IP address, determines which subnet you are attached to, and your unique host address on that subnet. In order to successfully communicate with other nodes on the LAN, you must define the subnet mask for your machine.

Subnet Mask Examples

```
255.255.0.0
255.255.255.0
255.255.254.0
```

Setting an IP Address and Subnet Mask with `set_parms`

When you connect your node to the LAN, your ISP or network manager will assign your host a unique IP address according to the address scheme implemented in your shop. You should also be given a subnet mask. When you boot your system for the first time, the `set_parms` utility automatically prompts for an IP address and several other network parameters. You can explicitly run `set_parms` any time to change your network parameters by typing:

```
# set_parms initial
```

Setting an IP Address and Subnet Mask with SAM

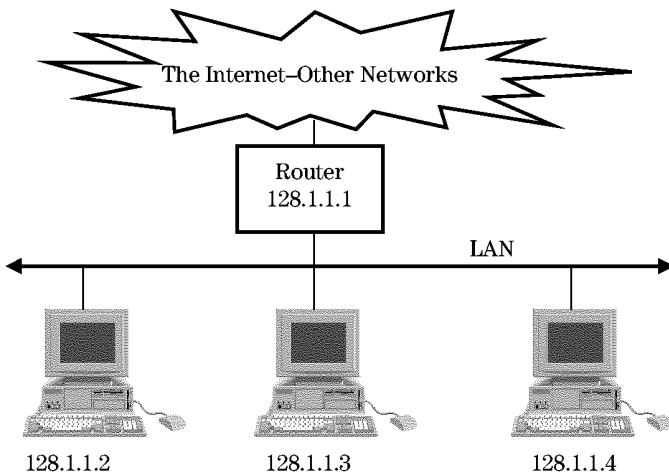
SAM also provides an intuitive interface for setting or changing your IP address and subnet mask:

```
# sam --> Networking & Communications
    --> Network Interface Cards
        (Select an interface card)
        Actions --> Configure
```

SAM displays a list of your system's LAN cards including the LAN card type, name, hardware path, status, and IP address. To change your subnet mask or IP address, select the interface card to change, select `Actions --> Configure`, and answer the questions in the dialog box that follows.

21-2. SLIDE: Setting a Default Route

Setting a Default Route



- Hosts can locally deliver data to nodes on own LAN
- Route is required to route data to other LANs
- Defined via SAM or set_parms

a669111

Student Notes

What Is a Default Route?

Without any further configuration, HP-UX should be able to deliver data to any node on your local network once your IP address and subnet mask have been set. However, if you wish to access nodes beyond your local network, your machine must have access to a **router**. A router is a device that is able to route data to other networks.

Your network manager should provide you with the IP address of a router on your LAN. Your host should pass all data destined to other networks through the default router specified by your network manager.

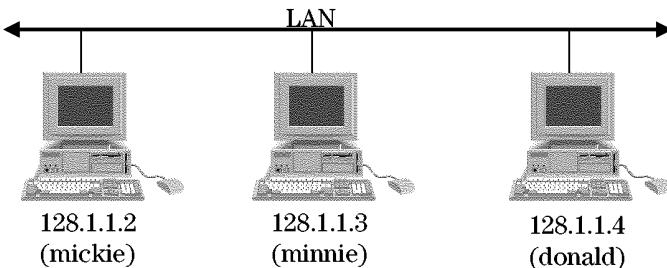
Defining a Default Route

The default router address may be set via either **set_parms** or SAM. Note that SAM uses the term **default gateway** interchangeably with **default router**:

```
# set_parms initial
# sam --> Networking & Communications
    --> Hosts
    --> Local Hosts File
        Actions --> Configure Default Gateway
```

21-3. SLIDE: Setting a System Hostname

Setting a System Hostname



Hostnames

- Mnemonic name assigned to each node
- Can be based on
 - Purpose of node
 - Primary users of node
 - Any other naming convention
- Defined via `set_parms`

a669112

Student Notes

What Is a Hostname?

Since many users find it difficult to remember IP addresses, most machines are also assigned a "hostname". A hostname is simply a unique alphanumeric "nickname" assigned to each host on the network. Hostnames are assigned based on any of the following:

- the purpose of the node
- the primary user of the node
- any other naming convention chosen by the network manager

Setting Your System Hostname

Your network manager will assign you a hostname, which must be defined on your machine using `set_parms` or SAM:

```
# set_parms initial  
  
# sam --> Networking & Communications  
--> Lan Interface Cards  
Actions --> Modify System Name
```

21-4. SLIDE: Resolving Hostnames to IP Addresses

Resolving Hostnames to IP Addresses

telnet mickie
→ 128.1.1.2

ftp minnie
→ 128.1.1.3

- Every outgoing packet requires an IP
- Three ways to resolve hostnames to IP addresses
 - /etc/hosts file
 - Domain Name Service (DNS)
 - Network Information Service (NIS)

a669113

Student Notes

Although hostnames are more memorable than IP addresses, a router must know a packet's IP address to successfully deliver data to a destination. For this reason, if a user or application specifies a packet's destination as a hostname, the system must translate the destination hostname to a destination IP address.

There are three ways to resolve hostnames to IP addresses on an HP-UX system:

- via the /etc/hosts file
- via the Domain Name Service (DNS)
- via the Network Information Service (NIS)

We will consider each method in the pages that follow.

21-5. SLIDE: Configuring /etc/hosts

Configuring /etc/hosts

Sample /etc/hosts:

128.1.1.1	mickie mailsrv
128.1.1.2	minnie
128.1.1.3	donald

- Maps: IP addresses ↔ hostnames
- Maintained on each individual host
- Usually used to resolve hostnames on small LANs
- Define via SAM or copy from another host

a669114

Student Notes

What Is the /etc/hosts File?

If you need to communicate only with hosts on your local network, your network manager may recommend using the `/etc/hosts` file to resolve hostnames to IP addresses. The `/etc/hosts` file is a configuration file maintained locally on each host to resolve hostnames to IPs.

Each line in the `/etc/hosts` file contains an IP address in the far left field, followed by the hostname associated with that IP address in the second field. In the example on the slide, hostname `donald` resolves to IP address 128.1.1.4. Hostname `minnie` resolves to IP address 128.1.1.3.

You may optionally define one or more **aliases** for each IP address, too. An alias is simply another name by which an IP can be referenced. In the example on the slide, both `mailsvr` and `mickie` resolve to IP 128.1.1.2.

Each line in `/etc/hosts` can contain a comment preceded by a # sign as well.

Configuring /etc/hosts

The `/etc/hosts` file must be modified anytime an IP address or hostname changes. Note that `/etc/hosts` is maintained locally on each host, so changes must be propagated to all hosts if you wish to maintain consistency across all machines on your network.

You can modify `/etc/hosts` several different ways:

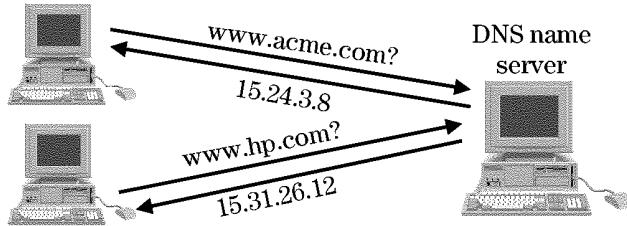
- Modify the local hosts file using SAM.
- Modify the local hosts file directly with vi.
- FTP an up-to-date `/etc/hosts` file from another host.

To get to the SAM interface for modifying your hosts file, select

```
# SAM --> Networking and Communications  
    --> Hosts  
        --> Local Hosts File  
            Actions --> Add/Modify/Remove
```

21-6. SLIDE: Configuring a DNS Client

Configuring a DNS Client



- Used to resolve hostnames on larger networks
- Clients request lookups from DNS servers
- To prevent duplicate names, each host assigned to a “domain”
- Must define the following via SAM or set_parms
 - Your DNS server address
 - Your DNS “domain” name

a669115

Student Notes

What Is DNS?

Although the `/etc/hosts` file is adequate for hostname resolution on small networks, your network manager will likely recommend using the Domain Name Service (DNS) for name resolution if you wish to access the outside Internet or other networks within your company.

Instead of referencing a locally stored `/etc/hosts` file that quickly becomes outdated, hosts using DNS send name resolution requests to a specially configured **DNS name server**. Your network manager can provide the IP addresses of a DNS name server that your host uses.

In addition to defining your local name server’s address, you need to define your host’s **DNS domain**. The slide shows hosts in two domains:

hp . com
acme . com

Each host on the internet can be uniquely identified by the concatenation of the host's hostname and domain. If two hosts have the same hostname, the domain name appended to the end of the hostname determines which host you wish to reference.

The example on the slide shows two hosts with hostname **www**. The two are differentiated from one another by the **hp.com** and **acme.com** domain names appended after **www**.

Specifying a DNS Nameserver

Configuring your host to use DNS is most easily accomplished via SAM. Go to:

```
# SAM --> DNS (BIND)
--> DNS Resolver
Actions --> Specify Name Servers
```

Up to three DNS nameservers may be defined. If one nameserver is unavailable, your system will automatically try another. Use the name server IP addresses provided by your network manager.

Specifying a DNS Domain Search List

While in SAM, you should also define your DNS domain. You may optionally list up to five other DNS domains to search when resolving hostnames. Defining additional search domains saves your users the trouble of typing long and cumbersome domain names.

Example

Without a domain search list defined, users in the **hp.com** domain may need to type:

```
# telnet corp.acme.com
```

Adding **acme.com** to the domain search list allows users to access **corp.acme.com** by simply typing:

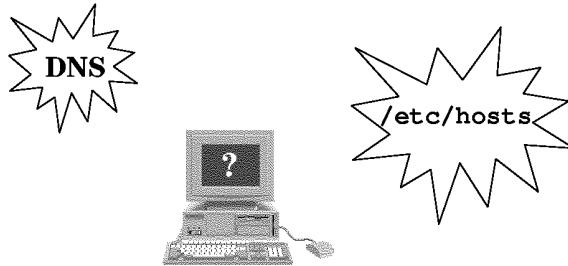
```
# telnet corp
```

SAM provides an intuitive interface for defining your own domain and a domain search list:

```
# SAM --> DNS (BIND)
--> DNS Resolver
Actions --> Set Default Domain
```

21-7. SLIDE: Choosing a Look-Up Service

Choosing a Look-Up Service



Q: Where do I look up hostnames?

A: Consult `/etc/nsswitch.conf`

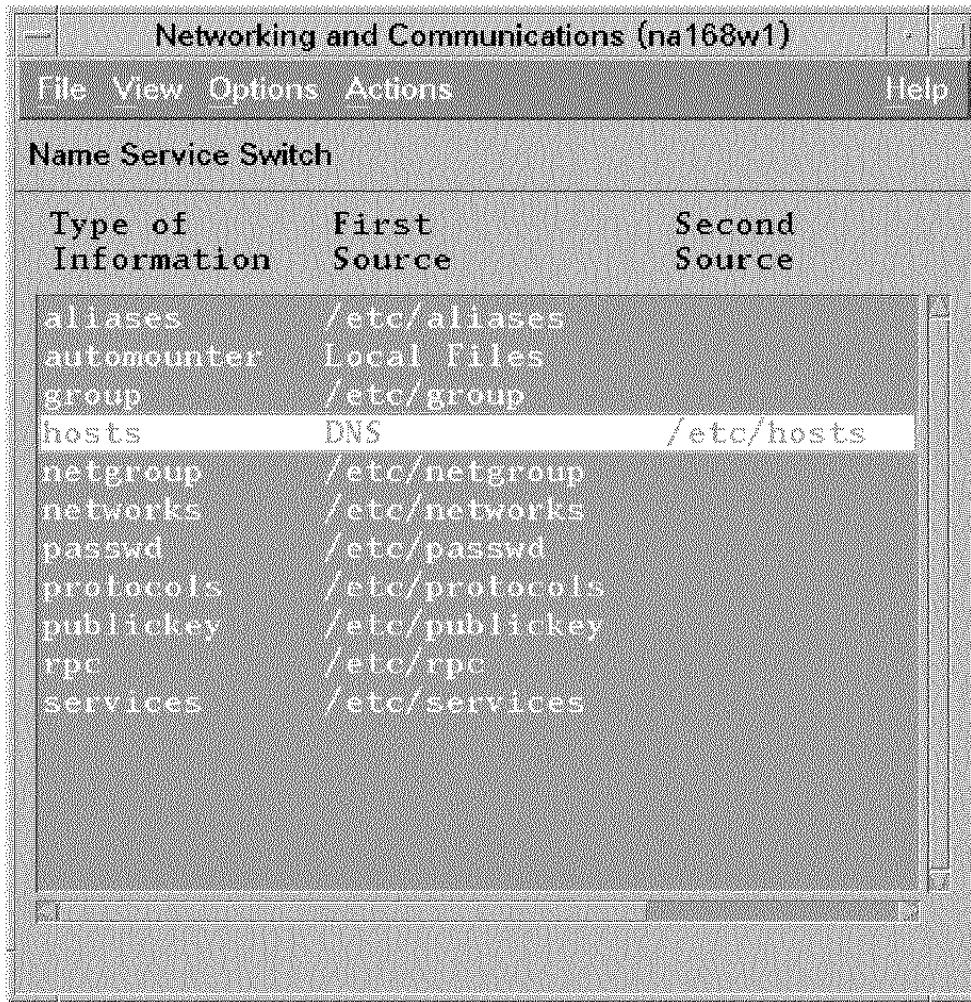
a669116

Student Notes

We've seen that hostnames may be resolved via the local `/etc/hosts` file or DNS. How does the system decide, then, which source to use? The `/etc/nsswitch.conf` file determines which source should be used for look-ups. This file can be edited manually, or can be configured via SAM:

```
# sam --> Networking and Communications
    --> Name Service Switch
        (Select a lookup type)
    Actions --> Configure Name Service Switch
```

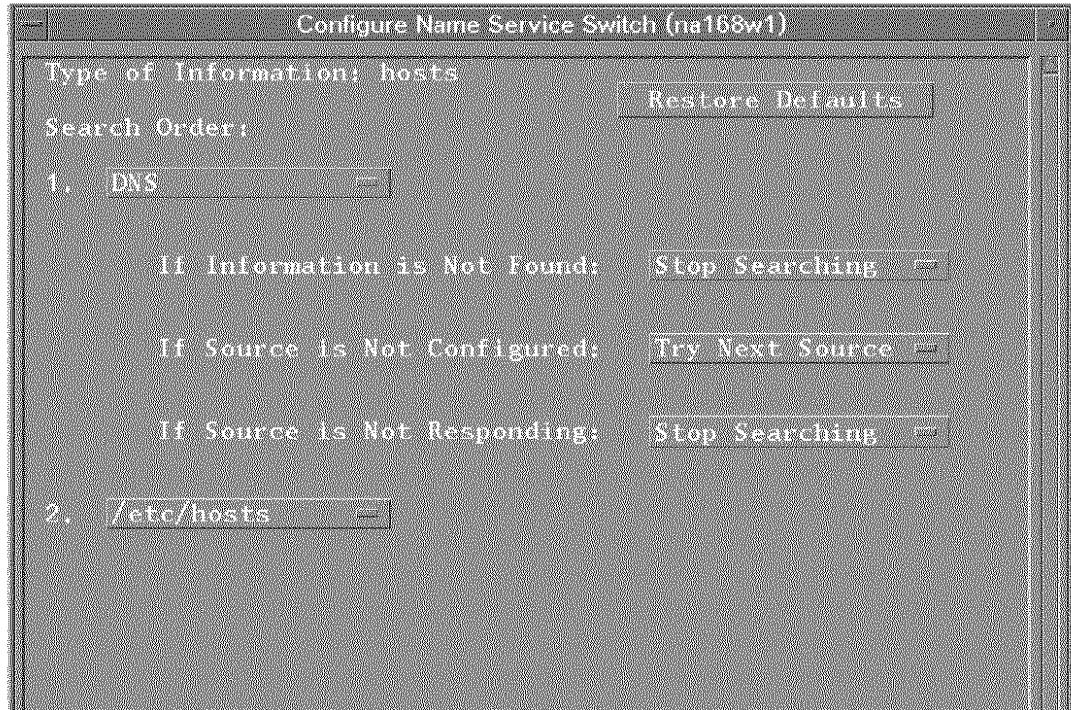
The sample system that provided the screen shot below is configured to use DNS as the primary source for hostname resolution. If the DNS server doesn't respond or isn't configured, the system resorts to using the local `/etc/hosts` file instead.



a669139

Figure 21-7. SAM Name Service Switch

SAM also lets you change the lookup source order. For instance, to force your system to resolve hostnames first from /etc/hosts, then from DNS, select the hosts line from the object list and click **Actions --> Configure Name Service Switch**. The screen that follows lets you determine which service is used when.



a669140

Figure 21-8. SAM Configure Name Service Switch

The `/etc/nsswitch.conf` file is used to configure the source order for password, group, and other types of look-ups as well, if you have access to a network information service (NIS) server. NIS configuration is not covered in this course.

21-8. SLIDE: Troubleshooting Tools

Troubleshooting Tools

Check connectivity

```
# ping 128.1.1.3  
# ping 128.1.1.4
```

Check hostname resolution

```
# nslookup minnie  
# nslookup donald
```

a669118

Student Notes

After configuring your network connection, check to ensure your configuration worked using `ping(1m)` and `nslookup(1)`.

Checking Connectivity with `ping`

The `ping` command is used to test connectivity to any IP address.

Examples

```
# ping 128.1.1.3  
# ping 128.1.1.4
```

`ping` sends a series of test packets to the specified IP and waits for a response. After a few seconds, hit `CTRL + C` and check to ensure that the packet loss shows a value of 0. Try several pings:

1. `ping` yourself

2. ping another host on your local network
3. ping your default router
4. ping a host on another network

If ping fails, check the following:

- Is the LAN card installed properly?
- Is the LAN cable connected properly?
- Did you set your IP properly?
- Did you configure a default gateway/router?
- Did you configure your subnet mask properly?

Check Hostname Resolution with nslookup

The **nslookup** utility can be used to ensure that your host is resolving hostnames properly. Given a hostname as an argument, **nslookup** responds with the corresponding IP address. **nslookup** also notes the source providing the IP address.

Example (on a host using the local /etc/hosts file)

```
# nslookup minnie
Using /etc/hosts on mickie

looking up FILES
Name:      minnie
Address:   128.1.1.3
```

If **nslookup** responds with an incorrect IP address, check the following:

1. Did you configure the name service switch properly?
2. If you are using DNS, is the name server IP address correct?
3. If you are using NIS, is the NIS domain name correct?
4. If you are using /etc/hosts, is the /etc/hosts file entry correct?

21-9. LAB: Connecting to the Network

Directions

In the exercise that follows, you will have an opportunity to explore the **set_parms** and SAM screens used to set network parameters on your system. Back in your shop, your IT department or ISP should provide you with your basic networking parameters. For this lab exercise, your instructor will provide the addresses you need:

Hostname:	
IP address:	
Subnet mask:	
Def. Router/Gateway IP address:	
DNS domain:	(optional)
DNS server address:	(optional)
Secondary DNS server address:	(optional)

There should be a shell script on your system called `/labs/clobber.net.sh`. Run the script. The script clobbers your network configuration files and reboots your machine. When your machine comes back up again, it will automatically run the **set_parms** utility, just as it would if you had just installed the operating system.

Part I: Configuring an IP Address and Hostname

1. PART I

set_parms first asks if you want to connect to the network. For this lab exercise, we do want to connect to the network, so click "OK."

2. Next, **set_parms** asks if you wish to obtain your network parameters via DHCP. DHCP is a service that can automatically provide your machine with an IP address and hostname, if a DHCP server exists on your network. For this exercise, click "NO".

3. Answer the prompts in the screens that follow to set your hostname, time zone information, and IP address. When asked if you wish to set additional network information (for example, a default gateway, netmask, and DNS and NIS information), answer NO. We will configure these parameters later via SAM.

NOTE:

If you make a mistake while entering any of the network parameters, you can run `set_parms` again any time by typing `set_parms initial`.

Part II: Checking your IP address and setting your netmask

1. PART II

After running `set_parms` and logging in, go to:

SAM -> Networking and Communications -> Network Interface Cards

to ensure that your IP is set correctly. What hardware slot number is your `1an0` card installed in? Is `1an0` enabled for use? What is `1an0`'s IP address? Are there any other LAN cards on your machine?

2. While still at the Network Interface Card screen, set your subnet mask. Without the subnet mask, you may not be able to communicate with other nodes on your LAN. Set your subnet mask for `1an0` via SAM.

Each machine on a network, regardless of the number of LAN interface cards, has a single hostname. If you have multiple LAN cards on a host, SAM requires you to configure a unique "Host Name Alias" for each LAN card. This is useful in troubleshooting situations where you may wish to access a specific LAN card on a target machine rather than the general hostname. If SAM asks you to set an alias, simply use your hostname with an "a" appended to the end.

Part III: Testing Your Network Connection

1. PART III

Try pinging your own IP address to see if your IP address is set properly. Did this work?

2. Next, try pinging a couple of neighboring machines' IP addresses. Did this work?

3. Next, try pinging a neighboring machine using its hostname. Can you guess why this fails? (Hint: Look in the `/etc/hosts` file.)

Part IV: Configuring `/etc/hosts`

You won't be able to access other hosts by hostname until you add entries for those machines to your `/etc/hosts` file. SAM offers the easiest method for viewing and modifying your hosts file. Use SAM to answer the questions that follow.

1. PART IV

Which hosts' entries are configured so far?

2. Add entries in your hosts file for the other machines in your row.

3. Try pinging your neighbors by hostname again. Does this work?

Part V (Optional): Configuring a Default Gateway and DNS

If you want to be able to access other networks or the Internet, you will have to define a default gateway and DNS server address. Both of these addresses can be configured via SAM.

1. PART V

Use SAM to configure your default gateway.

2. Use SAM to configure your default DNS domain name. Don't configure any "Other Domains to Search".

3. Now use SAM to configure the IP addresses of your DNS nameservers.

Part VI (Optional): Configure the Name Service Switch

1. PART VI

Did your DNS configuration work? If DNS is configured properly on your machine, and the DNS server is functioning, you should be able to lookup hostnames of machines outside your local network. Try an nslookup on `www.hp.com`. What happens?

2. Look carefully at the output from the nslookup you did in the previous exercise. Can you explain why you were unable to find an IP address for `www.hp.com`? (Hint: What lookup source is nslookup using at this point?)

3. Configure the name service switch via SAM to ensure that your system consults the DNS server if a hostname isn't found in the local `/etc/hosts` file.

4. After changing the nameservice switch, SAM reboots your machine. After the reboot, do an nslookup on `www.hp.com` again. Are you able to resolve `www.hp.com` now?

Solutions

1-6. REVIEW: Check Your Understanding

1. Describe the role of the system administrator.

Answer:

The system administrator is responsible for setting up and maintaining the system.

2. What does the system administrator need to understand in order to perform his or her duties appropriately?

Answer:

Not only must the administrator understand both hardware and software, but she/he must also understand the needs of the user community.

3. What are the three main categories of system administration responsibilities?

Answer:

- hardware responsibilities
- software responsibilities
- responsibilities to the users

4. What are two items in the system administrator's tool kit?

Answer:

- SAM
- Support Contract
- Instant Information
- Shell Scripts
- SupportLine

2-4. LAB: SAM

1. PART I

Log onto your workstation and start SAM. The first screen in SAM lists a number of functional areas. The following are just a few of the functional areas you may see listed:

accounts for users and groups
backup and recovery
disks and file systems
peripheral devices

Answer:

2. Which functional area would you choose to view a list of user accounts on your system? Select the appropriate area, and determine the number of accounts on your system. (Note: In some cases, a functional area may contain one or more sub-areas. You may have to drill down through several menus to get to the information you are looking for.) Explore some of the other SAM functional areas to find answers to the questions that follow.

Answer:

The functional area that lists user accounts is Accounts for Users and Groups -> Users

3. Are any tape drives configured on your system?

Answer:

The functional area that lists tape devices is Peripheral Devices -> Tape Drives

4. How many disk devices are attached to your system? Do you have any "Unused" disk devices?

Answer:

The functional area that lists disk devices is Peripheral Devices --> Disks and File Systems --> Disk Devices.

5. How many file systems are configured on your system?

Answer:

The functional area that lists file systems is Disks and File Systems --> File Systems.

6. Are there any currently scheduled automated backups on your system?

Answer:

The functional area that lists scheduled backups is Backup and Recovery --> Automated Backups.

1. PART II

Go to the **Users** object list in SAM. What type of information does the object list display for each user account? List four fields.

Answer:

Go to **Accounts for Users and Groups** --> **Users**. The users' object list by default shows the following fields:

Login Name
User ID
Real Name
Primary Group
Office Phone
Office Location

2. SAM doesn't always show all the information available for a given object list. For instance, by default, SAM doesn't list user home directory names in the Users object list.

You can modify the types and order of columns shown in the object list by launching the Column editor from the Menu Bar at the top of the screen: **View** --> **Columns**.

The **Home Directory** field is probably currently marked **Ignore**. Change **Ignore** to **5**, click **OK**, and note the change to your object list .

Answer:

A **Home Directory** column appears in the fifth column. Remaining fields automatically shift to the right.

3. Now use the column editor to put the **Start-up program** in the sixth column of the object list, and hide the **Office location** and **Office phone** fields.

Answer:

Choose **View** --> **Columns**.

Set the Column value for Startup Program to **6**.

Set the Column value for Office Location to **Ignore**.

Set the Column value for Office Phone to **Ignore**.

Click **OK**

4. You can also customize your object lists by changing the order in which they are sorted. Sort your Users object list by User ID in Descending order. To change the sort order, go to the Menu Bar at the top of the SAM screen and choose: **View** --> **Sort**. After selecting your sort order in the dialog box that appears, click **OK** and see what happens.

Answer:

The steps described sort the user object list in descending order by UID.

5. On large systems, even after defining a sort order, you may still find it tiresome to scroll through hundreds of user accounts to find an account that you need to modify. SAM allows you to filter object lists to show a subset of the available objects.

For the purpose of this lab exercise, define a filter in the Users object list that lists only user accounts with UIDs greater than 99. In the menu bar, choose: **View --> Filter**.

In the dialog box that appears, change the Operator for UID to Greater than, then type 99 in the Value box to the right. Click **OK** to apply the filter. Now look at the status bar at the top of the object list screen. How does SAM indicate that a filter has been applied?

Answer:

The steps described hide all user accounts with UIDs < 100. At the center of the status bar, SAM indicates the number of filtered objects.

6. Once you have customized your object list to your liking, save the current sort, column, and filter configuration as the default by selecting: **View --> Save view as default**

Answer:

7. Go to SAM's **File systems** functional area. It would be helpful to be able to view the amount of space in use in each of your file systems so you could determine if and when new disks may be needed on your system. Add the Percent used column to the object list, and sort the objects in descending order based on this new field. Save this new object list configuration as the default.

Answer:

Go to **Disks and File Systems --> File Systems**.
Choose **View --> Columns**.
Change **Ignore** to **1** in the Percent Used line.
Click **OK**.
Choose **View --> Sort**.
Find the Percent Used line.
Change **Ignore** to **1**, and **Ascending** to **Descending**.
Click **OK**.
Choose: **View --> Save View as Default**.

1. PART III

Return to the Users functional area in SAM. Without selecting a user from the object list, pull down the Actions menu. Can any actions be performed here without selecting a user?

Answer:

Several actions are available, even before selecting a user. For example, you can **Add** a new user account.

2. The **Add** action allows you to create a new user account. Select **Actions -->Add**. This should bring up a dialog screen requesting information about the new account. Set **operator**

as the user name for the new account, then click the **OK** button so SAM will take the defaults for the rest of the screen. After SAM prompts you for a password for the new account, check the object list to see what happened.

Answer:

After following the suggested steps and entering a password, an **operator** user is added to the object list.

3. Choose **Actions --> Add** again and enter user name `dbmgr`. Instead of pressing **OK**, try the **Apply** button. What is the difference between the **OK** and **Apply** buttons in a dialog box? Can you think of a situation where **Apply** might be a more efficient choice than **OK**?

Answer:

The difference between **Apply** and **OK** is that [Apply] leaves the dialog box open to add another user, whereas [OK] creates the requested user and closes the dialog box. This might be useful if you were creating many new user accounts and didn't want to repeatedly choose **Add** from the "Actions" menu.

4. Now try performing an action on an existing object. Using the mouse (GUI) or space bar (TUI), select `user1` from the object list. Now pull down the Actions menu. Now that you have selected a user, there should be many more actions to choose from. Choose **Modify user's password** and answer the questions that follow.

Answer:

5. In some cases, the dialog boxes that result from SAM actions may reference terms or concepts you haven't encountered before. Select `user1` from the object list again, then choose **Actions --> Modify**. This should open the **Modify a User** dialog box. The dialog box asks you to enter the user's Primary group and Start-up Program. For more information about terms and concepts you encounter in any SAM window, look for a **Help** button or Help menu on the menu bar. Click on the **Help** button at the bottom of the dialog box window. Skim the text, and try clicking on a couple of the underlined phrases (In the TUI, select one of the "See also" topics instead.) Any underlined phrase in a SAM help window is a link to additional information. When you have finished with the Help window, click **Close** to return to the **Modify a user** dialog box. We didn't really want to modify `user1`'s account after all, so click on **Cancel**.

Answer:

6. Occasionally, SAM will complain about an action you request. Try an experiment: From the Users object list, select **View--->Filter**. Set the User ID (UID) operator to **Any** and click on **OK**. Now select the root user account. Choose **Actions--->Remove**. Can you explain SAM's response to this action? Do you consider this to be a *feature* or a *bug*?

Answer:

A dialog box appears noting that SAM cannot remove root. This is a *feature* — removing root could be disastrous.

7. In other situations, SAM simply provides a warning message when you attempt a risky action. Try another experiment: From the Users screen, select the root user account, then

choose **Actions** --> **Modify**. You will get a warning box, to which the most prudent response at this point is "No"— do not modify root's account.

Answer:

1. PART IV

Choose **File** --> **Exit**. to exit your current SAM session. Then type `sam -r` to start the restricted SAM builder. (Note: you must be logged in as root to run the restricted SAM builder.)

Answer:

2. The SAM builder should display a list of user names and templates. Just accept the defaults for now and click **OK**.

Answer:

3. Next, you will see a window listing each of the SAM functional areas. Note that some functional areas are marked to be enabled (indicated by a green icon), some are disabled (indicated by a red icon), and some are partially enabled (indicated by a yellow icon). In the TUI interface, the status of a functional area is displayed in text form (enabled, disabled, partially enabled). Start by choosing **Actions** --> **Disable all**. What happens to the functional area icons?

Answer:

All the icons turn red.

4. Next, select the Process management icon with a single-click (GUI) or the space bar (TUI). Choose **Actions** --> **Enable**. What happens to the Process Management icon?

Answer:

The icon turns green.

5. Next, we need to enable automated backups. Go to the **Backup and recovery** functional area, and select the Automated backup icon. Again go to the Actions menu and choose: **Actions** --> **Enable**. Go back up to the main functional area window. What happened to the Backup and recovery icon? Why is this icon marked as partially enabled?

Answer:

The Backup and Recovery icon turns yellow. This indicates that some of the Backup and Recovery functionality is enabled, but some functionality is disabled.

6. You can enable and disable as many functional areas as you wish in the restricted SAM builder. Once you have enabled all the desired icons, choose **Actions** --> **Save privileges**.

Answer:

7. The "Save privileges screen defines which user(s) will have access to the functional areas you have selected. Select the **operator** user from the list , and click **OK**. (If you had multiple operators, you could select multiple user names from the list while holding the shift key. If the names are not together in thelist, hold the control key and select the names.) SAM saves the privileges you selected, then returns you to the icon window. Exit the restricted SAM builder, then log off your workstation or server.

Answer:

8. Log in as operator and try running SAM by typing: `/usr/sbin/sam`. Which functional areas are available for use by your operator user?

Answer:

Only the two functional areas we enabled appear when operator runs SAM.

9. Exit SAM, then try running SAM again by just typing `sam`. You will probably get an error message, `sh: sam: not found`. Try typing `/usr/sbin/sam`. Can you explain why SAM wouldn't start in the first case, but would start in the second?

Answer:

The `/usr/sbin` directory is not included in the user's PATH variable.

10. Log out as user operator, then log back in again as user root.

Answer:

1. PART V

Restart SAM. You will be in the functional area launcher. To view the SAM log, go to the Options menu, and choose `View SAM Log`.

Answer:

2. Experiment with some of the buttons in the dialog box that appears. Which Message level provides the most detail? Have any users other than root used SAM on your machine? When does your SAM log begin?

Answer:

The Commands Only message level gives the most detail. The box at the top of the screen indicates when the SAM log began.

3. On an active system, the SAM log can grow quickly. You may want to automatically trim the SAM log by choosing `Options --> SAM Log Options..` Configure your system so SAM automatically trims the SAM log to 100000 bytes.

Answer:

1. PART VI

From the functional area launcher, choose **Actions --> Add Custom Application**.

Answer:

2. Choose a name for your icon in the Label field.

Answer:

3. Enter the full pathname for the **whodo** command, **/usr/sbin/whodo**, in the Command field.

Answer:

4. **whodo** is not a graphical utility, so choose Terminal Environments as the user interface.

Answer:

5. Skip the optional fields and click **OK**.

Answer:

6. Try your new icon.

Answer:

7. Create another custom SAM application that automatically runs the **/usr/bin/cal** program for you to display a calendar of the current month.

Put your custom icon in the SAM's **Time** functional area.

Answer:

Go to: **Sam --> Time**.

Choose: **Actions --> Add Custom Application**.

Fill in the fields as follows:

Label: **Calendar**

Command: **/usr/bin/cal**

Execute using: User's ID

Supported interfaces: Terminal

3-11. REVIEW: Check Your Understanding

1. What steps is SAM taking when performing the following tasks?

Adding a new user

De-activating a user

Modifying a user's information

Adding a new group

Answer:

SAM is doing the following:

Adding a new user - Adding an entry to `/etc/passwd` and `/etc/group`, making a home directory, copying default setup files, and changing the owner and group of the login directory and default files.

De-activating a user - Replacing the user's password field (field two in the `/etc/passwd` file) with an asterisk .

Modifying a user's information - Modifying the appropriate information in the `/etc/passwd` file. Possibly changing the UID of the user's files.

Adding a new group - Adding an entry in the `/etc/group` file.

2. Describe the 7 fields of the `/etc/passwd` file

Answer:

- *user_name*:
- *encrypted password*:
- *user_id*:
- *group_id*:
- *comment_field*:
- *login_directory*:
- *startup_program*:

3. Describe the fields of the `/etc/group` file.

Answer:

- *group_name*:
- *password*:

- *group_id*:
 - *group_list*:
4. What does it mean to set up groups? Explain.

Answer:

Groups possibly allow users access to files that other members of their group own. It allows sharing of data amongst group members.

3-12. LAB: Hands-On Adding Users

1. Invoke **SAM** and add a user to your system. (You must be superuser to invoke SAM.) Use your name as a user name. Assign the user to a group called **class** and give him or her the POSIX shell.

Now, exit SAM and look at the **/etc/passwd** and **/etc/group** files. Do you see the user you added?

Answer:

Use **sam** to add a user.

2. Add a user to the system using HP-UX commands. This time, use your partner's name as the user name. (If you don't have a partner, pick any name.) Use a group called **class** and give the new user the C shell.

Look at the **/etc/passwd** and **/etc/group** files. Do you see the user added? Assign a password for the new account.

Answer:

```
useradd -m -s /usr/bin/csh -g class amy  
passwd amy
```

3. Run the commands to check the integrity of the **/etc/passwd** and the **/etc/group** files. Discuss your findings with the instructor.

Answer:

Use **pwck** and **grpck** commands. You may want to modify the **/etc/passwd** and **/etc/group** files to cause **pwck** and **grpck** commands to show some output.

4. Add a user called **date** that executes the **date** command. What would happen if you tried to log in using the user name **date**?

Answer:

Entry in /etc/passwd:

```
date::92:1:::/usr/bin/date
```

This could be created with **vipw**, SAM, or **useradd** and **passwd**. The command **date** is a valid login user name as indicated in the /etc/passwd file. more the /etc/passwd file and look at the *command* field (last entry) for these user names.

5. Use SAM to deactivate one of the new accounts you set up using SAM. Is the account still listed in /etc/passwd?

Answer:

The entry should still be in /etc/passwd with an encrypted password of *.

4-6. LAB: Customizing User Accounts

1. PART I

Modify the appropriate configuration file so that root will have access to command line editing at next login.

Answer:

```
# vi .profile
export EDITOR=vi
export HISTFILE=~/sh_history
export HISTSIZE=50
```

2. Modify the appropriate configuration file so that the system administrator's shell prompt displays the present working directory and user name after the next login.

Answer:

```
# vi .profile
export PS1='\$LOGNAME:\$PWD #'
```

3. If your system is running CDE, modify root's CDE login script to ensure that the system consults your .profile at next login.

Answer:

```
# vi .dtprofile
DTSOURCEPROFILE=true
```

4. Over the course of this week, you will run several scripts in the /labs directory. There should be a program in your /labs directory called **xroach**. What message do you get when you run **xroach** from your home directory by typing: **xroach**

Answer:

This will yield an `sh: xroach: not found` message.

5. Do whatever is necessary to ensure that you can run `xroach` and the other executables in `/labs` from any directory.

Answer:

```
# vi .profile
  export PATH=$PATH:/labs
```

6. Log out, then log back in again to see if your changes were successful.

Answer:

1. PART II

New users added to your system may appreciate having access to the same functionality you configured for `root` in Part I of this lab. How can you ensure that all new user accounts have the same custom functionality you configured for `root` in Part I? Make it so.

Answer:

```
# vi /etc/skel/.profile
  export EDITOR=vi
  export HISTFILE=~/sh_history
  export HISTSIZE=50
  export PS1='LOGNAME:$PWD #'
  export PATH=$PATH:/labs
```

2. `.profile` will be read only if your new users have a modified `.dtprofile`, too. How can you ensure that new home directories automatically get a copy of the modified `.dtprofile` that you created in Part I? Make it so.

Answer:

```
cp ~root/.dtprofile /etc/skel
```

3. Create a new user account using SAM or `useradd`; then log in as that new user and see if your customizations worked.

Answer:

```
# useradd -m mickie
# passwd mickie
```

```
# login mickie  
# exit
```

4. Do changes to `/etc/skel` affect already-existing accounts? Try logging in as `user24` to find out. Does `user24` have the custom prompt you configured in `/etc/skel/.profile`?

Answer:

```
# login user24  
# exit
```

`user24` was not affected by the new `/etc/skel/.profile`. The files in `/etc/skel` are copied only to new home directories.

5-5. LAB: HP-UX File System Hierarchy

1. Which of the following directories are dynamic?

/etc
/usr
/sbin
/dev
/tmp

Answer:

/etc
/dev
/tmp

2. Viewing a report on your disk space usage, you note that `/usr`, `/var`, and `/opt` are all nearing 90% capacity. Which of these directories should you be most concerned about? Why?

Answer:

`/var` deserves the most attention here because it is a dynamic file system that could grow quite quickly in case of an error condition that creates entries in the system log files. `/usr` and `/opt` are static file systems that are less likely to cause problems.

3. Match the directory with its contents:

- | | |
|--------------------------------|-------------------------------|
| 1. <code>/usr/share/man</code> | A. kernel, boot loader |
| 2. <code>/stand</code> | B. system configuration files |

- | | |
|-------------|--|
| 3. /var/adm | C. shareable operating system commands |
| 4. /etc | D. man pages |
| 5. /usr | E. application directories |
| 6. /opt | F. common admin files and logs |

Answer:

- | | |
|-------------------|--|
| 1. /usr/share/man | D. man pages |
| 2. /stand | A. kernel, boot loader |
| 3. /var/adm | F. common admin files and logs |
| 4. /etc | B. system configuration files |
| 5. /usr | C. shareable operating system commands |
| 6. /opt | E. application directories |

4. Where would you expect to find the `cp` and `rm` OS user executables? See if you are correct.

Answer:

Both are in `/usr/bin`, along with all the other user executables.

5. Where would you expect to find the `sam`, `useradd`, and `userdel` executables? See if you are correct.

Answer:

All three are in `/usr/sbin` along with many other administrative utilities.

6. The `pre_init_rc` utility executes in the early stages of the system start-up procedure to check for file system corruption. Where would you expect to find this executable? See if you are correct.

Answer:

`pre_init_rc` is in the `/sbin` directory, along with other files used during the boot process.

7. There is a system log file that maintains a record of system shutdowns. Where would you expect to find the shutdown log file? See if you are correct.

Answer:

The full pathname is `/var/adm/shutdownlog`.

Most OS log files are kept in `/var/adm`.

8. In which directory would you expect to find the "hosts" configuration file, which contains network hostnames and addresses? See if you are correct.

Answer:

The pathname for the hosts file is `/etc/hosts`.

9. Though many utilities and daemons maintain independent log files, many daemons and services write their errors and other messages to a log file called `syslog.log`. See if you can

find the path for this file, then check to see if any messages have been written to the file in the last day.

Answer:

```
# more /var/adm/syslog/syslog.log
```

10. Use the `whereis` command to search for the `xclock` executable. The executable is actually under `/usr/bin/X11/xclock`. Did `whereis` find this executable? Explain.

Answer:

```
# whereis xclock
```

`whereis` finds the man page, but probably doesn't find the executable. Remember that `whereis` doesn't search all directories: it searches only selected directories where executables are normally found. `whereis` does not search `/usr/bin/X11`, so it didn't find `xclock`.

11. Find all of the files (if any) under `/home` that are owned by root.

Answer:

```
find /home -user root
```

12. (Optional) Find all the files under `/tmp` that haven't been accessed within the last day.

Answer:

```
find /tmp -atime +1
```

13. (Optional) Find all the files on your system that are greater than 10000 bytes in size. If you needed to make some disk space available on your system, would it be safe to simply remove these large files?

Answer:

```
find / -size+10000c
```

6-8. LAB: Viewing the Configuration with `ioscan`

1. Which hardware path on this machine might be used for a serial (RS-232) modem?

Answer:

8/16/4 or 8/20/2 are the only serial RS-232 ports on this system.

2. Which hardware path on this machine might be used for a parallel printer?

Answer:

8/16/0 is the parallel interface card's address.

3. How many LAN interfaces does this machine have? What are their hardware paths?

Answer:

There are two LAN interfaces. One is at 8/16/6. The other is on the EISA bus at 8/20/5/1.

4. How many SCSI interfaces are available on this system? How many Fast/Wide SCSI interfaces are available?

Answer:

There is one single-ended SCSI interface at 8/16/5, and one Fast/Wide SCSI interface at 8/4.

5. How many disks are connected to the system shown in the ioscan above?

Answer:

There appear to be three "disk" class devices. Note in the description column, however, that one of the "disk" class devices is actually a CD-ROM.

6. What are the hardware addresses of the SEAGATE disks? What are the SCSI addresses of the SEAGATE disks?

Answer:

hardware paths: 8/4.5.0 (SCSI target 5)
 8/4.6.0 (SCSI target 6)

7. If you were to add another Fast/Wide SCSI disk to this system, what SCSI target address could you use for the new disk?

Answer:

0-15 are valid addresses for Fast/Wide SCSI devices. However, addresses 5-7 are already in use. Thus, the new device could be assigned any address in the range 0-4 or 8-15.

8. If you were to attach a new FW-SCSI disk at target address 4, what would be the resulting hardware path for the newly attached disk?

Answer:

8/4.4.0 would be the hardware path.

9. Is the tape drive on this system connected to the Fast/Wide SCSI chain, or the single-ended SCSI chain? What is the hardware address of the tape drive? What is the SCSI address of the tape drive?

Answer:

The tape on the single-ended SCSI chain, off the interface card at 8/16/5.

7-15. LAB: Device Files

1. PART I

Use **ioscan** to find the names of the device files for all of your disk class devices.

Answer:

```
# ioscan -fnC disk
```

2. Use **ioscan** to find the names of the device files for your system's LAN card(s).

Answer:

```
# ioscan -fnC lan
```

3. You should have a LAN card device file named **/dev/lan0**. Execute the command that lists the characteristics of this device file. What kernel driver is associated with this device file? What hardware path is associated with this device file?

Answer:

```
# lssf /dev/lan0
```

4. How does **lssf** know which kernel driver is associated with each device file?

Answer:

lssf determines the driver associated with a device file by examining the device's major number. **lssf** determines the hardware path of the device file by examining the device file's minor number.

5. Choose one of the disk device files in your **/dev/dsk** directory. Execute the command that lists the characteristics of this device file. What kernel driver is associated with this device file? What is the hardware path of the disk associated with this device file?

Answer:

```
# lssf /dev/dsk/c0t6d0 # the disk device filename will vary.
```

6. (Optional) – If you have a tape drive on your system, run **lssf /dev/rmt/***. Which device file(s) access your tape drive with the no-rewind feature?

Answer:

```
# lssf /dev/rmt/*
```

NOTE: File name typically ends in **n** or **nb**

1. PART II

Go to SAM --> Peripheral Devices --> Terminals and Modems. Are there any modems or terminals currently?

Answer:

2. First, configure a modem on your first available serial port.

1. Choose Actions -> Add Modem.
2. Choose a serial port/MUX interface hardware path.
3. Click Port Number to choose an available MUX port (Choose 0 on workstations).
4. Choose any baud rate (Doesn't really matter since we don't have a real modem.).
5. Choose to support both incoming and outgoing calls.
6. Click OK.

What device files does SAM create for you?

Answer:

SAM should have created three device files:

`/dev/cu10p0`

`/dev/ttyd0p0`

`/dev/cua0p0`

3. Now configure a hard-wired terminal on another available serial port.

1. Choose Actions -> Add Terminal.
2. Choose a serial port/MUX hardware path.
3. Click Port Number to choose an available MUX port.
4. Choose "H" as the baud rate.
5. Click OK.

What device files does SAM create for you?

Answer:

SAM should create a single device file for you: `/dev/tty1p0` (The actual device filename will vary, depending on the hardware path and port number you chose for your terminal.)

4. (Optional) If you have a tape drive on your machine, use SAM to create a non-standard device file for it. Specify a DDS1 density, no compression, Berkeley style semantics. Let the other options default.

This DDS1 device file may be useful if you create tapes that may need to be read on older tape drives that don't support the high density format used by "BEST" device files. What device file does SAM create?

Answer:

- Go to SAM → Peripheral Devices → Tape Drives.
 - Select a tape drive.
 - Choose Actions → Create Device Files → Create Custom Device File.
 - Click on the Density button and select DDS1 .
 - Take the defaults for the rest of the dialog box.
 - SAM should create a device file called `/dev/rmt/c0t0d0DDS1b`.
 - Click OK.
-

1. PART III

Remove the block device file for one of your disks.

Answer:

```
# rm /dev/dsk/c0t5d0
```

2. Do an `ioscan -funC disk` to ensure the device file is gone.

Answer:

3. Recreate the device file with `insf -evC disk`. The `e` option ensures that the device file is recreated even if the disk already exists in `/etc/ioconfig`. `v` stands for verbose, and `C` disk just recreates disk class devices.

Answer:

```
# insf -evC disk
```

4. Check to see that the device file was recreated.

Answer:

```
# ioscan -funC disk
```

`ioscan` should show that the `/dev/dsk` device file for your disk is back.

7-16. REVIEW: Check Your Understanding

1. What is a device file?

Answer:

A device file is used to associate a physical device with a logical file name.

2. What is the difference between a block and a character device file?

Answer:

The smallest unit that is exchanged during an I/O operation with a block device is a block (defined by `f_blksize` as returned by `ustat(2)`). The smallest unit that is exchanged during an I/O operation with a character device is a byte (eight bits).

3. Why do you need both block and character device files for disks?

Answer:

Because some commands like `mount` expect a block device and other commands like `newfs` expect a character device.

4. What are major and minor numbers?

Answer:

Major numbers describe the kind of device and the device driver. Minor numbers describe the location of the device and device dependent specialties.

5. What do the `insf` and `mksf` commands do?

Answer:

They create device files based on the information in `/etc/ioconfig`.

6. What is the difference between the `11` and the `lssf` commands related to device files?

Answer:

`11` lists major and minor numbers in their "encrypted" form, whereas `lssf` decrypts them into a readable form.

7. What does the `lsdev` command do?

Answer:

The `lsdev` command gives a listing of all device drivers and the corresponding block and character major numbers.

8-10. LAB: Logical Volume Manager

1. PART I

How many disks do you have on your system? Determine each disk's hardware path and device file names.

Answer:

```
# ioscan -func disk
```

2. Use **vgdisplay** to determine which of the disks on your system are already members of active volume groups. You should have at least one disk that isn't currently in a volume group. Note the free disk's device filename; you will create a new volume group using your free disk in the next part of the lab exercise.

Answer:

The free disk's hardware path and device filename will vary from system to system. For the sake of the solution that follows, we will assume that the device file name is **/dev/dsk/c0t5d0**. The **vgdisplay** command will show you which disks are already members of active volume groups on your system.

```
# vgdisplay -v
```

3. Before adding a disk to a volume group, you may want to check the size of the disk. This is accomplished via the **diskinfo** command. How large is your spare disk?

```
# diskinfo /dev/rdsk/c0t5d0
```

Answer:

```
# diskinfo /dev/rdsk/c0t5d0
```

The disk size will vary.

1. PART II

Configure your free disk as an LVM physical volume.

Answer:

```
# pvcreate /dev/rdsk/c0t5d0
```

2. Can you **pvdisplay** your disk at this point? Try it.

Answer:

```
# pvdisplay /dev/dsk/c0t5d0
```

Fails. You can't **pvdisplay** a disk until it is a member of a volume group.

3. Create a new **vg01** volume group using your newly created physical volume.

Answer:

```
# mkdir /dev/vg01  
# mknod /dev/vg01/group c 64 0x010000  
# vgcreate vg01 /dev/dsk/c0t5d0
```

4. Use **vgdisplay** and **pvdisplay** to check the status of your new physical volume and volume group. How many physical volumes are in the volume group at this point? How many logical volumes are in the volume group at this point? What is the extent size?

Answer:

```
# vgdisplay -v vg01 | more  
# pvdisplay /dev/dsk/c0t5d0
```

Currently there should be just one PV in the volume group, and no LVs. The PE size should be 4 MB (the default).

5. Create two 24-MB logical volumes in your new volume group. Name the first logical volume **cad** and the second **cam**.

Answer:

```
# lvcreate -L 24 -n cad vg01  
# lvcreate -L 24 -n cam vg01
```

6. Use **vgdisplay** and **lvdisplay** to ensure that your new logical volumes were actually created.

Answer:

```
# vgdisplay -v | more  
# lvdisplay -v /dev/vg01/cad  
# lvdisplay -v /dev/vg01/cam
```

7. Do an **ll** of the **/dev/vg01** directory. What is the name of the volume group device file for your new volume group? Each of your logical volumes should have two device files. Why?

Answer:

```
# ll /dev/vg01
```

The volume group device file should be called **/dev/vg01/group**.

Each logical volume requires both a raw and a block device file. Some commands used to access logical volumes require a block device file, while others require a raw device file. Both device file types are required for every device file.

8. Use SAM to remove your **vg01** volume group in preparation for the next portion of this lab. Volume groups can be managed in SAM from the Volume Groups object list:

SAM -> Disks and File Systems -> Volume Groups

Answer:

9. Try a **vgdisplay** back at the command line to ensure the volume group is gone. Did SAM remove the volume group device files, too?

Answer:

```
# vgdisplay -v vg01
```

The volume group is not found. In fact, SAM even removed the volume group device files.

1. PART III

What happens if you **pvcreate** a disk that is already in use by another active volume group? Try it. See what happens if you **pvcreate** your boot disk in **vg00**. Did this work? Try again using the **-f** "force" option on **pvcreate**. What is the result?

Answer:

```
# vgdisplay -v vg00 (Determine the name of your boot disk.)  
# pvcreate /dev/rdsk/c0t6d0  
# pvcreate -f /dev/rdsk/c0t6d0
```

Both **pvcreates** should give you error messages. LVM won't let you **pvcreate** a disk that's already claimed by another active volume group, even with **-f**.

2. Re-**pvcreate** the spare disk you used in the previous part of the lab. (This **pvcreate** *should* work.)

Answer:

```
# pvcreate /dev/rdsk/c0t5d0
```

3. For the sake of variety, create a new volume group called **vg02** using the spare disk you just **pvcreated**.

Answer:

```
# mkdir /dev/vg02  
# mknod /dev/vg02/group c 64 0x020000  
# vgcreate vg02 /dev/dsk/c0t5d0
```

```
# pvdisplay /dev/dsk/c0t5d0
```

```
# vgdisplay -v vg02
```

4. Now that you have a volume group, try creating a few logical volumes. Create a logical volume called `test1` in `vg02`. This time, though, don't specify a size for your logical volume. Based on the result of this experiment, what is the default logical volume size?

```
# lvcreate -n test1 vg02
```

```
# vgdisplay -v vg02
```

Answer:

The new logical volume is created with size 0 MB. The default logical volume size appears to be 0 MB.

5. What happens if you don't specify a logical volume name when you `lvcreate`? Try it. Create two new logical volumes of size 12 MB and 16 MB, leaving off the `-n` option in both cases. What names did LVM assign to your new logical volumes? Why?

Answer:

```
# lvcreate -L 12 vg02
# lvcreate -L 16 vg02
# vgdisplay --v vg02
```

By default, LVM uses the following naming convention for new logical volumes: `lvol1`, `lvol2`, `lvol3`, ... In this case, since these were the second and third logical volumes in `vg02`, LVM named them `lvol2` and `lvol3`. The number after `lvol` should match the last couple digits of the logical volume device file's minor number.

6. See what happens if you attempt to create an 11 MB logical volume in `vg02` called `test2`. Watch the output from `lvcreate` carefully. What size is your new logical volume? Explain.

Answer:

```
# lvcreate --L --n test2 vg02
# vgdisplay -v vg02
```

The new logical volume is 12 MB, not 11 MB. Space is allocated to logical volumes in units of extents. If you choose a size that is not a multiple of the extent size, LVM rounds up to the nearest extent boundary.

7. At some point in your UNIX career, you will almost certainly accidentally use `-1` instead of `-L` on `lvcreate`. What appears to be the difference between these two options? Try it and find out.

```
# lvcreate -1 12 -n test3 vg02
# lvcreate -L 12 -n test4 vg02
# vgdisplay -v vg02 | more
```

Answer:

The `-L` option defines a logical volume's size in megabytes, while the `-l` option defines a logical volume's size in extents. Thus, using `lvcreate -l 12` results in a logical volume that is 48 MB instead of 12 MB.

8. Once again, use SAM to remove the volume group you just created.

SAM --> Disks and File Systems --> Volume Groups

Answer:

1. PART IV

In preparation for the discussion of file systems in the following chapters, create a volume group `vg01` with your spare disk. Create three 12-MB logical volumes in `vg01` called `data`, `app`, and `tables`.

Answer:

```
pvccreate /dev/rdsk/c0t5d0 (device filenames may vary)
mkdir /dev/vg01
mknod /dev/vg01/group c 64 0x010000
vgcreate vg01 /dev/dsk/c0t5d0
lvcreate -L 12 -n data vg01
lvcreate -L 12 -n app vg01
lvcreate -L 12 -n tables vg01
```

8-11. REVIEW: Check Your Understanding

1. List two benefits of using LVM.

Answer:

- a. You can specify the size of a file system based on any need, or expand a file system by adding disk space anywhere on your system.
 - b. Data can exceed the capacity of a single disk.
2. Differentiate between a volume group and a logical volume.

Answer:

A volume group is a group of physical disks. A logical volume is a "virtual disk", a distribution of space within a volume group. A logical volume can span more than one disk, or represent only a portion of a disk.

3. What are the two reserved areas on a non-bootable disk?

Answer:

PVRA (Physical Volume Reserved Area) and VGRA (Volume Group Reserved Area). The VGRA is organized into VGDA (Volume Group Descriptor Area, VGSA (Volume Group Status Area, and MCR (Mirror Consistency Record).

4. What command do you use to initialize a disk as an LVM physical disk?

Answer:

`pvcreate`

5. What are the steps for creating a volume group?

Answer:

- a. Select disks for the volume group
- b. Make the disks physical volumes (`pvcreate`)
- c. Create a control directory for the volume group (`mkdir`)
- d. Create a "group" file in the volume group directory (`mknod`)
- e. Create the volume group (`vgcreate`)

6. Record the commands you use to perform these tasks:

- a. Make a disk an LVM disk.
- b. Create a volume group.
- c. Create a logical volume.

Answer:

- a. `pvcreate`
 - b. `vgcreate`
 - c. `lvcreate`
-

9-16. REVIEW: Check Your Understanding

1. List three types of file systems available on an HP-UX system.

Answer:

- HFS

- JFS
 - CDFS
 - NFS
2. List three types of file system metadata.

Answer:

- inodes
 - superblocks
 - directories
3. List three pieces of information in a superblock.

Answer:

- magic number
- file system size
- number of inodes
- map locations
- number of cylinder groups or allocation units
- block size and total number
- fragment size and total number (HFS only)
- total number of free data blocks
- total number of free inodes
- file system clean flag
- total number of free data blocks
- total number of free inodes
- file system clean flag

4. List some of the information found in an inode.

Answer:

- mode or permissions of the file
- type of file (that is, regular, directory, special)
- number of hard links to the file
- current owner of the file
- group associated with the file
- actual file size in bytes (more may be allocated as the file grows)
- time stamps relating to file activity
 - time/date of last file data change
 - time/date of last file access
 - time/date of last inode modification

- disk addresses, or pointers to disk addresses, where the file's data is stored
5. Describe what is created when the link command is executed.

Answer:

When the `ln` command is executed it creates a directory entry that has the new file name (the target name) associated with the same i-number as the original file. Both directory entries are then said to "point to the same file". A field in the inode is incremented to indicate the presence of the link. Only after the link count is reduced to zero does the actual data and inode get recycled.

6. What is the purpose of the JFS intent log?

Answer:

To ensure that an entire transaction is atomic. In other words, all the metadata updates associated with a transaction are recorded in the intent log with a single I/O operation.

10-8. LAB: Creating File Systems

1. PART I

The exercises in this lab assume that you already have the following three logical volumes:

`/dev/vg01/data`

`/dev/vg01/app`

`/dev/vg01/tables`

If you already have these logical volumes, you can skip ahead to Part II of the lab. Otherwise, create the `vg01` volume group and all three of the logical volumes listed above. Make each of the logical volumes 12 MB.

```
# pvcreate /dev/rdsk/c0t5d0
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgcreate vg01 /dev/dsk/c0t5d0
# lvcreate -L 12 -n data vg01
# lvcreate -L 12 -n app vg01
# lvcreate -L 12 -n tables vg01
# vgdisplay -v vg01
```

1. PART II

Create an HFS file system in the **data** logical volume. Create a JFS file system in the **app** logical volume. Don't mount your file systems yet.

Answer:

```
# newfs -F hfs /dev/vg01/rdata  
# newfs -F vxfs /dev/vg01/rapp
```

2. Do a **mount -v**. Why don't your new file systems appear at this point?

Answer:

```
# mount -v
```

The new file systems don't appear in the **mount -v** output because we haven't mounted them yet.

3. Create a mount point for each of your new file systems. Use **/data** as the mount point for the file system in the **data** logical volume, and **/app** as the mount point for the **app** logical volume. Again, don't mount your file systems, yet.

Answer:

```
# mkdir /data  
# mkdir /app
```

4. Add your new file systems to the **/etc/fstab** file so they will be mounted automatically at each system boot. Again, don't mount your file systems, yet.

Answer:

```
# vi /etc/fstab  
  
/dev/vg01/data /data hfs defaults 0 2  
  
/dev/vg01/app /app vxfs delaylog 0 2
```

5. Go ahead and mount your file systems now by doing a **mount -a**. Watch the resulting messages carefully. You should see several error messages indicating that **/dev/vg00/lvol1** and several other file systems are already mounted. Do the **mount -a** output messages offer any indication that your new file systems were successfully mounted?

Answer:

```
# mount -a
```

The output from **mount -a** notes that several other file systems are "already mounted", but there is no mention of the two new file systems. Oftentimes in UNIX, in the absence of an error message, you may assume that a command has succeeded. **mount -a** exemplifies this philosophy. Because **mount -a** didn't complain about your new file systems, you can assume that they mounted successfully.

6. Execute `mount -a` a second time and note the output messages again. Why did `mount -a` mention your new file systems in its output this time, but not when you did a `mount -a` in the previous exercise?

Answer:

In question 5, `mount -a` successfully mounted the new file systems as they weren't yet mounted. Executing `mount -a` a second time, however, generates an error message because the new file systems have already been mounted.

7. Use `mount -v` to see what file systems are now mounted. Did your new file systems mount properly? What other information can you glean from the `mount -v` output about your mounted file systems? List three fields presented in the `mount -v` output.

Answer:

Your file systems should all be mounted at this point.

`mount -v` presents several fields of information including:

- Device name
 - Mount point
 - File system type
 - Mount options
 - Mount time
-

1. PART III

The exercises in this section will be more meaningful if you have some real files in the `/data` and `/app` file systems. Towards that end, create a few files in these file systems using the following commands:

```
# cd /data; touch d1 d2 d3  
# cd /app; touch a1 a2 a3  
# ls /data /app  
# cd /
```

NOTE:

You may notice a `lost+found` directory in your new file systems. `newfs` creates this directory for you automatically. This special directory serves as a home for files that are victims of file system corruption. This directory will be discussed in some detail in a later chapter.

Answer:

2. Can you access files in a file system that is unmounted? Try an experiment to find out: Unmount `/data`, then do an `ls` of `/data`. Does the mount point still exist? Can you access `/data/d1` and `/data/d2`? Why or why not? Do whatever is necessary to regain access to `/data/d1` and `/data/d2`.

Answer:

```
# umount /data  
# ls /data  
# mount -a
```

After unmounting the file system, the mount point is still there, but the files in the file system are no longer visible under the mount point. The files in the file system cannot be accessed again until the file system is remounted.

3. Can you unmount a file system that is still in use? Try an experiment to find out: Open a second window on your screen. In this second window, `cd` to the `/data` file system. Back in your original window try unmounting the `/data` file system. What message do you get? Why?

Answer:

In a second window:

```
# cd /data
```

In your original window:

```
# umount /data
```

The `umount` fails. The `/data` file system cannot be unmounted until the processes using `/data` die.

4. Before you can unmount a file system, you will have to kill all processes accessing the file system. HP-UX provides a command to solve this very problem. Try the following command:

```
# fuser -u /dev/vg01/data
```

Each entry in the `fuser` output lists the PID of a process accessing the file system, a single letter code indicating how the process is using the file system ("c" indicates that a user has changed to a directory in the file system), and the name of the user that owns the offending process. `fuser` can also kill all of the offending processes:

```
# fuser -ku /dev/vg01/data
```

The effect of this command should be pretty dramatic. What happens? Now try unmounting `/data`.

Answer:

```
# fuser -u /dev/vg01/data # lists at least one process using the LV  
# fuser -ku /dev/vg01/data # kills the shell/window that was using /data
```

```
# umount /data # should work now.
```

5. What happens if you mount a file system on a directory that already contains files? Try an experiment to find out:

While the `/dev/vg01/data` file system is still unmounted, touch a few files under the `/data` mount point:

```
# touch /data/junk1 /data/junk2  
# ls /data
```

Now mount the `/dev/vg01/data` file system and list the contents of `/data` again:

```
# mount /data  
# ls /data
```

Do you still see `junk1` and `junk2`? Unmount `/data` again and check to see if `junk1` and `junk2` still exist. Why are `junk1` and `junk2` hidden while the data file system is mounted? Explain.

NOTE: File systems should always be mounted on empty mount point directories.

Answer:

While `/data` is mounted, `junk1` and `junk2` are hidden.

When `/data` is unmounted, `junk1` and `junk2` become visible again.

6. Remove the `junk` files you created in the previous exercise, and remount all of your file systems again.

Answer:

```
rm /data/junk*
```

```
mount -a
```

7. One last experiment: Can you unmount all of your file systems? Try a `umount -a` and explain the result.

Answer:

```
# umount -a
```

Though a couple of your file systems may successfully unmount, most will fail because they are in use by the system daemons.

8. Remount all of your file systems before continuing.

Answer:

```
# mount -a
```

1. PART IV

HP-UX supports two different types of disk-based file systems: HFS and JFS. In all of the examples you have tried so far, you have specified your desired file system type via the **-F** option. What happens if you forget the **-F**? Try it: do a **newfs** on the **/dev/vg01/tables** logical volume, but leave off the **-F** option. Look at the **newfs** output carefully. How does the system determine your default file system type? What is your default file system type?

Answer:

```
# newfs /dev/vg01/rtables
```

The default file system type is defined in **/etc/default/fs**. At version 11.x, **vxfs** is usually the default file system type.

2. The **newfs** command is a powerful tool for creating file systems, but it must also be treated with some respect. What happens if you accidentally **newfs** a logical volume or disk that already contains a file system? Try an experiment to find out: What happens if you attempt to create a new **hfs** file system on the data logical volume with the following command:

```
# newfs -F hfs /dev/vg01/rdata
```

Answer:

It fails. **newfs** will not overwrite a mounted file system.

3. What happens if you try to **newfs** a logical volume that contains an unmounted file system? Try it.

```
# umount /data
# newfs -F hfs /dev/vg01/rdata
# mount /data
# ls /data
```

Did you encounter any problems? What happened to the **d1**, **d2**, **d3** files you created earlier? Always use **newfs** with care.

Answer:

This sequence of commands should succeed without any problems. **newfs** *will* overwrite a file system in a logical volume if the file system is unmounted.

1. PART V

So far, you have created several file systems from the command line. File systems can be created and managed within SAM as well. Open SAM, then do the following:

1. Select: **SAM --> Disks and file systems --> Logical volumes**
2. Select: **Actions --> Create**

3. Click: **Select a Volume Group** (Choose vg01)
4. Click: **Define New Logical Volumes**
5. Fill in the blanks in the dialog box that appears:

LV name: app2
LV size (MB): 12
Usage: File System
Mount Dir: /app2

6. Click: **Modify FS defaults**.
7. Fill in the blanks in the dialog box that appears:

- a. Create a Journalized file system.
- b. Choose Now AND Every system boot options under When to mount.
- c. For now, take the defaults for the rest of the screen, and click OK.
8. You will be returned to the **Define new logical volumes** screen.
9. Click Add to add your new logical volume to the list of new LVs to create.
10. Click OK to go back to the **Create new logical volumes** step menu.
11. Click OK to create the new logical volume.

Answer:

2. Do a **mount -v** from the command line. Is the new file system mounted?

Answer:

Yes. The file system should be mounted.

3. Look at **/etc/fstab**. Did SAM ensure that the new file system will be mounted with every system boot?

Answer:

Yes. SAM automatically added the file system to **fstab**.

1. PART VI

Before moving on to the next chapter, use SAM to remove the file systems and logical volumes you created during this lab exercise:

1. Select: **SAM --> Disks and file systems --> Logical volumes**
2. While holding down **CTRL**, select each of the vg01 LVs. *Do not* remove any of the logical volumes in vg00.
3. Select: **Actions --> Remove**
4. When asked for confirmation, answer **Yes**.

Answer:

2. What did SAM do on your behalf?

Are the file systems unmounted?

Are the file systems still listed in `/etc/fstab`?

If you do a `vgdisplay`, do the logical volumes still appear?

Do the mount points still exist?

Answer:

SAM unmounted the file systems, removed them from `/etc/fstab`, and removed the logical volumes. The mount points, however, remain in tact.

3. You should also remove volume group `vg01` before continuing on to the next chapter.
Again, use SAM:

1. Select: `SAM --> Disks and file systems --> Volume groups`
 2. Select `vg01`
 3. Select: `Actions --> Remove`
-

11-7. LAB: `fsck`

1. PART I

Once in the simulator, choose simulation #1 from the menu. Simulation #1 asks you to run `fsck` on the `/app` file system.

Answer:

```
# simfsck  
Enter choice: 1
```

2. You will need to know the name of the logical volume containing the `/app` file system when you run `fsck`, so start by viewing a list of the currently mounted file systems.

Answer:

```
Sim: mount -v
```

3. File systems must be unmounted before running `fsck`. Unmount the `/app` file system.

Answer:

```
Sim: umount /app
```

4. Now run `fsck` on the raw logical volume containing `/app`. Be sure to specify the file system type. `fsck` will check the file system and ask which problems should be corrected.

Although one could conceivably answer "no" in response to **fsck**'s questions, you should generally answer "yes" – otherwise you will be left with a still-corrupted file system. Take note of the problems identified by **fsck**, but go ahead and answer "yes" to all of the prompts.

Answer:

```
Sim: fsck -F hfs /dev/vg01/rapp
```

5. Once **fsck** terminates, remount the file system.

Answer:

```
Sim: mount /app
```

6. After mounting the file system, you should check the **lost+found** directory to see if **fsck** identified any orphaned files. Are there any orphaned files in this case?

Answer:

```
Sim: ls /app/lost+found
```

In this case there were no orphaned files.

7. Did the **fsck** prompts indicate that **fsck** "REMOVE'd any files from **/app**? If you have a tape backup of the file system, you should restore these files at this point. In the case of our simulator, however, we don't have a tape backup so any removed files are lost forever.

Answer:

Yes. **fsck** noted an unallocated I=4, and prompted for permission to remove **/importantfile**. This file should be restored from a tape backup.

8. Go back to the main simulator by typing: "menu".

Answer:

```
Sim: menu
```

1. PART II

Choose simulation #2, which asks you to run **fsck** on the **/db** file system. Enter choice: 2

Answer:

2. Following the same procedure used in the previous exercise, **fsck** the **/db** file system and fix all the corruption identified.

Answer:

```
Sim: mount -v  
Sim: umount /db
```

```
Sim: fsck -F hfs /dev/vg02/rdb
Sim: mount /db
```

- Are there any files in `/db/lost+found` that need attention? If there are any files in `lost+found`, what can you do to find the file's owner and filename?

Answer:

```
Sim: cd /db/lost+found
Sim: ll \#0004
Sim: file \#0004
Sim: cat \#0004
```

There does appear to be one file in `lost+found` with inode#4. The `file` command tells us that the file contains ASCII text, `ll` shows us the owner is root, and the `cat` command shows that the file contains a list of key card holders. We can surmise, perhaps, that this file should be named "keycards".

- Did `fsck` "REMOVE" any files this time?

Answer:

No. No `REMOVE?` prompts appeared while running `fsck`.

- Return to the simulator menu by typing "menu".

```
Sim: menu
```

Answer:

1. PART III

Run simulation #3, which asks you to run `fsck` on the `/data` file system. Enter choice: 3

Answer:

- What happens if you specify the wrong file system type to `fsck`? Unmount the data file system, and try running `fsck -F hfs /dev/vg02/rdata`. What happens? Despite the worrisome message `fsck` offers here, the superblock really isn't damaged. Can you guess why you get this message?

Answer:

```
Sim: mount -v
Sim: umount /data
Sim: fsck -F hfs /dev/vg02/rdata
```

`fsck` was expecting an HFS-style superblock, but instead encountered a JFS-style superblock in the rdata logical volume. `fsck` interprets this inconsistency as a corrupted superblock.

3. Try running **fsck** again with the **-F vxfs** option. How does running **fsck** on a **vxfs** file system differ from running **fsck** on an **hfs** file system?

Answer:

```
Sim: fsck -F vxfs /dev/vg02/rdata
```

By default, **fsck** simply replays the intent log rather than doing a full check of the file system.

4. After running **fsck**, remount the file system and return to the main simulator menu.

Answer:

```
Sim: mount /data Sim: menu
```

1. PART IV

Run simulation #4, which asks you to run a "full" **fsck** on the **/data** file system. Menu choice: 4

Answer:

2. You saw in the previous simulation that, by default, **fsck** simply replays the JFS intent log. After a system crash, an intent log replay is all that is required. An intent log replay simply scans the intent log completes any pending transactions; it does not check the consistency of your superblock, inodes, and allocation units.

If you suspect more serious file system corruption in a JFS file system, you can perform a full check of the file system. Try the following:

```
Sim: mount -v
Sim: umount /data
Sim: fsck -F vxfs -o full /dev/vg02/rdata
Sim: mount /data
Sim: ll /data/lost+found
```

Answer:

3. How did this **fsck** differ from the **fsck** in the previous JFS simulation?

Answer:

Instead of simply replaying the intent log, **fsck** appears to do a multipass check of the file system's inodes, directories, superblock, and resource maps. After making several corrections, **fsck** clears the intent log.

1. PART V

In all of the examples tried thus far, you have manually run `fsck` to check and repair your file systems. However, the system automatically runs `fsck` for you each time you boot. Simulation #5 will show you console messages that appear while booting an E35 server. Run the simulation and watch the console messages that follow. Enter choice: 5

Answer:

2. Did the system `fsck` all of your file systems?

Answer:

No. `fsck` only checked `/app`: the other file systems were found to be "clean" and didn't require maintenance.

3. Every file system superblock contains a "file system clean flag". When a file system is mounted, it is marked "dirty." Properly unmounting the file system with `umount` toggles the flag back to a "clean" state. The `fsclean` utility that runs during the boot process only `fsck`'s dirty file systems. Why might a file system be left in a dirty state?

Answer:

An improper shutdown leaves file systems in a dirty state. Any file systems that are mounted at the time of a crash or improper shutdown will be automatically `fscked` at next boot. In the simulator example here, it appears that `/app` was the only file system mounted at the time of the improper shutdown.

4. Did `fsck` fix any problems in your "dirty" file system?

Answer:

Yes. Several minor problems were corrected in `/app`.

5. Do a `mount -v` to ensure that all of your file systems are properly mounted after the boot.

Answer:

Sim: `mount -v`

All file systems appear to be mounted.

1. PART VI

Try running simulation #6, another boot simulation. Start the boot simulation, and watch the console messages carefully. Enter choice: 6

Answer:

2. In this simulation, `fsck` again automatically identified and corrected several minor problems in the `/app` file system during the boot process. However, `fsck` also identified a problem that requires removal of a file. When running automatically during the boot process,

fsck will never make any repairs that cause loss of data. When **fsck** identifies problems that may require removal of data, you will be prompted to run **fsck** manually. Which file system requires a manual **fsck** this time?

Answer:

/app needs to be checked.

3. Run **fsck** on /app and fix any identified problems.

Answer:

```
Sim: fsck -F hfs /dev/vg01/app
```

4. When you complete the **fsck**, allow the boot process to complete. Return to the main menu, then exit out of the simulator.

Answer:

```
Sim: menu Enter choice: 0
```

1. PART VII

Now that you have had an opportunity to run **fsck** in the simulator a few times, try running **fsck** on some real file systems. Do a **mount -v** to determine which file systems are mounted where on your system.

Answer:

```
# mount -v
```

2. You should have an HFS file system mounted on /stand, and a JFS file system mounted on /home. Unmount both of these file systems, and run the appropriate **fsck** commands to check them both. Does **fsck** identify any problems?

Answer:

```
# umount /stand
# fsck -F hfs /dev/vg00/lvol1
# mount /stand
# umount /home
# fsck -F vxfs /dev/vg00/lvol4
```

There shouldn't be any problems in either file system.

3. If you haven't already remounted your file systems, do so now.

Answer:

```
# mount -a
```

12-7. LAB: File System Management

1. Occasionally, the `/tmp` file system fills up, causing problems on a system. Using SAM or the command line, list all of the files in `/tmp` that haven't been accessed within the last 2 days.

Answer:

```
# find /tmp -atime +2
```

or

SAM --> Routine Tasks --> Selective File Removal

2. `/var` is another file system which sometimes reaches 100%. Often this is a result of log files that haven't been properly trimmed. Trim the following log files back to size 0:

```
/var/adm/btmp  
/var/adm/wtmp
```

Answer:

```
# >/var/adm/btmp  
# >/var/adm/wtmp
```

3. The questions that follow give you an opportunity to fix a full file system. You should have a script on your system called `/labs/fillfs.sh`. The `fillfs.sh` script will fill one of your file systems to capacity. Run `fillfs.sh`. As the script executes, you may see a number of file system full messages scroll across your screen. Don't worry – yet!

Answer:

```
# /labs/fillfs.sh
```

4. Which file system appears to be full? How many kbytes were allocated for this file system? What percent of the space in that file system is in use?

Answer:

```
# bdf
```

`bdf` will show the status of each of your file systems. The `/home` file system should be nearly 100% full.

5. What happens at this point if a user tries to copy a file to the full file system? Is anything recorded in `syslog.log`? Copy a large file (e.g., `/stand/vmunix`) to `/home` to find out.

Answer:

```
# cp /stand/vmunix /home  
# tail /var/adm/syslog/syslog.log
```

The **cp** command will generate an error message on the screen, and the administrator will see a message in the syslog, too.

6. List two possible solutions to this full file system problem.

Answer:

Add additional disk space to the **/home** file system.
Remove core files.
Purge large, unneeded files from **/home**.

7. Are there any core files in the problem file system that could be removed? If so, remove them. What commands did you use to find the core files?

Answer:

```
# find /home -name core  
# rm /home/user5/core
```

8. Which directory under **/home** is taking the most space? Which command can you use to find out? Mail a message to the culprit asking him or her to purge some files.

Answer:

```
# du -sk /home/* | sort -rn  
# echo "remove some files, user5!" | mail user5
```

9. You could wait for your users to purge some old files, but in many cases, you will eventually need to add some additional space to the full file system. **pvcreate** your free disk and add it to **vg00**. Use **vgdisplay** to ensure that the new disk was successfully added to the volume group.

Answer:

```
# pvcreate /dev/rdsk/c0t5d0 # disk device filename may vary.  
# vgextend vg00 /dev/dsk/c0t5d0  
# vgdisplay -v vg00
```

10. Double the size of the logical volume containing **/home**. Which disk contains the new extents? What command did you use to find out?

Answer:

```
# lvdisplay -v /dev/vg00/lvol4 # lvol name may vary  
# lvextend -L 100 /dev/vg00/lvol4  
# lvdisplay -v /dev/vg00/lvol4  
# note the extents used by lvol4
```

The logical volume should use the first free extents available in the volume group.

11. Using **bdf**, check to see if the **/home** file system contained in the logical volume you just extended increased in size. Does extending a logical volume automatically extend the file system within the logical volume?

Answer:

bdf suggests that the home file system hasn't changed in size. The file system won't take advantage of the additional space in the logical volume until you do an **extendfs**.

12. Execute the commands necessary to extend the **/home** file system to take advantage of the additional space in the logical volume.

Answer:

```
# umount /home  
# extendfs /dev/vg00/rlvol4 # lv name may vary  
# mount /home
```

13. Unmount the file system and try doing **extendfs** on the logical volume containing **/home** again. Explain the resulting message.

Answer:

```
# umount /home  
# extendfs /dev/vg00/rlvol4 # lv name may vary  
# mount /home
```

The file system can't be extended any further because there isn't any more space in the logical volume. However, no harm was done.

14. Before moving on to the next chapter, remove all the "bigfiles" from user5's home directory.

Answer:

```
# rm /home/user5/bigfile*
```

12-8. REVIEW: Check Your Understanding

1. List and define two commands to monitor the free disk space on the system

Answer:

bdf and **du** are easy ways to monitor the disk file system loading. You can also use the **df** or the **diskusg** commands.

2. What are some different solutions to recover space on your file system?

Answer:

- Trim log files.

- Remove core files.
 - Remove large, old files
-

13-13. LAB: Backup and Recovery

1. PART I

If don't already have a `/var/adm/fbackupfiles` directory on your machine, create it. By default, this is the directory where `fbackup` maintains the "dates" log file of backups completed on your system. Make this directory your present working directory.

Answer:

```
mkdir -p /var/adm/fbackupfiles  
cd /var/adm/fbackupfiles
```

2. Run the `/labs/corp1.sh` script, which will create a few directories that you can practice backing up. Note the directories that are created by the script.

Answer:

```
/labs/corp1.sh
```

1. PART II

Create a graph file that includes everything in the `/corp` directory except `/corp/dept3`.

Answer:

```
# vi graph  
i /corp  
e /corp/dept3
```

2. Perform a full backup of `/corp` using the graph file you just created. Write the backup to a file called `tape1`, write an index of the files included in the backup to "index1", and use the `-u` option to ensure that the "dates" file records the time stamp of your backup.

Answer:

```
# fbackup -f tape1 -u0g graph -I index1
```

3. What was recorded in the "dates" file as a result of your `fbackup`? Were all of the directories under `/corp` backed up? Look at the "index1" file that was created, and explain what you see.

Answer:

```
# more dates  
# more index1
```

There should be a single entry in the **dates** file indicating the time at which the level 0 backup of "graph" was performed.

index1 should show that all the directories under **corp** were backed up except **/corp/dept3**. Since the **graph** file explicitly excluded **/corp/dept3**, this directory was omitted from the backup.

4. Run **/labs/corp2.sh**. This script should create a few new files under the **/corp** directory. Note the changes.

Answer:

```
/labs/corp2.sh
```

5. Do a level 1 backup of **/corp** using your **graph** file. Write the backup to **tape2**, create an index of your backup in **index2**, and use the **-u** option to ensure that the **dates** file is updated again.

Answer:

```
# fbackup -f tape2 -ulg graph -I index2
```

6. What was recorded in your **dates** file as a result of your backup? Were all of the files and directories under **/corp** backed up? Look at the index file for this backup and explain what you see.

Answer:

```
# more dates  
# more index2
```

A new line should have been added to your **dates** file indicating that a level 1 backup of the "graph" graph file was performed. In the **index2** file, note that only **/corp/dept1** and the new files under **/corp/dept1** were included in this backup. Since this was a level 1 incremental backup, only the files that changed were included in the backup.

7. Run the **/labs/corp3.sh** script. Note the newly created files.

Answer:

```
# /labs/corp3.sh
```

8. Run another level 1 backup. Write the backup to **tape3**. Ensure that the **dates** file is updated and create an index file called **index3**.

Answer:

```
# fbackup -f tape3 -ulg graph -I index3
```

9. What changed in your "dates" file as a result of your backup? Explain. Which files and directories under /corp backed up? Explain.

Answer:

```
# more dates  
# more index3
```

The previous level 1 backup in the dates file was replaced by this new level 1 backup. The previous level 1 backup is no longer needed to ensure a full recovery since the new level 1 includes ALL the files created or modified since the last FULL backup.

This is confirmed by the index file. The index for the most recent level 1 includes both the **report*** and **chart*** files.

10. Without running another **corp** script, try running a level 2 backup of /corp using your graph file. Again, ensure that **fbackup** updates the "dates" file and creates an index. What files and directories are included in this backup? Explain.

Answer:

```
# fbackup -f tape4 -u2g graph -I index4
```

fbackup adds an entry to the dates file to indicate that the backup was performed. However, since no files have been created or changed since the last level 1, no files were included in the backup.

1. PART III

Run the **/labs/corp4.sh** script, and note the effect this script has on your /corp directory.

Answer:

```
# /labs/corp4.sh
```

The script destroyed /corp and everything under it.

2. You did a total of four backups in the previous part of this lab. Is it necessary to restore all four tapes to fully recover your system? Explain.

Answer:

It is only necessary to restore the most recent backup at each backup level. The most recent level 0 backup is on tape1, the most recent level 1 backup is on tape3, and the most recent level 2 backup is on tape4. It is not necessary to restore tape2.

3. Do whatever is necessary to fully restore /corp.

```
# frecover -f tape1 -rv          # restore the most recent level 0  
# frecover -f tape3 -rv          # restore the most recent level 1
```

```
# frecover -f tape4 -rv          # restore the most recent level 2  
# find /corp                      # ensure that the recovery worked.
```

Answer:

4. Occasionally, users accidentally remove a file and expect the administrator to be able to restore the missing file from tape. Remove `/corp/dept1/reporta`. How can you restore this single file? Do it.

Answer:

```
# frecover -f tape3 -i /corp/dept1/reporta -xv
```

5. Will `frecover` overwrite newer data in a file with older data from a tape archive? Try it and find out.

```
# vi /corp/dept1/reporta          # Make a change to reporta  
# ll /corp/dept1/reporta         # Note the time stamp  
  
# frecover -f tape3 -i /corp/  
dept1/reporta -xv  
  
# cat /corp/dept1/reporta        # Were your changes overwritten by frecover?
```

Answer:

Note that `frecover` does *not* overwrite the newer version of the file with the older version of the file from the tape archive. By default, `frecover` always compares the time stamp of the archived version against the time stamp of the file on disk.

1. PART IV

Ensure that your server has granted your client(s) root access in the `~root/.rhosts` file. Also make sure there is a writable tape in the server's tape drive.

Answer:

```
server# vi ~root/.rhosts  
      clienthostname  
client# vi ~root/.rhosts  
      serverhostname
```

2. On the client, `cd` to the `/var/adm/fbackupfiles` directory.

Answer:

```
# cd /var/adm/fbackupfiles
```

3. From one of your clients, initiate a backup of the directories listed in the graph file you created for `corp` earlier in the lab. Do the backup to the tape device file `/dev/rmt/0m` on the server. Also create an index in "index5".

Answer:

```
client# fbackup -f serverhostname:/dev/rmt/0m -u0g graph -I index5
```

NOTE:

`fbackup` always begins writing at the beginning of the tape. Thus, if another client were to do a backup to the server at this point, your backup would be overwritten. Make sure you remove your tape when the backup is complete, before another client attempts to do a backup.

4. On which machine was the index file created? Which machine's dates file was updated as a result of the backup?

Answer:

The index and dates files are maintained on each individual client, not the backup server.

5. Remove the client's `/corp` directory structure. Then see if you can restore it from your tape.

Answer:

```
client# rm -r /corp/user4
```

```
client# frecover -f serverhostname:/dev/rmt/0m -rv
```

6. If time permits, do a backup from the other clients as well.

Answer:

14-5. LAB: cron

1. PART I

`cron` should start automatically during the boot process. Check to ensure that the `cron` daemon is running on your machine.

Answer:

```
# ps -ef | grep cron
```

2. Create and submit a `cron` job to display the current time and date to the console every minute.

Answer:

```
# crontab -e
* * * * * /usr/bin/date > /dev/console
```

3. Add another **cron** job that sends the output from the **who** command to the console every 10 minutes.

Answer:

```
# crontab -e
* * * * * /usr/bin/date > /dev/console
0,10,20,30,40,50 * * * * /usr/bin/who > /dev/console
```

4. List your scheduled **cron** jobs. Are they both there?

Answer:

```
@ # crontab -l
```

Both jobs should appear.

5. Edit your **cron** file again. See what happens if you precede the **/usr/bin/date** line with a **#** sign. Save your changes. What effect does the **#** sign have?

Answer:

```
# crontab -e
# * * * * * /usr/bin/date > /dev/console
0,10,20,30,40,50 * * * * /usr/bin/who > /dev/console
```

After saving the change, the **date** command stops executing.

6. Why would commenting a line out of the **cron** file be preferable to simply removing the line?

Answer:

Simply commenting the **date** line out of the **cron** file makes it easier to reschedule the job later: just delete the **#** sign.

7. How can you remove ALL of your scheduled cron jobs? Do it.

Answer:

```
# crontab -r
```

1. PART II

Create a graph file that includes /home, but nothing else. Name your graph file /home.graph.

Answer:

```
# vi /home.graph
i /home
```

2. Schedule a full backup of /home.graph to occur every Sunday night, and an incremental backup to occur every weekday night. Schedule both backups to run at 11 p.m. Redirect the error messages from your backups to a file called /home.err.

Answer:

```
# crontab -e
0 23 * * 0    fbackup -f /dev/rmt/0m -u0g /home.graph 2>> /home.err
0 23 * * 1,2,3,4,5  fbackup -f /dev/rmt/0m -u1g /home.graph 2>> /home.err
```

3. List your cron jobs to ensure they are properly scheduled. Leave your scheduled cron jobs in place, and check your /home.err file tomorrow morning to see if your cron job succeeded.
-

15-9. LAB: Swap

1. How much memory does your training system have? How much is lockable, available, and physical?

Answer:

Use the dmseg command to see real mem, avail mem, and lockable mem.

2. Create a 48M logical volume and dynamically add it as swap.

Answer:

```
lvcreate -L 48 -C y -r n -n swap /dev/vg00
swapon /dev/vg00/swap
```

3. Create a file system in a 20M logical volume. Mount the file system at data. Dynamically enable it as file system swap, limiting the amount of space that the paging system can take to 10M, and reserving 4M for files. Display swap space usage.

Answer:

```
lvcreate -L 20 -C y -r n -n data /dev/vg00
newfs /dev/vg00/rdata
```

```
mkdir /data  
mount /dev/vg00/data /data  
swapon -l 10M -r 4M /data  
swapinfo -mt
```

- How do you make sure that swap will be enabled each time the system is rebooted?

Answer:

Add an entry to `/etc/fstab`.

- Create a 12M logical volume using SAM. Use SAM to add this logical volume as swap. Enable the swap area now and at every boot. Examine `/etc/fstab`. What entry did SAM add?

Answer:

Choose **Disks and File Systems** from the SAM functional area launcher. Choose **Logical Volumes**. Choose **Create** from the **Actions** menu. Select a volume group for the swap area. Next choose **Add New Logical Volumes**. Fill in the menu. Be sure to choose swap space for **Usage**. Modify the logical volume defaults to turn bad block relocation off and contiguous allocation on. When you have finished with the menu select **Add**, then **OK**. Finally, choose **OK** to create the logical volume and enable swap.

- Can you unmount a file system that has file system swap enabled? Try it.

Answer:

```
# umount /data
```

Fails – "Device busy". A file system that is enabled for use as file system swap cannot be unmounted.

- You should have discovered in the previous question that once swap is enabled in a file system, it is impossible to unmount that file system. Is this a problem? Explain.

Answer:

Both the `extendfs` and `fsck` commands require a file system to be unmounted. Thus, once a file system is enabled for use as file system swap, you can't extend the file system with `extendfs`, or run `fsck`.

- Try to view the `man` page for `swapoff`. What happens?

NOTE:

To disable swap, edit the `/etc/fstab` file; comment out or delete the lines for the swap areas you wish to disable; and reboot. When the system reboots, the commented lines in `fstab` will be ignored. There is no way to disable swap without rebooting.

Answer:

"No manual entry for swapoff" — There is no **swapoff** command.

9. Remove the **/etc/fstab** entries for the swap areas you created in this lab, then reboot your machine by typing:

```
# vi/etc/fstab  
# cd /  
# shutdown -ry 0
```

16-12. LAB: Hands-On Adding Printers

1. If you have a printer available, use SAM to configure the printer. Add the printer to the class named **class1**.

If no printer is available use your terminal (or a window) as your printer. To find the device file associated with your terminal (or window) issue the command:

```
tty
```

This will return the device file name. Change the permissions on the device file so that **lp** can read and write to the file.

```
chmod 666 devicefile
```

You can now use SAM to configure your printer. You will add it as a local printer with "Nonstandard Device File". Use the **dumb** Model/Interface. Place your printer in the class named **class1**.

Exit SAM when after you have added your printer.

Answer:

Select Printers and Plotters from the main SAM menu. Select Printers and Plotters from the submenu. Select Add Local Printer/Plotter from the actions menu. Choose the appropriate type of printer. For this solution we will assume you used your terminal with a device file of **/dev/ttyp3**. Select Add Printer with Nonstandard Device File. Complete the menu as follows:

```
Printer Name: sparky  
[ Printer Model/Interface... ] dumb  
Printer Device File Name: /dev/ttyp3  
[ Printer Class... ] class1(optional)
```

```

Default Request Priority: [ 0 ->]

[x] Make This Printer the Default Destination
-----
[ OK ] [ Cancel ] [ Help ]

```

Then press **OK** and exit SAM.

2. List the status of the lp scheduler. Is the scheduler running? What is your system default destination? Is your printer enabled? Is your printer accepting requests?

Answer:

```
lpstat -t
```

SAM should have started the scheduler, enabled your printer and issued the accept command for you.

3. Try to print to your printer. Issue the command:

```
banner success|lp -d
```

your_destination

Try to print to the destination class1.

Answer:

```
banner success|lp -d sparky
```

```
banner This printer has class|lp -d class1
```

4. Set the fence priority on your printer to a priority higher than your default priority. Issue the lp command without the -p option. Did the request print? Display the status of the scheduler. Do you see your request on the queue? Raise the priority on your request to make it print.

Answer:

lpshut

The scheduler must first be stopped

lpfence sparky 5

Restart the scheduler

lpsched

banner fenced in|lp -dsparky

This request will not print yet

lpstat -t

Find the request id for your request

lpalt sparky-23 -p5

Substitute your request id for sparky-23

5. Use SAM to remove your printer.

Answer:

Select **Printers and Plotters** from the main SAM menu. Select **Printers and Plotters** from the submenu. Highlight your printer in the object list. Select **Remove** from the **Actions** menu. SAM will ask for confirmation. Press **Yes** to proceed.

6. Configure your neighbor's local printer as a remote printer for your machine. Print a copy of `/etc/passwd` on your new remote printer and see what happens.

Answer:

Use SAM.

7. If the equipment is available in the classroom, configure a network printer.

Answer:

Use SAM or jetadmin.

16-13. REVIEW: Check Your Understanding

1. What functions does the spooling system provide, and why are they required?

Answer:

The spooling system manages print requests. They are required for optimizing the usage of limited system resources.

2. Which of these functions are available to the administrator and which to normal users?

Answer:

Functions like `lp`, `lpstat`, `enable`, `disable` and `cancel` can be executed by a normal user, the other commands by the LP administrator only.

3. What is the difference between a device, a printer and a destination?

Answer:

A device refers to the physical device and its related device file. A printer is a logical name to represent a physical device within the spooling system. A destination is the name of a spooling queue. Printers, but also classes are destinations.

4. Is it possible, to install two printers for one device?

Answer:

Yes. And it might be useful, too. When you regularly intend to print on two different types of paper (for example, preprinted form sheets and normal blank paper), having only one print device, address each print request to the appropriate printer (for example, "prform"

and "prnorm"). Both printers are related to the same device, but only one of them can be **enabled** at the same time. The other collects its print requests until you change the type of paper. Prior to the change **disable** both, after the change **enable** the one waiting so far.

5. How would you cancel your own print request?

Answer:

`cancel request_id_number`

6. How would you cancel a print request owned by someone else?

Answer:

`cancel request_id_number`

7. What is an interface program?

Answer:

A link between the printer and its device file.

8. If you have stopped the scheduler (with **lpshut**), does printing continue?

Answer:

No

9. If you have stopped the scheduler (with **lpshut**), can you still use the **lp** command to add print requests to the queues?

Answer:

Yes

10. How can you tell other users that a printer is "broken"?

Answer:

By `disable -r"Printer is broken" printer`

11. How can you move print requests from the "printer1" queue to the "printer2" queue?

Answer:

By `lpmove printer1 printer2`.

17-14. LAB: Shutting Down and Rebooting Your System

1. PART I

If you are at a graphics terminal, it is good practice to shutdown X-windows before shutting down your system. Otherwise, console messages generated during shutdown are often garbled by the windows on your screen.

To shutdown X-windows, log out. Back at the CDE login screen, click on the [Options] button and choose "Command line login". Immediately hit **Return** and log in as root. (If you wait too long before logging in, X-windows restarts and you will have to try again.)

Answer:

[Exit]

[Options] -> Command line login

Return

2. Currently, your system should be in multiuser mode. Note which file systems are mounted, and how many processes are running.

Answer:

```
# mount -v  
# ps -ef | wc -l
```

3. Shut down to single-user mode. Then check to see what processes and mounted file systems remain. What differences do you see between single and multiuser modes?

Answer:

```
# cd /  
# shutdown -y 0  
# mount -v  
# ps -ef | more
```

There are few file systems mounted in single-user mode, and few processes running. Single-user mode is a good place to do kernel configuration, file system maintenance with **fsck**, and other maintenance activities that require a quiet system.

4. From single-user mode, take your machine down to the halt state. What can you do in the halt state? Why might it be necessary to take your system down to the halt state?

```
# reboot -h
```

Answer:

Nothing is running in the halt state, and no logins are possible. It is necessary to bring the system to a halt state before powering off the system to install new peripherals and interface cards.

5. Power-off your machine before continuing on to the next part of the lab.

Answer:

[Power]

1. PART II

Power-on your machine, but immediately begin hitting [ESC] to interrupt the autoboot process. Proceed to the boot admin menu.

Answer:

[Power]

[Escape]

Menu choice: a (some machines take you directly to boot admin)

2. Search for possible boot devices on your system. How many SCSI devices does your system have? If a system isn't booting properly, this is one way of determining if all your SCSI devices are properly connected and powered on.

Answer:

BOOT ADMIN> search

The number of devices on the system will vary.

3. Now search for disks that contain an IPL that you can boot from. How many of your disks appear to be bootable?

Answer:

BOOT ADMIN> search ipl

Most likely, there will be only one bootable disk.

4. Display your primary and alternate boot paths. Which disk is defined as your primary boot device?

Answer:

BOOT ADMIN> path

The primary and alternate boot paths will vary.

5. Boot to the ISL from your primary boot disk. *Be sure to specify that you want to interact with the ISL/IPL.*

Answer:

```
BOOT ADMIN> boot pri isl
```

Some models ask if you want to interact with the ISL. Answer "yes".

6. At the ISL prompt, get a list of valid ISL commands.

Answer:

```
ISL> help
```

7. Interacting with the ISL may be useful if your primary kernel is corrupted and you have to boot from a backup kernel. Do you currently have a backup kernel in your /stand directory? (Kernel filenames usually begin with **vmunix**.)

Answer:

```
ISL> hpx ls
```

There should be just one kernel at this point: **/stand/vmunix**.

8. By default, the system boots to multiuser mode using **/stand/vmunix**. Boot to single-user mode from the default kernel.

Answer:

```
ISL> hpx -is
```

OR

```
ISL> hpx -is /stand/vmunix
```

9. Did the system prompt you for a password when you were brought to single-user mode? When might this be helpful?

Answer:

No, single-user mode does not prompt for a password. The administrator is automatically logged in as root. This may be useful if the administrator forgets the root password.

1. PART III

Single-user mode is useful for some system administration tasks, but your users won't be able to log in until you bring the system to multiuser mode. During an autoboot, the system

automatically boots to the default run level, which is usually defined as run-level 3. What is your default system run-level?

Answer:

```
# cat /etc/inittab
```

The **initdefault** line probably defines "3" as the default run level.

2. In the steps that follow, we will bring your system up to multiuser mode one run-level at a time to see what happens at each step. To start, use **init** to bring your system to run level 1. Watch the console messages carefully.

Answer:

```
# init 1
```

3. The **init** daemon calls the **/sbin/rc** program, which is responsible for starting most of your system services and daemons. **/sbin/rc** generates a checklist of services needed at each run level. Based on the checklist on your console, put a "1" beside any services in the table below that **/sbin/rc** started while bringing your system to run level 1.

Table Solutions-1.

Level Service started	
	Mount file systems
	Set the system hostname
	Enable auxilliary (secondary) swap
	Start the syncer daemon to periodically flush buffer cache
	Start the "internet services deamon" that provides telnet, FTP access
	Start the mail daemon
	Start the LP "print spooler" daemon
	Start the clock (cron) daemon
	Start CDE

Answer:**Table Solutions-2.**

Level Service Started	
1	Mount file systems
1	Set the system hostname
1	Enable auxilliary (secondary) swap
1	Start the syncer daemon to periodically flush buffer cache
2	Start the "internet services deamon" that provides telnet, FTP access
2	Start the mail deamon
2	Start the LP "print spooler" daemon
2	Start the clock (cron) daemon
3	Start CDE

4. Now bring your machine to run level 2 and note the additional services that start. Update the table shown in question 3.

Answer:

init 2

5. Now bring your machine to run level 3 and note the additional services that start. Update the table in question 3. Your system should now be in a fully functional state.

Answer:

init 3

6. Based on the table you completed above, could you telnet to a machine that is in single-user mode?

Answer:

No. The internet service daemon (`inetd`) doesn't start until run level 2.

7. Based on the table you completed above, can you print a file while in single-user mode?

Answer:

No. The spooler doesn't start until run level 2.

8. Reboot your system again, but this time let it boot unattended using the default boot disk and kernel.

Answer:

```
# shutdown -r now 0
```

18-9. LAB: Kernel Configuration

1. PART I

SAM is the easiest tool for managing kernel drivers. Go to the SAM kernel driver screen by selecting: `SAM --> Kernel Configuration --> Drivers`. This should give you a list of all the drivers currently installed on your system. The `Current State` column indicates which of the drivers are actually configured in your kernel. The `Pending State` column indicates which drivers will be included in the next kernel rebuild. Based on this list, can your kernel successfully communicate with a tape drive that requires the `stape` driver? Can you dynamically load or unload the `stape` driver without rebooting, or is it a **static** kernel driver that requires a reboot?

Answer:

The `Current State` of the `stape` driver may be either `In` or `Out`. The "Type" column should show that this is a "Static" driver that would require a reboot.

2. Select a "static" driver that isn't yet configured in your kernel, and use the `Actions` menu to add the driver to the kernel.

Answer:

Select `stape` (or any other static driver) from the driver object list.

Select `Actions --> Add Driver to Kernel`.

3. What changed on the SAM screen to indicate that the static driver you selected will be included in the next kernel rebuild?

Answer:

The **Pending State** changes from "Out" to "In". However, since this is a static driver, the "Current State" remains "Out" until the kernel is rebuilt.

4. Dynamically load the hwgw DLKM driver:

- Select the hwgw driver from SAM's list of available drivers.
- Select **Actions --> Add driver to kernel**.
- Click **Yes** to confirm that you want to proceed.

You should be sitting in the **Kernel Module Attributes** window at this point. DLKM drivers may be statically or dynamically loaded in the kernel. If you set the module type to **Static**, the new module will become available only at the next kernel build and reboot.

If you choose **Loadable**, the driver becomes available immediately. If you choose **Load automatically at boot**, the kernel will load the DLKM automatically during the boot process. Otherwise, the kernel will load only the DLKM as needed.

- Set the **Module Type** to **Loadable**.
- Leave **Load automatically at boot** set to **No**.
- Click **Modify**.
- Click **OK**.
- Click **Yes** in the confirmation box that follows.

Answer:

5. What changed on the SAM screen to indicate that the module is immediately available for use by devices on your system?

Answer:

The hwgw **Current State** and **Pending State** columns both changed to "In".

1. PART II

Choose **List --> Subsystems**. Is your kernel currently configured to support LVM? Is your kernel currently configured to support a LAN connection? How can you tell?

Answer:

The **Current State** of both the LVM and the LAN/9000 subsystems should be "In", indicating that the functionality is available.

2. Choose a subsystem that isn't currently configured and add it to your kernel.

Answer:

Select **Diagnostics**.

Select **Actions --> Add Subsystem to Kernel**.

3. Is the new subsystem functionality added immediately? How can you tell?

Answer:

The **Pending State** changed to "In," suggesting that the new functionality will become available after the next kernel rebuild. However, the **Current State** is still "Out."

1. PART III

Choose **List --> Configurable Parameters**. What is the maximum number of processes your kernel can support? How many files can be open simultaneously?

Answer:

Look for the **nproc** and **nfile** parameters. The parameter values will vary.

2. Double the current value of the **maxusers** parameter. Does this change take effect immediately? How can you tell?

Answer:

Select **Maxusers**.

Select **Actions --> Modify Configurable Parameter**.

Double the parameter value. Click **OK**.

3. After changing the value of **maxusers**, note that the pending value for **nproc** changed as well. Select the **nproc** parameter and select **Actions --> Modify Configurable Parameter**. How does the definition of this parameter differ from the **maxusers** parameter? Why do you think **nproc** is defined in this manner?

Answer:

nproc is a calculated value, based on the value of **maxusers**. Since the number of processes allowed on the system largely depends on the number of users, this scheme makes some sense. Note that you can modify or replace the formula if you wish.

1. PART IV

Now that you have marked several pending changes to the kernel, choose **Actions --> Process New Kernel**. Before moving the kernel into place, look at the next question.

Answer:

Select **Actions --> Process New Kernel**.

Wait a few minutes while the new kernel is being processed.

2. Shortly, SAM will ask if you want to move the new kernel into place and reboot. SAM built your new kernel in the **/stand/build** directory. Which three files or subdirectories in **/stand/build** must be moved into place in **/stand** before rebooting with the new kernel?

Answer:

```
/stand/build/system.SAM  
/stand/build/vmunix_test  
/stand/build/dlkm.vmunix_test
```

3. Allow SAM to move the new kernel into place and reboot.

Answer:

Select **Move New Kernel into Place and Shutdown/Reboot System Now**.

Check **Overwrite /stand/system**.

Click **OK**.

Click **OK** again to confirm that you wish to overwrite **/stand/system**.

4. When your system returns, try the following:

```
# lsdev    # Lists currently configured static kernel drivers  
# sysdef  # Lists currently running your kernel's tunable parameters
```

Did your changes take effect?

Answer:

Hopefully your changes were successful.

1. PART V

Reboot your system to single user mode using the backup kernel.

Answer:

```
# shutdown -ry 0  
[ESC]
```

```
BOOTADMIN> boot pri isl
ISL> hpx ls
ISL> hpx -is /stand/vmunix.prev
```

2. Use **lsdev** and **sysdef** to check your parameter values and driver list again. Which kernel appears to be running? (Note: The **lsdev** and **sysdef** executables are in **/usr/sbin**, so you will need to do a **mount -a** first.)

Answer:

```
# mount -a
# sysdef
# lsdev
```

Hopefully, the driver and parameter lists reflect the original kernel, not the kernel you tuned in the earlier part of this lab.

3. Move your backup kernel and associated files back into place as the default kernel and reboot again.

Answer:

```
# kmupdate /stand/vmunix.prev
#cp /stand/system.prev /stand/system
# shutdown ---ry 0
```

19-13. LAB: Hands-On, Using the Software Distributor

1. Verify that the **swagentd** is running.

Answer:

```
ps -ef | grep swagentd
```

Listing Software: The following tasks can be performed using either the command line interface or by using SAM.

2. List the products installed on your system.

Answer:

Command line:

```
swlist -l product
```

3. Remove the Keyshell product. When removal has completed check the log file. If the product was successfully removed, the file **/usr/bin/keysh** should be gone.

Answer:

If using **SAM** choose **Software Management** from the **SAM** functional area launcher. Choose **Remove Local Host Software**. **SAM** will start an interactive **swremove** session.

Command line:

```
swremove           # to run interactively  
swremove Keyshell # to run interactively
```

Once you are in the **swremove** menu select **Change Software View** from the **View** menu. Select **Start with Products**. Highlight Keyshell. Select **Mark for Remove** from the **Action** menu. Select **Remove (Analysis)** from the **Action** menu. Once the analysis has completed, press **OK**. SD will ask for confirmation. Press **Yes**. Once removal has completed press **Logfile**.

You can also examine the logfile by looking at the file **/var/adm/sw/swremove.log**.

Installing Software:

4. Reinstall the Keyshell product. When installation has completed, check the log file. If the product was successfully installed, the file **/usr/bin/keysh** should be back.

Answer:

If using **SAM** choose **Software Management** from the **SAM** functional area launcher. Choose **Install Software to Local Host**. **SAM** will start an interactive **swinstall** session.

Command line:

```
swinstall           # to run interactively  
swinstall -s depot_name Keyshell # to run interactively
```

Once you are in the **swinstall** menu, specify your depot location. Do not select any Software Filter. Select **Change Software View** from the **View** menu. Select **Start with Products**. Highlight the Keyshell. Select **Mark for Remove** from the **Action** menu. Select **Install (Analysis)** from the **Action** menu. Once the analysis has completed, press **OK**. SD will ask for confirmation. Press **Yes**. Once installation has completed press **Logfile**. You can also examine the logfile by looking at the file **/var/adm/sw/swinstall.log**.

20-9. LAB: Patch Management

1. PART I

Run the **swinstall** utility to install the EchoApp utility on your machine. EchoApp should be available in a local directory depot on your machine called **/labs/depots/echoapp.dep**ot.

Answer:

```
# swinstall
```

```
Depot Type: Local Directory
Depot Path: /labs/depots/echoapp.depot
```

Select EchoApp. Choose **Actions --> Install Analysis**. After the install completes, exit out of **swinstall**.

2. List the products installed on your host. Is EchoApp listed? Try running **echoapp** to see that it works: # /opt/echoapp/bin/echoapp

Answer:

```
# swlist -l product EchoApp
# /opt/echoapp/bin/echoapp
```

1. PART II

Oftentimes administrators download and install patches from HP's SupportLine web site. For this lab exercise, that won't be necessary. The patch you will install, PHSS_01111, is already in your **/labs** directory. To start, copy the patch to the **/tmp** directory.

Answer:

```
# cp /labs/PHSS_01111 /tmp
```

2. **cd** to the **/tmp** directory and unpack the patch's **shar** archive. Read the **.text** file. What problem does this patch fix?

Answer:

```
# cd /tmp @ # sh PHSS_01111
# more PHSS_01111.text
```

The patch makes the **echoapp** output a little more interesting.

3. Follow the directions in the patch's **.text** file to install the patch. Since we may need to remove this patch eventually, do NOT to use the **patch_save_files=false** option on **swinstall**. Try running **echoapp** again. Did the patch seem to work?

Answer:

```
# swinstall -x autoreboot=true \
           -x patch_match_target=true \
           -s /tmp/PHSS_01111.depot

# /opt/echoapp/bin/echoapp
```

The output should look a little different.

4. `swinstall` keeps copies of all files that have been replaced by patches in the `/var/adm/sw/save` directory. Use the `find` command to take a look at the PHSS_01111 files under this directory. Which two files were replaced by the PHSS_01111 patch?

Answer:

```
# find /var/adm/sw/save
```

`/echoapp/doc/echoapp_pic` and `/opt/echoapp/bin/echoapp` were both replaced by the patch.

5. Now run `swremove` to remove the patch. Again look at the files under `/var/adm/sw/save`. What changed?

Answer:

```
# swremove PHSS_01111
# find /var/adm/sw/save
```

Note that the `/var/adm/sw/save/PHSS_01111` directory is gone.

6. Now remove the EchoApp product as well.

Answer:

```
swremove EchoApp
```

1. PART III

Let's try installing EchoApp again, but this time install the product from `/labs/depots/echoapp+patch.dep`t. Note that this depot contains both the EchoApp product and the PHSS_01111 patch.

Answer:

```
# swinstall
  Source Depot Type: Local Directory
  Source Depot Path: /labs/depots/echoapp+patch.dep
```

Select the EchoApp product. Choose **Actions --> Install Analysis**.

2. Run `echoapp`. Based on this experiment, what is the advantage of having products and their associated patches in the same depot? (Note: This feature is new at 11.x.)

Answer:

```
# /opt/echoapp/bin/echoapp
```

When installing a product, `swinstall` automatically selects the associated patches for install, too.

3. Now remove the EchoApp product. Afterwards, use `swlist` to see if PHSS_01111 is still installed on your system. When you remove a product, what happens to the patches associated with that product?

Answer:

```
# swremove EchoApp
```

When a product is removed with `swremove`, all patches associated with that product are automatically removed as well.

4. Confirm your conclusion in the previous question by checking the `/var/adm/sw/save` directory.
-

1. PART IV

Install EchoApp and the PHSS_01111 patch again from the echoapp+patch.depot. Use the "du" command to check the amount of space occupied by PHSS_01111 in the `/var/adm/sw/save` directory.

Answer:

```
# swinstall
  Source Depot Type: Local Directory
  Source Depot Path: /labs/depots/echoapp+patch.depot
```

Select the EchoApp product.

Choose Actions --> Install Analysis.

```
# du -sk /var/adm/sw/save/PHSS_01111
```

2. As you install patches on your system, the `/var/adm/sw/save` directory can consume a significant amount of disk space. You can save some space in `/var/adm/sw/save` by committing a patch. Try committing the PHSS_01111 patch. What effect does the `swmodify` command have on the `/var/adm/sw/save` directory? Use `du` to find out.

```
# swmodify -x patch_commit=true 'PHSS_01111.*'
# du -sk /var/adm/sw/save/PHSS_01111
```

Answer:

Committing the patch removes all of the `save` files for the patch, thus saving disk space in `/var`.

3. Try to remove the patch. What appears to be the downside of committing a patch?

Answer:

```
# swremove PHSS_01111
```

Fails! After a patch is committed, it can no longer be removed.

4. Can you still remove the product associated with the patch? Try it. Then **swlist** for both the EchoApp product and the patch.

Answer:

```
# swremove EchoApp
# swlist -l product EchoApp
# swlist -l product PHSS_01111
```

After a patch is committed, you can't remove the patch. However, you can still remove the product, which removes the patch as well.

21-9. LAB: Connecting to the Network

1. PART I

set_parms first asks if you want to connect to the network. For this lab exercise, we do want to connect to the network, so click "OK."

Answer:

Click **OK**.

2. Next, **set_parms** asks if you wish to obtain your network parameters via DHCP. DHCP is a service that can automatically provide your machine with an IP address and hostname, if a DHCP server exists on your network. For this exercise, click "NO".

Answer:

Click **NO**.

3. Answer the prompts in the screens that follow to set your hostname, time zone information, and IP address. When asked if you wish to set additional network information (for example, a default gateway, netmask, and DNS and NIS information), answer **NO**. We will configure these parameters later via SAM.

NOTE:

If you make a mistake while entering any of the network parameters, you can run **set_parms** again any time by typing **set_parms initial**.

Answer:

Click **NO**.

1. PART II

After running `set_parms` and logging in, go to:

SAM → Networking and Communications → Network Interface Cards

to ensure that your IP is set correctly. What hardware slot number is your `lan0` card installed in? Is `lan0` enabled for use? What is `lan0`'s IP address? Are there any other LAN cards on your machine?

Answer:

Answers will vary, depending on the system configuration.

2. While still at the Network Interface Card screen, set your subnet mask. Without the subnet mask, you may not be able to communicate with other nodes on your LAN. Set your subnet mask for `lan0` via SAM.

Each machine on a network, regardless of the number of LAN interface cards, has a single hostname. If you have multiple LAN cards on a host, SAM requires you to configure a unique "Host Name Alias" for each LAN card. This is useful in troubleshooting situations where you may wish to access a specific LAN card on a target machine rather than the general hostname. If SAM asks you to set an alias, simply use your hostname with an "a" appended to the end.

Answer:

SAM → Networking and Communications

→ Network Interface Cards

Select `lan0`.

Actions → Modify

Enter the subnet mask.

Click OK.

SAM may still prompt you to add a hostname alias as described above.

1. PART III

Try pinging your own IP address to see if your IP address is set properly. Did this work?

Answer:

ping 128.1.1.1 # use your IP address.

This should work.

2. Next, try pinging a couple of neighboring machines' IP addresses. Did this work?

Answer:

```
# ping 128.1.1.2 # use your neighbors' IP addresses.  
# ping 128.1.1.3
```

This should work, too.

3. Next, try pinging a neighboring machine using its hostname. Can you guess why this fails? (Hint: Look in the `/etc/hosts` file.)

Answer:

The `/etc/hosts` file doesn't contain an entry for your neighbor's machine yet.

1. PART IV

Which hosts' entries are configured so far?

Answer:

```
SAM --> Networking and Communications  
      --> Hosts  
          --> Local Hosts File
```

`localhost` is a special hostname that always references your local machine. This entry should exist in every `/etc/hosts` file.

The other entry in your `/etc/hosts` file defines your own hostname and IP address. Since this entry exists, you should be able to ping yourself by hostname.

2. Add entries in your hosts file for the other machines in your row.

Answer:

`Actions --> Add`

Enter your neighbor's IP and hostname.

Click `OK`.

3. Try pinging your neighbors by hostname again. Does this work?

Answer:

```
# ping neighbor                      # Neighbor's hostname will vary.
```

This should work.

1. PART V

Use SAM to configure your default gateway.

Answer:

SAM -> Networking and Communications

-> Hosts

Actions -> Configure Default Gateway

Enter the Default Gateway address suggested by your instructor

Optionally, you can enter the hostname of the gateway.

Click OK.

2. Use SAM to configure your default DNS domain name. Don't configure any "Other Domains to Search".

Answer:

SAM -> Networking and Communication

-> DNS/BIND

-> DNS Resolver

Actions -> Set Default Domain

Enter the DNS Domain Name suggested by your instructor

Don't enter any Other Domains to Search

Click OK

3. Now use SAM to configure the IP addresses of your DNS nameservers.

Answer:

SAM -> Networking and Communications

-> DNS/BIND

-> DNS Resolver

Actions -> Specify Name Servers

Enter the IP address(es) of the DNS server(s) suggested by your instructor.

Click "OK".

1. PART VI

Did your DNS configuration work? If DNS is configured properly on your machine, and the DNS server is functioning, you should be able to lookup hostnames of machines outside your local network. Try an `nslookup` on `www.hp.com`. What happens?

Answer:

```
# nslookup www.hp.com
```

Fails! "No address information available."

2. Look carefully at the output from the `nslookup` you did in the previous exercise. Can you explain why you were unable to find an IP address for `www.hp.com`? (Hint: What lookup source is `nslookup` using at this point?)

Answer:

`nslookup` is using the `/etc/hosts` file on your machine. The local hosts file doesn't contain an entry for `www.hp.com`, so the lookup fails.

3. Configure the name service switch via SAM to ensure that your system consults the DNS server if a hostname isn't found in the local `/etc/hosts` file.

Answer:

Select `SAM --> Networking and Communications`.

Select `--> Name Service Switch`.

Select `Hosts`.

Select `Actions --> Configure Name Service Switch`.

Set `/etc/hosts` as the first service.

Set `DNS` as the second service.

4. After changing the nameservice switch, SAM reboots your machine. After the reboot, do an `nslookup` on `www.hp.com` again. Are you able to resolve `www.hp.com` now?

Answer:

```
# nslookup www.hp.com
```

This should succeed, assuming your DNS server is configured properly.