Student Kit

# HP-UX System and Network Troubleshooting

H4264 (Version D.01)

HP education training www.hp.com/education

# HP-UX System and Network Troubleshooting

H4264S D.01

HP Training

# Student guide

invent

Copyright 2005 Hewlett-Packard Development Company, L.P.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

This is an HP copyrighted work that may not be reproduced without the written permission of HP. You may not use these materials to deliver training to any person outside of your organization without the written permission of HP.

UNIX® is a registered trademark of the Open Group.

Printed in the USA

**HP-UX System and Network Troubleshooting**
Student guide
February 2005

# Contents

**Module 1 — Troubleshooting Resources**
1–1. SLIDE: Course Contents ..... 1-2
1–2. SLIDE: Module Objectives ..... 1-3
1–3. SLIDE: Troubleshooting Resources ..... 1-4
1–4. SLIDE: Troubleshooting Philosophy ..... 1-5
1–5. SLIDE: Troubleshooting Techniques ..... 1-8
1–6. SLIDE: Manuals, Websites, and Newsgroups ..... 1-10
1–7. SLIDE: Man Page Sections and `catman -w` ..... 1-12
1–8. SLIDE: Printouts (Output from Commands) 1 of 2 ..... 1-15
1–9. SLIDE: Printouts (Output from Commands) 2 of 2 ..... 1-19
1–10. SLIDE: Printouts (Configuration Files) ..... 1-21
1–11. SLIDE: System Log Book ..... 1-24
1–12. SLIDE: System Log Files ..... 1-25
1–13. SLIDE: Module Objectives ..... 1-28
1–14. QUIZ: Review Questions ..... 1-29
1–15. LAB: System Log Files ..... 1-30
1–16. LAB: MANPATH Issue ..... 1-35

**Module 2 — System Recovery with Ignite-UX**
2–1. SLIDE: Module Objectives ..... 2-2
2–2. SLIDE: System Recovery Choices ..... 2-3
2–3. SLIDE: What's On the Tape? ..... 2-5
2–4. SLIDE: Creating a Minimum Core Recovery Tape ..... 2-7
2–5. SLIDE: Creating a Full Core Recovery Tape ..... 2-9
2–6. SLIDE: Adding to the Archive ..... 2-10
2–7. SLIDE: `make_tape_recovery` Caveats ..... 2-12
2–8. SLIDE: System Recovery Options ..... 2-13
2–9. SLIDE: Verifying the Recovery Tape ..... 2-15
2–10. SLIDE: Booting the Recovery Tape ..... 2-16
2–11. SLIDE: The `make_net_recovery` Tool ..... 2-19
2–12. SLIDE: Initiating `make_net_recovery` from an Ignite Server ..... 2-20
2–13. SLIDE: Selecting Files for the Client Archive ..... 2-21
2–14. SLIDE: Monitoring Creation of Client Archive ..... 2-22
2–15. SLIDE: Initiating `make_net_recovery` on the Client ..... 2-23
2–16. SLIDE: Using the Archive to Recover a Client ..... 2-24
2–17. SLIDE: Module Objectives ..... 2-26
2–18. LAB: Creating a `make_net_recovery` archive ..... 2-27
2–19. LAB: Creating a `make_tape_recovery` archive (Optional) ..... 2-28

**Module 3 — Troubleshooting Non-Bootable LVM Disks and LVM Structures**
3–1. SLIDE: Module Objectives ..... 3-2
3–2. SLIDE: LVM –Logical Volume Manager - Review ..... 3-3
3–3. SLIDE: LVM Structures ..... 3-4
3–4. SLIDE: Recovering Lost or Damaged Structures ..... 3-6
3–5. SLIDE: Missing Device Files: Symptoms ..... 3-8
3–6. SLIDE: Missing Device Files: Solution ..... 3-10
3–7. SLIDE: Corrupt `/etc/lvmtab`: Symptoms ..... 3-12
3–8. SLIDE: Corrupt `/etc/lvmtab`: Solution ..... 3-14


© 2005 Hewlett-Packard Development Company, L.P.

3–9. SLIDE: Outdated Kernel Structures: Symptoms ..... 3-16
3–10. SLIDE: Outdated Kernel Structures: Solution ..... 3-18
3–11. SLIDE: Failed Disks: Symptoms (1 of 2) ..... 3-19
3–12. SLIDE: Failed Disks: Symptoms (2 of 2) ..... 3-21
3–13. SLIDE: Activating a Volume Group that Contains Failed Disks ..... 3-23
3–14. SLIDE: Restoring a Powerfailed Disk ..... 3-25
3–15. SLIDE: Replacing a Failed Disk ..... 3-26
3–16. SLIDE: Backing up the PVRA/VGRA ..... 3-27
3–17. SLIDE: Restoring the PVRA/VGRA ..... 3-29
3–18. SLIDE: Restoring the PVRA/VGRA: Complete Procedure ..... 3-30
3–19. SLIDE: Restoring Unmirrored File System Data (1 of 2) ..... 3-32
3–20. SLIDE: Restoring Unmirrored File System Data (2 of 2) ..... 3-33
3–21. SLIDE: Restoring Mirrored File System Data (1 of 2) ..... 3-34
3–22. SLIDE: Restoring Mirrored File System Data (2 of 2) ..... 3-35
3–23. SLIDE: Removing Corrupted Physical Volumes ..... 3-37
3–24. SLIDE: Removing Corrupted Volume Groups ..... 3-40
3–25. SLIDE: Module Objectives ..... 3-41
3–26. LAB: Recovering LVM Disks and Structures ..... 3-43

**Module 4 — Troubleshooting a PA-RISC System Boot Disk**

4–1. SLIDE: Module Objectives ..... 4-2
4–2. SLIDE: Overview of System Startup ..... 4-3
4–3. SLIDE: Problems at Boot Time ..... 4-6
4–4. SLIDE: PA-RISC Boot Disk Structure Details ..... 4-11
4–5. SLIDE: Potential Boot Disk Failures ..... 4-13
4–6. SLIDE: Using the Install Media for Repairs ..... 4-15
4–7. SLIDE: The LABEL File ..... 4-16
4–8. SLIDE: Boot Data Reserve Area (BDRA) ..... 4-17
4–9. SLIDE: Corrupt LABEL File or BDRA ..... 4-18
4–10. SLIDE: The `lvlnboot` Command ..... 4-19
4–11. SLIDE: Recovery Shell ..... 4-26
4–12. SLIDE: Normal Boot ..... 4-27
4–13. SLIDE: Booting into the Recovery Shell ..... 4-29
4–14. SLIDE: Booting into the Recovery Shell — Logical Picture ..... 4-30
4–15. SLIDE: Using the HP-UX Install Media ..... 4-31
4–16. SLIDE: HP-UX System Recovery Main Menu ..... 4-35
4–17. SLIDE: HP-UX Recovery Menu ..... 4-37
4–18. SLIDE: Repairing Damage — Missing / Corrupt LIF Utilities ..... 4-41
4–19. SLIDE: Repairing Damage — Corrupt LABEL File ..... 4-44
4–20. SLIDE: Repairing Damage — Missing Kernel ..... 4-46
4–21. SLIDE: Repairing Damage — Rebuilding Kernel ..... 4-47
4–22. SLIDE: System Startup — Binary Files ..... 4-49
4–23. SLIDE: Binary Startup Files ..... 4-50
4–24. SLIDE: Repairing Damage — Missing Binary File(s) ..... 4-51
4–25. SLIDE: Corrupted PVRA/VGRA Structures on the Boot Disk ..... 4-61
4–26. SLIDE: LVM Maintenance Mode ..... 4-63
4–27. SLIDE: Booting without Quorum ..... 4-64
4–28. SLIDE: Recovering a Mirrored Boot Disk ..... 4-66
4–29. SLIDE: Module Objectives ..... 4-68
4–30. QUIZ: Review Questions ..... 4-69
4–31. LAB: Booting: Corrupt Stable Storage and Bad LIF Area ..... 4-70
4–32. LAB: Recovering an Unbootable System — Missing kernel ..... 4-72


© 2005 Hewlett-Packard Development Company, L.P.

**Module 5 — Booting HP-UX 11i v2 via the EFI**


5–1. SLIDE: Reviewing the Module Objectives ..... 5-2
5–2. SLIDE: Integrity/EFI Boot Concepts ..... 5-3
5–3. SLIDE: Integrity Boot Disk Structure ..... 5-4
5–4. SLIDE: Integrity Boot Disk System Partition Structure ..... 5-6
5–5. SLIDE: Integrity Boot Disk OS Partition Structure ..... 5-8
5–6. SLIDE: Integrity Boot Disk HP Service Partition Structure ..... 5-9
5–7. SLIDE: PARISC / EFI Boot Process Comparison ..... 5-10
5–8. SLIDE: Interacting with the EFI Boot Manager ..... 5-12
5–9. SLIDE: Initiating an Integrity System Boot ..... 5-13
5–10. SLIDE: Choosing a Boot Device ..... 5-15
5–11. SLIDE: Choosing an Alternate Kernel/Mode ..... 5-16
5–12. SLIDE: Interacting with the EFI Shell ..... 5-18
5–13. SLIDE: Accessing and Leaving the EFI Shell ..... 5-19
5–14. SLIDE: EFI Shell Command Overview ..... 5-21
5–15. SLIDE: Executing EFI Commands ..... 5-25
5–16. SLIDE: Viewing System Information via the EFI ..... 5-27
5–17. SLIDE: Viewing EFI Devices via the EFI ..... 5-31
5–18. SLIDE: Viewing EFI Devices via HP-UX ..... 5-34
5–19. SLIDE: Managing the EFI Boot Menu via the EFI ..... 5-35
5–20. SLIDE: Managing the EFI Boot Menu via HP-UX ..... 5-37
5–21. SLIDE: Managing EFI File Systems via the EFI ..... 5-39
5–22. SLIDE: Managing EFI File Systems via HP-UX ..... 5-42
5–23. SLIDE: Booting from an Alternate Disk via the EFI ..... 5-44
5–24. SLIDE: Booting Alternatives from the HP-UX Boot Loader ..... 5-45
5–25. SLIDE: Booting from the Install DVD ..... 5-47
5–26. SLIDE: Booting from an Ignite-UX Server ..... 5-48
5–27. SLIDE: Booting from a make_tape_recovery Tape ..... 5-49
5–28. SLIDE: Booting from the HPSP or Diagnostics CD ..... 5-53
5–29. SLIDE: Mirroring an Integrity Boot Disk ..... 5-57
5–30. SLIDE: Configuring the Partition Table ..... 5-58
5–31. SLIDE: Configuring the System Partition ..... 5-60
5–32. SLIDE: Configuring the OS Partition ..... 5-62
5–33. SLIDE: Configuring the EFI Boot Manager Menu ..... 5-64
5–34. SLIDE: Booting from the Alternate Mirror ..... 5-65
5–35. SLIDE: Replacing a Failed Integrity Boot Disk Mirror ..... 5-67
5-36. SLIDE: Rebuilding the Partition Table ..... 5-68
5–37. SLIDE: Rebuilding the System Partition ..... 5-70
5–38. SLIDE: Rebuilding the OS Partition ..... 5-72
5-39. SLIDE: Reviewing the Module Objectives ..... 5-73
5–40. SLIDE: For Further Study ..... 5-74
5–41. LAB: Booting HP-UX 11i v2 via the EFI ..... 5-76


**Module 6 — Booting: Missing or Corrupt ASCII Startup Files**


6–1. SLIDE: Module Objectives ..... 6-2
6–2. SLIDE: System Startup ..... 6-3
6–3. SLIDE: System Configuration Files ..... 6-4
6–4. SLIDE: Interrupting the `rc` Start Scripts ..... 6-6
6–5. SLIDE: System Startup Links ..... 6-8
6–6. SLIDE: The `/etc/rc.config.d` Directory ..... 6-10
6–7. SLIDE: Single-User Mode ..... 6-12
6–8. SLIDE: Booting to Single-User Mode on PA-RISC ..... 6-14




© 2005 Hewlett-Packard Development Company, L.P.

6–9. SLIDE: Module Objectives ..... 6-16
6–10. QUIZ: Review Questions ..... 16-7
6–11. LAB: Recovering an ASCII Startup File, Start Link, and Startup-Script Configuration File ..... 6-18

**Module 7 — Troubleshooting Login Problems**
7–1. SLIDE: Module Objectives ..... 7-2
7–2. SLIDE: Login Methods ..... 7-3
7–3. SLIDE: ASCII Login (Direct Connect) ..... 7-4
7–4. SLIDE: ASCII Login (Network) ..... 7-7
7–5. SLIDE: GUI Login (CDE Session) ..... 7-10
7–6. SLIDE: GUI Login (CDE Failsafe Login) ..... 7-12
7–7. SLIDE: Protecting the System with `/etc/shadow` ..... 7-14
7–8. SLIDE: Pluggable Authentication Modules ..... 7-17
7–9. SLIDE: Module Objectives ..... 7-19
7–10. QUIZ: Review Questions ..... 7-20
7–11. LAB: Problems Logging On ..... 7-21

**Module 8 — Troubleshooting File Systems**
8–1. SLIDE: Module Objectives ..... 8-2
8–2. SLIDE: HP-UX File Systems ..... 8-3
8–3. SLIDE: Example of File System Corruption ..... 8-4
8–4. SLIDE: How the File System Becomes Corrupt ..... 8-5
8–5. SLIDE: `fsck` ..... 8-7
8–6. SLIDE: What Is `fsck` Checking? ..... 8-8
8–7. SLIDE: `fsck` on HFS versus JFS ..... 8-10
8–8. SLIDE: Introducing `fsdb` ..... 8-13
8–9. SLIDE: Using `fsdb` ..... 8-14
8–10. SLIDE: Using `fsadm` to Defragment a File System ..... 8-17
8–11. SLIDE: Module Objectives ..... 8-20
8–12. QUIZ: Review Questions ..... 8-21
8–13. LAB: Recovering a Corrupt VxFS File System ..... 8-22
8–14. LAB: Using `fsdb` ..... 8-24
8–15. LAB: Using `fsadm` ..... 8-29

**Module 9 — Troubleshooting Process Problems**
9–1. SLIDE: Module Objectives ..... 9-2
9–2. SLIDE: What Is a Process? ..... 9-3
9–3. SLIDE: Processes, Thread States, and Transitions ..... 9-6
9–4. SLIDE: Looking at Processes with ps ..... 9-11
9–5. SLIDE: Identifying the Hierarchy of a Process ..... 9-18
9–6. SLIDE: The `kill` Command ..... 9-19
9–7. SLIDE: Zombie Processes ..... 9-25
9–8. SLIDE: Clearing Zombie Processes ..... 9-28
9–9. SLIDE: Identifying CPU-Intensive Processes ..... 9-30
9–10. SLIDE: Troubleshooting CPU-Intensive Processes ..... 9-31
9–11. SLIDE: Module Objectives ..... 9-33
9–12. QUIZ: Review Questions ..... 9-34
9–13. LAB: Process Hierarchy ..... 9-36
9–14. LAB: Zombie Processes ..... 9-37
9–15. LAB: CPU-Intensive Processes ..... 9-39


© 2005 Hewlett-Packard Development Company, L.P.

**Module 10 — Troubleshooting an HP-UX 11i v2 Kernel**


10-1. SLIDE: Module Objectives ..... 10-2
10-2. SLIDE: Kernel Troubleshooting Overview ..... 10-3
10-3. SLIDE: Kernel Recovery Roadmap ..... 10-4
10-4. SLIDE: Using the kcweb GUI Interface ..... 10-6
10-5. SLIDE: Understanding the kcweb Architecture ..... 10-8
10-6. SLIDE: Launching kcweb via a Local Browser ..... 10-9
10-7. SLIDE: Using kcweb via a Remote Browser ..... 10-11
10-8. SLIDE: Viewing the Kernel Change Log via kcweb ..... 10-13
10-9. SLIDE: Viewing the Kernel Change Log via kclog ..... 10-14
10-10. SLIDE: Monitoring kcweb Logs ..... 10-16
10-11. SLIDE: Booting from an Alternate Kernel ..... 10-17
10-12. SLIDE: Booting to Tunable Maintenance Mode ..... 10-19
10-13. SLIDE: Module Objectives ..... 10-21
10-14. LAB: Troubleshooting the HP-UX 11i v2 Kernel ..... 10-22


**Module 11 — Troubleshooting a Kernel Crash Dump**


11–1. SLIDE: Module Objectives ..... 11-2
11–2. SLIDE: What is a Kernel Crash? ..... 11-3
11–3. SLIDE: Configuring Selective and Compressed Kernel Dumps ..... 11-6
11–4. SLIDE: Kernel Crash Dump Overview ..... 11-7
11–5. SLIDE: Kernel Crash Dump Configuration Overview ..... 11-8
11–6. SLIDE: Configuring Dump Devices ..... 11-9
11–7. SLIDE: Dump Device Concepts ..... 11-10
11–8. SLIDE: How Much Dump Space is Required? ..... 11-11
11–9. SLIDE: Configuring Dump Devices via `crashconf` ..... 11-13
11–10. SLIDE: Configuring Selective Crash Dumps ..... 11-15
11–11. SLIDE: Selective Crash Dump Concepts ..... 11-16
11–12. SLIDE: Configuring Selective Dumps via `crashconf` ..... 11-18
11–13. SLIDE: Configuring Compressed Dumps ..... 11-19
11–14. SLIDE: Compressed Crash Dump Concepts ..... 11-20
11–15. SLIDE: Configuring Compressed Dumps via `crashconf` ..... 11-22
11–16. SLIDE: Configuring `savecrash` Functionality ..... 11-23
11–17. SLIDE: `savecrash` Concepts ..... 11-24
11–18. SLIDE: Configuring Post-dump Behavior via `savecrash` ..... 11-26
11–19. SLIDE: Configuring Post-dump Behavior via `savecrash` (Continued) ..... 11-27
11–20. SLIDE: When a Panic Occurs ..... 11-29
11–21. SLIDE: Manually Initiating a Dump ..... 11-30
11–22. SLIDE: Enabling Selective Dump at Dump Time ..... 11-31
11–23. SLIDE: Enabling Compression at Dump Time ..... 11-32
11–24. SLIDE: Terminating a Dump ..... 11-33
11–25. SLIDE: Converting the Dump with `crashutil` ..... 11-34
11–26. SLIDE: Viewing the Dump Contents with `q4` ..... 11-36
11-27. SLIDE: Working with the Response Center ..... 11-38
11–28. SLIDE: Removing the Dump ..... 11-44
11–29. SLIDE: Review Module Objectives ..... 11-45
11–30. QUIZ: Review Questions ..... 11-46
11–31. LAB: Kernel Crash Dump Configuration ..... 11-47




© 2005 Hewlett-Packard Development Company, L.P.

Contents

**Module 12 — Troubleshooting Patch Problems**


12–1. SLIDE: Module Objectives ..... 12-2
12–2. SLIDE: Managing Patches ..... 12-3
12–3. SLIDE: New Attribute: **`is_patch`** ..... 12-4
12–4. SLIDE: New Attribute: **`patch_state`** ..... 12-5
12–5. SLIDE: New Attributes: **`supersedes`**, **`superseded_by`** ..... 12-6
12–6. SLIDE: Location of Rollback Files ..... 12-7
12–7. SLIDE: Committing Patches ..... 12-8
12–8. SLIDE: Managing Patches with **`swinstall`** ..... 12-9
12–9. SLIDE: Additional Patch Management Tools ..... 12-11
12–10. SLIDE: The **`cleanup`** Tool ..... 12-12
12–11. SLIDE: The **`show_patches`** Tool ..... 12-13
12–12. SLIDE: The **`check_patches`** Tool ..... 12-14
12–13. SLIDE: The **`security_patch_check`** Tool ..... 12-18
12–14. SLIDE: **`security_patch_check`** ..... 12-19
12–15. SLIDE: Setting Up **`security_patch_check`** ..... 12-21
12–16. SLIDE: Using **`security_patch_check`** ..... 12-22
12–17. SLIDE: Working with SD-UX Logfiles ..... 12-23
12–18. SLIDE: The Installed Products Database (IPD) ..... 12-24
12–19. SLIDE: Repairing a Corrupt IPD ..... 12-25
12–20. SLIDE: Module Objectives ..... 12-27
12–21. LAB: Troubleshooting Patch Installation ..... 12-28
12–22. LAB: Troubleshooting a corrupt IPD ..... 12-34


**Module 13 — Troubleshooting Network Problems: Part 1**


13–1. SLIDE: Module Objectives ..... 13-2
13–2. SLIDE: Addressing the Correct Layer ..... 13-3
13–3. SLIDE: Troubleshooting LAN Card Configuration Problems ..... 13-4
13–4. SLIDE: Solving the LAN Card Configuration Problem ..... 13-5
13–5. SLIDE: Troubleshooting: Using the **`lanadmin`** Command ..... 13-6
13–6. SLIDE: The Subnetting Concept ..... 13-8
13–7. SLIDE: IP Addresses in a Subnetted Network ..... 13-10
13–8. SLIDE: Netmasks in a Subnetted Network ..... 13-11
13–9. SLIDE: Subnet Addresses ..... 13-13
13–10. SLIDE: Host IP Addresses on a Subnet ..... 13-15
13–11. SLIDE: Troubleshooting Subnet Mask Problems ..... 13-17
13–12. SLIDE: The Routing Concept ..... 13-19
13–13. SLIDE: Troubleshooting Routing Problems ..... 13-20
13–14. SLIDE: Monitoring the Network with Ethereal ..... 13-22
13–15. SLIDE: Tuning the Network with **`ndd`** (HP-UX 11.x) ..... 13-24
13–16. SLIDE: Module Objectives ..... 13-25
13–17. QUIZ: Review Questions ..... 13-26
13–18. LAB: Using **`ndd`** to Set Network Parameters ..... 13-27
13–19. LAB: Configuring the LAN Card ..... 13-28
13–20. LAB: Host Communication Problems ..... 13-31


**Module 14 — Troubleshooting Network Problems: Part II**


14–1. SLIDE: Module Objectives ..... 14-2
14–2. SLIDE: NFS Functionality ..... 14-3
14–3. SLIDE: NFS Server Setup Issues ..... 14-4
14–4. SLIDE: Configuring the NFS Server ..... 14-6




© 2005 Hewlett-Packard Development Company, L.P.

14–5. SLIDE: Unmounting an Active Exported File System ..... 14-8
14–6. SLIDE: NFS Client Issues ..... 14-10
14–7. SLIDE: DNS Functionality ..... 14-13
14–8. SLIDE: DNS Configuration Files and Setup ..... 14-15
14–9. SLIDE: Troubleshooting a DNS Problem (Example) ..... 14-19
14–10. SLIDE: Testing Name Servers with **`nslookup`** ..... 14-20
14–11. SLIDE: Testing Resolvers with **`nsquery`** ..... 14-22
14–12. SLIDE: Module Objective ..... 14-24
14–13. QUIZ: Review Questions ..... 14-25
14–14. LAB: NFS Problems ..... 14-26
14–15. LAB: DNS Data File Problem ..... 14-32

**Solutions**


© 2005 Hewlett-Packard Development Company, L.P.

# Overview

## Course Description

The 5-day *HP-UX System and Network Troubleshooting* course is aimed at experienced HP-UX system administrators who need to be able to isolate and solve problems found on their HP 9000 systems.

This course examines some of the problems that can arise in key system areas. It also presents useful tools that can help in finding the cause of problems and discusses tools that can be used to fix these problems. The course breaks the system down into a number of areas and examines some of the problems that can arise in these areas. It presents some of the tools that can be used to find the cause of the problems and the tools that can be used to fix the problems. The course includes lectures as well as labs.

## Course Goals

The course goals are to prepare the student to be a more effective and efficient system administrator of an HP 9000 workstation or server running HP-UX release 11i. The course will help system administrators diagnose and solve problems.

## Student Performance Objectives

### Module 1 — Troubleshooting Resources

- Describe the course contents and state the course goals.
- List four HP-UX troubleshooting resources.
- Create **`/usr/share/lib/whatis`** for use with the **`man -k`** keyword search.
- Correlate your troubleshooting technique to sample troubleshooting techniques.

### Module 2 — System Recovery with Ignite-UX

- Understand system recovery choices
- Describe the layout of a system recovery tape
- Describe how to create a system recovery tape
- Understand system recovery options
- Describe how to verify a system recovery tape
- Describe how to boot from a system recovery tape
- Understand how to create a **`make_net_recovery`** archive


© 2005 Hewlett-Packard Development Company, L.P.

**Module 3 — Troubleshooting Non-Bootable LVM Disks and LVM Structures**

- Describe the critical structures in an LVM environment.
- Diagnose and restore missing LVM device files.
- Diagnose and rebuild a missing or corrupt **`/etc/lvmtab`** file.
- Update the kernel LVM structures when missing disks return.
- Identify failed LVM disks.
- Backup LVM data structures using **`vgcfgbackup`**.
- Restore LVM data structures using **`vgcfgrestore`**.
- Restore file system and user data when a non-mirrored data disk fails.
- Restore file system and user data when a mirrored data disk fails.
- Remove a corrupted physical volume from a volume group.
- Remove a corrupted volume group from a host.
- List critical recovery information that should be recorded in a system log book.

**Module 4 — Troubleshooting a PA-RISC System Boot Disk**

- Describe PA-RISC system startup
- Describe the boot disk structure
- Describe potential boot failures
- Understand the **`LABEL`** file and the **`BDRA`**
- Understand the operation of the **`lvlnboot`** command
- Describe the operation of the recovery shell and its capabilities
- Understand the implications of missing or corrupt binary startup files
- Describe the recovery process for a missing kernel

**Module 5 — Booting HP-UX 11i v2 via the EFI**

- Describe the major steps in the HP-UX 11i v2 boot process
- Describe the major components of an HP-UX 11i v2 boot disk
- Initiate an HP-UX 11i v2 boot from a command prompt, the MP, or the EFI shell


© 2005 Hewlett-Packard Development Company, L.P.

**Overview**

- Navigate and manage EFI file systems
- Use the EFI help utility
- Boot from the default boot disk
- Boot from an alternate boot disk
- Boot from a DVD or CDROM
- Boot from an Ignite-UX install/recovery server
- Boot from an Ignite recovery tape
- Boot from an alternate kernel
- Boot to single-user, LVM maintenance, or tunable maintenance mode
- Configure a mirrored 11i v2 boot disk
- Recover a failed mirrored 11i v2 boot disk

**Module 6 — Booting: Missing or Corrupt ASCII Startup Files**

- Describe the system startup process after the operating system is loaded and binary startup files are executed.
- Identify important system configuration files and start links.
- Identify function and importance of `/etc/rc.config.d`.
- Boot into single-user mode.
- Systematically identify, check, and fix corrupted system configuration files and missing start links.
- Explain the significance of an invalid or corrupted configuration file existing in the `/etc/rc.config.d`.

**Module 7 — Troubleshooting Login Problems**

- Describe the login process as it pertains to a `getty`, `telnet`, and the common desktop environment (CDE) session.
- Identify and fix login problems relating to a `getty`, `telnet`, and CDE session.
- Explain the benefits of password shadowing


© 2005 Hewlett-Packard Development Company, L.P.

**Module 8 — Troubleshooting File Systems**

- State the purpose and describe the functionality of **`fsck`**.
- Determine what **`fsck`** is auditing and repairing during each pass of a file system (HFS/VxFS).
- Perform an **`fsck`** on a file system (HFS/VxFS).
- Perform a full **`fsck`** on a VxFS file system.
- Understand **`fsdb`** basics
- Describe the steps to defragment a VxFS file system

**Module 9 — Troubleshooting Process Problems**

- Define a process.
- Display a user's process hierarchy.
- Use the **`kill`** command.
- Identify and clear zombie processes.
- Troubleshoot CPU-intensive processes.

**Module 10 — Troubleshooting an HP-UX 11i v2 Kernel**

- Describe the major steps in recovering from a bad kernel
- Describe how to access **`kcweb`**.
- Describe how to view kernel changes via **`kcweb`** `and` **`kclog`**
- Describe other useful logfiles for troubleshooting kernel problems
- Boot from an alternate kernel
- Boot to tunable maintenance mode

**Module 11 — Troubleshooting a Kernel Crash Dump**

- Describe the purpose of kernel crash dumps
- Configure dump devices
- Configure selective crash dumps
- Configure compressed crash dumps


© 2005 Hewlett-Packard Development Company, L.P.

**Overview**

- Configure **`savecrash`** functionality
- Describe the crash dump directory structure
- Uncompress and view a crash dump via **`crashutil`** and **`q4`**.

**Module 12 — Troubleshooting Patch Problems**

- List four new patch management attributes introduced with HP-UX 11.00.
- List three additional patch management tools that can be added to an HP-UX 11.x system through a patch.
- Describe the procedure for committing a patch.
- Describe options when installing patches.
- Describe the use and purpose of **`security_check_patch`** tool.
- Describe the purpose of various SD-UX logfiles.
- Understand the purpose of the Installed Products Database (IPD).

**Module 13 — Troubleshooting Network Problems: Part 1**

- Relate the OSI 7-Layer Model to network addressing, display, and test command output.
- Identify and resolve LAN card configuration problems.
- Identify and resolve subnetting problems.
- Identify and resolve routing problems.
- Display and modify tunable network parameters with **`ndd`**.

**Module 14 — Troubleshooting Network Problems: Part II**

- Discuss NFS server setup issues.
- Troubleshoot a stale NFS file handle.
- Explain DNS server configuration files and setup issues.
- Understand the usage of **`nslookup`** and **`nsquery`**


© 2005 Hewlett-Packard Development Company, L.P.

## Student Profile and Prerequisites

The student should have taken *one* of the following system administration courses:

- *HP-UX System and Network Administration for Experienced UNIX® System Administrators* (H5875S)

or

- *HP-UX System and Network Administration I* (H3064S)

and

- *HP-UX System and Network Administration II* (H3065S)

- Knowledge of and experience administering an HP-UX system is required.

## Agenda

| | | |
|---|---|---|
| Day 1 | Module 1 | Troubleshooting Resources |
| | Module 2 | System Recovery with Ignite-UX |
| | Module 3 | Troubleshooting Non-Bootable LVM Disks and LVM Structures |
| Day 2 | Module 4 | Troubleshooting a PA-RISC System Boot Disk |
| | Module 5 | Booting HP-UX 11i v2 via the EFI |
| | Module 6 | Booting: Missing or Corrupt ASCII Startup Files |
| Day 3 | Module 7 | Troubleshooting Login Problems |
| | Module 8 | Troubleshooting File Systems |
| | Module 9 | Troubleshooting Process Problems |
| Day 4 | Module 10 | Troubleshooting an HP-UX 11i v2 Kernel * |
| | Module 11 | Troubleshooting a Kernel Crash Dump |
| | Module 12 | Troubleshooting Patch Problems |
| Day 5 | Module 13 | Troubleshooting Network Problems: Part I |
| | Module 14 | Troubleshooting Network Problems: Part II |

* Above agenda is suggested only, pending availability of Integrity systems for labs.


© 2005 Hewlett-Packard Development Company, L.P.

# Module 1 — Troubleshooting Resources

## Objectives

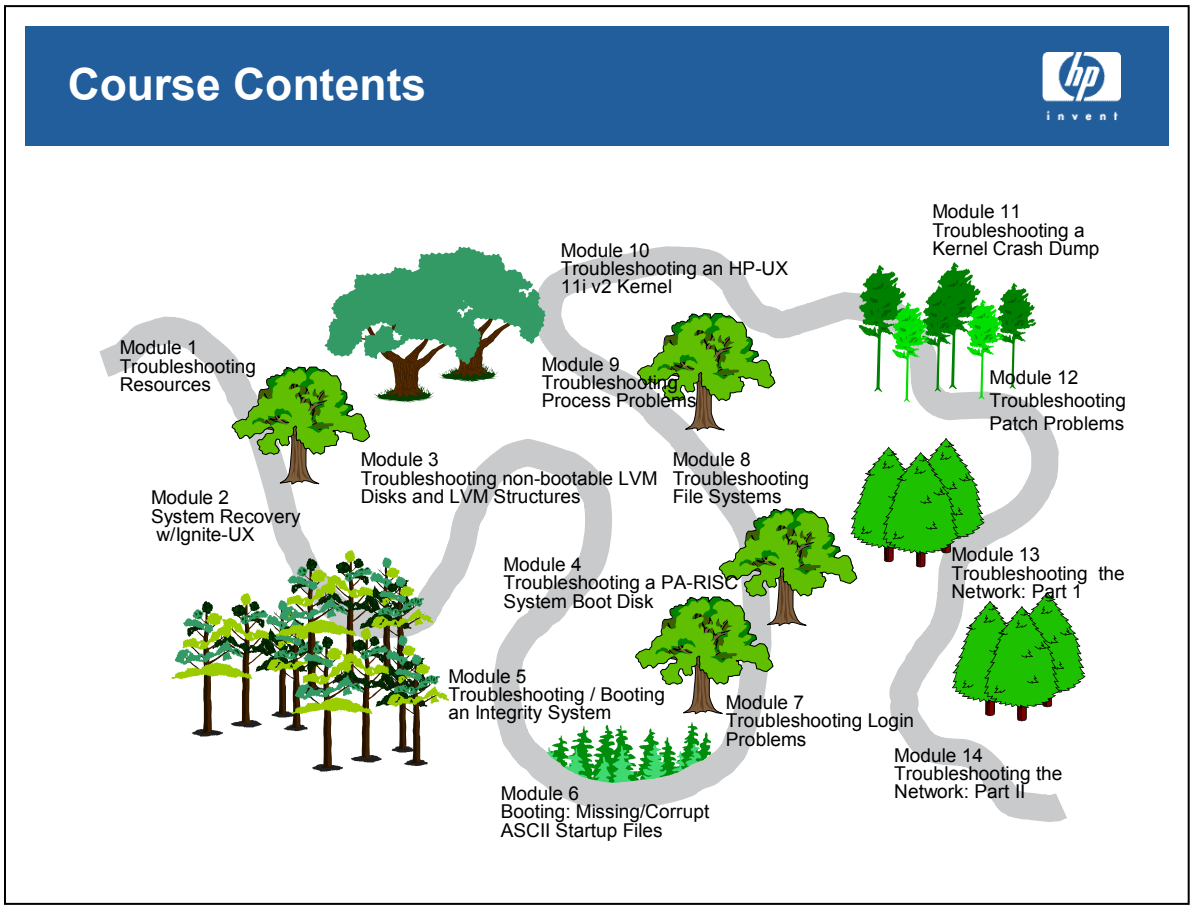
Upon completion of this module, you will be able to do the following:

- Describe the course contents and state the course goals.
- List four HP-UX troubleshooting resources.
- Create **`/usr/share/lib/whatis`** for use with the **`man -k`** keyword search.
- Correlate your troubleshooting technique to sample troubleshooting techniques.




© 2005 Hewlett-Packard Development Company, L.P.

## 1–1. SLIDE: Course Contents




## Student Notes

A network of computer systems is a little like a forest, with winding paths, many trees, and numerous places to get lost or get in trouble.

Troubleshooting is a system that involves knowledge, experience, and intuition. A course in troubleshooting, especially a short one, is clearly limited in the material it can cover. It cannot hope to solve all of the system administrator's problems or provide all the necessary tools, but it can provide an effective starting point.

This course discusses some proactive troubleshooting measures (such as record keeping), and some common HP-UX system administration trouble spots.

Bring your own problem solving experience to the lectures and lab exercises.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 1–2. SLIDE: Module Objectives

### Module Objectives

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

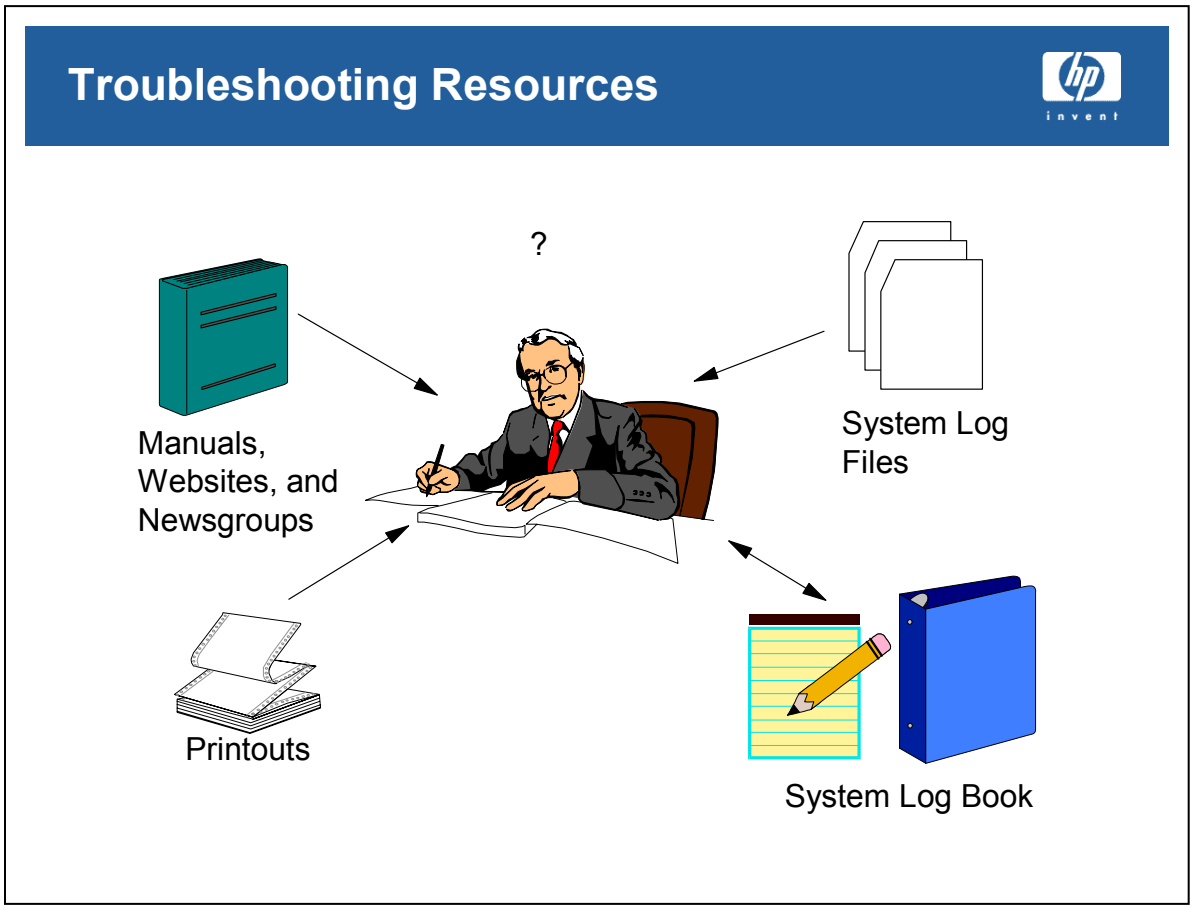
- Describe the course contents and state the course goals.
- List four HP-UX troubleshooting resources.
- Create `/usr/share/lib/whatis` for use with the `man -k` keyword search.
- Correlate your troubleshooting technique to sample troubleshooting techniques.



## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 1–3. SLIDE: Troubleshooting Resources




## Student Notes

As a system administrator, there are a number of resources to help troubleshoot your system.

| | |
|---|---|
| Manuals, websites, and newsgroups | Product documentation is available in hard copy as well as online. Most system administrators are already familiar with a number of these resources. Even if you are not in the habit of researching with manuals, keep in mind that information exists here. |
| Printouts | Some of your most valuable troubleshooting information may be in the form of printouts generated from your system and kept on file. Examples: printouts of customized configurations, hardware device paths, and others. |
| Activity records | Records that log or describe system configuration changes, system usage, and problem solutions, can provide valuable troubleshooting information. |
| System log files | Messages the system writes to various files can be another useful troubleshooting tool. |


**© 2005 Hewlett-Packard Development Company, L.P.**

## 1–4. SLIDE: Troubleshooting Philosophy

### Troubleshooting Philosophy

**Thoughts and preparation**

- GOAL: IDENTIFY and SOLVE the PROBLEM
- Avoid distractions.
- Gather as many symptoms as possible.
  - What works is just as important as what doesn't.
- Gather configuration information from testing and monitoring command output.
  - ASCII configuration files may have been edited but not applied.
- Has it EVER worked?
- Document your steps.
- Don't let fear impede progress.

## Student Notes

Your goal in troubleshooting is clearly to solve the problem. However, the journey to that end can either be a methodical and rewarding one –or- it can be a painful and frustrating process.

A number of people address problems by just "jumping in" and trying things. However, unless you've seen (and solved the problem) before, it might behoove you to instead gather information before acting. This can be accomplished via testing, looking at files, and/or talking to others that were present when the problem occurred.

One thing to remember: What IS working can be just as important as what is NOT. We'll discover this soon enough in this course.

Another key question to ask: Has it EVER worked?
If not, there will likely be a completely different set of steps to follow to get it working.

A simple problem may not require documenting your troubleshooting steps, but as problems become more and more complex, documenting your actions becomes ever more important. Picture a problem where you have performed 5 or 6 troubleshooting steps, the symptoms seemingly change a little with each step, but after the last step, the original problem seems to


© 2005 Hewlett-Packard Development Company, L.P.

have re-appeared. Can you re-trace your steps from memory? Having written them down sure makes backtracking easier.

Distractions come in many guises and can really foul the troubleshooter's path. An easy way to be led astray during troubleshooting is discovering other "could have been done better" things that have nothing to do with the problem under investigation. Learning at least to control, or at best avoid, distractions is a secret weapon of the most successful problem-solvers. It can mean the difference between hours and days in the troubleshooting loop.

It is also important to avoid being distracted by what appears to be associated facts, which in fact, have nothing to do with the problem. The initial problem report from the user can be very misleading. The following story is a good example.

### Car Allergic to Vanilla Ice Cream

The Pontiac Division of General Motors received a complaint:

"This is the second time I have written you, and I don't blame you for not answering me, because I kind of sounded crazy, but it is a fact that we have a tradition in our family of ice cream for dessert after dinner each night. But the kind of ice cream varies so, every night, after we have eaten, the whole family votes on which kind of ice cream we should have and I drive down to the store to get it. It is also a fact that I recently purchased a new Pontiac and since then, my trips to the store have created a problem. You see, every time I buy vanilla ice cream, when I start back from the store my car won't start. If I get any other kind of ice cream, the car starts just fine. I want you to know I'm serious about this question, no matter how silly it sounds: 'What is there about a Pontiac that makes it not start when I get vanilla ice cream, and easy to start whenever I get any other kind?'"

The Pontiac President was understandably skeptical about the letter, but sent an engineer to check it out anyway. The latter was surprised to be greeted by a successful, obviously well-educated man in a fine neighborhood. He had arranged to meet the man just after dinner time, so the two hopped into the car and drove to the ice cream store. It was vanilla ice cream that night and, sure enough, after they came back to the car, it wouldn't start.

The engineer returned for three more nights. The first night, the man got chocolate. The car started. The second night, he got strawberry. The car started. The third night he ordered vanilla. The car failed to start.

Now the engineer, being a logical man, refused to believe that this man's car was allergic to vanilla ice cream. He arranged, therefore, to continue his visits for as long as it took to solve the problem. And toward this end he began to take notes: he jotted down all sorts of data, time of day, type of gas used, time to drive back and forth, etc.

In a short time, he had a clue: the man took less time to buy vanilla than any other flavor. Why? The answer was in the layout of the store.


**© 2005 Hewlett-Packard Development Company, L.P.**

> Vanilla, being the most popular flavor, was in a separate case at the front of the store for quick pickup. All the other flavors were kept in the back of the store at a different counter where it took considerably longer to find the flavor and get checked out.
>
> Now the question for the engineer was why the car wouldn't start when it took less time. Once time became the problem -- not the vanilla ice cream -- the engineer quickly came up with the answer: vapor lock. It was happening every night, but the extra time taken to get the other flavors allowed the engine to cool down sufficiently to start. When the man got vanilla, the engine was still too hot for the vapor lock to dissipate.
>
> Moral of the story: even problems that appear ridiculous are sometimes real.

Finally, do not be afraid — of the system, the symptoms, the methods, or the problem. That may sound strange, but fear can cripple a troubleshooter. A clear head and a relaxed attitude start the path to problem solution on an upswing. If you get frustrated or confused along the way, walk away for a few minutes. It will do wonders to get a breath of fresh air or a fresh point of view. Then it is a deep breath, and back into the troubleshooting loop. You can do it — you and your resources.




© 2005 Hewlett-Packard Development Company, L.P.

## 1–5. SLIDE: Troubleshooting Techniques

### Troubleshooting Techniques

**Techniques**

- Top down
  - Gather application-specific symptoms first.
- Bottom up
  - Test and verify hardware first.
- Divide and conquer.

## Student Notes

In addition to “shooting from the hip,” there are three philosophies for troubleshooting — top-down, bottom-up, and "divide and conquer". In all three, troubleshooters are looking for two things — what works and what does not.

Using the top down method, a troubleshooter starts with a “known good” as far away from the operating system and hardware as possible. This is usually the application. The troubleshooter looks for clues in logs, error messages, and unusual application behaviors. The problem is then traced backwards or downwards toward application configurations, release, revision, or patch changes, and installation. If necessary, the troubleshooting continues down through the operating environment of the user, then on to operating system logs, error messages, unusual behaviors, configuration, release, revision, or patch changes, and finally to the hardware.

Bottom-up troubleshooters start at the lowest level, usually the hardware — even as far down as checking power. After all, if there is no power, nothing else matters, right?

"Divide and conquer", also known as "half-splitting" is another technique in which the troubleshooter knows the operational flow of the application and system. Testing starts in the middle and progresses either up or down, depending on test results.


© 2005 Hewlett-Packard Development Company, L.P.

Any way you choose, the desired results are the same:

- Find something that works adjacent to something that does not.
- Use all the tools, tricks, and expertise you can find.
- Gather as many symptoms as you can; what works is just as important as what does not.

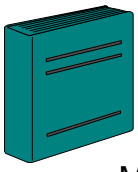



© 2005 Hewlett-Packard Development Company, L.P.

## 1–6. SLIDE: Manuals, Websites, and Newsgroups

### Manuals, Websites, and Newsgroups

- Manuals (Documentation)
  - Dynatext Instant Information CD
  - Manual Pages
  - Release Notes
- Websites
  - www.hp.com
  - www.itrc.hp.com
  - docs.hp.com
  - software.hp.com
  - hpux.cs.utah.edu
  - www.interex.org
  - www.cert.org
  - forums.itrc.hp.com
  - google.com



Manuals and Websites

Newsgroups
- comp.sys.hp.hpux
- comp.sys.hp
- comp.unix.admin
- comp.unix

## Student Notes

While the online (or hardcopy) manual pages can certainly assist with understanding command syntax and operation, we often need more information.

There are many websites, both HP and non-HP that can provide useful information.

For many administrators, the first stop is the I.T. Resource Center (http://www.itrc.hp.com). The ITRC is now the single "go-to" place to obtain information, get patches, find solutions, and even sign up for further education through Hewlett Packard. If you cannot find the information you seek, you can even open a call with the Hewlett-Packard Response Center and get assistance.

Another popular website: http://docs.hp.com is now the "go-to" place to get most kinds of documentation related to HP technology. While it is still possible to purchase hardcopies of many of HP's equipment or product manuals, a large number of these manuals are available for free online.

Other sources of information on HP equipment and products can be found on some non-HP websites. For example, the ITRC sponsors a forum for users of HP equipment, on which it is possible to find unique solutions to problems. Often, where the product in question is no


© 2005 Hewlett-Packard Development Company, L.P.

longer supported by HP, it is likely that somebody has developed a solution related to that product. HP encourages customers to visit this site, as long as the customer understands that HP is not offering these solutions as supported. Instead, our philosophy is: "Here's how another customer has done this. Perhaps it might also work for you."

**`http://www.cert.org`** and **`news:comp.sys.hp.hpux`** can also be valuable. Instead, the administrator may wish to enter their query into http://google.com. While Google may ultimately refer the administrator to one of these other sites, it's still a popular (and quite effective) site for finding just about anything. Remember though, since these are not HP sites, HP does not officially support them. While much of what is posted on the Internet is helpful, it is a public forum, and mistakes are made.

The release notes and white papers in **`/usr/share/doc`** are also useful sources of information about new features and changes in the current release of HP-UX. Files with a **`.ps`** extension are PostScript files. Files with a **`.txt`** extension are ASCII text files. A **`.Z`** extension signifies compressed files (use the **`uncompress`** command to expand them). Check the sizes of these files before you print them (for text files, you can use **`wc -l`** ***`filename`***, and for the PostScript files, you can use non-HP tools such as **`ghostview`** or **`ps2pdf`** and a **`pdf`** reader), as some are extremely long. Release notes are cumulative, meaning all revisions within the current release are included.

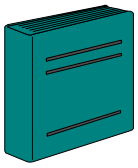



**© 2005 Hewlett-Packard Development Company, L.P.**

## 1–7. SLIDE: Man Page Sections and `catman -w`

### man Page Sections and `catman -w`

- `/usr/share/man/cat?.Z /usr/share/man/man?.Z`
  - Section directories of compressed man pages
- `/usr/share/man/man?`
  - Section directories of uncompressed, formatted man pages
- `/usr/share/lib/whatis`
  - ASCII index of man page one-line headlines
  - Create/update with `/usr/sbin/catman -w`
- Man page sections
  - 1 users' commands
  - 2 programmers' system calls
  - 3 programmers' language-specific subroutines
  - 1m system administrators' commands
  - 4 system administrators' file formats
  - 5 miscellaneous and overviews
  - 6 unused
  - 7 system administrators' special files (device, network, pipe)
  - 9 glossary



## Student Notes

Online documentation (the **man** pages) is among the most valuable, handiest tool a troubleshooter can use. Unfortunately, it is often bypassed because the administrator finds himself asking, " I don't know what command to use, so how can I use the man pages to find information about it?" That is where knowing what is in each section of the man pages, **catman -w** and **man -k** come in. They are powerful tools on which effective troubleshooters rely.

Before you can use **man**'s keyword search (**man -k**), you need an up-to-date library of **man** page headlines for the **man** utility to reference in the file **/usr/share/lib/whatis**. **/usr/sbin/catman -w** creates and updates **/usr/share/lib/whatis** with the headlines from all properly formatted files it finds in directories given in the superuser's **MANPATH** shell variable. (It must be exported.) Any time a product is added that has documentation the **man** command can use, **/usr/share/lib/whatis** should be updated with **/usr/sbin/catman -w**. Once created, all users, super and otherwise, can use **man**'s keyword search to find just the commands and information they need, by typing: **man -k *keyword***, where *keyword* is any word (case insensitive) contained in a **man** page headline. For example, **man -k shutdown** displays all man pages whose headlines include the string `shutdown` without case sensitivity. To get documentation from a specific section of the **man** pages, include the section number (as given on the slide) as the first argument to


**© 2005 Hewlett-Packard Development Company, L.P.**

the **man** command. For example, **man 1 passwd** displays documentation on the command used to change a user's password, but **man 4 passwd** displays documentation on the format of the **/etc/passwd** file. Man page sections are searched in the order given on the slide; (The admin commands section was originally section 8. HP renamed it to section 1m), so **man 1m shutdown** is required to see information on using the **shutdown** command. There also is a **shutdown** system call, documented in section two. For many administrative commands, there are corresponding system calls. It is good practice to type the section number.

Knowing what is in each section of the **man** pages, and using the keyword search function can make all the difference in using **man** as a key troubleshooting resource effectively.

Another issue with using the online man pages effectively is the maintenance of the MANPATH environment variable. Most software installed onto an HP-UX system will provide its own directory for its man page files, typically **/opt/<<APP_NAME>>/share/man**. However, in order for the **man** and **catman** commands to know about these directories, they need to be added to the **/etc/MANPATH** file. While Software Distributor provides a function for installation to do this, it is not always done. So, it is important for the administrator to check this file.

The format of the file with all entries on a single line can be a bit awkward to manage manually. You can maintain a second file with one entry per line, and then use a script to build the proper file.

```
#! /usr/bin/awk -f

function generatePATH(infile,outfile)
{
  start=1;

  while ( (stat = getline < infile) > 0 ) {
   if ( start ) {
      start=0;
   } else {
      printf(":") > outfile;
   }
   printf("%s", $0) > outfile;
  }

  if ( stat == -1 ) {
   printf( "the file %s is missing\n", infile );
   printf( "running %s initialize will create one\n",ARGV[0] );
  }

  close infile;
  close outfile;
}

function generateLOCAL(infile,outfile)
{
  getline < infile;

  ndirs=split($0,dirs,":");
```




© 2005 Hewlett-Packard Development Company, L.P.

```
  for (i=1;i<=ndirs;i++) {
   print dirs[i] > outfile;
  }

  close infile;
  close outfile;
}


BEGIN {

  if (( ARGC == 2 ) && ( ARGV[1] ~ /^i/ )) {
   ### initializing the .local files from the current path
   ### files
   generateLOCAL("/etc/PATH","/etc/PATH.local");
   generateLOCAL("/etc/MANPATH","/etc/MANPATH.local");
   generateLOCAL("/etc/SHLIB_PATH","/etc/SHLIB_PATH.local");
  } else {
   generatePATH("/etc/PATH.local","/etc/PATH");
   generatePATH("/etc/MANPATH.local","/etc/MANPATH");
   generatePATH("/etc/SHLIB_PATH.local","/etc/SHLIB_PATH");
  }

  exit 0;
}
```

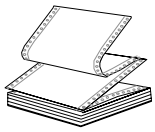

**© 2005 Hewlett-Packard Development Company, L.P.**

## 1–8. SLIDE: Printouts (Output from Commands) 1 of 2

### Printouts (Output from Commands) 1 of 2

Keep a hard copy of information displayed by commands such as the following:

- `ioscan -fn`
- `vgdisplay -v`
- `pvdisplay -v` *$each_disk*
- `lvdisplay -v` *$each_logical_volume*
- `lvlnboot -v`
- `kmtune (PA-RISC) / kctune (Integrity)`
- `lanscan`
- `ifconfig lan#`
- `hostname`
- `setboot`



Printouts



## Student Notes

Record keeping can be a valuable, proactive troubleshooting measure. The system administrator may want to keep the following information on file:

**`ioscan`** Scans system hardware, usable I/O system devices, or kernel I/O system data structures as appropriate, and lists the results.

By default, **`ioscan`** displays the hardware path to the hardware module, the class of the device driver, and a brief description for each hardware module on the system.

**`-n`** Lists device file names in the output. Special files are listed only in the **`/dev`** directory. The subdirectories are also listed.

**`-f`** Generates a full listing, displaying the driver's class, instance number, hardware path, driver, software state, hardware type, and a brief description. It also causes the system to scan right down to the hardware level, so it can be slow. Adding the “-u” option gets **`ioscan`** to just recall this information from memory.


© 2005 Hewlett-Packard Development Company, L.P.

**`vgdisplay -v`** Displays information about logical volume manager (LVM) volume groups. The **`vgdisplay`** command displays information about volume groups. For each **`vg_name`** specified, **`vgdisplay`** displays information for that volume group only. If no **`vg_name`** is specified, **`vgdisplay`** displays names and corresponding information for all defined volume groups.

The volume group must be activated (see **`vgchange(1M)`**) before it can be displayed.

**`-v`** Displays additional information about each logical volume, physical volume, and physical volume group.

**`pvdisplay`** Displays information about physical volumes within an LVM volume group. The **`pvdisplay`** command displays information about each physical volume specified by a **`pv_path`** parameter.

**`-v`** Displays the logical volumes that have extents allocated on the physical volume and the usage of all the physical extents.

**Example**

Displays the status, characteristics and allocation map of a physical volume: **`pvdisplay -v /dev/dsk/c2t0d0`**

For most systems to keep a printout of the **`pvdisplay`** information would be impractical, as it would be too large. The section that gives the distribution of logical volumes and a summary or pictorial representation of the PV ⇔ LV mapping is a better choice.

**`lvdisplay`** Displays information about LVM logical **`volumes`**. The **`lvdisplay`** command displays the characteristics and status of each logical volume specified by **`lv_path`**.

**`-v`** Displays the physical volume distribution, and the mapping of the logical extents onto the physical extents of the physical volumes for each logical volume.

As with the output of **`pvdisplay`**, keeping a printout of the **`lvdisplay -v`** command is not practical.


**© 2005 Hewlett-Packard Development Company, L.P.**

**`lvlnboot`**

The lvlnboot command manages the BDRA and LABEL files on bootable LVM disks. The BDRA lists the disks within the root volume group and also details the root, primary swap and dump volumes. The LABEL file has an emergency cut down version of some of this information and also the details of the boot logical volume.

**`-v`** Prints verbose messages. With no other arguments present, it prints information on root, boot, swap, and dump logical volumes. If a combined root-boot volume is configured, no information for the boot volume is displayed.

**`kmtune`** | **`kctune`**

The **`kmtune`** command can be used to report and modify the values of kernel parameters. With no arguments, it reports the current value of all kernel parameters, indicates whether they can be changed online, and what the planned (next reboot) value will be.
The **`kctune`** command has replaced **`kmtune`** on HP's Integrity servers.

**`lanscan`**

Displays LAN device configuration and status. **`lanscan`** displays the following information about each LAN device with software that is supported on the system:

- Hardware path
- Active station address (also known as Physical address)
- Card instance number
- Hardware state

Network Interface "NamePPA" — The Network Interface "Name" and the physical point of attachment (PPA) number are concatenated. A single hardware device may have multiple NamePPA identifiers, which indicates that multiple encapsulation methods can be supported on the device. For Ethernet/IEEE 802.3 links, the Name LAN is used to designate Ethernet encapsulation, and snap for IEEE 802.3 encapsulation. For other links, such as (FDDI and token ring), only the LAN encapsulation designation is used.

- Network management ID
- MAC type
- HP DLPI supported. — Indicates whether the LAN device driver will work with HP's common data link provider interface.
- DLPI major number
- Extended station address for those interfaces that require more than 48 bits. This is displayed only when the **`-v`** option is selected.
- Encapsulation methods the network interface supports. This is displayed only when the **`-v`** option is selected.
- The driver name for the LAN card. This is displayed only when the **`-v`** option is selected.




© 2005 Hewlett-Packard Development Company, L.P.

**`ifconfig`** The **`ifconfig`** command displays information about IP address configuration and software status of the specified network interface, including status flags, IP address, subnet mask, and broadcast address. **`netstat -in`** can be used to display the address and network number of all LAN interfaces.

**`hostname`** Sets or displays the name of the current host system. The **`hostname`** command displays the name of the current host, as given in the **`gethostname()`** system call (see **`gethostname(2)`**). Users who have appropriate privileges can set the hostname by giving the argument **`name_of_host`**; this is usually done in the startup script **`/sbin/init.d/hostname`**. The **`name_of_host`** argument is restricted to MAXHOSTNAMELEN characters as defined in **`/usr/include/sys/param.h`**, and should not be set when using a windowing GUI, as doing so could cause the window session to hang.

**`setboot`** Sets or displays the hardware addresses of the primary and alternate boot paths, and the states (ON/OFF) of **`Autoboot`** and **`Autosearch.`**

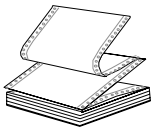

**© 2005 Hewlett-Packard Development Company, L.P.**

## 1–9. SLIDE: Printouts (Output from Commands) 2 of 2

### Printouts (Output from Commands) 2 of 2

Keep a hard copy of information displayed by commands such as the following:

- `lsdev`
- `lpstat -t`
- `bdf`
- `print_manifest`
- `mount -v`
- `swapinfo -tm`
- `uname -a`
- `model`
- `netstat -i`
- `netstat -rn`
- `swlist`



Printouts



## Student Notes

**`lsdev`** Displays a list of the currently available device drivers with major numbers, driver name, and class

**`lpstat -t`** Displays the status of the line printer scheduler, system default printer, queued print jobs, queue and printer status, and fence value.

**`bdf`** Displays statistics about currently mounted file systems, including device file, total used, and available data space, percentage of user-available space used, and mount point directory.

**`print_manifest`** This utility is one of the many commands available under the Ignite-UX suite. **`print_manifest`** will display many parameters on your system, all in one output. For example, the output will show networking parameters, software installed, hostname, OS revision, and much more.

**`mount -v`** While having a hardcopy of the **`/etc/fstab`** file is also useful, capturing a printout of **`mount -v`** will show a little more information.


© 2005 Hewlett-Packard Development Company, L.P.

| | |
|---|---|
| **`swapinfo -tm`** | Shows current swap utilization and swap devices. While the output of this command is dynamic, capturing a list of swap devices may prove to be valuable. |
| **`uname -a`** | Displays the system hostname, OS revision and Software Identifier |
| **`model`** | Displays the system model |
| **`netstat -i[n]`** | Displays the name, MTU, IP networks, IP address, inbound and outbound packet counts, for all configured LAN cards. |
| **`netstat -r[n]`** | Displays network routing information |
| **`swlist`** | Displays software bundles installed on the system. If a products list is desired, execute as follows: **`swlist -l product`** |

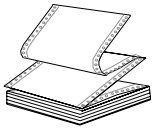

**© 2005 Hewlett-Packard Development Company, L.P.**

## 1–10. SLIDE: Printouts (Configuration Files)

### Printouts (Configuration Files)

Keep a hardcopy of output from the following configuration files:

- `/etc/inittab`
- `/etc/fstab`
- `/var/adm/sbtab`
- `/etc/hosts`
- `/etc/nsswitch.conf`
- `/etc/rc.config.d/netconf`
- `/etc/inetd.conf` and `/var/adm/inetd.sec`
- `/etc/rc.config.d/namesvrs`
- `/etc/resolv.conf`
- `/etc/lvmtab`
- `/stand/system`
- Application configuration files



Printouts



## Student Notes

The configuration files listed on the slide may be useful as a troubleshooting reference.

**inittab**

Script for the run-level **init** process. The **/etc/inittab** file supplies the script to the **init** daemon in its role as a general process dispatcher. (See **init(1M)**.) The process that constitutes the majority of boot **init**'s process dispatching activities is the line process, **/sbin/rc,** which references the symbolic links in the run-level directories to run/halt scripts in **/sbin/init.d**. Entries in **inittab** are position-dependent and have the following format:

```
id:rstate:action:process
```

**fstab**

Displays static information about the file systems. **fstab** is an ASCII file that resides in the **/etc** directory. Programs read it, but do not write to or from it. System administrators are responsible for creating and maintaining this file properly. **/etc/fstab** contains a list of mountable file system entries, secondary swap spaces, and secondary dump spaces. Each entry appears on a separate line, and consists of fields, separated


© 2005 Hewlett-Packard Development Company, L.P.

by one or more white spaces. Syntax errors can cause system hangs at boot time.

**`sbtab`** When an HFS file system is created, the addresses of the redundant superblock copies are appended to this file. These addresses can be used as arguments to **`fsck -b`** to check and repair the HFS file system when the primary superblock is corrupted or inaccessible.

**`hosts`** Hostname database. The file **`/etc/hosts`** associates Internet (IP) addresses with official host names and aliases. It allows a user to refer to a host by a symbolic name instead of an Internet address.
The usefulness of this file depends on how your site resolves hostnames. If you do use the host file on a large network, then this file can be very large. If so, a printout would be wasteful (you wouldn't want to re-type it would you?) Instead, we would rely on recovering the file from backup.

**`nsswitch.conf`** This file controls name services on the system. (The translation between names and numbers for such things as hostnames and IP addresses, service names and port numbers, password information, etc.)

**`netconf`** At boot time, this file is read to initialize the hostname local networking addresses, routing, etc.

**`inetd.conf`** Configuration file for **`inetd`** on invocation. The **`inetd`** daemon reads its configuration information from the **`/etc/inetd.conf`** configuration file, and possibly at some later time in response to a SIGHUP signal (normally by issuing the **`inetd -c`** command on HP-UX) (see **`inetd(1M)`**). Each line in the file is treated either as a comment or as configuration information for a given service. Comments are denoted by a **#** at the beginning of a line. Uncommented lines contain seven or nine required fields, depending on the service name specified in the first field. Tabs or spaces separate fields. A line can be continued if it terminates with a \. Each configuration line in the file contains the following fields in the order indicated:

- Service name
- Socket type
- Protocol
- Wait/nowait
- User
- Server program
- Program number (NFS, RPC services only)
- Version number (NFS, RPC services only)
- Server program arguments

**`inetd.sec`** Configures specific network services to be accessible to specific clients or groups of clients.

**`namesvrs`** At boot time, this file is used to initialize and configure DNS and NIS.


**© 2005 Hewlett-Packard Development Company, L.P.**

**`resolv.conf`** This file controls DNS domain, search, and server preferences.

**`lvmtab`** This is a binary data file, so it should not be printed directly. It holds the LVM configuration information. To print: **`strings /etc/lvmtab | lp`** . The **`strings`** command extracts strings of 5 or more "printable characters" from the file. It is quite possible that as well as getting the names of the volume groups and the physical volumes, the odd bit of garbage may also be output. This can probably be filtered by **`grep`**ing for **`/dev`**.

**`system`** This file holds the system-specific kernel tunable parameters. Under HP-UX 11i, the changes to kernel parameters are stored in the "Kernel Registry Service." The KRS maintains data in files under **`/stand/krs`**. These files, however, are binary files, so cannot be printed directly. Unlike the **`lvmtab`** file, these files cannot be dumped using strings. Instead, **`kmtune`** can be used to display the current and planned values for kernel tunables. If the kernel is rebuilt from the system file, any changes stored in the KRS will be lost.




© 2005 Hewlett-Packard Development Company, L.P.

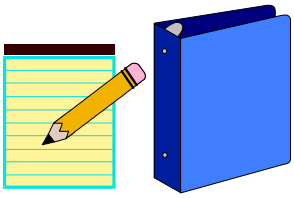
## 1–11. SLIDE: System Log Book

### System Log Book

A system log book should contain information specific to how the system has been managed in the past.

Examples of information contained in the book include:

- History of *problems* and their documented *solutions*
- Documentation of all system configuration *changes*
- Printouts of system configuration information (commands and configuration files)
- Index



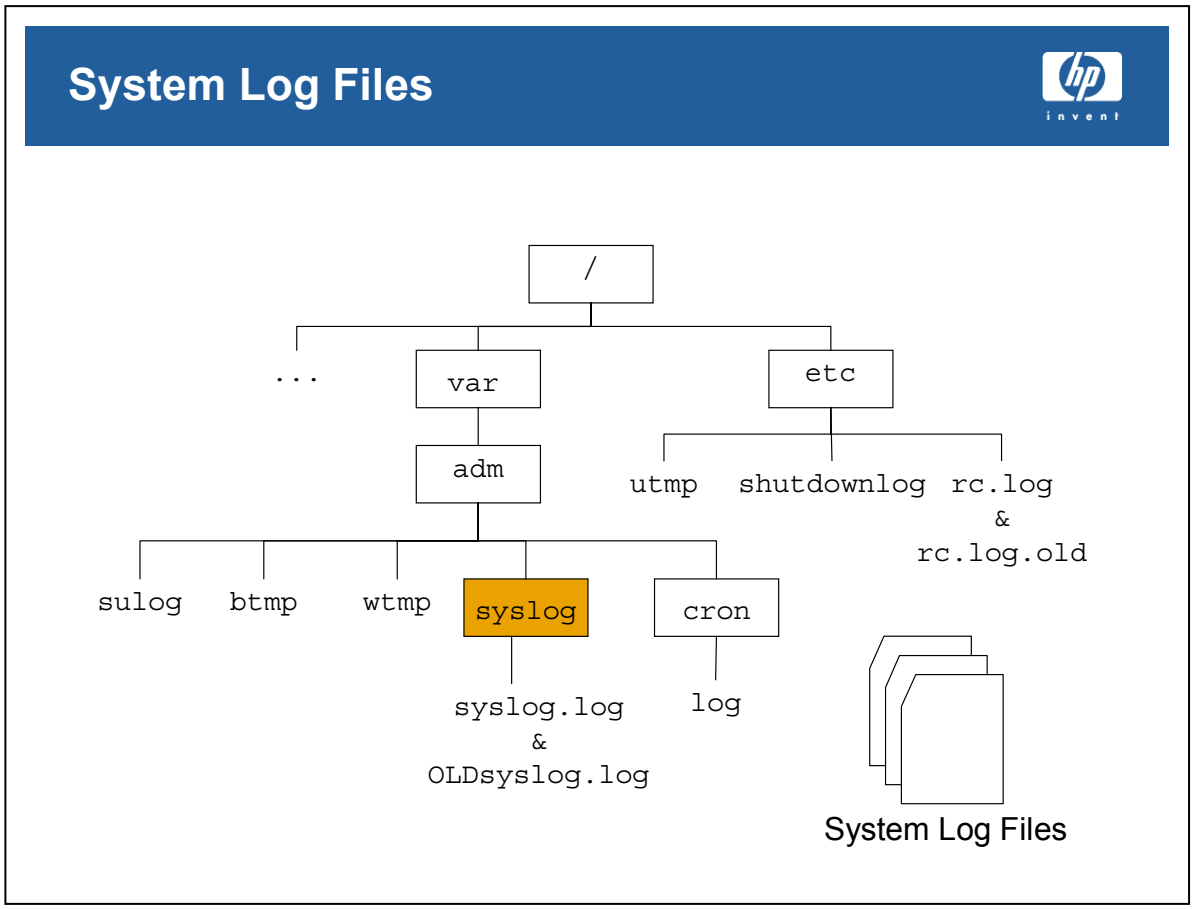
System Log Book

## Student Notes

A system log book can be an important reference regarding past problems and how they were resolved.


© 2005 Hewlett-Packard Development Company, L.P.

## 1–12. SLIDE: System Log Files




## Student Notes

Knowing the names and locations, and understanding the contents of system log files can prove critical to troubleshooting success. System administrators should view system log files periodically in order to gain familiarity with them and recognize abnormalities.

There are utilities that can be used to scan or monitor some of these log files to draw your attention to more urgent matters.

**utmp** — The **utmp** file is not strictly a log file. Its main job is to list the users currently logged onto the system. It also details the last change to the system clock and logs the actions of **init**.

**wtmp** — The log of users logging in and out of the system, activity of init, changes in run level and changes to the system clock.

**btmp** — The log of bad login attempts. For security reasons it is VITAL that this file is not readable by normal users.

**shutdownlog** — A log of the system shutdowns, written to by the **reboot** command during


© 2005 Hewlett-Packard Development Company, L.P.

normal shutdowns (shutdown calls reboot), or by the **savecrash** routines after a system failure.

**rc.log** A log of the activities of "**rc**" since the system was booted. This includes the output of the start scripts and any **kill** scripts that may have been run if the run level has been lowered.

**rc.log.old** The **rc.log** from the previous time the system was booted.
If you need to examine what is happening at shutdown time, this is usually where you have to look.

*NOTE:* The old **rc.log.old** is lost by default. If you wished to change this behavior to keep these logs for longer periods, then either a change to the **/sbin/rc.utils** script is needed, or a new startup script can be added, which appended the current **rc.log.old** to another file, preferably not in the **/etc** directory.

---

**syslog.log** Many parts of HP-UX use the **syslogd** utility to log status, log, and diagnostic messages. The standard HP-UX configuration for **syslogd**, is to place all messages related to email into the file **/var/adm/syslog/mail.log**, and all other messages into the file **/var/adm/syslog/syslog.log**.

From a troubleshooting perspective, there are two issues with the **syslog.log** file on HP-UX:

- The file is a textual log file. UNIX, in common with nearly all operating systems, can make a file smaller only by deleting the end, not the beginning. Log files are appended to the end, so there is no easy way to control the size of the file. HP-UX does not attempt to limit the file's size while the system is running. At boot time, the file is renamed to **OLDsyslog.log**.

  It is therefore useful to have a **cron** job that regularly cleans out the file. However, care must be taken, because the file is always open while **syslogd** is running. A script such as the following can be run from a **crontab** entry just after midnight each night. This will then keep a week's worth of log messages.

```
#!/usr/bin/sh

cd /var/adm/syslog
# Get yesterday's name, since we want to run at just
# after midnight
set -A DAYS ERROR Sun Mon Tue Wed Thu Fri Sat
day=${DAYS[$(date +%u)]}
mv -f syslog.log syslog.$day

kill -HUP $(< /var/run/syslog.pid )
echo "syslog.log mv'd to syslog.$day" | \
    logger -t "trimmer" -p local1.info
```


**© 2005 Hewlett-Packard Development Company, L.P.**

If older logs are needed, they can be loaded from the backups.

The **cron** entry might look like:

```
5 0 * * *        /usr/contrib/sbin/cleanSyslog
```

The system startup script **/sbin/init.d/syslogd** would then need to be modified so that it no longer moves (**mv**) the **syslog.log** file to **OLDsyslog.log**.

- The second problem with the default management of the **syslog** file concerns the renaming of **syslog.log** to **OLDsyslog.log**. It only maintains a one deep history. The old **OLDsyslog.log** information is lost. If the machine is rebooted twice, the logs are lost.




© 2005 Hewlett-Packard Development Company, L.P.

## 1–13. SLIDE: Module Objectives

### Module Objectives

Your should now be able to do the following:

- Describe the course contents and state the course goals.
- List four HP-UX troubleshooting resources.
- Create `/usr/share/lib/whatis` for use with the `man -k` keyword search.
- Correlate your troubleshooting technique to sample troubleshooting techniques.

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 1–14. QUIZ: Review Questions

### Directions

Answer the following questions. These questions serve as a review for this module.

1. List four HP-UX troubleshooting resources.

2. Which troubleshooting resource do you value more than others? Why?




© 2005 Hewlett-Packard Development Company, L.P.

# 1–15. LAB: System Log Files

## Purpose

This lab is designed to give you experience with logfiles.

## Directions

Read and answer the following questions, which serve as a review of the module on "Your Troubleshooting Resources."

1. Normally, HP-UX's **`inetd`** does not log connections to **`/var/adm/syslog/syslog.log`**. Executing **`inetd -l`** enables the logging of network connections through **`syslogd`**. If this functionality is needed all the time, then it can be enabled at boot time by setting the option

   **`export INETD_ARGS="-l"`** *(the letter "ell")*

   in the file **`/etc/rc.config.d/netdaemons`**.

   First, examine the **`/var/adm/syslog/syslog.log`** file. Note the kinds of information it contains:

   ```
   # vi /var/adm/syslog/syslog.log
   ```

   | | |
   |---|---|
   | *NOTE:* | We really do not want to accidentally write to the system log file via vi. vi makes a copy of the file, edits it, and then writes back to the original. Any messages logged while in vi would be lost if you saved the file. |

   a. How much physical memory does your system contain?

   b. Is the **`ftp`** service available?

   c. Is the **`rlogin`** service available?

   d. Is the **`telnet`** service available?

   e. Is the **`remsh`** service available?

   f. Are there any **`telnet`** requests from other machines?


© 2005 Hewlett-Packard Development Company, L.P.

2. Look at the **inetd** process using the standard **ps** command:

   ```
   # ps -ef | grep inetd
   ```

   Turn on more extensive logging.

   Properly kill the **inetd** process:

   ```
   # inetd -k
   ```



3. Restart **inetd**, but this time with logging enabled, by using the letter "*ell*" command-line option (not the number "*one*" ):

   ```
   # inetd -l
   ```

   | | |
   |---|---|
   | *NOTE:* | You can enable logging for **inetd** simply by executing **inetd -l**, but at the next shutdown, logging will be lost. Later in this lab, we will enable logging permanently. |

   Repeat the same **ps** command you used earlier in this same lab question:

   ```
   # ps -ef | grep inetd
   ```

4. After obtaining permission, **telnet** to a system near you. Login in as root; do a simple **ll** command; and then log out. After the person near you has **telnet**ed to your machine and logged out, re-examine **syslog.log**:

   ```
   # cd /var/adm/syslog
   ```

   ```
   # vi syslog.log
   ```

   Do you find the **telnet** entries for the person sitting near you?

   Turn on the logging capability so that at the next reboot, **inetd** will restart automatically with logging. First, change to the subdirectory where most of the startup configuration information is kept:

   ```
   # cd /etc/rc.config.d
   ```


© 2005 Hewlett-Packard Development Company, L.P.

In this subdirectory, **vi** the **netdaemons** file. Notice that the comments near the top of the file instruct us to add the **-l** ("ell") option to turn on logging at bootup. Using **vi**, add this option:

```
# vi netdaemons
```

5. Properly shutdown and reboot your machine:

   ```
   # shutdown -ry 0
   ```

   When the machine has completely rebooted, look again at the **inetd** process using the standard **ps** command:

   ```
   # ps -ef | grep inetd
   ```

6. **telnet** a second time to a person's machine sitting near you. Login as root, do a simple **ll** command, then log out. After the person near you has **telnet**ed to your machine and logged out, re-examine **syslog.log** file:

   ```
   /var/adm/syslog/syslog.log
   ```

   Do you find the **telnet** entries for the person sitting near you?

7. Set up your machine as an anonymous **ftp** site. The easiest way to do this is with SAM. First, access SAM, and then follow the instructions to get to SAM's correct submenu:

   ```
   # sam

   From the top level of SAM, double-click on "Networking and Communications"
   From the second level of SAM, double-click on "Network Services"
   From the third level of SAM, SINGLE-click on "Anonymous FTP", and then
                                SINGLE-click on "Actions", and then
                                SINGLE-click on "Enable".
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

8. Notice that a **/home/ftp** subdirectory now exists. Copy just *one* of the following files into the **pub** subdirectory below **/home/ftp**:

   **/etc/passwd** **/etc/hosts**
   **/etc/group** **/etc/inetd.conf**
   **/var/adm/syslog/syslog.log** **/etc/rc.config.d/netdaemons**

   Be sure that the file you copy has permissions so that it can be read by anyone.

   ```
   # chmod 644 /home/ftp/pub/< filename >
   ```

   Now enable logging within the **ftp** daemon itself.

   ```
   vi /etc/inetd.conf
   ```

   Find the line for the **ftp** daemon and change the option **-l** to **-L**.

   ```
   ftp      stream tcp nowait root /usr/lbin/ftpd     ftpd -L
   ```

   Save the file, and restart **inetd**.

   ```
   inetd -c
   ```

9. Ask a person sitting near you to use anonymous **ftp** to grab a copy of the file you placed in the **pub** directory. That person should do the following:

   a. **ftp** to your machine.

   b. At the login prompt, login as **anonymous** or **ftp**.

   c. At the **passwd** prompt, enter your complete email address.

      From the **ftp>** prompt, **cd** into the **pub** directory:

      ```
      ftp> cd pub
      ```

      In the **pub** directory, execute the **ftp "dir"** command to see the files currently in the **pub** directory:

      ```
      ftp> dir
      ```

      While still in the **pub** directory, **get** the file the other person previously placed there:

      ```
      ftp> get < file_name >
      ```

      And then exit **ftp**:

      **ftp> bye**




© 2005 Hewlett-Packard Development Company, L.P.

10. Finally, **`telnet`** a second time to a system near you. Login as root, do a simple **`ll`** command, and then log out. After the person near you has **`telnet`**ed to your machine and logged out, re-examine **`syslog.log`** file:

    ```
    # vi /var/adm/syslog/syslog.log
    ```

    Do you find the **`telnet`** entries for the person sitting near you? Did you notice that, in addition to showing the anonymous **`ftp`** login, **`syslog.log`** also shows you what FTP operations the person did while there!


**© 2005 Hewlett-Packard Development Company, L.P.**

# 1–16. LAB: MANPATH Issue

## Purpose

This lab will investigate the MANPATH variable.

1. Change to the **`/home/labs`** directory.

2. Run the script: **`./mod1.lab1.sh`**

3. Then log out, and log back in, and try the following command:

   ```
   # man make_net_recovery
   ```

4. What happened? What do you need to do to fix this?




© 2005 Hewlett-Packard Development Company, L.P.

**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 2 — System Recovery with Ignite-UX

## Objectives

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

- Understand system recovery choices
- Describe the layout of a system recovery tape
- Describe how to create a system recovery tape
- Understand system recovery options
- Describe how to verify a system recovery tape
- Describe how to boot from a system recovery tape
- Understand how to create a **`make_net_recovery`** archive




© 2005 Hewlett-Packard Development Company, L.P.

## 2–1. SLIDE: Module Objectives

### Module Objectives

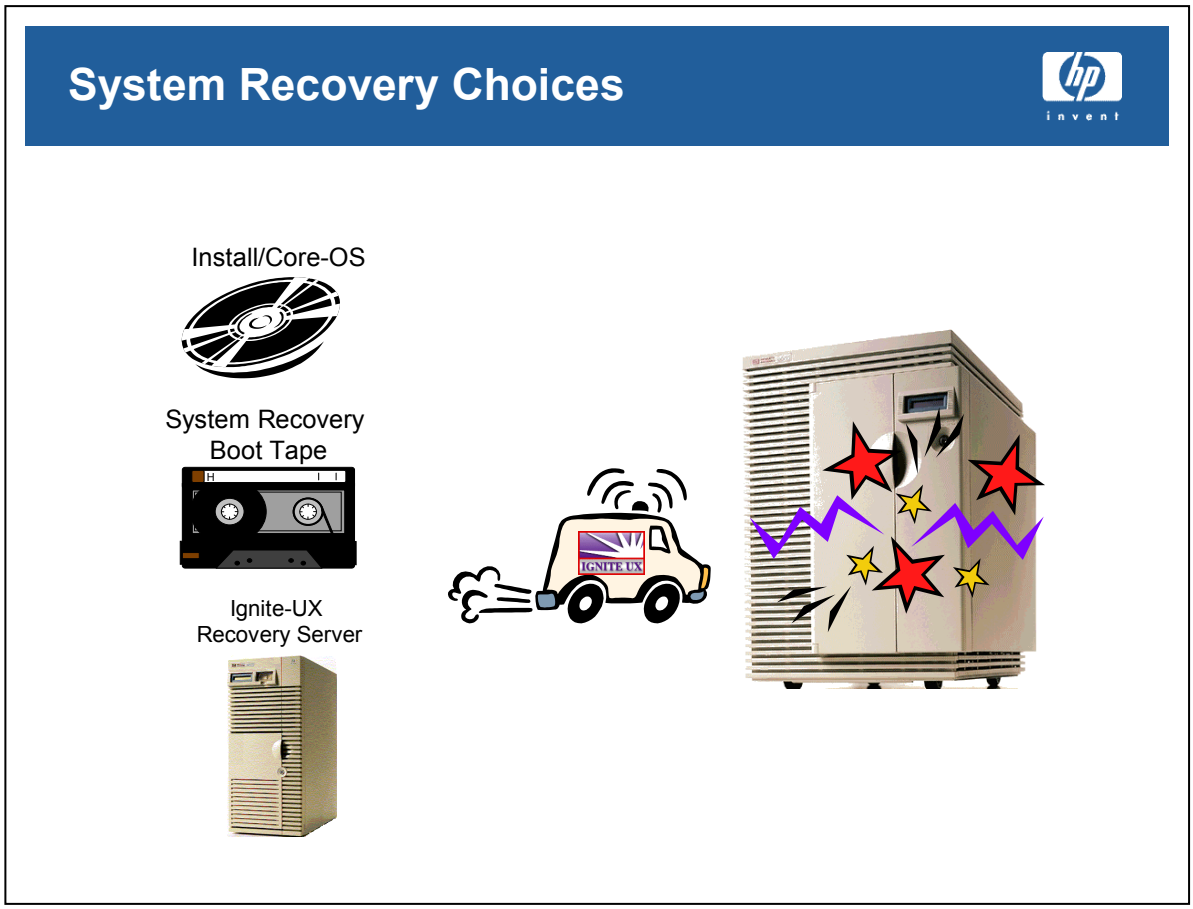
Upon completion of this module, you will be able to do the following:

- Understand system recovery choices
- Describe the layout of a system recovery tape
- Describe how to create a system recovery tape
- Understand system recovery options
- Describe how to verify a system recovery tape
- Describe how to boot from a system recovery tape
- Understand how to create a `make_net_recovery` archive

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 2–2. SLIDE: System Recovery Choices




## Student Notes

Today, you have several choices for recovery from a failed system, as a result of a disk failure. Your choices include:

- Hot Pluggable disks
- Disk Arrays (Auto-Raid)
- MC/ServiceGuard High Availability Cluster
- Installation of a minimum system + Restore from your backups
- System Recovery Boot Tape + Restore from your backups
- Mirror Disk/UX (LVM mirroring) + Spare

All of the choices listed above have their own strengths and weaknesses, as well as costs. Your choice of recovery will depend upon a number of factors:

- How well documented are your configurations?
  - hardware
  - software
- What does your disk infrastructure support for recovery?




© 2005 Hewlett-Packard Development Company, L.P.

- How reliable are your backups?
- How up-to-date are your system backups?
- Can you re-create your environment, completely?
- How long can you afford to have your system(s) out of service?

Careful planning and testing is required to successfully restore any system after a failure.

## Creation of a System Recovery Boot Tape

In this section we'll focus on the **`make_tape_recovery`** tool for system recovery. The **`make_tape_recovery`** command is used to create the **`system_recovery_boot_tape`**.

| | |
|---|---|
| *NOTE:* | **`make_tape_recovery`** is just one part of a complete disaster recovery solution. |

## Uses for `make_tape_recovery`

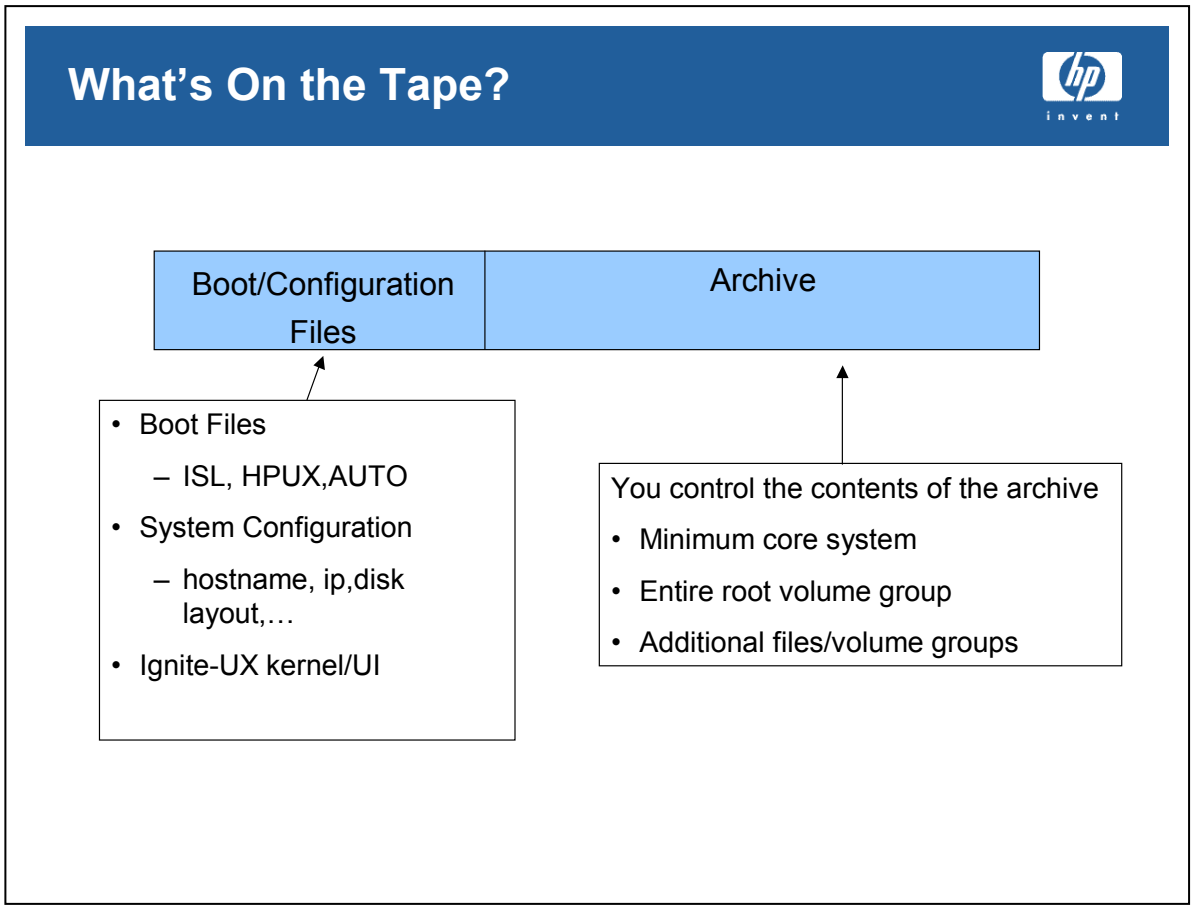
The **`make_tape_recovery`** tool will perform an Ignite-UX installation of your system. Data that is stored in the archive (created by **`make_tape_recovery)`** can be used for the following:

- Restore a non-bootable system with little or no intervention.
- Restore a system in the event of a root disk (volume group) failure.
- Convert file systems in the root volume from hfs to **`vxfs`** (logical volumes).
- Modify the root file system size (logical volume).
- Modify the size of primary swap (logical volume).
- Clone software from one system to another (requires nearly identical hardware).

The option to interact with the install process will be covered in more detail later in this section.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 2–3. SLIDE: What's On the Tape?




## Student Notes

**`make_tape_recovery`** is a command provided by Ignite-UX. Ignite-UX is available on the HP Application Media or from the web site **`//http:www.software.hp.com/products/IUX`** .

**`make_tape_recovery`** will create a bootable, system recovery tape. The **`make_tape_recovery`** tape is actually an Ignite-UX installation tape that will restore your system to a configured state automatically when used as the boot tape.

The contents of the system recovery tape are as follows:

- Boot Image (HP-UX LIF)
- System Configuration information
- Configuration files that control Ignite processing
- An Ignite-UX kernel and User Interface
- Archive

The contents of the archive can be customized. You can control how much, or how little actually goes onto the tape. You can choose:




© 2005 Hewlett-Packard Development Company, L.P.

- Minimum Core-OS
- Entire root volume group
- Additional files from both root and non-root volume groups

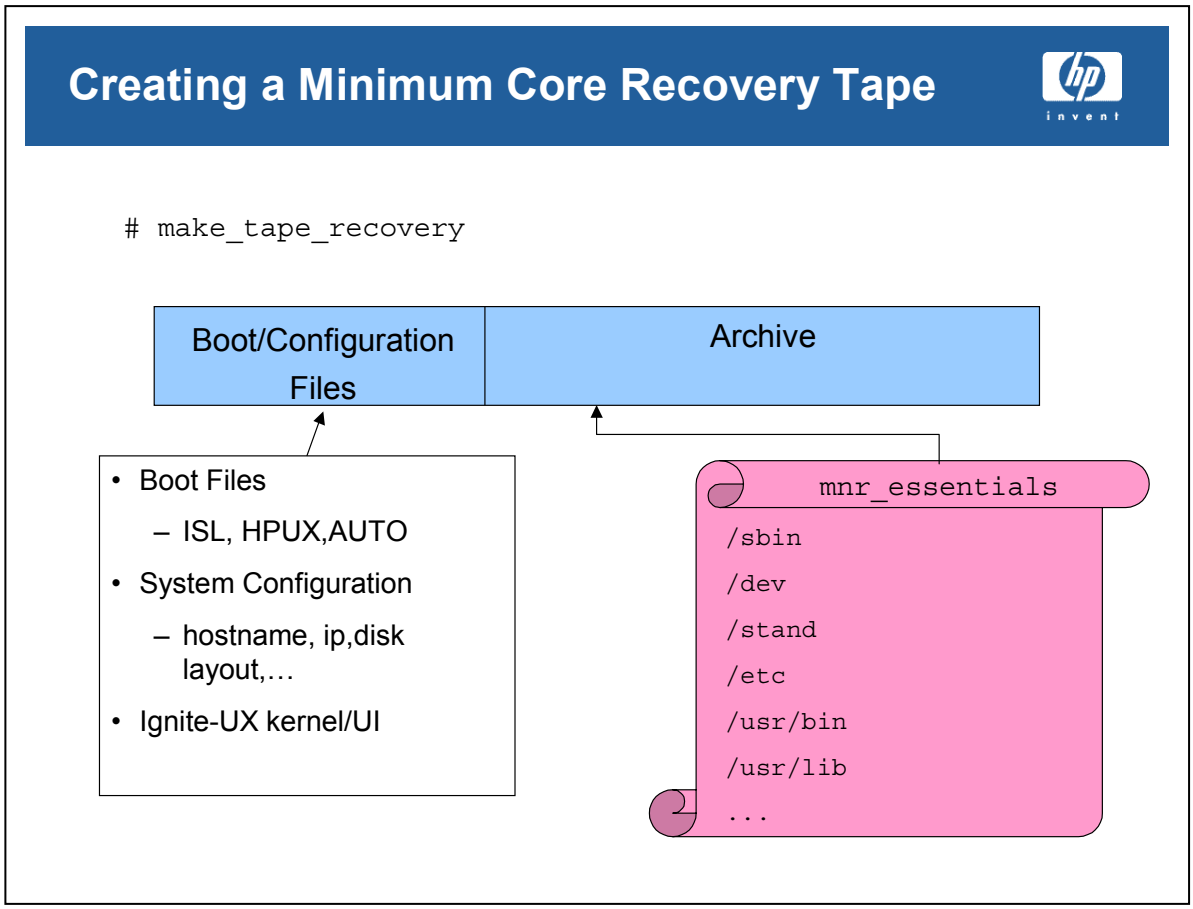
The type of recovery tape that you create will depend upon what other means you have to recover your system. For example, if you have a very robust 3rd party backup and restore solution, such as HP OmniBack, you may opt for a minimum image and restore the rest of your data and configurations using OmniBack.

| | |
|---|---|
| *NOTE:* | When booting from a **`make_tape_recovery`** tape the entire root volume group and any other included volume groups will be re-created. You will not be able to do selective restores. |

When **`make_tape_recovery`** is executed, the choice of the archive contents depends on the command options that you choose. These options are discussed on next several pages.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 2–4. SLIDE: Creating a Minimum Core Recovery Tape




## Student Notes

When choosing the type of recovery tape to create, you must take into consideration the speed of the recovery, and how much additional work must be done to restore the rest of the system.

To create a minimum system, simply use the **`make_tape_recovery`** command with no options. The archive will include the files listed in the file **`/opt/ignite/recovery/mnr_essentials`**. You should not modify this file.

If you wish to modify the list of essential files (perhaps you need to add the files necessary to restore your backup/restore program), you should copy **`/opt/ignite/recovery/mnr_essentials`** to **`/var/opt/ignite/recovery/mnr_essentials`** and edit the file there. **`make_tape_recovery`** will always check the file in `/var` first.

The following is a listing of the **`/opt/ignite/recovery/mnr_essentials`** file:

```
# @(#) $Revision: 10.9 $
#
# /opt/ignite/recovery/mnr_essentials
```



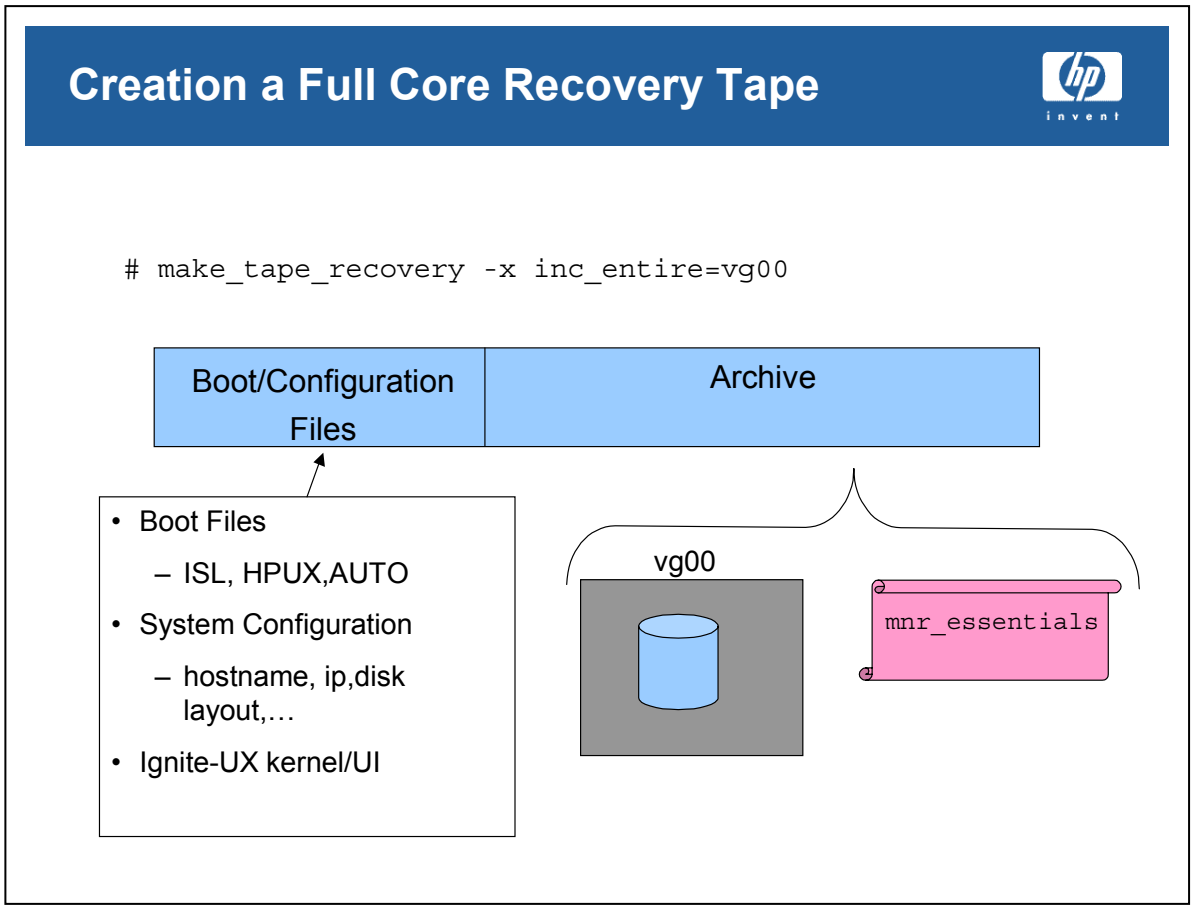

© 2005 Hewlett-Packard Development Company, L.P.

```
#
# WARNING: This file should not be modified, for the following
reasons:
#    - Changes to this file will be overwritten with the next
revision.
#    - It can lead to including more than you really need without
#      the ablity to remove what you add via make_net_recovery(1M).
#
# If you must modify the list of essential files, copy this file
# to the following location, then make the necessary modifications:
# /var/opt/ignite/recovery/mnr_essentials
#
# /var/opt/ignite/recovery/mnr_essentials is checked before
# /opt/ignite/recovery/mnr_essentials when looking for essential
files.
#
#
# For information about this file, see make_net_recovery(1M).

/sbin
/dev
/stand
/stand/vmunix
/usr/bin
/usr/ccs
/usr/conf
/usr/lbin
/usr/lib
/usr/newconfig
/usr/sbin
/usr/sam
/usr/share
/usr/obam

/bin
/lib
/etc
```

The boot/configuration files must be temporarily assembled on disk before they are copied to the tape. The space requirement is approximately 32M. If you do not have 32M free in **/var** and wish to specify an alternate location for storing the boot/configuration files, use the **-B** ***<boot_destination_file>*** option.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 2–5. SLIDE: Creating a Full Core Recovery Tape




## Student Notes

The creation of a full recovery tape for the root volume group will include all of the contents of vg00, or the root disk(s). The command to create a full recovery tape is:

```
make_tape_recovery -x inc_entire=vg00
```

This creates a full archive of the root volume group (**vg00**) on the default tape device: **/dev/rmt/0m.** If there are any files in the essential list that are not part of vg00 they will be backed up as well.

If you wish to include the entire contents of all volume groups containing essential files, use the command:

```
make_tape_recovery -A
```




© 2005 Hewlett-Packard Development Company, L.P.

## 2–6. SLIDE: Adding to the Archive

### Adding to the Archive

- Files/Directories

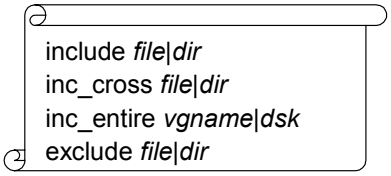
```
-x include=<file|directory>
-x inc_cross=<file|directory>
-x exclude=<file|directory>
```

- Disks/Volume Groups

```
-x inc_entire=<vgname>|/dev/dsk/<name>
```

- Use a contents file

```
-f <contents_file>
```




- Archive all volume groups that contain essential files

```
make_tape_recovery -A
```

## Student Notes

The essential files listed in **/opt/ignite/recovery/mnr_essentials** are always included in the archive. You can further customize the contents of the archive by using the options listed on the slide.

The **-x include** option includes the specified file or directory. Directories will be copied recursively, until a mount point is reached, while the **-x inc_cross** includes the file or directory recursively and crosses mount points.

The **-x exclude** option excludes files or directories. When a directory is specified, no files beneath the directory will be stored in the archive.

When modifying the contents of the archive, keep the following points in mind:

- No essential files or directories can be excluded. Exclusions of essential files or directories will be ignored.
- Exclusions take precedence over inclusions. Anything that is both excluded and included will be excluded from the archive.


**© 2005 Hewlett-Packard Development Company, L.P.**

- The ordering of inclusions and exclusions is not significant.

Examples:

Create a minimum recovery tape but also include the **/home** directory:

```
make_tape_recovery -x include=/home
```

Create a recovery tape that includes all files from file systems in vg00 and vg01, but exclude **/tmp**.

```
make_tape_recovery -x inc_entire=vg00 -x inc_entire=vg01 -x
exclude=/tmp
```

Create a recovery tape that includes all files/directories in the volume group(s) containing the files specified in the essentials list (**/opt/ignite/recovery/mnr_essentials** or **/var/opt/ignite/recovery/mnr_essentials**).

```
make_tape_recovery -A
```




© 2005 Hewlett-Packard Development Company, L.P.

## 2–7. SLIDE: `make_tape_recovery` Caveats

### `make_tape_recovery` Caveats

- LVM disk mirrors not restored
  - must re-mirror manually
- VxVM disk groups are not included in the archive
  - disk groups will be left undisturbed and will be re-integrated to the system when recovery is complete.

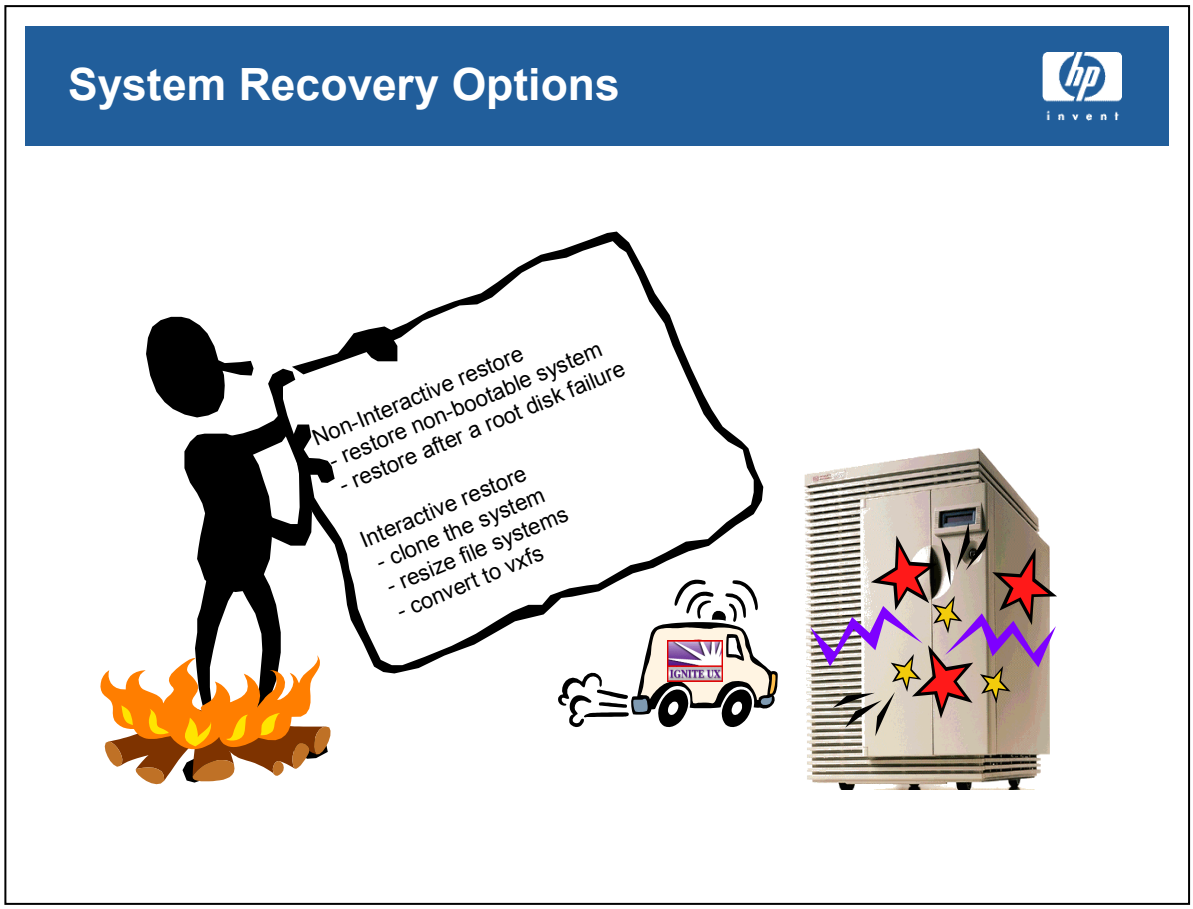
## Student Notes

The **`make_tape_recovery`** tool will create a recovery backup for a system with mirrored disks but it will not restore the mirrored disk configuration. If the system is later recovered, previously mirrored volumes will no longer be mirrored. They can be manually re-mirrored after the system is up. For more details, see the white paper **`/opt/ignite/share/doc/diskmirror.pdf`**.

The disk groups managed by VERITAS Volume Manager (VxVM) cannot be included in the Ignite-UX archive. If they are included, **`make_tape_recovery`** will error out. Those disk groups will be left undisturbed and re-integrated to the system after the recovery is complete.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 2–8. SLIDE: System Recovery Options




## Student Notes

### Uses for make_tape_recovery

The `make_tape_recovery` tool will create a tape that can be used to perform a non-interactive recovery or Ignite-UX installation of your system. Data stored in the archive (created by `make_tape_recovery`) can be used for the following:

**Non-interactive Recovery:**

- Restore a non-bootable system with little or no intervention.
- Restore a system in the event of a root disk (volume group) failure.

**Interactive Recovery:**

- Convert file systems in the root volume from `hfs` to `vxfs` (logical volumes).
- Modify the root file system size (logical volume).
- Modify the size of primary swap (logical volume).
- Clone software from one system to another (requires nearly identical hardware).




© 2005 Hewlett-Packard Development Company, L.P.

## Cloning Systems

Using a recovery tape to clone systems will only work if the two systems are capable of running the same software configuration. This means that the source system must contain a version of HP-UX with all necessary patches and driver software required by the clone system.

If the two systems are different hardware models, a new kernel will automatically be built to suit the new hardware. If the two systems are the same model, the clone will by default use the kernel from the original system. You may force a kernel to be built by using the "Additional" screen in the Advanced user interface to set the "Cloning to different HW?" selector to TRUE.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 2–9. SLIDE: Verifying the Recovery Tape

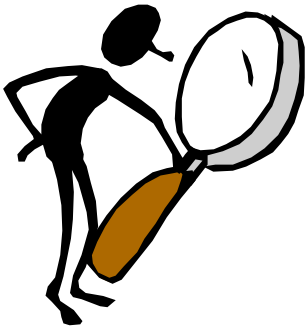
### Verifying the Recovery Tape

Log File:

```
/var/opt/ignite/recovery/<date,time>/recovery.log
```

Tape Table of Contents:

```
# mt -t /dev/rmt/0mn fsf 1
# tar tvf /dev/rmt/0m
```





## Student Notes

After you have created the archive on tape using **`make_tape_recovery`** you should check the log file in:

**`/var/opt/ignite/recovery/<date,time>/recovery.log`**

You can also get a table of contents from the archive by using the commands listed on the slide. You may want to redirect the contents of the **`tar`** command to a file. To view what was supposed to be archived, check the file:

**`/var/opt/ignite/recovery/latest/flist`**


© 2005 Hewlett-Packard Development Company, L.P.

## 2–10. SLIDE: Booting the Recovery Tape

### Booting the Recovery Tape

- Power up the system ( or <cntl> B and RS )
- Interrupt the boot
- Load the recovery tape
- Search for bootable devices
- Boot from the recovery tape
- Automatic or interactive restore

## Student Notes

The tape created by **`make_tape_recovery`** is called the **`system_recovery_boot_tape`**. This tape is really an Ignite-UX custom installation medium. This tape can be used to boot your system and re-install the created image automatically, or you can interact to customize the installation of the system.

To recover a non-bootable system:

1. Power up the system (or **<cntl> B** and **RS** )
2. Interrupt the boot
3. Load the recovery tape into the tape drive
4. Search for bootable devices **`(Main Menu> sea ipl )`**
5. Boot from the recovery tape
6. Let the system recover automatically, or interact to customize the installation


© 2005 Hewlett-Packard Development Company, L.P.

## Searching for Bootable Devices

How you proceed here depends entirely on your system and PDC version. To access the interface menu, you'll need to interrupt the boot process. Pressing the SPACE bar after power on is sufficient for this. Once you interrupt the boot, you'll want to execute the search command to find bootable devices (unless you remember the hardware path to the tape drive).

## Example: Booting from the Recovery Tape

*(Ignite-UX recovery for a D-220)*

**…Power up sequence…**

```
Processor is booting from first available device.

To discontinue, press any key within 10 seconds.

<any key>

Boot terminated.

------ Main Menu ----------------------------------------
Command                                Description
 --------------                          ----------------
Boot  [PRI|ALT|<path>]                 Boot from specified path
Path  [PRI|ALT|CON|KEY]  [<path>]      Display or modify a path
SEArch  [Display|IPL][<path>]          Search for boot devices

Configuration [<command>]              Access Configuration menu/commands
Information [<command>]                Access Information menu/commands
SERvice [<command>]                    Access Service menu/commands

Display                                Redisplay the current menu
Help [<menu>|<command>]                Display help for menu or command
RESET                                  Restart the system
```

Main Menu: Enter command > **sea ipl** *(Search for possible boot devices)*

```
Searching for device(s) with bootable media …
This may take several minutes.

To discontinue search, press any key (termination may not be immediate).
```

| Path Number | Device Path (dec) | Device Type and Utilities |
|---|---|---|
| P0 | 8/16/5.6 | Random access media IPL |
| P1 | 8/16/5.5 | Random access media IPL |
| **P2** | **8/16/5.0** | **Sequential access media IPL** |
| P3 | 8/16/6.0 | LAN Module |

Main Menu: Enter command **> boot p2**

Interact with IPL (Y, N, or Cancel)?> **n**




© 2005 Hewlett-Packard Development Company, L.P.

```
Booting …
Boot IO Dependent Code (IODC)  revision 144


SOFT Booted.
ISL Revision A.00.38   June 30, 2001

ISL booting  hpux (;0):INSTALL

Boot
: tape(8/16/5.0.0.0.0.0;0):INSTALL
```

Unless you used the **–I** option when you created the tape, the restore will call for a non-interactive restore. You will have an opportunity to cancel the non-interactive installation by hitting the return key when the following messages are displayed:

```
Warning: The configuration information calls for a non-interactive
         installation.

Press <Return> within 10 seconds to cancel batch-mode installation:
```

If you cancel the non-interactive installation, or you created the tape for an interactive installation, the Ignite-UX Welcome screen will be presented.

Select the option **[Install HP-UX]**

Select the option **[ Advanced Installation ]**

You can then make any desired changes to file systems, hostname, IP address, timezone, root password, DNS server, and gateway information.

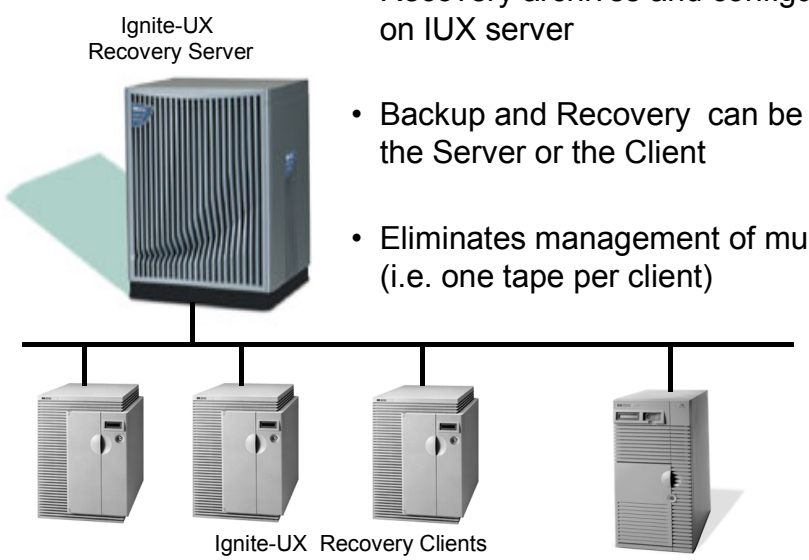
Select **[GO]** to proceed with the installation.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 2–11. SLIDE: The `make_net_recovery` Tool

### The `make_net_recovery` Tool

- Centralized control over many systems
- Recovery archives and configuration stored on IUX server
- Backup and Recovery can be initiated from the Server or the Client
- Eliminates management of multiple tapes (i.e. one tape per client)






## Student Notes

While the **`make_tape_recovery`** command is useful to for making recovery tapes for individual systems, the **`make_tape_recovery`** command offers the advantages listed on the slide.

Rather than creating a separate tape for each managed system, the archives are stored either on the Ignite-UX server, or on any system on the network.

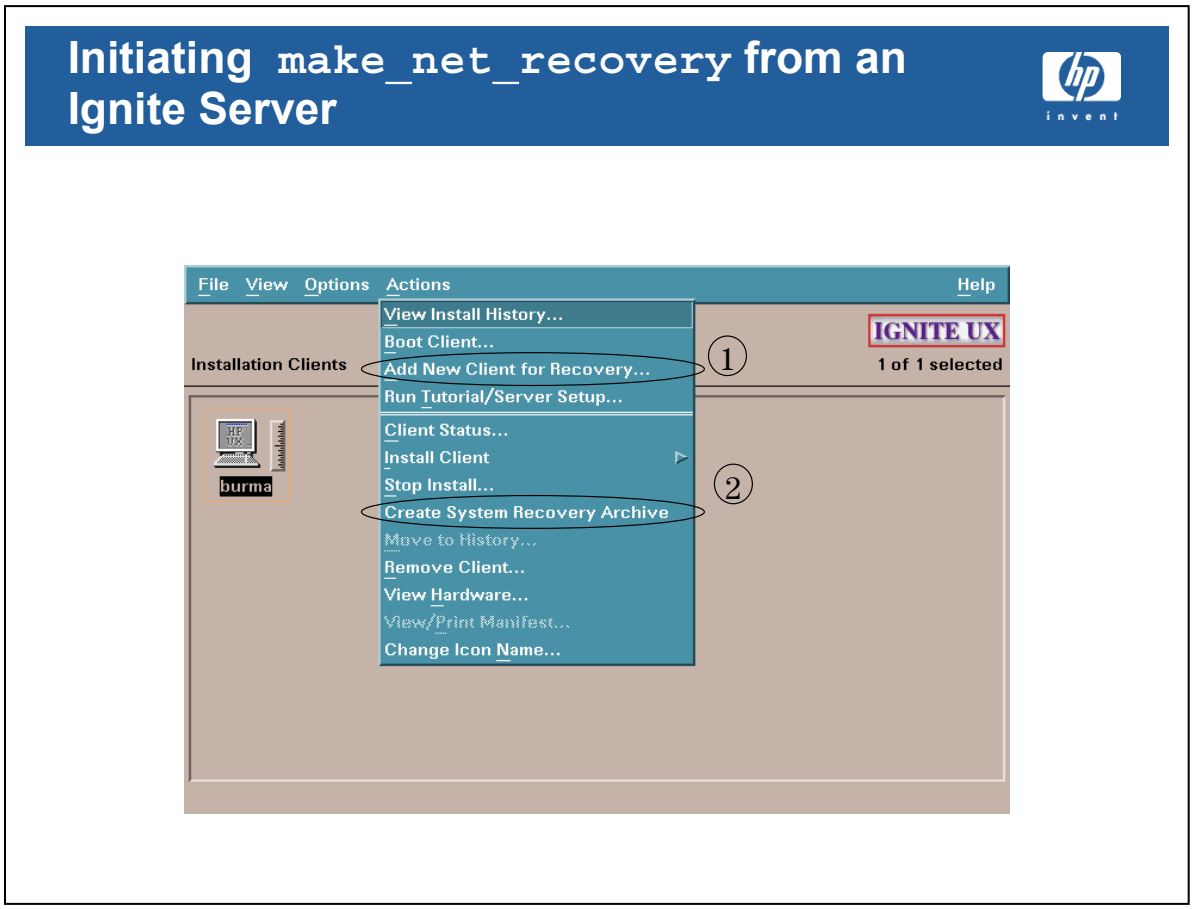
The configuration files necessary to do a restore are stored on the Ignite-UX server. Each client will have a directory similar to the **`/var/opt/ignite/recovery/<date,time>`** directory used by **`make_tape_recovery`** on the Ignite-UX server called **`/var/opt/ignite/clients/0x{LLA}/recovery/<date,time>`**

If a client restore becomes necessary, the client boots over the network from the Ignite-UX server.

The syntax and procedures for using **`make_net_recovery`** and **`make_tape_recovery`** are almost identical.


© 2005 Hewlett-Packard Development Company, L.P.

## 2–12. SLIDE: Initiating make_net_recovery from an Ignite Server




## Student Notes

With the **make_net_recovery** tool, an image of a clients' file systems (including files from outside the **/dev/vg00** volume group) can be archived to the disks of the Ignite-UX server.

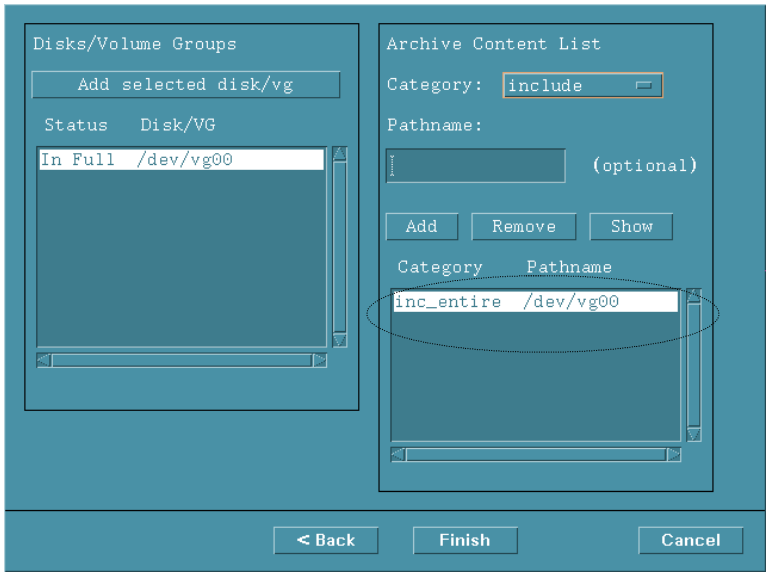
The **make_net_recovery** archive can be created from the client or from the Ignite-UX server. To initiate the creation of the archive from the Ignite-UX server, enter the Ignite-UX GUI with the **ignite** command. From within the Ignite-UX GUI, perform the following to steps:

1. **Add New Client for Recovery**… This adds a client icon to the Installation Client window. Once the client icon is created, highlight the client icon and perform step 2.

2. **Create System Recovery Archive**. This causes a number of screens to display related to the creation of the client archive on the Ignite-UX server. Among the different screens is one that allows the specification of files to be included in the client archive (see next page).


© 2005 Hewlett-Packard Development Company, L.P.

## 2–13. SLIDE: Selecting Files for the Client Archive




## Student Notes

The above slide shows the screen for specifying which files to include in the client archive.

The example on the slide shows the entire **vg00** volume group selected.

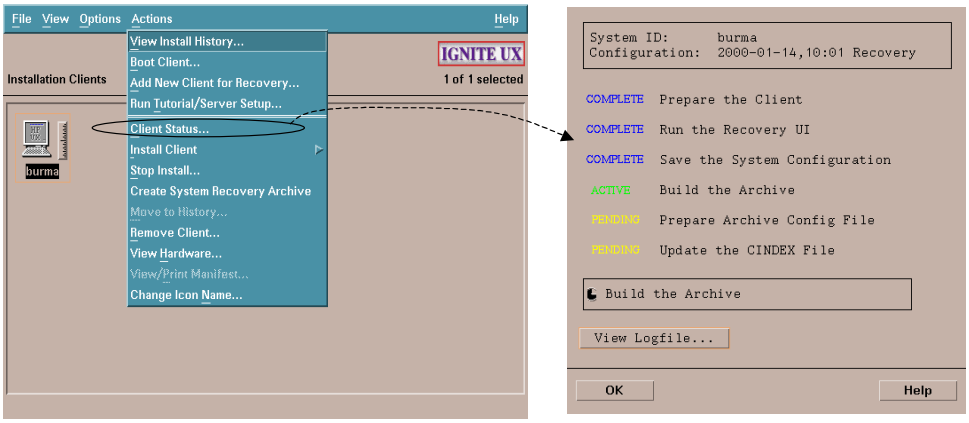



© 2005 Hewlett-Packard Development Company, L.P.

## 2–14. SLIDE: Monitoring Creation of Client Archive

### Monitoring Creation of Client Archive

- Clients can be monitored from the central Ignite-UX server
- Detailed log file is available for each client on the IUX server




## Student Notes

One advantage of initiating the creation of the client archives from the Ignite server is the ability to monitor multiple client archive creations from a single display. Many client archives can be created in parallel and monitored, without ever having to leave the Ignite-UX server's display.

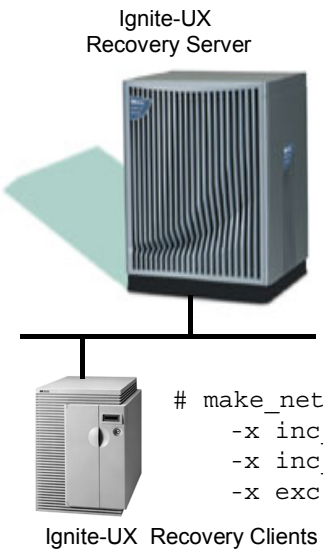
Detailed status of an individual client archive can also be obtained from the GUI by highlighting the client icon and selecting **`Client Status`**... from the **`Action`** pulldown menu.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 2–15. SLIDE: Initiating `make_net_recovery` on the Client

### Initiating `make_net_recovery` on the Client

- Backups can be initiated from the client
- New options to specify data to archive
- Can archive more than just `/dev/vg00`
- Backups can be automated through `cron`




```
# make_net_recovery -s IUX_server_name \
    -x inc_entire=vg00 \
    -x inc_entire=vg01 \
    -x exclude=/depots
```

## Student Notes

In addition to being able to initiate the creation of a client archive from the Ignite-UX server, the creation of the archive can also be initiated on the client itself.

A command line executable (**`make_net_recovery`**) is available so the client can locally initiate the creation of the archive. This allows the client to determine when a new archive is needed, rather than having the server initiating new archives periodically even though they may not be necessary.

The syntax of the **`make_net_recovery`** command is quite similar to the **`make_tape_recovery`** command. One advantage to running **`make_net_recovery`** on the client is the ability to automate the process through **`cron`**.




© 2005 Hewlett-Packard Development Company, L.P.

## 2–16. SLIDE: Using the Archive to Recover a Client

### Using the Archive to Recover a Client

To recover a failed root disk or volume group using the system recovery archive:

- Boot the client system, listing the Ignite-UX server as the boot source
  - Client PDC prompt:
    `bo lan.IP_Address_of_Ignite_server  install`
- Do not interact with IPL
- Select: [ Install HP-UX ], Select "Advanced Installation" .
- If prompted, fill in the appropriate networking information
- Select the recovery configuration for your client

## Student Notes

The procedure to recover a failed system using the system recovery archive is shown on the slide:

1. Boot the client system.
2. During the boot, interrupt the session and boot to the Ignite-UX server. Do NOT interact with the IPL/ISL.
3. From the Installation and Recovery menu, select “Install HP-UX”.
4. Once the client boots to the Ignite-UX server, an icon for the client will appear in the Ignite-UX server GUI. At this point, the installation could proceed from either the Ignite Server or the client. On the server, highlight the client icon.
5. Once the client icon has been highlighted, select from the action menu, Install/New Install.
6. From this window, select the appropriate archive to restore, then click OK.


© 2005 Hewlett-Packard Development Company, L.P.

The restoration of the client archive will begin. Allow restoration to continue until completion.




© 2005 Hewlett-Packard Development Company, L.P.

## 2–17. SLIDE: Module Objectives

### Module Objectives

You should now be able to do the following:

- Understand system recovery choices
- Describe the layout of a system recovery tape
- Describe how to create a system recovery tape
- Understand system recovery options
- Describe how to verify a system recovery tape
- Describe how to boot from a system recovery tape
- Understand how to create a `make_net_recovery` archive

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 2–18. LAB: Creating a `make_net_recovery` archive

### Directions

Create a **`make_net_recovery`** archive of your system. Use the server designated by your instructor.

1. Verify that the Ignite-UX software is loaded. If not, install it. Ask your instructor for a depot location.

   ```
   # swlist -l product |grep Ignite-UX
   ```

2. Run **`make_net_recovery`** and choose a full vg00 archive.

   ```
   # make_net_recovery -s Ignite_Server -x inc_entire=vg00 \
           -x exclude=/tmp
   ```

3. When the above command completes, logon to the Ignite server and inspect the **`/var/opt/ignite/clients/<your client>/recovery/<date,time>`** directory. You should see several files that correspond to your archive.

   When the **`make_net_recovery`** completes check the file **`/var/opt/ignite/clients/<your client>/recovery/<date,time>/flist`** to see how many files will be archived?

   ```
   # wc -l flist
   ```

   Will **`/home`** be included in the archive?

   ```
   # grep '^[0-9][0-9]* tmp/' flist
   ```

4. This archive will function as a backup in case something goes wrong in a future module.




© 2005 Hewlett-Packard Development Company, L.P.

## 2–19. LAB: Creating a `make_tape_recovery` archive (Optional)

### Directions

If you have a tape drive available and if you have time, perform a **`make_tape_recovery.`**

1. Insert a tape in the tape drive. Run **`make_tape_recovery`** in preview mode, specifying that you wish to create a minimum core recovery tape. (Ignore any vxvm errors)

   ```
   # make_tape_recovery -p
   ```

2. When the above command completes change directories to **`/var/opt/ignite/recovery`**. Do a long listing. Change to the **`latest`** directory. Do another long listing.

   ```
   # cd /var/opt/ignite/recovery
   # ll
   # cd latest
   # ll
   ```

   When the **`make_tape_recovery`** completes, check the file **`/var/opt/ignite/recovery/latest/flist`** to see how many files will be archived?

   ```
   # wc -l flist
   ```

   Will **`/home`** be included in the archive?

   ```
   # grep '^[0-9][0-9]* home/' flist
   ```

3. Run **`make_tape_recovery`** in preview mode, specifying that you wish to create an archive containing all the files and directories in the root volume group excluding **`/tmp`**. (Ignore any vxvm errors)

   ```
   # make_tape_recovery -p -x inc_entire=vg00 -x exclude=/tmp
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

4. When the above command completes change directories to **/var/opt/ignite/recovery**. Do a long listing. Change to the latest directory. Do another long listing. When **make_tape_recovery** runs in preview mode it renames the directory "**latest**" therefore you will need to **cd** out of the renamed "**latest**" directory and then **cd** back to latest to ensure that you are looking at the current information.

```
# cd /var/opt/ignite/recovery
# ll
# cd latest
# ll
```

How many files will be archived?

```
# wc -l flist
```

Will **/home** be included in the archive?

```
# grep '^[0-9][0-9]* home/' flist
```

Will **/tmp** be included in the archive?

```
# grep '^[0-9][0-9]* tmp/' flist
```

5. Create a recovery tape that includes all the files in all the volume groups containing the essential files, calling for an interactive restore. This step will take quite a while.

```
# make_tape_recovery -A -I
```

6. After the tape has been created, check the log file in **/var/opt/ignite/recovery/latest/recovery.log**.




© 2005 Hewlett-Packard Development Company, L.P.

## Optional

Check with your instructor for a convenient time to try to restore from your tape.


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 3 — Troubleshooting Non-Bootable LVM Disks and LVM Structures

## Objectives

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

- Describe the critical structures in an LVM environment.
- Diagnose and restore missing LVM device files.
- Diagnose and rebuild a missing or corrupt **`/etc/lvmtab`** file.
- Update the kernel LVM structures when missing disks return.
- Identify failed LVM disks.
- Backup LVM data structures using **`vgcfgbackup`**.
- Restore LVM data structures using **`vgcfgrestore`**.
- Restore file system and user data when a non-mirrored data disk fails.
- Restore file system and user data when a mirrored data disk fails.
- Remove a corrupted physical volume from a volume group.
- Remove a corrupted volume group from a host.
- List critical recovery information that should be recorded in a system log book.




© 2005 Hewlett-Packard Development Company, L.P.

## 3–1. SLIDE: Module Objectives

### Module Objectives

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

- Describe the critical structures in an LVM environment.
- Diagnose and restore missing LVM device files.
- Diagnose and rebuild a missing or corrupt `/etc/lvmtab` file.
- Update the kernel LVM structures when missing disks return.
- Identify failed LVM disks.
- Backup LVM data structures using `vgcfgbackup`.
- Restore LVM data structures using `vgcfgrestore`.
- Restore file system and user data when a non-mirrored data disk fails.
- Restore file system and user data when a mirrored data disk fails.
- Remove a corrupted physical volume from a volume group.
- Remove a corrupted volume group from a host.
- List critical recovery information that should be recorded in a system log book.

## Student Notes

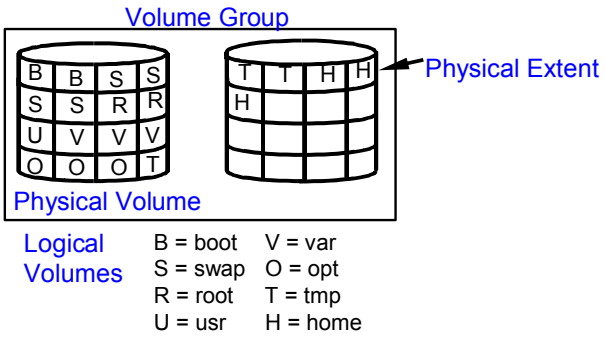

**© 2005 Hewlett-Packard Development Company, L.P.**

## 3–2. SLIDE: LVM –Logical Volume Manager - Review

### LVM — Logical Volume Manager - Review

LVM provides:

- Disk grouping
- Logical partitioning
- Ability to span disk drives
- Bad block replacement
- Disk mirroring (optional product)




## Student Notes

The logical volume manager (LVM) allows disks (also known as physical volumes) to be grouped together into volume groups. Space from these groups can then be allocated to logical volumes in increments of physical extents, which provides a very flexible environment to manage disk resources.

The logical volumes behave exactly like physical disk partitions. They have block and character devices. Normal file systems are built within the logical volumes. These are managed with the same commands as other file systems. The only real difference is that the size of a logical volume and the file systems it contains can be extended; physical disk partitions cannot be easily extended.

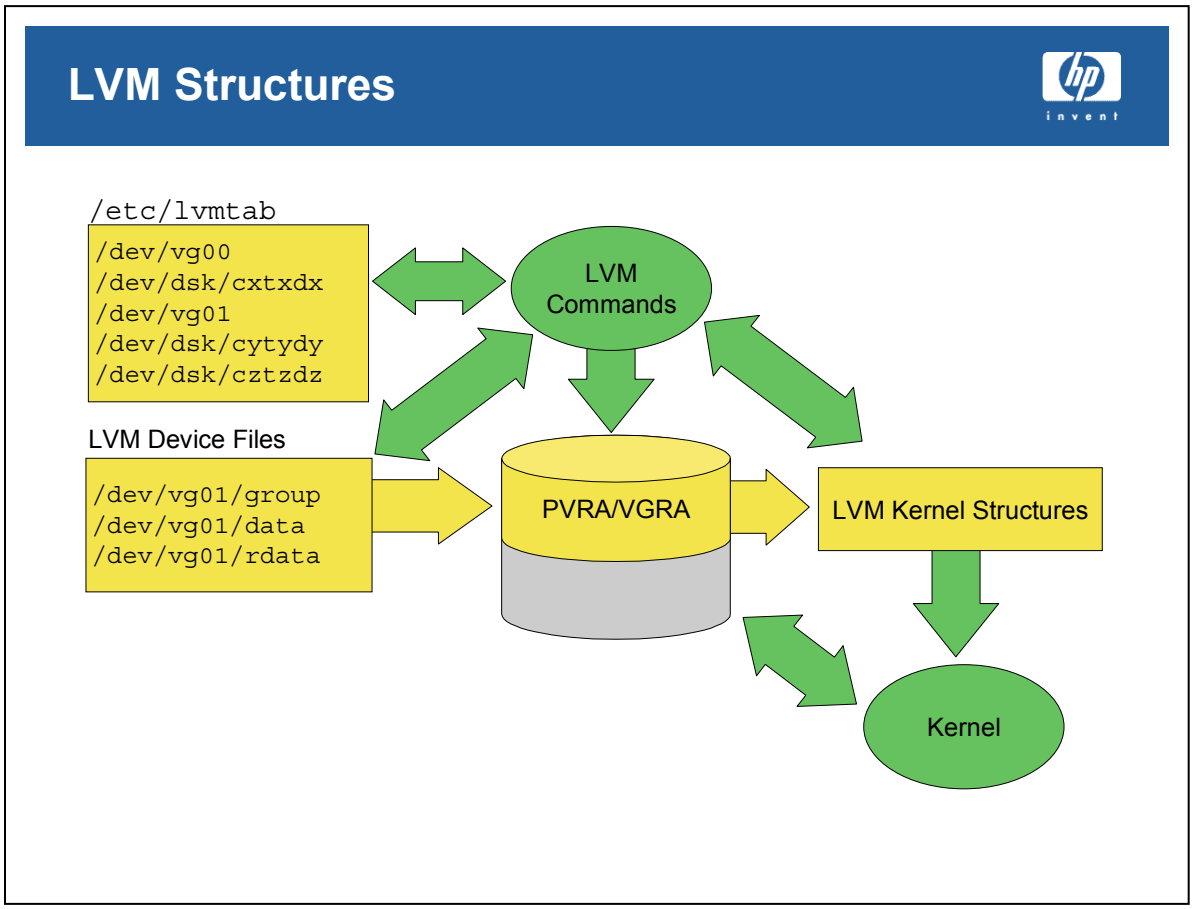
Disk mirroring permits you to store identical copies of data on separate disks, thereby increasing safety and availability of data.

*NOTE:* In order to use mirroring, you will need HP MirrorDisk/UX software (product number B2491A), which can be ordered separately.




© 2005 Hewlett-Packard Development Company, L.P.

## 3–3. SLIDE: LVM Structures




## Student Notes

LVM configuration information is stored in a variety of structures. This slide provides an overview of the LVM structures; future slides in the chapter will consider each structure's purpose, vulnerabilities, and recovery mechanisms.

**`/etc/lvmtab`**

**`/etc/lvmtab`** cross references disk devices with volume groups. This file is referenced at boot time to determine which volume groups should be activated, and as a sanity check for every LVM command executed thereafter. Without the **`/etc/lvmtab`** file, none of the standard LVM commands will work.

Device Files

LVM device files provide the administrator a mechanism for identifying and accessing volume groups, logical volumes, and physical volumes. Without LVM device files, already-mounted logical volumes will remain accessible, but you will be unable to mount logical volumes or make changes to the logical volumes and volume groups whose device files are missing.


**© 2005 Hewlett-Packard Development Company, L.P.**

| | |
|---|---|
| PVRA/VGRA | These on-disk structures store the volume group extent maps and other configuration information. The structures are referenced at activation time to populate the LVM kernel structures. If the PVRA/VGRA are corrupted, already-activated volume groups will be unaffected, but future attempts to activate the volume groups containing the damaged disks will likely fail. |
| LVM Kernel Structures | When a volume group is activated, LVM uses the volume group's on-disk PVRA/VGRA structures to populate the LVM kernel structures. The LVM pseudo-driver then uses these kernel structures to determine your system's LVM configuration, and to map logical extents to actual disk locations. It is fairly rare for these structures to become corrupted, though they occasionally become outdated. |




© 2005 Hewlett-Packard Development Company, L.P.

## 3–4. SLIDE: Recovering Lost or Damaged Structures

### Recovering Lost or Damaged Structures

| Structure | Recovery Strategy |
|---|---|
| Device Files | Restore from tape backup *or*<br>Recreate using `insf` and `mknod` |
| `/etc/lvmtab` | Restore from tape backup *or*<br>Rebuild using information from PVRAs/VGRAs via `vgscan` |
| Kernel Structures | Re-scan PVRAs/VGRAs using `vgchange -a y` |
| PVRA/VGRA | Restore from `/etc/lvmconf` backup directory |

## Student Notes

LVM was designed to be robust, and loss of any single structure will rarely lead to loss of data if you follow the recommendations presented in this chapter. The table on the slide provides a conceptual overview of the recovery mechanism that may be used to restore each of the LVM structures mentioned on the previous page.

Device Files — May be restored from a backup tape, or may be recreated using **`insf`** and **`mknod`**.

**`/etc/lvmtab`** — May be restored from a backup tape, or may be recreated using the **`vgscan`** command. **`vgscan`** scans the PVRA/VGRA header information on your system's disks to determine which disks belong to which volume groups.

Kernel LVM Structures — Simply execute the **`vgchange -a y`** command. **`vgchange`** re-reads your PVRA/VGRA headers and updates the kernel LVM structures accordingly.


**© 2005 Hewlett-Packard Development Company, L.P.**

| | |
|---|---|
| PVRA/VGRA | LVM maintains backup copies of your VGRA/PVRA header information in the `/etc/lvmconf` directory. The `vgcfgrestore` command may be used to restore these structures if a disk fails. |




© 2005 Hewlett-Packard Development Company, L.P.

## 3–5. SLIDE: Missing Device Files: Symptoms

### Missing Device Files: Symptoms

When device files are missing, file systems that are already mounted will remain accessible, but new attempts to mount or modify the affected LVs will fail.

```
# vgdisplay -v vg01
vgdisplay: "/dev/vg01/group": not a character device.
vgdisplay: Cannot display volume group "vg01".

# lvdisplay -v /dev/vg01/data
lvdisplay: "/dev/vg01/data": No such file or directory

# pvdisplay /dev/dsk/c0t8d0
pvdisplay: "/dev/dsk/c0t8d0": No such file or directory
```

## Student Notes

LVM device files provide the administrator a mechanism for identifying and accessing volume groups, logical volumes, and physical volumes.

### Causes of Lost Device Files

Lost device files typically are the result of administrator error. Be very careful when using the **rm** command while logged in as root!

### Symptoms of Lost Device Files

If you accidentally remove LVM device files, already-mounted logical volumes will remain accessible, but you will be unable to access, mount, or modify to the logical volumes and volume groups whose device files are missing.

```
# vgdisplay -v vg01
vgdisplay: "/dev/vg01/group": not a character device.
vgdisplay: Cannot display volume group "vg01".

# lvdisplay -v /dev/vg01/data
lvdisplay: "/dev/vg01/data": No such file or directory
```


© 2005 Hewlett-Packard Development Company, L.P.

```
# pvdisplay /dev/dsk/c0t8d0
pvdisplay: "/dev/dsk/c0t8d0": No such file or directory
```




© 2005 Hewlett-Packard Development Company, L.P.

## 3–6. SLIDE: Missing Device Files: Solution

### Missing Device Files: Solution

Missing device files may be restored from your last tape backup, or may be recreated using insf (for PVs) or mknod (for VGs and LVs). Don't forget to re-create both the block *and* character device files for LVs!

Re-create PV device files:

```
# ioscan -C disk
# insf -eC disk
```

Re-create a VG device file:

```
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# chmod 755 /dev/vg01
```

Re-create LV device files:

```
# mknod /dev/vg01/data b 64 0x010001
# mknod /dev/vg01/rdata c 64 0x010001
# chmod 640 /dev/vg01/*
```

## Student Notes

Missing device files may be restored from your last tape backup, or may be recreated using **`ioscan`** and **`insf`** (for physical volumes) or **`mknod`** (for volume groups and logical volumes). Don't forget to restore or re-create both the block *and* character device files for logical volumes and physical volumes! Volume groups only have character device files.

Re-create physical volume device files via autoconfiguration:

```
# ioscan -C disk
# insf -eC disk
```

Re-create a volume group character device file:

```
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# chmod 755 /dev/vg01
```

Re-create logical volume block and character device files:

```
# mknod /dev/vg01/data b 64 0x010001
# mknod /dev/vg01/rdata c 64 0x010001
# chmod 640 /dev/vg01/*
```


**© 2005 Hewlett-Packard Development Company, L.P.**

Recall that the first two digits in a volume group or logical volume minor number identify the number of the volume group that the device file is associated with. The last two digits in a logical volume minor number identify individual logical volumes within a volume group.

If you aren't sure which logical volume names and minor numbers were being used in a volume group, you can use a short shell script to create device files for all 255 available minor numbers:

```
#!/usr/bin/sh
vgname=vg01
vgnum=01
mkdir /dev/$vgname
for m in 0 1 2 3 4 5 6 7 8 9 a b c d e f
do
   for n in 0 1 2 3 4 5 6 7 8 9 a b c d e f
   do
      if [[ $m$n = 00 ]]; then
         mknod /dev/$vgname/group     c 64 0x${vgnum}00$m$n
      else
         mknod /dev/$vgname/lvol$m$n  b 64 0x${vgnum}00$m$n
         mknod /dev/$vgname/rlvol$m$n c 64 0x${vgnum}00$m$n
      fi
   done
done
```

Then run **`vgdisplay`**. Logical volumes with size 0M may be safely removed. Mount the remaining logical volumes to determine their contents and guess their names. Then unmount and rename the device files using the mv command.

```
# mv /dev/vg01/lvol1 /dev/vg01/data
# mv /dev/vg01/rlovl1 /dev/vg01/rdata
```




© 2005 Hewlett-Packard Development Company, L.P.

## 3–7. SLIDE: Corrupt `/etc/lvmtab`: Symptoms

### Corrupt `/etc/lvmtab`: Symptoms

`/etc/lvmtab` is used as a sanity check for all LVM commands, and to determine which VGs to activate at boot time. Without `/etc/lvmtab`, LVM commands will fail, but already mounted file systems remain accessible.

```
# vgdisplay -v vg01
vgdisplay: /etc/lvmtab: No such file or directory
vgdisplay: "/etc/lvmtab" could not be read into memory.

# lvcreate -n data -L 16 vg01
The version of /etc/lvmtab file is not correct.
Remove /etc/lvmtab file and execute vgscan command to recreate the
file.
lvcreate: "/etc/lvmtab" could not be read into memory.
```

## Student Notes

**`/etc/lvmtab`** cross references disk device file names with their associated volume groups. The file is referenced at boot time to determine which volume groups should be activated, and as a sanity check for every LVM command executed thereafter.

### Causes of a Lost or Corrupted `/etc/lvmtab`

Problems with the **`/etc/lvmtab`** file are rare, and are often the result of administrator error.

### Symptoms of a Lost or Corrupted `/etc/lvmtab`

Without **`/etc/lvmtab`**, LVM commands will fail, but already mounted file systems remain accessible.

```
# vgdisplay -v vg01
vgdisplay: /etc/lvmtab: No such file or directory
vgdisplay: "/etc/lvmtab" could not be read into memory.
```


© 2005 Hewlett-Packard Development Company, L.P.

```
# lvcreate -n data -L 16 vg01
The version of /etc/lvmtab file is not correct.
Remove /etc/lvmtab file and execute vgscan command to recreate
the file.
lvcreate: "/etc/lvmtab" could not be read into memory.
```

Recall that **/etc/lvmtab** is a binary file, though you can view portions of the file using the strings command.

```
# strings /etc/lvmtab
/dev/vg00
/dev/dsk/c1t15d0
/dev/vg01
/dev/dsk/c0t8d0
/dev/dsk/c0t1d0
/dev/dsk/c0t12d0
/dev/dsk/c0t14d0
```

Sometimes administrators mistakenly believe they have **/etc/lvmtab** file corruption, when in fact the file is fully functional. Consider the following example:

```
# pvcreate -f /dev/rdsk/c0t1d0
pvcreate: The physical volume "/dev/dsk/c0t8d0" is already
recorded in the "/etc/lvmtab”.
```

This doesn't mean that there's a problem with the **/etc/lvmtab** file – it means that you are trying to **pvcreate** a disk that's already in a volume group!



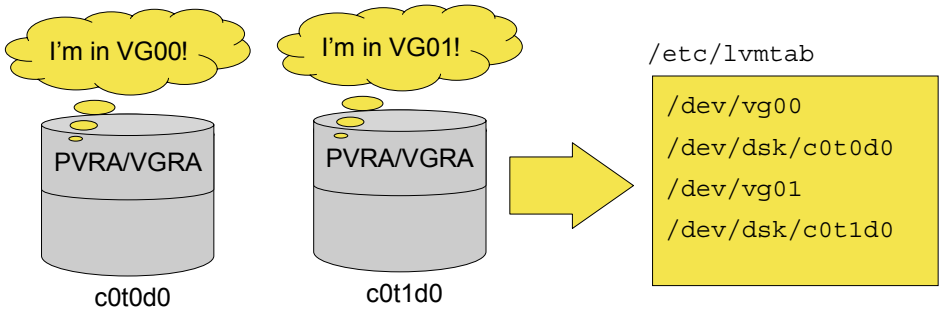

© 2005 Hewlett-Packard Development Company, L.P.

## 3–8. SLIDE: Corrupt `/etc/lvmtab`: Solution

### Corrupt `/etc/lvmtab`: Solution

If `/etc/lvmtab` is corrupted, make a backup copy, then use `vgscan` to scan your PVRAs/VGRAs and device files, and make corrections as necessary.

```
# cp /etc/lvmtab /etc/lvmtab.bkp
# vgscan
# strings /etc/lvmtab
```




## Student Notes

If **/etc/lvmtab** is corrupted or lost, make a backup copy of the file, then run **vgscan** to scan your PVRAs/VGRAs and device files, and make corrections as necessary. You may wish to run the command first in preview mode (**-p**), before running the command a second time to actually update the file. The **-v** option provides verbose output.

```
# cp /etc/lvmtab /etc/lvmtab.bkp
# vgscan -p
# vgscan
```

**vgscan** should be run only in the event of a catastrophic error such as the deletion of the **/etc/lvmtab** file or the mismatch of names of the physical volumes in the **/etc/lvmtab** file to the actual physical volume path configuration. If the **/etc/lvmtab** file exists, the information contained in the file is used to assist in rebuilding the file, but the existing file is updated with the new corrected configuration.


**© 2005 Hewlett-Packard Development Company, L.P.**

**vgscan** searches each physical volume connected to the system, looking for LVM headers. It matches physical volumes to their associated volume groups using the PVID and VGID information in the PVRAs, then searches the **/dev** directory for all group device files, and tries to match device files with the logical volumes' information found on the physical volumes, and updates the **/etc/lvmtab** file accordingly.

You may get a message similar to the following after you run **vgscan**:

```
Following Physical Volumes belong to one Volume Group.
Unable to match these Physical Volumes to a Volume Group.
Use the vgimport command to complete the process.
/dev/dsk/c0t12d0
/dev/dsk/c0t14d0
```

This simply indicates that **vgscan** found one or more disks that had LVM headers that didn't have corresponding group device files. These are likely disks that belong to volume groups that were exported, but weren't properly cleaned via **dd** or **mediainit**.

### Warning for Users of PV Links

If you use disk arrays on your HP-UX system, you may choose to configure multiple redundant paths from HBAs on your host system to controllers on your array. These "Physical Volume Links" will be discussed in detail in the Disk Array chapter at the end of the course.

If you have multiple physical paths to your disk devices, the **vgscan** command may cause several side effects that may have an impact on your system.

- **vgscan** may reverse the order of your primary and alternate links in the **/etc/lvmtab** file.
- Alternate links will be added to the **/etc/lvmtab** file even if they weren't configured in the volume group initially.
- Boot information may no longer be correct due to the different order of disks in the new **/etc/lv**mtab file.

In order to correct the above problems, do the following:

1. Use **vgchange** with **-a** option to activate all volume groups.
2. Use **lvlnboot** with **-R** option to correct boot information on disk.
3. Use **vgreduce** to reduce any alternate links that were added to the **/etc/lvmtab** file by **vgscan**, but they were not needed.
4. If the original primary path of a disks become an alternate path after **/etc/lvmtab** file is reconstructed, the order can be easily reverted by using **vgreduce** to remove the primary path and use **vgextend** to add the path back again.




© 2005 Hewlett-Packard Development Company, L.P.

## 3–9. SLIDE: Outdated Kernel Structures: Symptoms

### Outdated Kernel Structures: Symptoms

The kernel structures are updated automatically when LVs/VGs are created or modified.
Occasionally, though, the PV Status field may inaccurately report a PV "unavailable":
- When a PV that is non-responsive at activation time comes back online
- When a failed/corrupted PV is replaced without deactivating/reactivating the VG

```
# vgdisplay -v vg01
   --- Physical volumes ---
   PV Name        /dev/dsk/c0t1d0
   PV Status      unavailable
   Total PE       4340
   Free PE        4340
   Autoswitch     On
```

This PV may be marked "unavailable" even though it is fully functional!

## Student Notes

When a volume group is activated, LVM uses the volume group's on-disk PVRA/VGRA structures to populate the LVM kernel structures. The LVM pseudo-driver then uses these kernel structures to determine your system's LVM configuration, and to map logical extents to actual disk locations. When you use the **`vgdisplay`**, **`lvdisplay`**, or **`pvdisplay`** commands, these utilities obtain their information from the LVM kernel structures

### Causes and Symptoms of Outdated Kernel Structures

The kernel table is updated automatically when logical volumes and volume groups are created or modified. Occasionally, though, the PV Status field may inaccurately report a PV "unavailable":

- When a PV that is non-responsive at activation time comes back online
- When a failed/corrupted PV is replaced without deactivating/reactivating the VG


© 2005 Hewlett-Packard Development Company, L.P.

In the example below, c0t1d0 has been recovered, but the LVM kernel structures believe the disk is still unavailable.

```
# vgdisplay -v vg01
   --- Physical volumes ---
   PV Name       /dev/dsk/c0t1d0
   PV Status     unavailable
   Total PE      4340
   Free PE       4340
   Autoswitch    On
```




© 2005 Hewlett-Packard Development Company, L.P.

## 3–10. SLIDE: Outdated Kernel Structures: Solution

### Outdated Kernel Structures: Solution

When a missing disk comes back online, execute `ioscan` to force the OS to recognize the returned disk, then run `vgchange` to force LVM to update the disk's "PV Status".

```
# ioscan
# vgchange -a y vg01
# vgdisplay -v vg01
   --- Physical volumes ---
   PV Name       /dev/dsk/c0t1d0
   PV Status     available
   Total PE      4340
   Free PE       4340
   Autoswitch    On
```

`vgdisplay` now correctly reports that the disk is "available".

## Student Notes

When a missing disk comes back online, execute **`ioscan`** to force the OS to recognize the returned disk, then run **`vgchange`** to force LVM to update the disk's "PV Status".

```
# ioscan -fC disk
  Class      I  H/W Path    Driver S/W State   H/W Type  Description
  ==============================================================
  disk       0  0/0/1/0.8.0 sdisk  CLAIMED     DEVICE    HP 18.2G

# vgchange -a y vg01
  Volume group "vg01" has been successfully changed.

# vgdisplay -v vg01
  --- Physical volumes ---
  PV Name       /dev/dsk/c0t1d0
  PV Status     available
  Total PE      4340
  Free PE       4340
  Autoswitch    On
```


© 2005 Hewlett-Packard Development Company, L.P.

## 3–11. SLIDE: Failed Disks: Symptoms (1 of 2)

### Failed Disks: Symptoms (1 of 2)

Monitor the `syslog.log` file for POWERFAIL messages indicating that a disk is no longer responding. Determine why the disk lost power.
If the POWERFAIL message is followed by a "power returned" message, you may simply need to increase the PV timeout value.

```
# tail /var/adm/syslog/syslog.log
vmunix: LVM: VG 1 : PV 1 (device 0x1f070500) is POWERFAILED
```

Also watch for SCSI lbolt errors in the `syslog.log`, which typically indicate that a disk or interface card needs to be replaced.

```
# tail /var/adm/syslog/syslog.log
vmunix: SCSI: Request Timeout; Abort -- lbolt: 137056, dev: bc0da000
vmunix: SCSI: Request Timeout; Abort -- lbolt: 167456, dev: bc0da000
The above message repeats 37 times.
```

## Student Notes

Although device files, **`/etc/lvmtab`**, and the LVM kernel structures are important, physical volumes themselves are perhaps the most vulnerable components of any LVM environment. LVM provides a number of tools that may be used to monitor and recover physical volumes.

### Monitoring for POWERFAIL Messages in `syslog.log`

**`/var/adm/syslog/syslog.log`** is excellent source of information when you suspect possible hardware problems. Monitor **`syslog.log`** for POWERFAIL messages indicating that a disk is no longer responding. Determine why the disk lost power.

```
# tail /var/adm/syslog/syslog.log
vmunix: LVM: VG 1 : PV 1 (device 0x1f070500) is POWERFAILED
```

If the error is accompanied by a message similar to "**`pv[#] returned to vg[#]`**", then the error can usually be attributed to abnormally high I/O activity coupled with a low timeout value on the disk driver. By default, this timeout is 30 seconds. You can change the timeout with the **`pvchange`** command.

```
# pvchange -t 180 /dev/dsk/disk_device
```




© 2005 Hewlett-Packard Development Company, L.P.

Increasing the timeout will not affect I/O performance on the disk.

### Monitoring for `lbolt` Messages in `syslog.log`

Also watch for **`lbolt`** messages in **`syslog.log`**, which often indicate that a disk or interface card needs to be replaced.

```
# tail /var/adm/syslog/syslog.log
vmunix: SCSI: Request Timeout; Abort -- lbolt: 137056, dev: 1f070500
vmunix: SCSI: Request Timeout; Abort -- lbolt: 167456, dev: 1f070500
The above message repeats 37 times.
```

### Understanding Hexadecimal Hardware Addresses in `syslog.log`

The hexadecimal code at the end of the syslog SCSI error messages (1f070500 in the example above) indicates which device may have caused the error.

The instructions below explain how to decode the hexadecimal address in the error message to a disk device address. Consider the sample 1f070500 address mentioned in the message above.

1f: The first two hexadecimal digits identify the driver that generated the error. In this case, hexadecimal 1f is equivalent to decimal 31. **`lsdev`** tells us that 31 is the block major# for the **`sdisk`** SCSI driver.

07: The next two digits identify the SCSI card instance number. In this case, hexadecimal 07 is equivalent to decimal 07, which indicates that the error came from one of the **`/dev/dsk/c7txdx`** devices.

0: The next digit identifies the device target address. This error apparently came from device **`/dev/dsk/c7t0dx`**.

5: The next two digits identify the LUN number. This error came from **`/dev/dsk/c7t0d5`**.

00: The last two numbers identify device-specific options for tape drives. For disks, the last two digits should always be 00.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 3–12. SLIDE: Failed Disks: Symptoms (2 of 2)

### Failed Disks: Symptoms (2 of 2)

If a PV contains extents for a mirrored LV, and the PV fails for any reason, `vgdisplay` and `pvdisplay` will mark the disk "unavailable".

```
# vgdisplay -v vg01
   --- Physical volumes ---
   PV Name            /dev/dsk/c0t1d0
   PV Status           unavailable
```

If `vgdisplay` says "couldn't query disk …", that typically means that the VGRA/PVRA headers weren't accessible when the VG was activated.

```
# vgdisplay vg01
vgdisplay: Warning: couldn't query physical volume "/dev/dsk/c0t1d0":
The specified path does not correspond to physical volume attached
to this volume group
```

See if you can access the disk using the `dd` command

```
# dd if=/dev/rdsk/c0t1d0 of=/dev/null bs=1024k count=1024
```

## Student Notes

If a physical volume contains extents for a mirrored LV, and the PV fails for any reason, **`vgdisplay`** and **`pvdisplay`** will mark the disk "unavailable".

```
# vgdisplay -v vg01
  --- Physical volumes ---
  PV Name          /dev/dsk/c0t1d0
  PV Status        unavailable
  Total PE         4340
  Free PE          1785
  Autoswitch       On
```

If **`vgdisplay`** and **`pvdisplay`** say "couldn't query disk …", that typically means that the VGRA/PVRA headers weren't accessible when the volume group was activated.

```
# vgdisplay vg01
  vgdisplay: Warning: couldn't query physical volume
  "/dev/dsk/c0t1d0": The specified path does not correspond to
  physical volume attached to this volume group
```




© 2005 Hewlett-Packard Development Company, L.P.

See if you can access the disk using the **dd** command. If you get an error message, or if the **dd** command doesn't successfully read all 1024 records, the disk may be physically corrupted.

```
# dd if=/dev/rdsk/c0t1d0 of=/dev/null bs=1024k count=1024
1024+0 records in
1024+0 records out
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 3–13. SLIDE: Activating a Volume Group that Contains Failed Disks

### Activating a Volume Group that Contains Failed Disks

By default, activation quorum must be met in order to activate a VG.
If <51% of the disks in a VG are functional, activation fails.

```
# vgchange -a y vg01
vgdisplay: Warning: couldn't query physical volume "/dev/dsk/c0t1d0":
Quorum not present.  Activation failed.
```

You can override the quorum restriction using with the `-q n` option.

```
# vgchange -a y -q n vg01
vgdisplay: Warning: couldn't query physical volume "/dev/dsk/c0t1d0":
Quorum not present.  Proceeding…
```

Overriding quorum allows the VG to be activated, but non-redundant LVs may not be available. To make the VG fully functional, the failed disk must be recovered.

## Student Notes

By default, activation quorum must be met in order to activate a volume group. If less than 51% of the disks in a volume group are functional, activation fails.

```
# vgchange -a y vg01
vgdisplay: Warning: couldn't query physical volume "/dev/dsk/c0t1d0":
Quorum not present.  Activation failed.
```

If you want to activate the volume group nonetheless, you can override this quorum restriction with the **`vgchange -q n`** option.

```
# vgchange -a y -q n vg01
vgdisplay: Warning: couldn't query physical volume "/dev/dsk/c0t1d0":
Quorum not present.  Proceeding…
```




© 2005 Hewlett-Packard Development Company, L.P.

If the vg00 quorum check fails during the boot process, a similar option is available on the **`hpux`** (PARISC) and **`hpux.efi`** (IPF) kernel loader utilities that are responsible for activating vg00 during the boot process.

PARISC: `ISL> hpux -lq`
IPF: `HPUX> boot vmunix -lq`

Overriding quorum allows a volume group to be activated, but non-redundant LVs may not be available. To make the volume group fully functional, the failed disk(s) must be recovered.

---

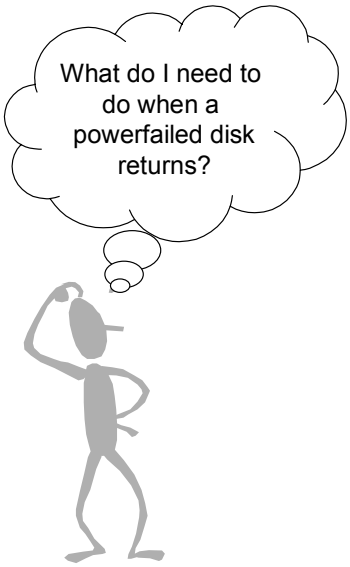
WARNING! Technically, LVM will allow you to create, remove, and extend logical volumes when a volume group is missing one or more physical volumes, as long as running quorum is present. However, doing so may complicate recovery later. When a disk fails, it is best to troubleshoot and restore the failed disk before you make other changes to your volume group configuration.

---


**© 2005 Hewlett-Packard Development Company, L.P.**

## 3–14. SLIDE: Restoring a Powerfailed Disk

### Restoring a Powerfailed Disk




If a disk powerfails *after* activation, no administrator intervention is required when the disk returns.

If a disk is powerfailed *before* activation, execute `ioscan`, `vgchange`, and `vgsync` when the disk returns.

```
# ioscan -fC disk
# vgchange -a y vg01
# vgsync vg01
```

## Student Notes

If a disk powerfails in an already activated volume group, then later returns, LVM should recognize the returned disk automatically, synchronize mirrors if necessary, and make the logical volumes on the disk available.

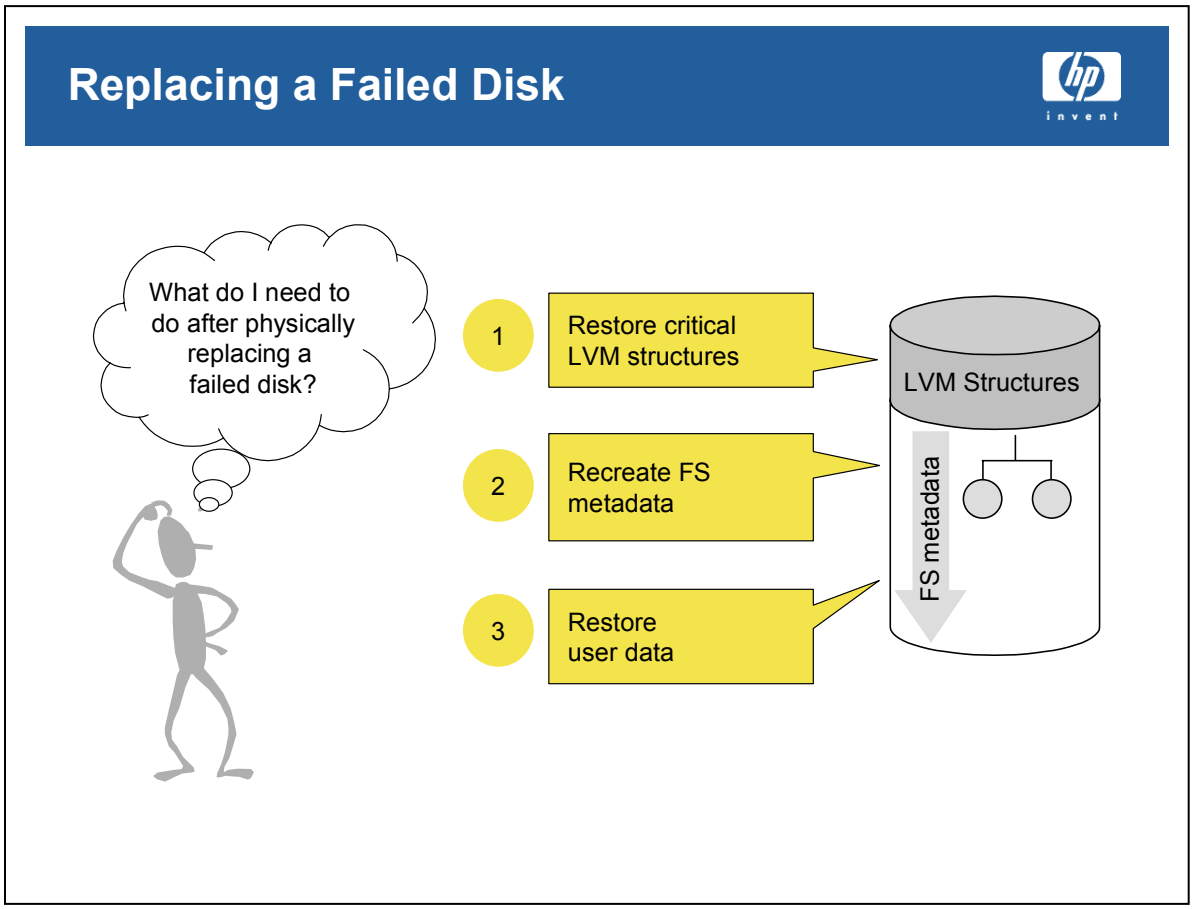
If a disk is powerfailed when a volume group is activated, it will be necessary to execute **`ioscan`**, **`vgchange`**, and **`vgsync`** when the disk returns. Note that it isn't necessary to deactivate the volume group; executing **`vgchange -a y`** on an already-activated volume group simply causes LVM to re-scan the disks.

```
# ioscan -fC disk
# vgchange -a y vg01
# vgsync vg01
```




© 2005 Hewlett-Packard Development Company, L.P.

## 3–15. SLIDE: Replacing a Failed Disk




## Student Notes

If a physical volume is corrupted or damaged, it may be necessary to replace the disk. After replacing the disk, you will have to recreate or restore all the structures and data on the disk, including:

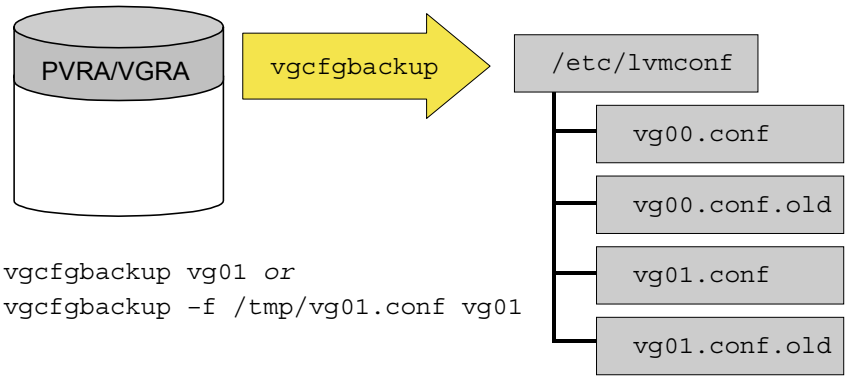
- LVM PVRA/VGRA headers
- File system metadata
- User and application data in your logical volumes


© 2005 Hewlett-Packard Development Company, L.P.

## 3–16. SLIDE: Backing up the PVRA/VGRA

### Backing up the PVRA/VGRA

- `vgcfgbackup` may be used to backup the PVRA/VGRA to `/etc/lvmconf`.
- `vgcfgbackup` runs automatically when the LVM configuration is modified.
- `vgcfgbackup` may be run directly by the administrator any time.




```
# vgcfgbackup vg01 or
# vgcfgbackup -f /tmp/vg01.conf vg01
```

## Student Notes

Normal backup utilities backup files in your file systems, but do not backup the LVM PVRA/VGRA headers. LVM includes a pair of utilities that are specifically designed to backup and restore these structures.

The **`vgcfgbackup`** command stores two backup copies of each volume group's PVRA/VGRA information in the **`/etc/lvmconf`** directory. A volume group's most recent backup is stored in **`/etc/lvmconf/`*`vgname`*`.conf`**. An archive of the volume group's configuration prior to the most recent LVM configuration change is stored in **`/etc/lvmconf/`*`vgname`*`.conf.old`**.

By default, **`vgcfgbackup`** runs every time the volume group configuration is modified. Most LVM commands support **`a -A n`** option, which skips the automated configuration backup. If you choose to create, extend, or reduce logical volumes while a disk is missing from the volume group (not recommended!) the **`-A n`** option will be required since **`vgcfgbackup`** fails when physical volumes are missing.




© 2005 Hewlett-Packard Development Company, L.P.

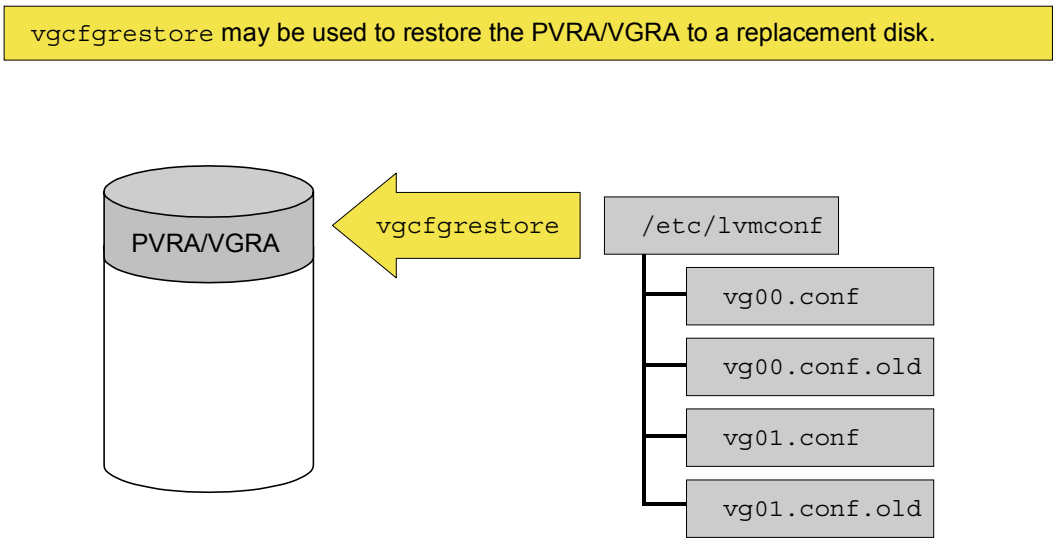
You can manually run the **`vgcfgbackup`** command, too.

| | |
|---|---|
| `# vgcfgbackup vg01` | (stores the backup in **`/etc/lvmconf/`**) |
| `# vgcfgbackup -f /tmp/vg01.conf vg01` | (stores the backup elsewhere) |


**© 2005 Hewlett-Packard Development Company, L.P.**

## 3–17. SLIDE: Restoring the PVRA/VGRA

### Restoring the PVRA/VGRA



`vgcfgrestore` may be used to restore the PVRA/VGRA to a replacement disk.



```
# vgcfgrestore -n vg01 /dev/rdsk/cxtxdx or
# vgcfgrestore -f /tmp/vg01.conf /dev/rdsk/cxtxdx
```

## Student Notes

When you replace a failed LVM disk, use the **`vgcfgrestore`** command to restore the PVRA/VGRA headers from the **`/etc/lvmconf`** directory to the replacement disk. The first example on the slide restores vg01's default PVRA/VGRA backup to a replacement disk at location ***`cxtxdx`***. The second restores a PVRA/VGRA backup from a non-default directory to the ***`cxtxdx`*** replacement disk. Note that the replacement disk's raw disk device file must be used in both cases.

```
# vgcfgrestore -n vg01 /dev/rdsk/cxtxdx or
# vgcfgrestore -f /tmp/vg01.conf /dev/rdsk/cxtxdx
```

The **`vgcfgrestore`** command is just one of several commands required to fully restore LVM headers on a replacement disk. The next slide describes the replacement procedure in detail.




© 2005 Hewlett-Packard Development Company, L.P.

## 3–18. SLIDE: Restoring the PVRA/VGRA: Complete Procedure

### Restoring the PVRA/VGRA: Complete Procedure

The procedure below describes the complete process for restoring PVRA/VGRA headers

1. Force the OS to recognize the replacement disk
   `# ioscan -fC disk; insf -C disk`
2. Restore the PVRA/VGRA headers to the replacement disk
   `# vgcfgrestore -n vg01 [-o /dev/dsk/cwtwdw] /dev/rdsk/cxtxdx`
3. Force LVM to recognize the replacement disk
   `# vgchange -a y vg01`
4. Extra steps required if the replacement disk is at a different hardware location:
   ```
   # vgchange -a n vg01
   # vgexport -s -v -m /etc/lvmconf/vg01.map vg01
   # vgimport -s -v -m /etc/lvmconf/vg01.map vg01
   # vgchange -a y vg01
   ```
5. Extra steps required if the replacement disk is larger than the original disk:
   ```
   # pvmove -n /dev/vg01/data /dev/dsk/cxtxdx /dev/dsk/cytydy
   # vgreduce vg01 /dev/dsk/cxtxdx
   # pvcreate -f /dev/rdsk/cxtxdx
   # vgextend vg01 /dev/dsk/cxtxdx
   ```
6. Proceed with file system and user data recovery …

## Student Notes

The cookbook below describes the complete procedure required to replace a failed physical volume.

1. Force the OS to recognize the replacement disk.

   ```
   # ioscan -fC disk; insf -eC disk
   ```

2. Restore the PVRA/VGRA headers to the replacement disk (**/dev/rdsk/*cxtxdx*** in this case). If the replacement disk has a different hardware path than the original disk, specify the old disk's hardware path (**/dev/dsk/*cwtwdw*** in this case) with the **-o** option, and be sure to follow the instructions in step 4 below. If the replacement disk is at the same hardware path as the original disk, you can leave off the **-o** and skip step 4.

   ```
   # vgcfgrestore -n vg01 [-o /dev/dsk/cwtwdw] /dev/rdsk/cxtxdx
   ```

3. Force LVM to recognize the replacement disk.

   ```
   # vgchange -a y vg01
   ```


© 2005 Hewlett-Packard Development Company, L.P.

4. Extra steps are required if the replacement disk is at a different hardware location. The old hardware path must be removed from **/etc/lvmtab** via the **vgexport** command, and the new hardware path must be added via the **vgimport** command. Note that file systems in the volume group must be unmounted before you deactivate the volume group.

```
# vgchange -a n vg01
# vgexport -s -v -m /etc/lvmconf/vg01.map vg01
# mkdir /dev/vg01
# mknod /dev/vg01/group c 64 0x010000
# vgimport -s -v -m /etc/lvmconf/vg01.map vg01
# vgchange -a y vg01
```

5. Extra steps are required, too, if the replacement disk is larger than the original disk. Move all of the extents off the disk, take the disk out of the volume group, then add it back in again. If you wish, you can move the logical volumes back to the replacement disk after the **vgextend**.

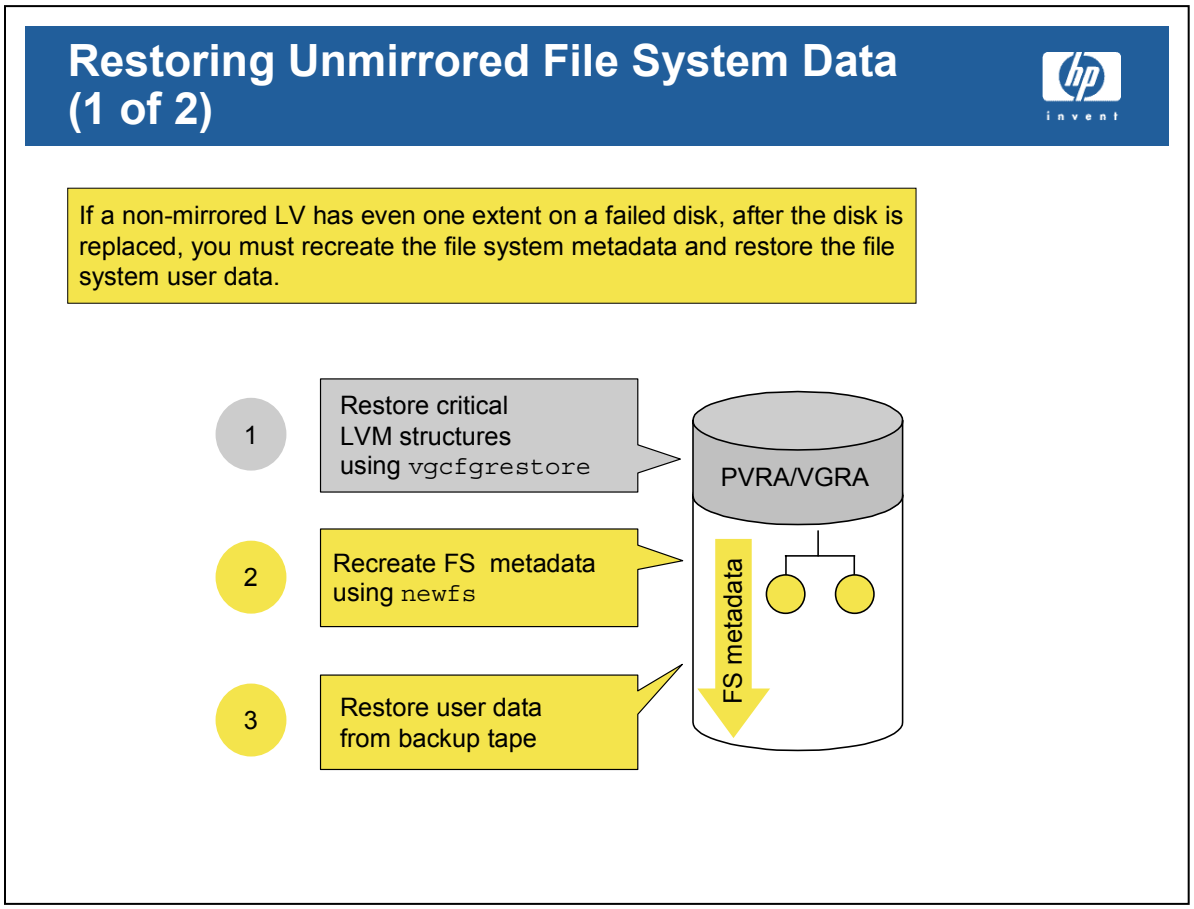
```
# pvmove -n /dev/vg01/data /dev/dsk/cxtxdx /dev/dsk/cytydy
# vgreduce vg01 /dev/dsk/cxtxdx
# pvcreate -f /dev/rdsk/cxtxdx
# vgextend vg01 /dev/dsk/cxtxdx
```

6. Proceed with file system and user data recovery. The procedure described here only restores the LVM headers. File system metadata and user data must be restored using separate procedures described in the pages that follow.




© 2005 Hewlett-Packard Development Company, L.P.

## 3–19. SLIDE: Restoring Unmirrored File System Data (1 of 2)




## Student Notes

If a non-mirrored logical volume has even one extent on a failed disk, the file system metadata and user data must be recreated after the disk has been replaced.


© 2005 Hewlett-Packard Development Company, L.P.

## 3–20. SLIDE: Restoring Unmirrored File System Data (2 of 2)

### Restoring Unmirrored File System Data (2 of 2)

- Use the `newfs` command to re-create the file system metadata structures.
- Use `frecover` or your preferred tape recovery solution to recover your user data.

1. Re-create the file system metadata
   `# newfs /dev/vg01/rdata`
2. Re-mount the file system
   `# mount /data`
3. Restore user data from tape
   `# frecover -rv`

## Student Notes

Use the **`newfs`** command to recreate the file system metadata structures, then mount the file system and restore your user data from tape.

1. Re-create the file system metadata.

   ```
   # newfs /dev/vg01/rdata
   ```

2. Re-mount the file system.

   ```
   # mount /data
   ```

3. Restore user data from tape.

   ```
   # frecover -rv
   ```

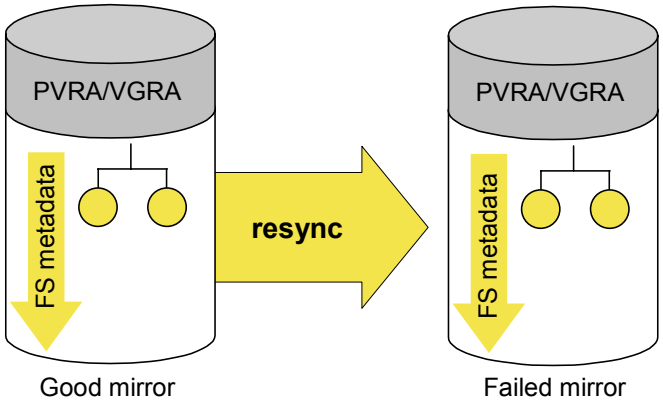



© 2005 Hewlett-Packard Development Company, L.P.

## 3–21. SLIDE: Restoring Mirrored File System Data (1 of 2)

### Restoring Mirrored File System Data (1 of 2)

File system metadata and user data on mirrored LVs may be restored by resyncing from the remaining mirror. There is no need to `newfs` or `frecover` (though `vgcfgrestore` is still required)!




## Student Notes

Mirrored file system metadata and user data may be restored by resyncing from the remaining mirror. There is no need to **`newfs`** or **`frecover`** (though **`vgcfgrestore`** is still required)!


**© 2005 Hewlett-Packard Development Company, L.P.**

## 3–22. SLIDE: Restoring Mirrored File System Data (2 of 2)

### Restoring Mirrored File System Data (2 of 2)

"Stale" extents will be resync'ed automatically if …
- The VG is deactivate/reactivated *or*
- The PV in an activated VG simply powerfailed, then returned

"Stale" extents must be manually resync'ed with `vgsync` or `lvsync` if …
- A PV is replaced without deactivating/reactivating the VG
- A PV is powerfailed when a VG is initially activated, but later returns

1. Ensure the VG is activated
```
# vgchange -a y vg01
```
2. Synchronize stale extents
```
# vgsync vg01 or
# lvsync /dev/vg01/data
```
3. Verify all extents are synchronized
```
# lvdisplay -v /dev/vg01/data
LV Status:  available/stale
   --- Logical extents ---
LE     PV1                PE1    Status1  PV2                PE2     Status2
00000  /dev/dsk/c0t1d0    00000  current  /dev/dsk/c0t2d0    00000   current
00001  /dev/dsk/c0t1d0    00001  current  /dev/dsk/c0t2d0    00001   current
00002  /dev/dsk/c0t1d0    00002  current  /dev/dsk/c0t2d0    00002   stale
```

## Student Notes

After a mirrored physical volume has been replaced and **`vgcfgrestore`**d, the mirrors must be resync'd.

"Stale" extents will be resync'ed *automatically* if …

- The VG is deactivate/reactivated *or*
- The PV simply powerfailed, then returned

"Stale" extents must be *manually* resync'ed with **`vgsync`** or **`lvsync`** if …

- A PV is replaced without deactivating/reactivating the VG
- A PV is powerfailed when a VG is initially activated, but later returns

In either case, your users can continue accessing the logical volume while the resync proceeds.




© 2005 Hewlett-Packard Development Company, L.P.

The commands required to manually resynchronize a mirrored logical volume are described below.

1. Ensure the volume group is activated.

```
# vgchange -a y vg01
```

2. Synchronize stale extents. **`vgsync`** synchronizes all mirrored logical volumes in the volume group. This may monopolize your buses, so some administrators prefer to manually synchronize one logical volume at a time with **`lvsync`**.

```
# vgsync vg01 or
# lvsync /dev/vg01/data
```

3. Verify all extents are synchronized.

```
# lvdisplay -v /dev/vg01/data
  LV Status:  available/stale
  --- Logical extents ---
  LE    PV1             PE1   Status1  PV2             PE2    Status2
  00000 /dev/dsk/c0t1d0 00000 current  /dev/dsk/c0t2d0 00000  current
  00001 /dev/dsk/c0t1d0 00001 current  /dev/dsk/c0t2d0 00001  current
  00002 /dev/dsk/c0t1d0 00002 current  /dev/dsk/c0t2d0 00002  stale
```

Synchronization may take some time, but eventually, you should see all of the extents marked “current” on both physical volumes.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 3–23. SLIDE: Removing Corrupted Physical Volumes

### Removing Corrupted Physical Volumes

If a PV fails and you have no intention of replacing it, you can forcefully remove it from `/etc/lvmtab` using the procedure below. Use this procedure with care!

1. Obtain a list of LVs in the VG and determine which ones touch the non-responsive disk
```
# vgdisplay -v vg01
# lvdisplay -v /dev/vg01/data | grep "???"
```
2. Remove these LVs from VG
```
# lvremove -A n /dev/vg01/data
```
3. Forcefully remove pointers to non-responsive disks from other disks' VGRAs
```
# vgreduce -f vg01
```
4. Rebuild `/etc/lvmtab` to reflect the new configuration
```
# mv /etc/lvmtab /etc/lvmtab.bkp
# vgscan
```
5. Back up the new configuration
```
# vgcfgbackup vg01
```
6. If you removed any LVs in step 2, recreate them, and restore their data from tape
```
# lvcreate -L 32 -n data vg01
# newfs /dev/vg01/rdata
# mount -a
# frecover -i /data -xv
```

## Student Notes

When a physical volume fails, the best solution is to replace the disk and use the **`vgcfgrestore`** recovery procedures described on the preceding slides. However, if you choose not to replace the failed disk, or can't replace the failed disk, the slide above describes an alternative procedure that removes the failed physical volume from the on-disk LVM structures and the **`/etc/lvmtab`** file.

WARNING! This procedure should only be used as a last resort when a failed physical volume can't be replaced and restored using the normal **`vgcfgrestore`** mechanism discussed previously.




© 2005 Hewlett-Packard Development Company, L.P.

Follow the steps below to remove a physical volume from a volume group if you choose not to replace the disk using **vgcfgrestore**.

1. Obtain a list of logical volumes in the volume group and determine which ones touch the non-responsive disk.

   ```
   # vgdisplay -v vg01
   # lvdisplay -v /dev/vg01/data | grep "???"
   ```

2. Remove the logical volumes that have "???" extents from volume group. The **-A n** skips the automated **vgcfgbackup** that occurs after every LVM configuration change. **vgcfgbackup** would fail anyway since a disk is missing in the volume group.

   ```
   # lvremove -A n /dev/vg01/data
   ```

3. Forcefully remove pointers to non-responsive disks from other disks' VGRAs.

   ```
   # vgreduce -f vg01
   ```

   If the **vgreduce -f** command doesn't work, or if **vgdisplay** still shows that Cur PV and Act PV disagree then use the following steps to **vgexport** and vgicmport the volume group. This procedure isn't recommended in vg00.

   A. Determine the volume group's minor number, and list the physical volumes in the volume group.

      ```
      # ll /dev/vg01/group
      # vgdisplay -v vg01
      ```

   B. Activate the volume group.

      ```
      # vgchange -a y vg01
      ```

   C. Export the volume group.

      ```
      # vgexport -m /etc/lvmconf/vg01.map vg01
      ```

   D. Recreate the volume group directory and device file. Use the minor number you recorded above.

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

   E. Import the volume group. Explicitly list the physical volumes in the volume group; *exclude* the physical volume you wish to remove.

      ```
      # vgimport -m /etc/lvmconf/vg01.conf vg01 /dev/dsk/c0t0d0
      ```

   F. Retry the **vgreduce** command.

      ```
      # vgreduce -f vg01
      ```


**© 2005 Hewlett-Packard Development Company, L.P.**

4. Rebuild **/etc/lvmtab** to reflect the new configuration.

```
# mv /etc/lvmtab /etc/lvmtab.bkp
# vgscan
```

5. Back up the new configuration.

```
# vgcfgbackup vg01
```

6. If you removed any logical volumes in step 2, recreate them, and restore their data from tape.

```
# lvcreate -L 32 -n data vg01
# newfs /dev/vg01/rdata
# mount -a
# frecover -i /data -xv
```




© 2005 Hewlett-Packard Development Company, L.P.

## 3–24. SLIDE: Removing Corrupted Volume Groups

### Removing Corrupted Volume Groups

If a VG becomes unrecoverable for some reason, or if you want to remove a VG with minimal effort, simply export it!

1. Unmount any affected file systems
   `# umount /data`
2. Remove affected file systems from `/etc/fstab`
   `# vi /etc/fstab`
3. Deactivate the VG
   `# vgchange -a n vg01`
4. Export the VG
   `# vgexport vg01`
5. Clobber any remaining data on the affected disks
   `# dd if=/dev/zero of=/dev/rdsk/cxtxdx bs=8192`

## Student Notes

If a volume group becomes unrecoverable for some reason, or if you want to remove a VG with minimal effort, simply export it!

1. Unmount any affected file systems.

   ```
   # umount /data
   ```

2. Remove affected file systems from **`/etc/fstab`**.

   ```
   # vi /etc/fstab
   ```

3. Deactivate the volume group.

   ```
   # vgchange -a n vg01
   ```

4. Export the volume group.

   ```
   # vgexport vg01
   ```

5. Clobber any remaining data on the affected disks.

   ```
   # dd if=/dev/zero of=/dev/rdsk/c0t1d0 bs=8192
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 3–25. SLIDE: Module Objectives

### Module Objectives

You should now be able to do the following:

- Describe the critical structures in an LVM environment.
- Diagnose and restore missing LVM device files.
- Diagnose and rebuild a missing or corrupt `/etc/lvmtab` file.
- Update the kernel LVM structures when missing disks return.
- Identify failed LVM disks.
- Backup LVM data structures using `vgcfgbackup`.
- Restore LVM data structures using `vgcfgrestore`.
- Restore file system and user data when a non-mirrored data disk fails.
- Restore file system and user data when a mirrored data disk fails.
- Remove a corrupted physical volume from a volume group.
- Remove a corrupted volume group from a host.
- List critical recovery information that should be recorded in a system log book.

## Student Notes

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

- Describe the critical structures in an LVM environment.
- Diagnose and restore missing LVM device files.
- Diagnose and rebuild a missing or corrupt **`/etc/lvmtab`** file.
- Update the kernel LVM structures when missing disks return.
- Identify failed LVM disks.
- Backup LVM data structures using **`vgcfgbackup`**.
- Restore LVM data structures using **`vgcfgrestore`**.
- Restore file system and user data when a non-mirrored data disk fails.
- Restore file system and user data when a mirrored data disk fails.




© 2005 Hewlett-Packard Development Company, L.P.

- Remove a corrupted physical volume from a volume group.
- Remove a corrupted volume group from a host.
- List critical recovery information that should be recorded in a system log book.


**© 2005 Hewlett-Packard Development Company, L.P.**

# 3–26. LAB: Recovering LVM Disks and Structures

## Directions

Carefully follow the instructions below. If you are using an Integrity system for this lab, check with your instructor for the appropriate disks to use for your new volume group.

## Part 1: Preliminary Steps

1. Configure a two-disk vg01 volume group. If you already have a vg01 volume group, you can use it in this lab, but be sure that the volume group has only two disks, and no logical volumes.

2. Create a 16M non-mirrored logical volume called "nonmirrored" on the first disk in the volume group. Create a VxFS file system in the logical volume, and mount the file system on **`/nonmirrored`**, and copy a few files to the file system from **`/usr/bin`**.

3. Create a 16M mirrored logical volume called "mirrored". Create a VxFS file system in the logical volume, mount the file system on /mirrored, and copy a few files to the file system from **`/usr/bin`**.




© 2005 Hewlett-Packard Development Company, L.P.

## Part 2: Recreating Missing Device Files

1. Remove all the device files in the **`/dev/vg01`** directory.

```
# rm /dev/vg01/*
```

2. How are the following commands impacted?

```
# ls /mirrored
# vgdisplay -v vg01
# lvextend -L 32 /dev/vg01/mirrored
```

3. Recreate the volume group device file.

4. You should be able to **`vgdisplay -v vg01`** now, but something is missing from the output. What is it?

5. Recreate the logical volume device files, too.

6. Verify the creation of the logical volume device files by executing step 4 again.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 3: Rebuilding a Missing /etc/lvmtab

What happens when the **/etc/lvmtab** file is lost or corrupted? In this part of the lab you will find out!

1. Move **/etc/lvmtab** to **/etc/lvmtab.bkp**

2. What impact does this have on the following commands?

```
# ls /mirrored
# vgdisplay -v vg01
# lvextend -L 32 /dev/vg01/mirrored
```

3. What command can you use to recreate the **/etc/lvmtab** file? Make it so! How does this command determine which disks belong to which volume groups?

4. Verify your recovery succeeded. How can you view the contents of the **/etc/lvmtab** file?




© 2005 Hewlett-Packard Development Company, L.P.

## Part 4: What Happens when a Disk Fails?

This portion of the lab demonstrates the effect of a downed disk on volume group activation. We will simulate a disk failure by overwriting the LVM headers on the first disk in your volume group.

1. Write 64M of zeroes to the first disk in your volume group. This should be more than adequate to clobber the LVM headers.

   ```
   # dd if=/dev/zero of=/dev/rdsk/cxtxdx bs=8192 count=8192
   ```

2. Does **`vgdisplay`** indicate that there is a problem? Explain!

3. Unmount your **`/mirrored`** and **`/nonmirrored`** file systems, deactivate the volume group, then try to reactivate. Does this succeed? Explain!

4. What can you do to activate the volume group without quorum? Make it so!

5. Now does **`vgdisplay`** indicate that there is a problem?

6. Can you access your logical volumes? Attempt to mount **`/dev/vg01/nonmirrored`** and **`/dev/vg01/mirrored`**. What happens? Why?


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 5: Replacing a Failed Disk

Now let's see if you can restore the failed LVM disk.

1. Use **`vgcfgrestore`** to restore the LVM headers to your failed disk.

2. Does LVM recognize the fact that the headers were restored? Execute **`vgdisplay -v`** to find out.

3. Do whatever is necessary to force LVM to recognize that the disk has been recovered – without taking the volume group offline again.

## Part 6: Restoring Logical Volumes

You restored the LVM headers, but your logical volumes still require some attention.

1. View the **`lvdisplay -v`** output for the **`/dev/vg01/nonmirrored`** logical volume. What is the LV Status?

2. What is the status of the file system in the logical volume? Try running **`fsck`** to find out!

3. Recreate the file system metadata structures in **`/dev/vg01/rnonmirrored`**, then remount the file system. How can you restore the data that was in this file system?




© 2005 Hewlett-Packard Development Company, L.P.

4. Now view the **`lvdisplay -v`** output for the **`/dev/vg01/mirrored`** logical volume. What is the LV Status?

5. Recall that you already successfully mounted the file system in this logical volume, even while one of the mirrors was unavailable. Did the user data survive the disk failure, too?

6. Although you probably discovered that the data in **`/dev/vg01/mirrored`** survived the disk failure, the mirror on the failed drive is no longer participating in the logical volume updates. Long term, you will probably want to resynchronize the logical volume. Make it so!

7. Wait a minute or two, then re-run the **`lvdisplay`** command to verify that the command succeeded.

## Part 7: Cleanup

1. Before you continue on to the next chapter, remove the logical volumes that you created during this lab. However, you can keep the **`vg01`** volume group for use in the next lab.

```
# umount /mirrored
# umount /nonmirrored
# lvremove -f /dev/vg01/mirrored
# lvremove -f /dev/vg01/nonmirrored
```


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 4 — Troubleshooting a PA-RISC System Boot Disk

## Objectives

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

- Describe PA-RISC system startup
- Describe the boot disk structure
- Describe potential boot failures
- Understand the **`LABEL`** file and the **`BDRA`**
- Understand the operation of the **`lvlnboot`** command
- Describe the operation of the recovery shell and its capabilities
- Understand the implications of missing or corrupt binary startup files
- Describe the recovery process for a missing kernel




© 2005 Hewlett-Packard Development Company, L.P.

## 4–1. SLIDE: Module Objectives

### Module Objectives

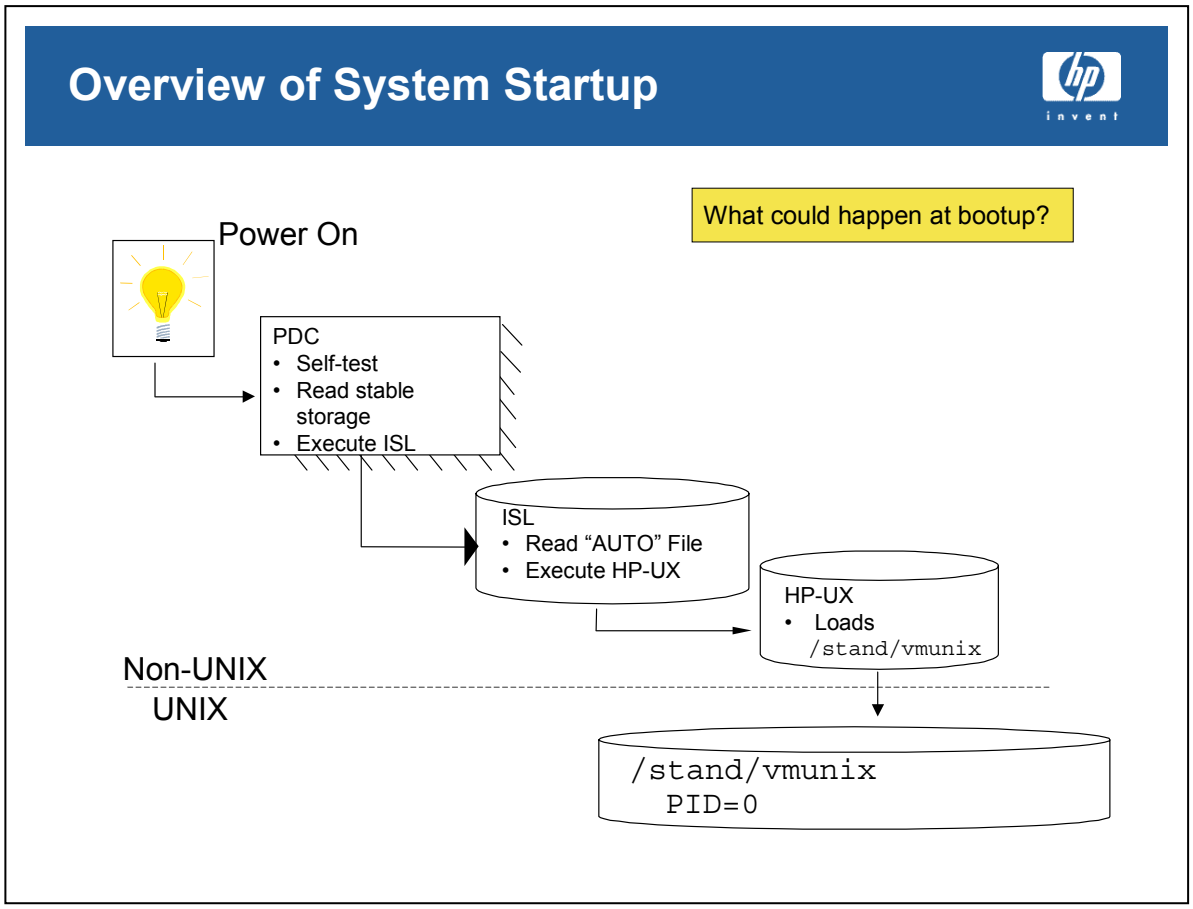
By the end of the module, you should be able to…

- Describe PA-RISC system startup
- Describe the boot disk structure
- Describe potential boot failures
- Understand the `LABEL` file and the `BDRA`
- Understand the operation of the `lvlnboot` command
- Describe the operation of the recovery shell and its capabilities
- Understand the implications of missing or corrupt binary startup files
- Describe the recovery process for a missing kernel

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 4–2. SLIDE: Overview of System Startup




## Student Notes

The boot process is started when the system is power cycled. Afterwards, a series of steps is performed:

### 1. Processor Dependent Code (PDC)

The processor-dependent code (PDC) is code on the boot-ROM chip. It is responsible for performing hardware self tests, reading stable storage, and loading the ISL from the LIF Area. The behavior of the PDC code will vary between the different server and workstation families.

The PDC hardware self-test results are reported as a 4 digit HEX code either on the console (if your system has an MP (Maintenance Processor), or access port, or to a display on the system. For earlier workstations, the results are displayed to a bank of LEDs located on the front panel. If your system fails to boot, record the code or LED pattern and report it to your HP service professional. Only after all tests have passed will you see a message on the console that the search for an operating system has begun.

At this stage, the boot can be interrupted and you can interact with the PDC environment. The facilities available will vary depending upon your system.




© 2005 Hewlett-Packard Development Company, L.P.

## 2. Stable Storage

Stable storage is non-volatile memory that is read by the PDC code during the initial stages of the boot process. Its role is to provide information pertaining to paths and auto flags. It also stores low-level diagnostic information that HP service personnel can access to investigate problems. The paths will reflect hardware addresses of your boot disk, console, keyboard, etc. The auto flags indicate if you should attempt booting the system automatically, as well as automatically searching for bootable devices.

If stable storage is corrupt, your system may not boot. This can be fixed by interacting directly with the PDC menus in interactive mode. Common commands that can be used are as follows:

- **`search`** Search for boot devices.
- **`path`** Display or modify a path.
- **`boot`** Boot from specified path.
- **`config`** Display or modify an auto flag and set other parameters.

## 3. Initial System Loader (ISL)

ISL is loaded and executed after the system successfully completes the self-test and initialization. ISL defines, loads, and passes control to one of several possible secondary loaders, the most common of which is HPUX.

When you boot the system manually, you can interact with the ISL. The ISL allows you to perform tasks such as loading the HP-UX kernel into memory, changing the primary or alternate boot paths, and turning **`autoboot`** on or off.

The following list describes several commonly used ISL commands:

- **`primpath`** Modify the primary boot path.
- **`altpath`** Modify the alternate boot path.
- **`conspath`** Modify the console path.
- **`autoboot`** Enable or disable the autoboot sequence.
- **`?`** or **`help`** List commands and available utilities.
- **`listf`** or **`ls`** List available utilities.
- **`autosearch`** Enable or disable the **`autosearch`** sequence.
- **`lsautofl`** List contents of the **`autoexecute`** file.
- **`display`** Display the primary boot, alternate boot, and console paths.

Note that some of these functions replicate functions in PDC. Earlier PDC versions were much more limited and, for instance, were not able to set the paths.

Remember that the primary boot path is generally set to the system root disk. The alternate boot path is generally set as a mirror for the boot disk, if mirroring is in use, or it may point to a tape device used for installations and updates. The console path is set to the hardware address of the system console.


**© 2005 Hewlett-Packard Development Company, L.P.**

For more information, see man page **`isl(1M)`**.

## 4. HPUX

HPUX is called the secondary loader. A secondary loader is software that loads the operating system into memory. It is needed as ISL is an operating system independent loader. It is not exclusively used with HP-UX; consequently, it does not know how to load an HP-UX kernel from a UNIX file system. HPUX loads **`/stand/vmunix`**. It supports the operations summarized below:

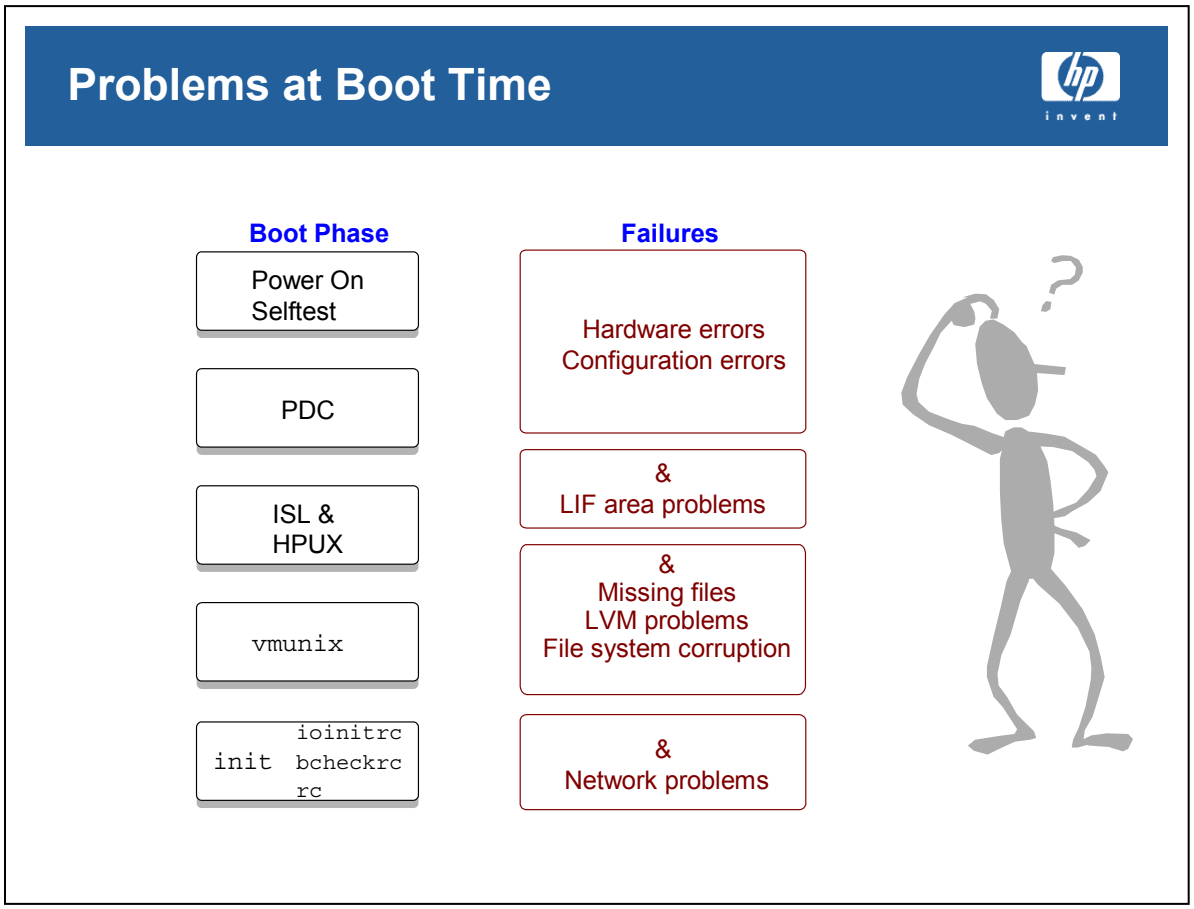
| | |
|---|---|
| **`boot`** | Loads an object file from an HP-UX file system or raw device and transfers control to the loaded image. (Note: The boot operation is position-dependent.) This is the default. |
| **`ll`** | Lists the contents of HP-UX directories in a format similar to **`ls -aFln`**.<br>(See **`ls(1)`**; **`ls`** only works on a local disk with an HFS file system.) |
| **`ls`** | Lists the contents of HP-UX directories.<br>(See **`ls(1)`**; **`ls`** only works on a local disk with an HFS file system.) |
| **`show autofile`** | Displays the contents of the **`autoexecute`** file. |
| **`set autofile`** | Changes the contents of the **`autoexecute`** file to that specified by string. |
| **`-v`** | Displays the release and version numbers of the **`hpux`** utility. |
| **`restore`** | Recovers the system from a properly formatted bootable tape. (Series 700 specific)<br>This form of recovery is not recommended these days. Instead, the use of Ignites recovery tapes is preferred. These recovery tapes were fully usable only on identical disk models. |

HP-UX commands can be given interactively from the keyboard, or provided in an **`isl autoexecute`** file.




© 2005 Hewlett-Packard Development Company, L.P.

## 4–3. SLIDE: Problems at Boot Time




## Student Notes

The system can fail during any of these boot phases. Over the following modules, we are going to cover the possible failure situations.

### Power on Self-Test and PDC

Failures during the earliest stages are most likely caused by hardware problems.

If the system fails to even power up, the first thing to check is the power source: Is the system plugged in? This is a much more common problem than would be expected. A common scenario: a member of the cleaning or maintenance staff unplugged the system, because they wanted to use that electrical socket. If there is adequate power to the system, it will most likely be necessary to contact an HP hardware engineer.

Once the system starts to boot, status messages are displayed 1) on the console (where an MP or access port is installed), 2) to a 4 digit HEX display, or 3) on the earlier workstations, a series of LEDs on the front of the system (possibly behind a drop down panel). Hardware engineers can use these codes to help locate the problem.

At the end of this phase, messages should be displayed to the console. One possible reason for failure to see these messages is configuration of the console path. PDCs stable storage


© 2005 Hewlett-Packard Development Company, L.P.

may have been configured so that the console messages are being sent to a different location. For instance, workstations typically have their console configured to be the graphics display, but it can also be moved to one of the serial ports.

Once PDC has displayed the boot time messages to the console, the next step is to load ISL from the LIF area.

## ISL and HP-UX

Here, possible problems include hardware errors such as disk drive failures or other issues in the I/O subsystem, such as SCSI cabling, card failures, or configuration errors, such as having the wrong boot paths configured. Also, since ISL needs to be loaded from the LIF area of the disk, problems with the LIF area would also prevent ISL from being loaded.

If the boot path is completely misconfigured, PDC will indicate that the path is not valid.

```
Main Menu: Enter command or menu > boot 1/2/3/4
Interact with IPL (Y, N, or Cancel)?> n

Booting...
Not a Valid Boot Path.
```

If the path makes sense but does not point to a real device, the error is less obvious.

```
Main Menu: Enter command or menu > boot 0/0/1/1.14
Interact with IPL (Y, N, or Cancel)?> n

Booting...
Cannot find ENTRY_TEST.

Failed to initialize.

ENTRY_INIT
Status = -7

FFFFFFF9 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000012 00000000 00000000 454E4841 00000000 00000000 454E4841
00000000 00000000 00000000 00000000 00000200 00000000 00000000 00000000

Failed to initialize.
```

With errors like these, the PDC **`search`** command lets you look for bootable devices, or even for those that contain an ISL.

```
Main Menu: Enter command or menu > sea

Searching for potential boot device(s)...
This may take several minutes.

To discontinue search, press any key (termination may not be
immediate).


Path#  Device Path (dec)  Device Path (mnem)  Device Type
```




© 2005 Hewlett-Packard Development Company, L.P.

```
-----  -----------------  ------------------  -----------
P0     0/0/0/0            lan.10.100.0.254   LAN Module
P1     0/0/1/1.15         intscsia.15         Random access media
P2     0/0/2/0.2          extscsib.2          Random access media
P3     0/0/2/0.0          extscsib.0          Sequential access media
P4     0/0/2/1.15         intscsib.15         Random access media

Main Menu: Enter command or menu >
```

If the device is valid, but the LIF area is missing or unrecognizable, PDC will give an error relating to a bad magic[1] number in the LIF area.

```
Main Menu: Enter command or menu > boot 0/0/2/1.15
Interact with IPL (Y, N, Q)?> n

Booting...
Boot IO Dependent Code (IODC) revision 1

IPL error: bad LIF magic.

Error Reading IPL
```

If the LIF area is okay, but ISL is missing, the IPL error will address IPL rather than LIF.

```
Main Menu: Enter command or menu > boot 0/0/2/0.2
Interact with IPL (Y, N, or Cancel)?> n

Booting...
Boot IO Dependent Code (IODC) revision 1

IPL error: bad IPL checksum.

Error Reading IPL
```

Once ISL has been loaded, you should see an ISL prompt. From the ISL prompt, it is possible to check for the HPUX secondary loader and the contents of the **AUTO** file.

```
Booting...
Boot IO Dependent Code (IODC) revision 1


HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL> ls
```

[1] Magic number is a technique frequently used in UNIX to recognize file formats. You simply store a chosen bit of information, typically a number, at a specific location in a file. If you want to know the type of file, go to that location and look for the number. See the file **/etc/magic** for some examples. LIF stores 0x8000 in the first 2 bytes of the file.


**© 2005 Hewlett-Packard Development Company, L.P.**

```
    Utilities on this system are:

filename   type   start    size     created
=====================================================
HPUX       -12928 904      848      02/05/10 16:25:44

ISL> lsautofl

   Auto-execute file contains:
    hpux

ISL>
```

If the **AUTO** file is missing or unusable, a manual boot of the system should work. Once the system is up, the **mkboot(1m)** command can then be used to rebuild the **AUTO** file.

For example: **mkboot –a "hpux"  /dev/rdsk/cxtydz**

If ISL or HP-UX is unusable, it will be necessary to boot from an alternate source, such as the mirror of the root disk, an alternate boot disk, or the recovery shell.

## Loading and Starting the Kernel

If the HP-UX boot loader fails to find or load the **vmunix** file, the HP-UX boot loader has the ability to list the contents of the **/stand** directory.

```
HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL booting  hpux

Boot
: disk(0/0/2/0.2.0;0)/stand/vmunix
disk(0/0/2/0.2.0;0)/stand/vmunix: cannot open, or not executable



ISL>
ISL> hpux ll

ls
: disk(0/0/2/0.2.0;0)/.
drwxrwxrwx   10 2            2                  1024 ./
drwxrwxrwx   10 2            2                  1024 ../
drwxr-xr-x    2 0            0                  8192 lost+found/
-r--r--r--    1 0            3                   969 system
drwxr-xr-x    2 0            3                  1024 system.d/
-r--r--r--    1 0            3                    82 kernrel
-rw-rw-rw-    1 0            3                  2452 ioconfig
-rw-r--r--    1 0            3                    19 bootconf
drwxr-xr-x    4 0            3                  2048 build/
-rw-------    1 0            0                    12 rootconf
drwxr-xr-x    2 0            0                  1024 krs_lkg/
```




© 2005 Hewlett-Packard Development Company, L.P.

```
drwxr-xr-x    2 0            0                  1024 krs/
drwxr-xr-x    2 0            0                  1024 krs_tmp/
drwxrwxrwx    5 0            3                  1024 dlkm/
drwxrwxrwx    5 0            0                  1024 dlkm.vmunix.prev/
-rwxr-xr-x    1 0            0              20332040 vmunix.prev*


ISL>
```

If an alternate kernel exists, it can be used to boot the system, preferably into single user mode. If no alternate kernel exists, it will be necessary to boot from an alternate source.

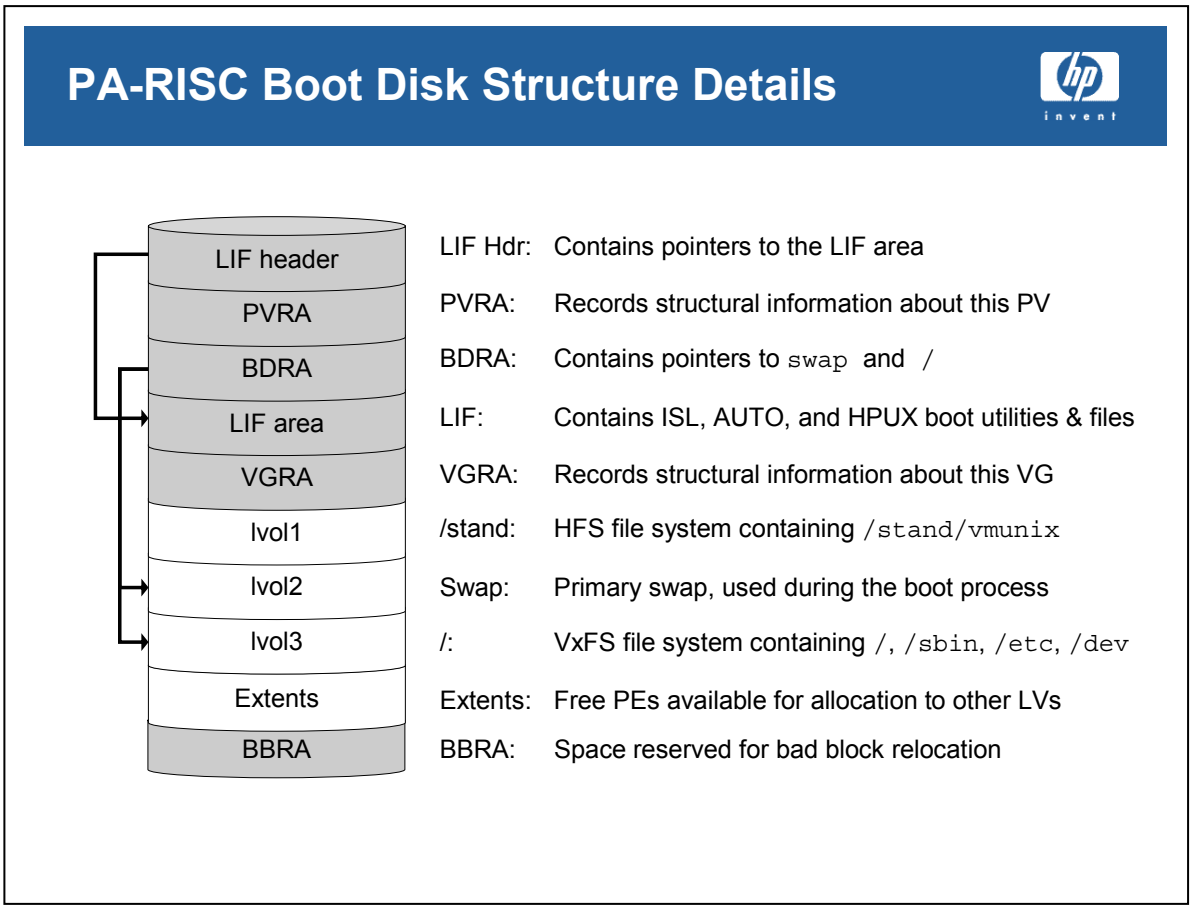
If the **`vmunix`** file exists, there is still no guarantee that it will boot, as the file itself may be corrupt. A successful boot also relies on the LVM configuration, the kernel's own configuration, the root file system, **`pre_init_rc`**, **`init`**, and a handful of other files.

In this module, we will look at issues with the boot disk.

We will discuss LVM and file system problems later in the course.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–4. SLIDE: PA-RISC Boot Disk Structure Details




## Student Notes

LVM boot physical volumes include all the reserved areas typically found on an LVM data physical volume, and several additional structures that are required to support the system boot process. These structures are described in detail below.

- At the top of every boot disk, there is a Logical Interchange Format (LIF) header structure. This structure contains information necessary to access and read the boot LIF area described below. If the LIF header is corrupt or bad, this disk will not be noted as a bootable disk.

- The PVRA on a boot disk serves the same purpose as the PVRA on a data disk.

- The Boot Data Reserved Area (BDRA) contains pointers to two of the three key logical volumes used in the early stages of the boot process: primary swap and **/**. The HPUX kernel loader is unable to read LVM extent maps; instead it determines the locations of these logical volumes using the pointers in the BDRA.

- The Logical Interchange Format (LIF) area on a boot disk contains several boot utilities and files. The **`ISL`** executable is responsible for determining which kernel to boot. It does this by reading the ASCII **`AUTO`** file, which is also stored in the LIF area. The HPUX



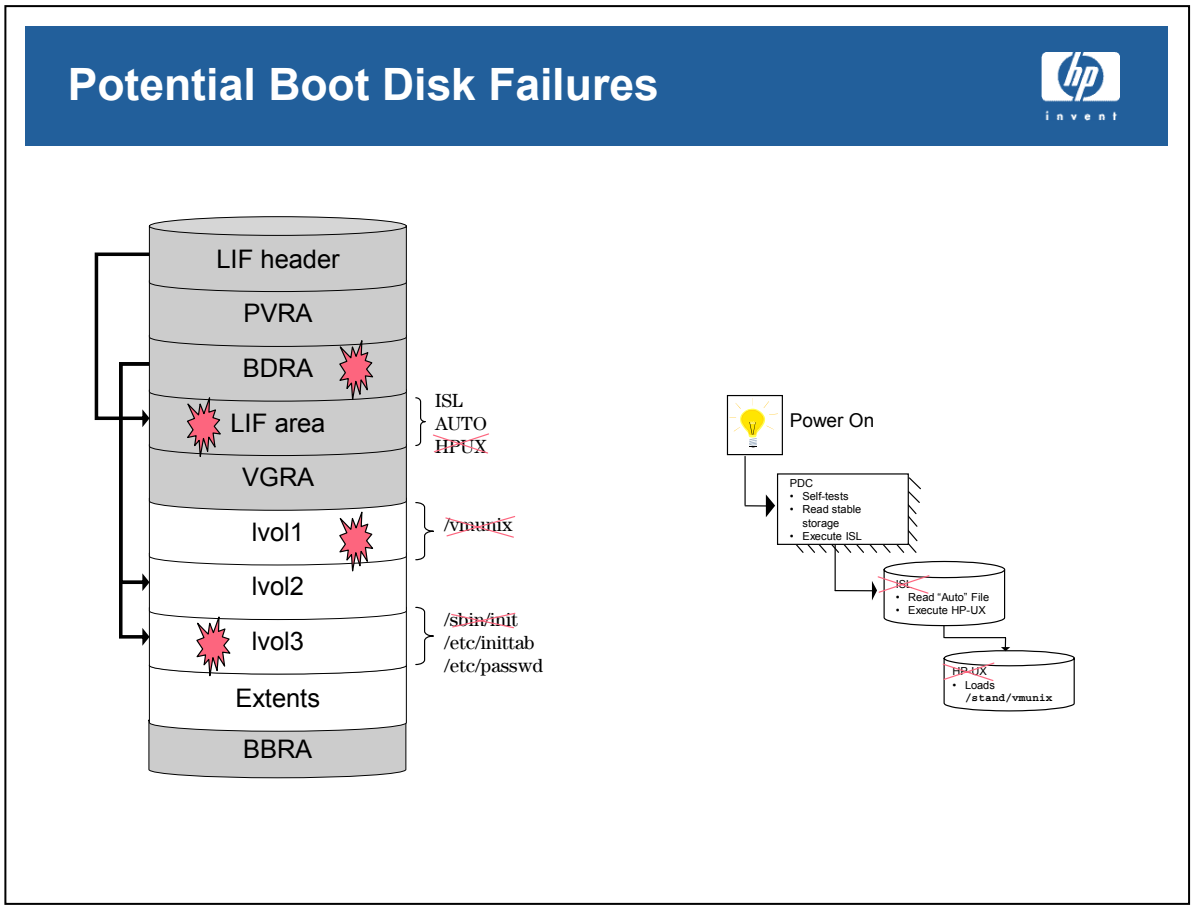

© 2005 Hewlett-Packard Development Company, L.P.

utility is responsible for actually loading the kernel in memory. The LIF area files are stored in a special LIF file format that isn't readable using standard UNIX commands like **`cp`**, **`ls`**, and **`cat`**. The utilities required to manage the LIF area will be described later in the chapter.

- The VGRA on a boot disk also serves the same purpose as the VGRA on a data disk.
- Three logical volumes are required on the boot disk: **`lvol1 (/stand`**, which includes the **`/stand/vmunix`** kernel), **`lvol2`** (primary swap), and **`lvol3`** (**`/`**, which must include **`/dev`**, **`/etc`**, and **`/sbin`**). Other file systems such as **`/usr`**, **`/var`**, and **`/tmp`** may optionally be included on the boot disk as well, or they may be stored on a separate disk.
- The BBRA on the end of the boot disk serves the same purpose as the BBRA on a data disk.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–5. SLIDE: Potential Boot Disk Failures




## Student Notes

If your logical volume manager (LVM)-managed boot disk has corrupt or missing boot utilities in the LIF area, your system may not boot.

In the above example, 3 utilities/files are either missing or corrupt.

The HPUX boot utility is missing from the LIF area. This utility is critical to the boot process. Without it, your system will not boot.

Additionally, `vmunix` is either missing or corrupt. Since this is the kernel, our system could not boot.

Finally, **`/sbin/init`** is either missing or corrupt. **`/sbin/init`** is responsible for executing the numerous startup scripts required to bring the system to Multi-User Mode.

How do you fix this problem? If there is an alternate boot disk, it might be possible to boot from it and execute certain commands to repair the damage on the original boot disk. If no alternate boot disk is available, booting into a recovery shell could be the next best option.



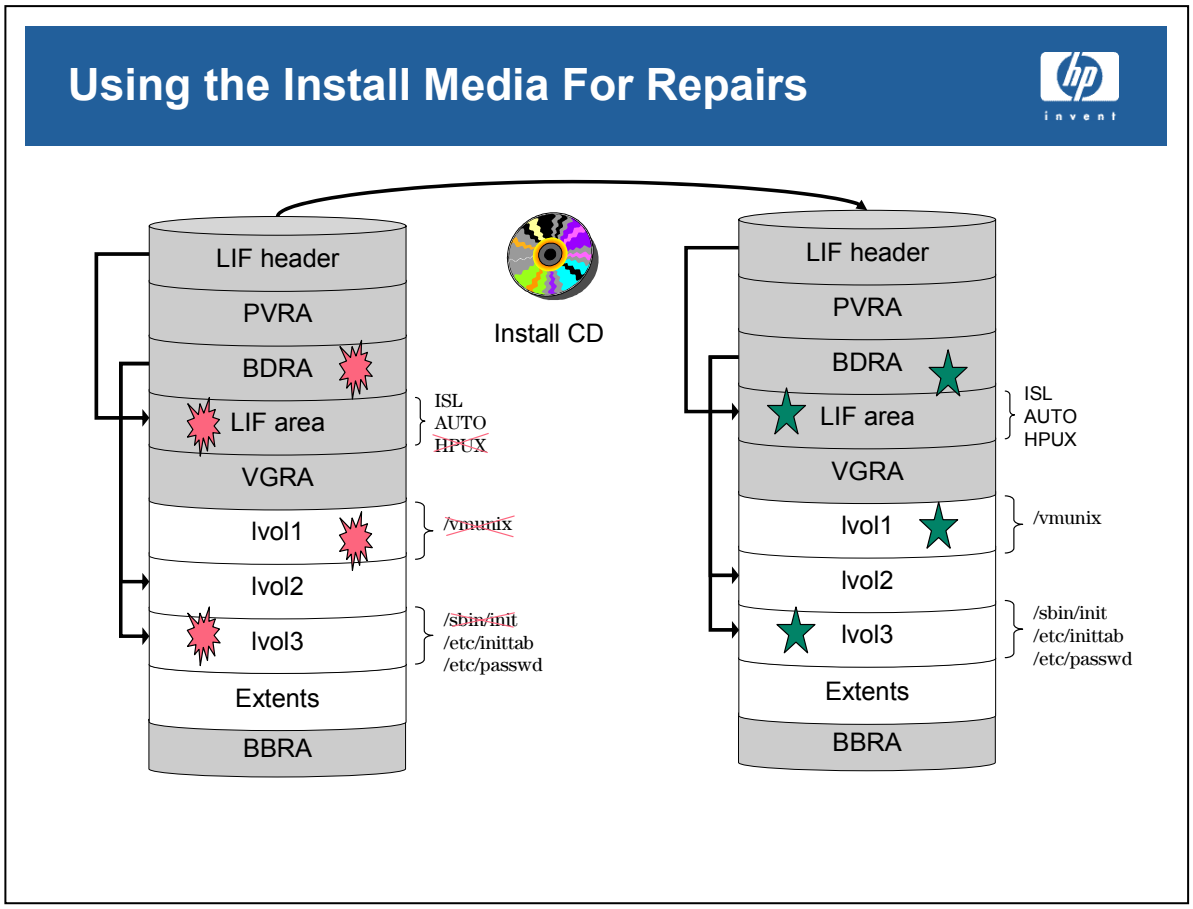

© 2005 Hewlett-Packard Development Company, L.P.

Keep in mind that when encountering a situation such as this, you probably won't immediately know the extent of the damage. Additionally, while you may feel that you can fix the problem, it is important to consider the amount of time it might take to do so.

Always be mindful that, while you may not learn what happened that caused the damage in the first place, reinstalling from a **`make_net_recovery`** or **`make_tape_recovery`** image will often make the system available much faster, and that is important.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–6. SLIDE: Using the Install Media for Repairs




## Student Notes

You can use the Install Media to initiate the recovery process. When booting from the install media, there is a "recovery shell" option. This is a program that allows you to diagnose and fix problems when the operating system cannot be booted from the system disk. Its most common uses are to repair the LIF area, restore a missing/damaged kernel, and restore critical files needed for completion of the boot process.

In this module, the recovery shell will be used to repair a number of problems with a boot disk.




© 2005 Hewlett-Packard Development Company, L.P.

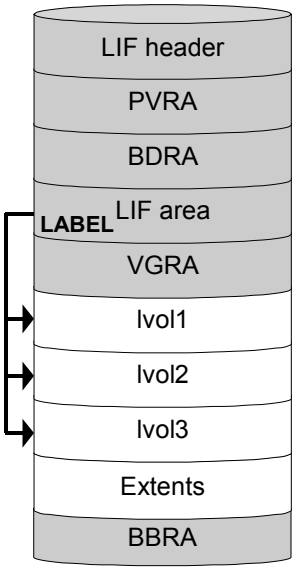
## 4–7. SLIDE: The LABEL File

### The LABEL File

LABEL File

- Contains pointers to the starting points and sizes of boot relevant logical volumes.
- Resides in the LIF volume.
- Information stored within LABEL is used by the secondary loader to load the operating system.

```
# lifls -l /dev/rdsk/c0t6d0
```




## Student Notes

The **LABEL** file is one of the boot files found in the LIF area. It contains pointers to the starting points and sizes of the boot-relevant logical volumes. To load the operating system, the HP-UX secondary loader uses information found in the **LABEL** file. Without an intact **LABEL** file, the system will not boot.

The **LABEL** file is maintained by the **lvlnboot** and **lvrmboot** commands, respectively.

If the **LABEL** file is missing or corrupt, you will need to boot into the recovery shell.

In later slides, we will walk through the steps to repair/replace a **LABEL** file.


© 2005 Hewlett-Packard Development Company, L.P.

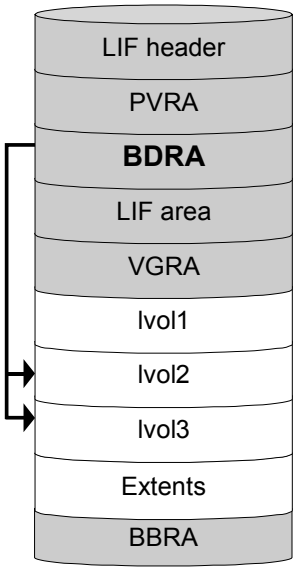
## 4–8. SLIDE: Boot Data Reserve Area (BDRA)

### Boot Data Reserved Area (BDRA)

The BDRA contains the following records about the system's root logical volumes:

- Root volume group ID
- Number of disks in root volume group
- A list of hardware addresses of the LVM disks in the root volume group
- Information needed to select the correct logical volumes for boot, primary swap, and dump

The operating system uses the information in the BDRA to activate the root volume group and mount the `root` file system.




## Student Notes

To boot the system, the kernel activates the volume group to which the system's root logical volumes belong. Pointers the locations of the root logical volumes are stored in the **boot data reserve area** (BDRA). The boot data reserve area contains information on the locations and sizes of LVM disks in the root volume group. It also has vital information needed to configure root, primary swap, and dump logical volumes (but not the boot volume), and to mount the root file system.

Information about the LVM disk data structures in the BDRA is maintained with the **`lvlnboot(1M)`** and **`lvrmboot(1M)`** commands.



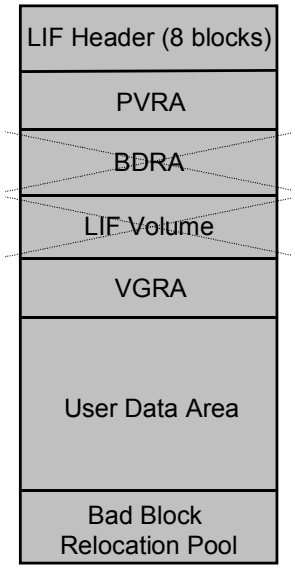

© 2005 Hewlett-Packard Development Company, L.P.

## 4–9. SLIDE: Corrupt `LABEL` File or `BDRA`

### Corrupt LABEL File or BDRA

If the `LABEL` file or BDRA is corrupt:

- The system will *not* boot.
- To recover, the system must be booted into LVM Maintenance mode or to the recovery shell.
- A command called `lvlnboot` is needed to fix the problem.




## Student Notes

If the **`LABEL`** file or BDRA is corrupt or missing, the system will not boot. The severity of the problem determines whether booting into LVM maintenance mode or booting into a recovery shell is needed. Either way, the **`lvlnboot`** command is used to fix the bad **`LABEL`** file or bad BDRA.

When the **`LABEL`** file or BDRA is corrupt, the autoboot process fails with the following error message:

```
Exec failed: Cannot find /stand/vmunix or /vmunix
```

Attempting to enter maintenance mode will load the kernel, but it then panics with the message:

```
all VFS_MOUNTROOTs failed: NEED DRIVERS ?????
```

Therefore, booting into a recovery shell is necessary.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–10. SLIDE: The `lvlnboot` Command

### The `lvlnboot` Command

`lvlnboot`:

- Used to update a corrupted `LABEL` file or BDRA with valid data.
- The syntax of the command is:
  `lvlnboot [-b|-r|-s|-d] lvol_name`
- Examples of the command:

```
# lvlnboot -b /dev/vg00/lvol1
# lvlnboot -r /dev/vg00/lvol3
# lvlnboot -s /dev/vg00/lvol2
# lvlnboot -d /dev/vg00/lvol2
```

## Student Notes

The **`lvlnboot`** command updates the BDRA & LABEL files on the bootable disks within the volume group. It can be used to (re) set the pointers to the root, boot, primary swap, and dump logical volumes. To change the current root or primary swap areas, it is necessary to use **`lvrmboot`** first to de-configure the current volumes. Be aware that de-configuring the current root logical volume will also de-configure the boot, primary swap and all dump volumes!

If changes are made to any of these logical volumes, the **`lvlnboot`** command needs to be rerun. Unless the **`-A n`** option is passed to the **`lv???`** commands, this will happen automatically.

Specifying **`lvlnboot -v`** displays current information on root, boot, primary swap, and dump logical volumes. If the information is incorrect, it can be fixed with any combination of the following commands:

- Boot declaration

  `/sbin/lvlnboot -b /dev/vg00/lvol1`




© 2005 Hewlett-Packard Development Company, L.P.

- Root declaration

  `/sbin/lvlnboot -r /dev/vg00/lvol3`

- Primary swap declaration

  `/sbin/lvlnboot -s /dev/vg00/lvol2`

- Dump declaration

  `/sbin/lvlnboot -d /dev/vg00/lvol2`

Before we go too much further, let's familiarize ourselves with the menus and options from the Boot Console Handler (BCH) prompt and the **ISL>** prompt. Then, in the next slide, we'll explore booting from the Install Media and then look at the Recovery Shell.

1) Power system on and interrupt the automatic boot process.

```
Firmware Version  42.19

Duplex Console IO Dependent Code (IODC) revision 1

-------------------------------------------------------------------------
(c) Copyright 1995-2001, Hewlett-Packard Company, All rights reserved
-------------------------------------------------------------------------

Processor   Speed            State           CoProcessor State  Cache Size
Number                                       State              Inst
Data
------  --------   ---------------------  -----------------  ------------
 0      650  MHz   Active                 Functional         750 KB 1.5 MB

  Central Bus Speed (in MHz)  :        120
  Available Memory            :     655360  KB
  Good Memory Required        :      25820  KB

   Primary boot path:    0/0/1/1.15
   Alternate boot path:  0/0/2/0.2
   Console path:         0/0/4/1.643
   Keyboard path:        0/0/4/0.0


Processor is booting from first available device.

To discontinue, press any key within 10 seconds.


Pressing any key at this point will terminate the automatic boot
process


Boot terminated.
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
Main Menu -------------------------------------------------------------

     Command                            Description
     -------                            -----------
     BOot [PRI|ALT|<path>]              Boot from specified path
     PAth [PRI|ALT] [<path>]            Display or modify a path
     SEArch [DIsplay|IPL] [<path>]      Search for boot devices

     COnfiguration menu                 Displays or sets boot values
     INformation menu                   Displays hardware information
     SERvice menu                       Displays service commands

     DIsplay                            Redisplay the current menu
     HElp [<menu>|<command>]            Display help for menu or command
     RESET                              Restart the system
----
Main Menu: Enter command or menu >
```

Let's look around a bit.
The **pa** command will display our current boot and console paths

```
Main Menu: Enter command or menu > pa

   Primary boot path:    intscsia.15
                         0/0/1/1.15
                         0/0/01/01.f    (hex)

   Alternate boot path:  extscsib.2
                         0/0/2/0.2
                         0/0/02/0.2     (hex)

   Console path:         0/0/4/1.643
                         0/0/04/01.283  (hex)

   Keyboard path:        0/0/4/0.0
                         0/0/04/0.0     (hex)

Main Menu: Enter command or menu > in
```

The **in** command will display an Information menu. If we wanted more info, we could obtain it from within this Information Menu.

```
Information Menu ------------------------------------------------------

     Command                            Description
     -------                            -----------
     ALL                                Display all system information
     BootINfo                           Display boot-related information
     CAche                              Display cache information
     ChipRevisions                      Display revisions of major VLSI
     COprocessor                        Display coprocessor information
     FRU                                Display FRU information
     FwrVersion                         Display firmware version
     IO                                 Display I/O interface information
     LanAddress                         Display Core LAN station address
```




© 2005 Hewlett-Packard Development Company, L.P.

```
     MEmory                            Display memory information
     PRocessor                         Display processor information
     WArnings                          Display selftest warning messages

     BOot [PRI|ALT|<path>]             Boot from specified path
     DIsplay                           Redisplay the current menu
     HElp [<command>]                  Display help for specified command
     RESET                             Restart the system
     MAin                              Return to Main Menu
----
Information Menu: Enter command >
```

Note that we're still in the Information Menu. Type **ma** to return to the Main Menu.

```
Information Menu: Enter command > ma

Main Menu ---------------------------------------------------------------

     Command                           Description
     -------                           -----------
     BOot [PRI|ALT|<path>]             Boot from specified path
     PAth [PRI|ALT] [<path>]           Display or modify a path
     SEArch [DIsplay|IPL] [<path>]     Search for boot devices

     COnfiguration menu                Displays or sets boot values
     INformation menu                  Displays hardware information
     SERvice menu                      Displays service commands

     DIsplay                           Redisplay the current menu
     HElp [<menu>|<command>]           Display help for menu or command
     RESET                             Restart the system
----
```

Let's look at the Configuration Menu. Here, we can set boot parameters such as Autoboot and AutoSearch.

```
Main Menu: Enter command or menu > co

Configuration Menu ------------------------------------------------------

  Command                           Description
  -------                           -----------
  AUto [BOot|SEArch|STart] [ON|OFF] Display or set specified flag
  BootID [<proc>] [<bootid>]        Display or set Boot Identifier
  BootINfo                          Display boot-related information
  BootTimer [0 - 200]               Seconds allowed for boot attempt
  CPUconfig [<proc>] [ON|OFF]       Config/Deconfig processor
  DEfault                           Set the system to predefined values
  FastBoot [ON|OFF]                 Display or set boot tests execution
  PAth [PRI|ALT] [<path>]           Display or modify a path
  SEArch [DIsplay|IPL] [<path>]     Search for boot devices
  TIme [c:y:m:d:h:m:[s]]            Read or set the real time clock in GMT

  BOot [PRI|ALT|<path>]             Boot from specified path
  DIsplay                           Redisplay the current menu
  HElp [<command>]                  Display help for specified command
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
  RESET                              Restart the system
  MAin                               Return to Main Menu
----
Configuration Menu: Enter command > ma

Main Menu ---------------------------------------------------------------

     Command                            Description
     -------                            -----------
     BOot [PRI|ALT|<path>]              Boot from specified path
     PAth [PRI|ALT] [<path>]            Display or modify a path
     SEArch [DIsplay|IPL] [<path>]      Search for boot devices

     COnfiguration menu                 Displays or sets boot values
     INformation menu                   Displays hardware information
     SERvice menu                       Displays service commands

     DIsplay                            Redisplay the current menu
     HElp [<menu>|<command>]            Display help for menu or command
     RESET                              Restart the system
----
```

Let's search for bootable devices.

```
Main Menu: Enter command or menu > sea

Searching for potential boot device(s)
This may take several minutes.

To discontinue search, press any key (termination may not be immediate).


   Path#  Device Path (dec)  Device Path (mnem)  Device Type
   -----  -----------------  ------------------  -----------
   P0     0/0/0/0            lan.10.100.0.254    LAN Module
   P1     0/0/1/1.15         intscsia.15         Random access media
   P2     0/0/2/0.2          extscsib.2          Random access media
   P3     0/0/2/0.0          extscsib.0          Sequential access media
   P4     0/0/2/1.15         intscsib.15         Random access media
```

Oddly enough, this seems to be a list of **all** disks and tapes on the system.
To view a more realistic picture of bootable devices (devices that contain an Initial Program Loader, or IPL), try the following:

```
Main Menu: Enter command or menu > sea ipl

Searching for device(s) with bootable media
This may take several minutes.

To discontinue search, press any key (termination may not be immediate).


   Path#  Device Path (dec)  Device Path (mnem)  Device Type and Utilities
   -----  -----------------  ------------------  -------------------------
   P0     0/0/1/1.15         intscsia.15         Random access media
                                                    IPL
```




© 2005 Hewlett-Packard Development Company, L.P.

```
   P1     0/0/2/0.2          extscsib.2          Random access media
                                                    IPL


Main Menu: Enter command or menu >
```

Let's boot the system to **ISL**. We'll boot from the primary path.

```
Main Menu: Enter command or menu > bo pri
Interact with IPL (Y, N, or Cancel)?> y

Booting...
Boot IO Dependent Code (IODC) revision 1


HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL>
```

Let's look around here.

```
ISL> he

    HELP          Help Facility
    LS            List ISL utilities
    AUTOBOOT      Set or clear autoboot flag in stable storage
    AUTOSEARCH    Set or clear autosearch flag in stable storage
    PRIMPATH      Modify primary boot path in stable storage
    ALTPATH       Modify alternate boot path in stable storage
    CONSPATH      Modify system console path in stable storage
    DISPLAY       Display boot and console paths in stable storage
    LSAUTOFL      List contents of autoboot file
    FASTSIZE      Sets or displays FASTSIZE
    800SUPPORT    Boots the s800 Support Kernel from the boot device
    700SUPPORT    Boot the s700 Support Kernel from the boot device
    READNVM       Displays contents of one word of NVM
    READSS        Displays contents of one word of stable storage
    LSBATCH       List contents of batch file
    BATCH         Execute commands in batch file
    LSEST         List contents of EST (Extended Self Test) file
    EST           Execute commands in EST (Extended Self Test) file

Enter 'LS' to see a list of the ISL utilities.

ISL>
```

Note that we can also set the **AUTOBOOT** and **AUTOSEARCH** flags here, in addition to setting our boot paths.

```
ISL> LS


    Utilities on this system are:
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
filename   type   start    size     created
=====================================================
HPUX       -12928 904      848      02/05/10 16:25:44
```

Note the secondary loader, HPUX.

From here, to boot the system, we execute the following:

```
ISL> hpux
```




© 2005 Hewlett-Packard Development Company, L.P.

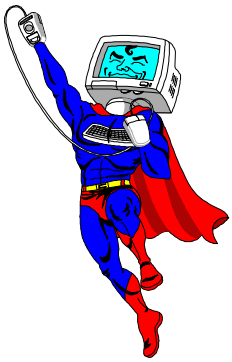
## 4–11. SLIDE: Recovery Shell

### Recovery Shell

The recovery shell is used to access the root file system and fix problems when the boot disk is “unbootable.”

Reasons the boot disk becomes “unbootable” include:

- Bad or damaged LIF area
- Bad or damaged binary start-up files
- Bad or damaged ASCII start-up files
- Corrupt / Missing Kernel



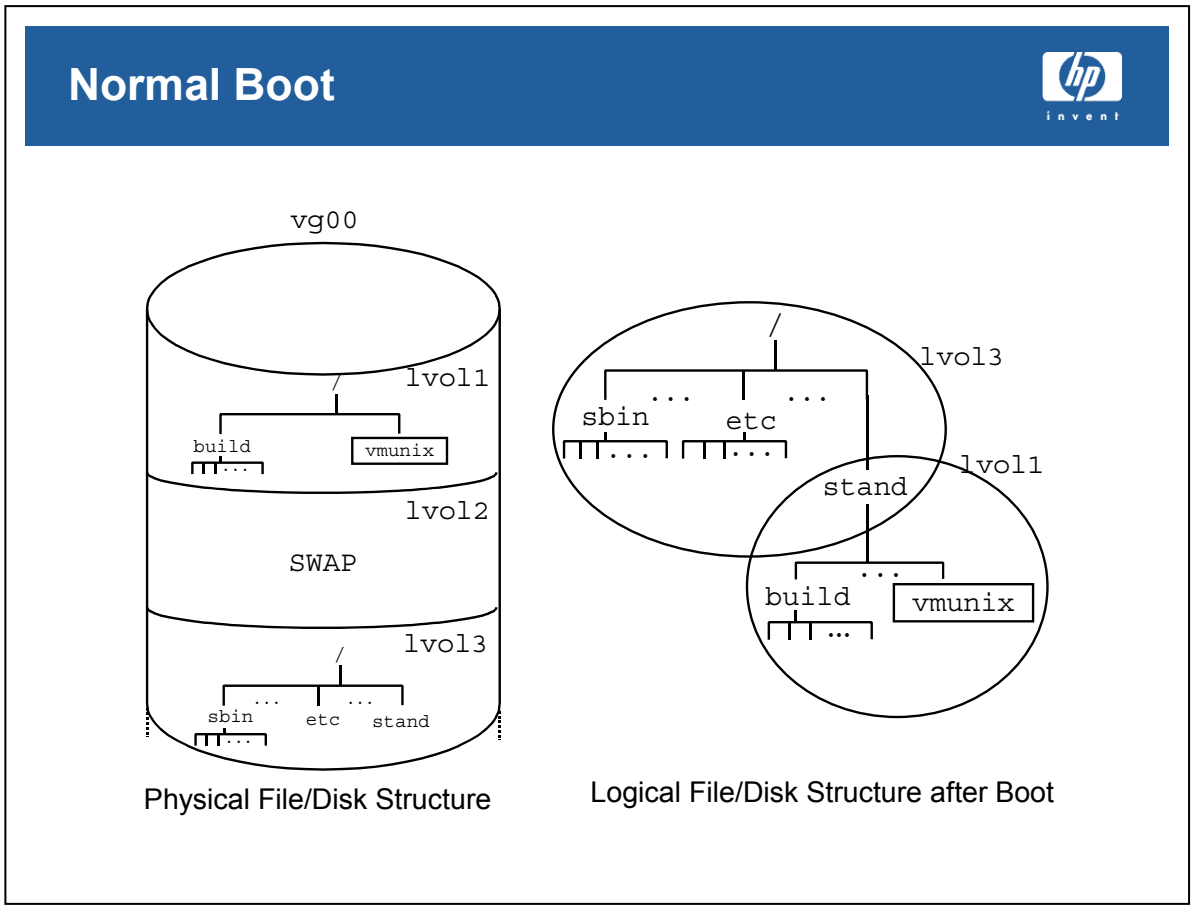
*“Recovery shell to the rescue”*

## Student Notes

To repair the problems listed on the previous slides, we can boot from the Install Media and access the Recovery Shell.


© 2005 Hewlett-Packard Development Company, L.P.

## 4–12. SLIDE: Normal Boot




## Student Notes

The normal boot sequence is entirely automatic and is started when the system is first powered up. The read-only memory (ROM) causes the hardware to be tested and files to be loaded from the disk to random-access memory (RAM).

The default layout of the boot disk is for **`lvol1`** to be the boot file system, which holds the kernel and support files and any other information that is or might be needed before the kernel has finished initializing.

The default file system for HP-UX 11i is JFS, but the PA-RISC boot loader is unable to access this type of file system, so the kernel must be loaded from an HFS file system. Therefore, a minimal HFS structure resides on **`lvol1`**,

The root file system then resides on **`lvol3`** and is of type JFS.

If the normal boot fails, the system administrator needs to boot the system in some other way. The type of failure and the way the system is configured usually dictates which approach to use.




© 2005 Hewlett-Packard Development Company, L.P.

Available options are:

- Boot from the other half of a mirrored disk configuration.
- Boot into single user mode.
- Boot into LVM maintenance mode.
- Boot from an alternative "full" boot disk.
- Boot from a recovery shell, either from the Install Media or an Ignite server.
- Boot from a recovery tape or server.
- Reinstall HP-UX from scratch.

Where disk mirroring is used, the system is typically configured so that it will automatically attempt to boot from the second boot disk if the primary fails at the ISL stage. If three way mirroring is used, and both the primary and secondary disks have failed, a manual boot is needed to boot from the tertiary copy.

However, this automatic rollover to the alternate boot path is used only if the device at the primary boot path fails to respond. If the primary boot disk hardware is good and ISL loads, the system will attempt to boot from it and not from the alternate. Problems with the HP-UX boot loader, LABEL file, or AUTO files will need to be addressed manually.

It is also possible that the copy of the file system on the primary disk is corrupt, but not on the secondary. This is caused either by hardware errors on the device, or by directly writing to the disk.

If the failure to boot properly is late in the boot sequence, after **`init`** has started, booting into single user mode is the easiest way to tackle the problem.

Certain LVM problems can be addressed from a maintenance mode boot.

If the system is configured with an alternate boot environment in a different volume group, it is easy to boot the system, get a shell, then manually repair it.

Where no alternative local boot environment is available, booting to the recovery shell from either the Install Media or an Ignite server provides an environment to repair the current installation.

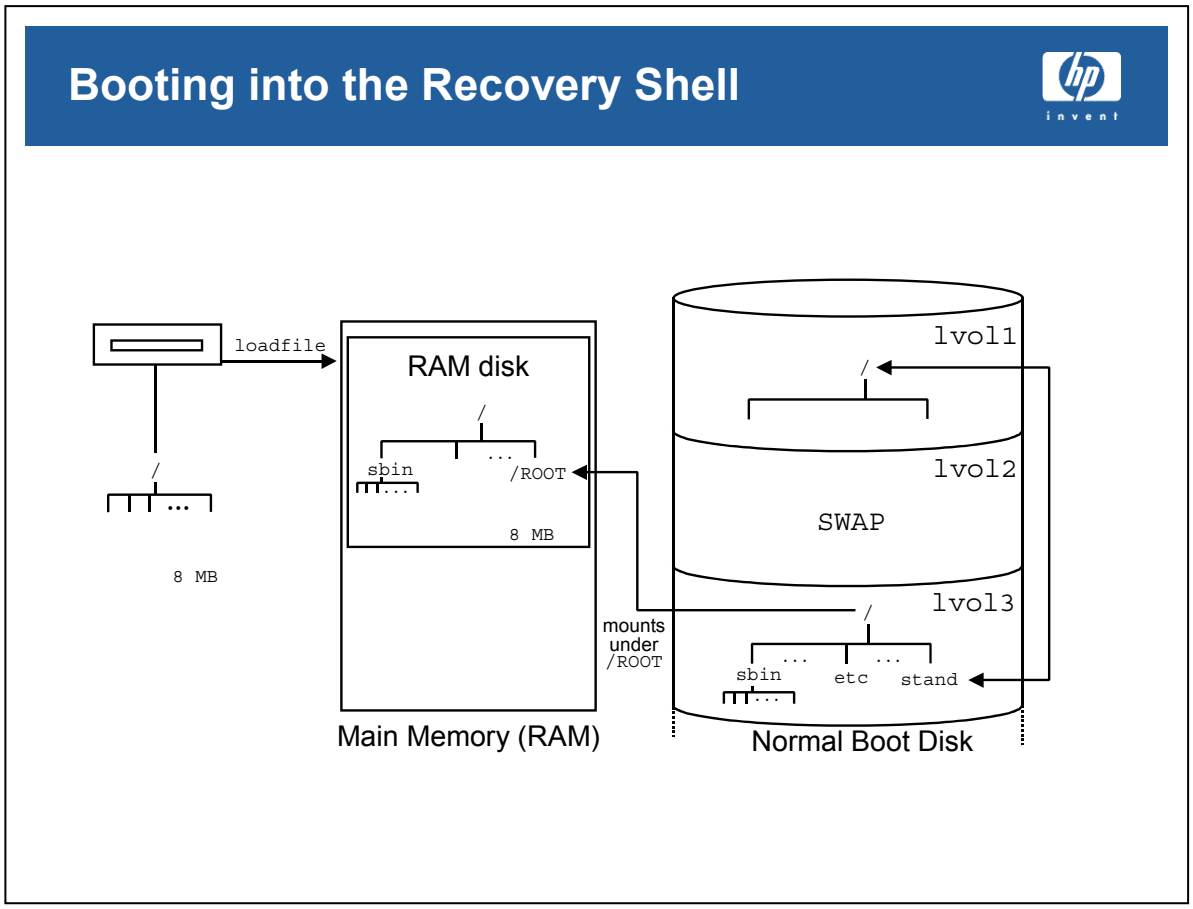
Alternately, the current installation can be overwritten using an Ignite recovery tape or **`make_net_recovery`** image (if one has been made), and then reaching for the backups.

Finally, if no recovery tape (or **`make_net_recovery`** image) was made, a full re-install may be required.

In this module, we are looking at the option of booting the recovery shell.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–13. SLIDE: Booting into the Recovery Shell




## Student Notes

When a normal boot is not successful, the system administrator must intervene by booting from some recovery medium into a **recovery shell**.

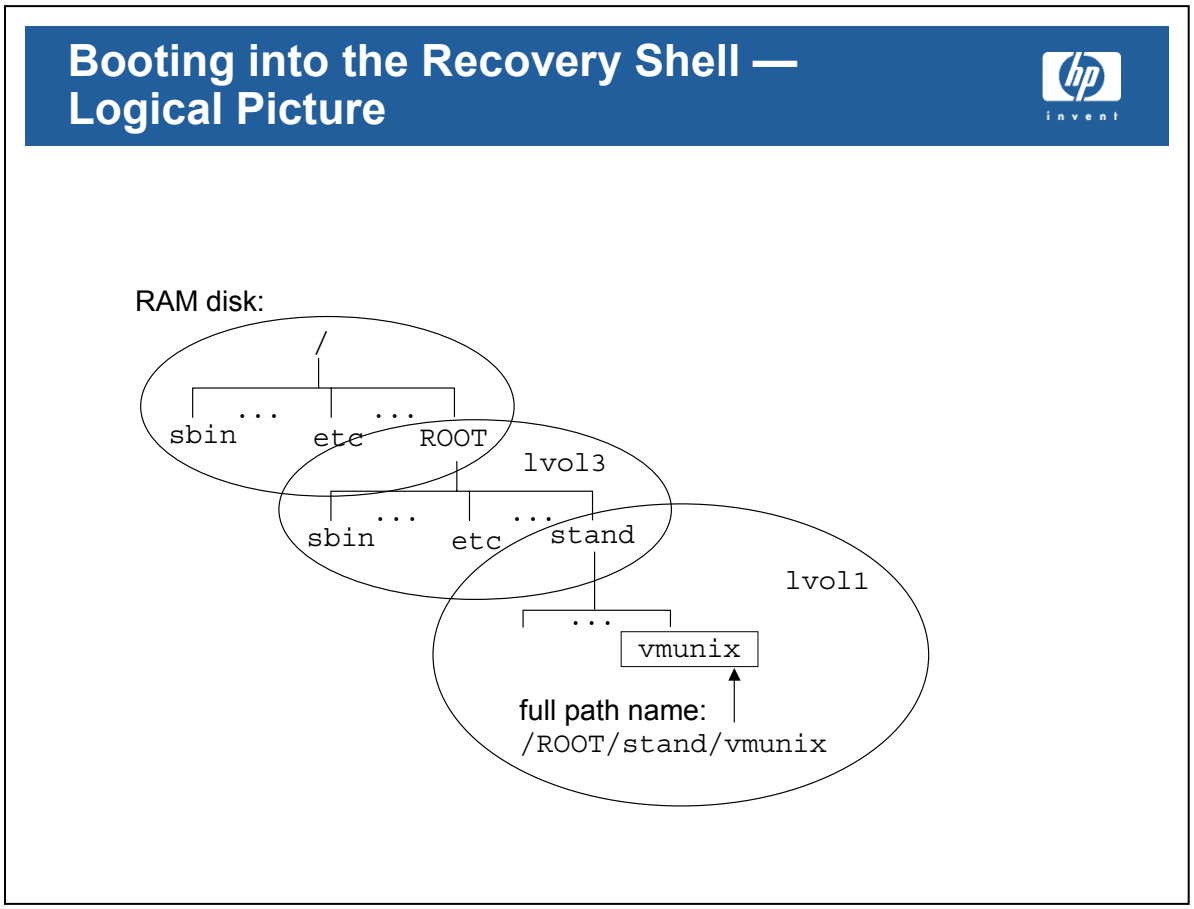
As shown in the slide, the program **`loadfile`** is run from the recovery medium (CD/ Ignite server). This loads essential files into the RAM-based file system (also known as the memory-based file system or the in-core file system). Only 8 MB are available for these files, so only the essential ones are retrieved. These files are in volatile memory and will disappear at the next boot.

Entering **`chroot /ROOT /sbin/sh`** makes the **`/ROOT`** file system your root file system.




© 2005 Hewlett-Packard Development Company, L.P.

## 4–14. SLIDE: Booting into the Recovery Shell — Logical Picture



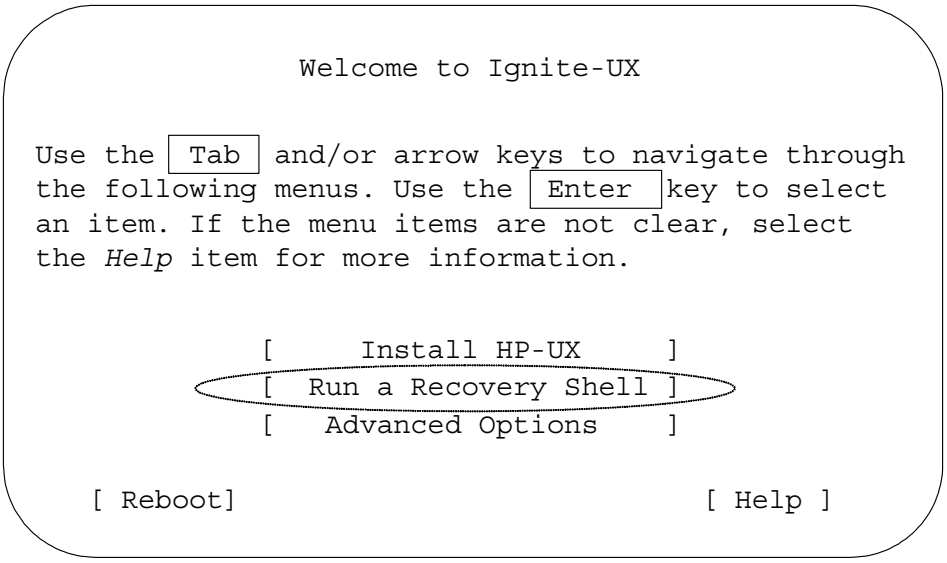

## Student Notes

This slide reiterates the logical result of booting into a recovery shell and choosing the options mentioned in the previous slide.


© 2005 Hewlett-Packard Development Company, L.P.

## 4–15. SLIDE: Using the HP-UX Install Media

### Using the HP-UX Install Media




## Student Notes

Following are the steps to load the recovery shell:

Power system up and interrupt the automatic boot process.

```
Firmware Version  42.19

Duplex Console IO Dependent Code (IODC) revision 1

-------------------------------------------------------------------------
(c) Copyright 1995-2001, Hewlett-Packard Company, All rights reserved
-------------------------------------------------------------------------

Processor   Speed            State           CoProcessor State  Cache Size
Number                                       State              Inst
Data
------  --------   ---------------------  -----------------  ------------
 0      650  MHz   Active                 Functional         750 KB 1.5 MB

  Central Bus Speed (in MHz)  :        120
  Available Memory            :     655360  KB
```




© 2005 Hewlett-Packard Development Company, L.P.

```
  Good Memory Required        :      25820  KB

   Primary boot path:    0/0/1/1.15
   Alternate boot path:  0/0/2/0.2
   Console path:         0/0/4/1.643
   Keyboard path:        0/0/4/0.0


Processor is booting from first available device.

To discontinue, press any key within 10 seconds.
```

Pressing any key at this point will terminate the automatic boot process

```
Boot terminated.


Main Menu -------------------------------------------------------------

     Command                            Description
     -------                            -----------
     BOot [PRI|ALT|<path>]              Boot from specified path
     PAth [PRI|ALT] [<path>]            Display or modify a path
     SEArch [DIsplay|IPL] [<path>]      Search for boot devices

     COnfiguration menu                 Displays or sets boot values
     INformation menu                   Displays hardware information
     SERvice menu                       Displays service commands

     DIsplay                            Redisplay the current menu
     HElp [<menu>|<command>]            Display help for menu or command
     RESET                              Restart the system
----
Main Menu: Enter command or menu >
```

We previously set our Alternate Path to the CDROM. We can now boot from the Install Media as follows:

```
Main Menu: Enter command or menu > bo alt
Interact with IPL (Y, N, or Cancel)?> n

Booting...
Boot IO Dependent Code (IODC) revision 1


HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL booting  hpux (;0):INSTALL

Boot
: disk(0/0/2/0.2.0.0.0.0.0;0):WINSTALL
11538432 + 1941592 + 2610520 start 0x201468
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
alloc_pdc_pages: Relocating PDC from 0xf0f0000000 to 0x27b00000.
gate64: sysvec_vaddr = 0xc0002000 for 2 pages
NOTICE: nfs3_link(): File system was registered at index 4.
NOTICE: autofs_link(): File system was registered at index 6.
NOTICE: cachefs_link(): File system was registered at index 7.

    System Console is on the Built-In Serial Interface
    Swap device table:  (start & size given in 512-byte blocks)
        entry 0 - auto-configured on root device; ignored - no room
WARNING: no swap device configured, so dump cannot be defaulted to primary
swap.
WARNING: No dump devices are configured.  Dump is disabled.
Starting the STREAMS daemons-phase 1
Create STCP device files
          $Revision: vmunix:    vw: -proj    selectors: CUPI80_BL2000_1108
-c 'V
w for CUPI80_BL2000_1108 build' -- cupi80_bl2000_1108 'CUPI80_BL2000_1108'
Wed Nov  8 19:24:56 PST 2000 $
Memory Information:
 physical page size = 4096 bytes, logical page size = 4096 bytes
 Physical: 655360 Kbytes, lockable: 447056 Kbytes, available: 527232Kbytes

===  01/17/04 20:04:00 EST  HP-UX Installation Initialization. (Sat Jan 17
         20:04:00 EST 2004)
         @(#) Ignite-UX Revision B.4.4.27
         @(#) install/init (opt) $Revision: 10.277 $
       * Scanning system for IO devices...
       * Querying disk device: 0/0/1/1.15.0 ...
       * Querying disk device: 0/0/2/1.15.0 ...
       * Setting keyboard language.
NOTE:
      The console firmware terminal type is currently set to "vt100". If
      you are using any other type of terminal you will see "garbage" on
      the screen following this message.
      If this is the case, you will need to either change the terminal
      type set in the firmware via MP (if your MP firmware version
      supports this feature), or change your terminal emulation to match
      the firmware. In either case you will need to restart if your
      terminal and the firmware terminal type do not match.
         Press the 'b' key if you want to reboot now.



              Welcome to the HP-UX installation/recovery process!

  Use the <tab> key to navigate between fields, and the arrow keys
  within fields.  Use the <return/enter> key to select an item.
  Use the <return/enter> or <space-bar> to pop-up a choices list.  If the
  menus are not clear, select the "Help" item for more information.

  Hardware Summary:         System Model: 9000/800/A400-6X
+---------------------+---------------+-------------------+ [Scan Again]
| Disks: 2  ( 67.8GB) |  Floppies: 0  | LAN cards:   1    |
| CD/DVDs:        1   |  Tapes:    1  | Memory:     640Mb |
| Graphics Ports: 0   |  IO Buses: 4  | CPUs:        1    |[ H/W Details]
```




© 2005 Hewlett-Packard Development Company, L.P.

```
+---------------------+----------------+-------------------+

                 [      Install HP-UX       ]
                 [   Run a Recovery Shell   ]
                 [    Advanced Options      ]


     [  Reboot  ]                            [  Help  ]

-----------------------------------------------------------------------
```

**Once here, tab to and select `"Run a Recovery Shell"`**


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–16. SLIDE: HP-UX System Recovery Main Menu

### HP-UX System Recovery Main Menu

```
HP-UX CORE MEDIA SYSTEM RECOVERY
MAIN MENU

s.  Search for a file
b.  Reboot
l.  Load a file
r.  Recover an unbootable HP-UX system
x.  Exit to shell

This menu is for listing and loading the tools
contained on the core media. Once a tool is
loaded, it may be run from the shell. Some tools
require other files to be present in order to
successfully execute.

Select one of the above:
```

## Student Notes

```
HP-UX CORE MEDIA SYSTEM RECOVERY
MAIN MENU


s.  Search for a file
b.  Reboot
l.  Load a file
r.  Recover an unbootable HP-UX system
x.  Exit to shell


This menu is for listing and loading the tools contained on the core
media. Once a tool is loaded, it may be run from the shell. Some tools
require other files to be present in order to successfully execute.

Select one of the above: r
Loading commands for recovery ....
```

---

Troubleshooting a PA-RISC System Boot Disk


© 2005 Hewlett-Packard Development Company, L.P.

You are now prompted for the hardware path to your root disk. What is displayed may or may not be correct, so be sure to read through this carefully before selecting an option.

```
        DEVICE FILE VERIFICATION MENU

    This menu is used to specify the path of the root file system.
    When the information is correct, select 'a'.

        INFORMATION to verify:
          Device file used for '/'(ROOT) is c1t15d0
          The hardware path to disk is 0/0/1/1.15.0

        Select one of the following:
          a.  The above information is correct.
          b.  WRONG!! The device file used for '/'(ROOT) is incorrect.

          m.  Return to the 'HP-UX Recovery MENU.'
          x.  Exit to the shell.

     Selection: a
The disk is LVM
Setting boot and root device file for c1t15d0...


-------------------------------------------------------------------
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–17. SLIDE: HP-UX Recovery Menu

### HP-UX Recovery Menu



```
                HP-UX RECOVERY MENU

  Select one of the following:
  a. Mount the root disk and exit to a shell only.
  b. Recover the bootlif/os partitions.
  c. Replace the kernel on the root file system.
  d. Both Options: b and c
  v. Read information about LVM Recovery (LVM.RECOVER)

  m. Return to the 'HP-UX Recovery Media Main Menu'
  x. Exit to the shell.

 Use this menu to select the level of recovery desired.

 Selection:
```

## Student Notes

```
                HP-UX Recovery MENU

     Select one of the following:
     a.  Mount the root disk and exit to a shell only.
     b.  Recover the bootlif/os partitions.
     c.  Replace the kernel on the root file system.
     d.  Both Options: b and c
     v.  Read information about LVM Recovery (LVM.RECOVER)

     m.  Return to 'HP-UX Recovery Media Main Menu'.
     x.  Exit to the shell.

 Use this menu to select the level of recovery desired.

 Selection: a
```

Here we are! Note the different options in the `HP-UX Recovery Menu`.
While it may appear that we can simply choose to automatically recover the bootlif or the kernel, these options are not always practical, so we will focus on a recovery that is.




© 2005 Hewlett-Packard Development Company, L.P.

First, let's look a little more into what the recovery shell does.

Choosing the **"a. Mount the root disk and exit to a shell only"** will mount the root disk and take us to a shell.
Here are the next menu's and prompts:

```
Entering file system checking ...


               FILE SYSTEM CHECK
                    MENU
       The file system check '/sbin/fs/vxfs/fsck -y
/dev/rdsk/c1t15d0s1lvm'
       will now be run.

       Select one of the following:
        a.  Run fsck -y.
        b.  Prompt for the fsck run string on c1t15d0s1lvm.
        m.  Return to the 'HP-UX Recovery MENU.'

        Selection: a
```

Again, note the non-standard names for the device files. This is OK as we have chosen the correct disk.

```
** /dev/rdsk/c1t15d0s1lvm
** Last Mounted on /stand
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
49 files, 0 icont, 28995 used, 114578 free (114 frags, 14308 blocks)
Going to mount /dev/dsk/c1t15d0s1lvm to /ROOT/stand as read only
Umount /ROOT/stand
Doing fsck on /ROOT file system
/sbin/fs/vxfs/fsck -y /dev/rdsk/c1t15d0s2lvm
file system is clean - log replay is not required
 Mount /ROOT file system
/sbin/fs/vxfs/mount /dev/dsk/c1t15d0s2lvm /ROOT
Mount /ROOT/stand again
/sbin/fs/hfs/mount  -F hfs /dev/dsk/c1t15d0s1lvm /ROOT/stand

=================================================================
The root disk has been mounted under /ROOT, if you want to chroot
to this directory type the commands:
     loadfile chroot
     chroot /ROOT /sbin/sh

#
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
=======  07/03/01 06:13:10 EDT  HP-UX Installation Initialization. (Tue Jul 03

         06:13:10 EDT 2001)
         @(#) Ignite-UX Revision B.3.0.502
         @(#) install/init (opt) $Revision: 10.207 $
       * EISA configuration has completed.  Following the completion of a
         successful HP-UX installation, please check the
"/etc/eisa/config.err"

         file for any EISA configuration messages.
       * Scanning system for IO devices...
       * Querying disk device: 8/0/19/0.2.0 ...
       * Querying disk device: 8/0/19/0.5.0 ...
       * Querying disk device: 8/0/19/0.6.0 ...
       * Querying graphics device: 10/0/4/0 ...
       * Setting keyboard language.
```

At this point, let's look around:

```
# ls -l
total 196
-rw-------   1 root       sys          76858 Jan 17 20:47 .sh_history
drwxr-xr-x   2 root       sys             24 Jan 17 20:27 CDROM
drwxr-xr-x  17 root       sys           1024 Jan 17 17:55 ROOT
d---------   2 bin        bin             24 Jul 23 17:44 core
drwxr-xr-x   8 root       root          1024 Jan 17 20:27 dev
drwxr-xr-x   2 root       root          1024 Jan 17 20:27 dev.old
drwxr-xr-x   2 bin        bin             24 Jul 23 17:44 disc
dr-xr-xr-x   4 bin        bin           1024 Jan 17 20:27 duped_root
dr-xr-xr-x   4 bin        bin           1024 Jan 17 20:41 etc
drwxr-xr-x   2 root       root          4096 Jul 23 17:44 lost+found
drwxr-xr-x   3 bin        bin             40 Jul 23 17:44 opt
drwxr-xr-x   4 bin        bin           1024 Jan 17 20:37 sbin
drwxr-xr-x   2 bin        bin           1024 Jan 17 20:41 stand
drwxrwxrwx   2 bin        bin           1024 Jan 17 20:27 tmp
drwxr-xr-x   7 bin        bin             96 Jul 23 17:44 usr
drwxr-xr-x   5 bin        bin             60 Jul 23 17:44 var
#
```

The **/ROOT** directory, and everything below it, corresponds to our root disk. Everything else in this directory is the 8MB ram-based file system.
Just before we executed the **ls -l**, we saw the directions that suggested we execute a few more commands. The message:

```
The root disk has been mounted under /ROOT, if you want to chroot
to this directory type the commands:
     loadfile chroot
     chroot /ROOT /sbin/sh
```

We need to **loadfile chroot** because the **chroot** command isn't currently in our 8MB ram-based file system.

Entering **chroot /ROOT /bin/sh** makes the **/ROOT** file system our root file system.




© 2005 Hewlett-Packard Development Company, L.P.

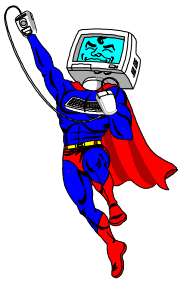
From there, we can poke around on the boot disk and view/manipulate almost any file we want. If, for example, the **`LABEL`** file or the BDRA was corrupt, we could simply execute the appropriate **`lvlnboot`** commands at this point. Further, if the kernel needed to be rebuilt, we could do that now, too. But let's look at that a little later.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–18. SLIDE: Repairing Damage — Missing / Corrupt LIF Utilities

### Repairing Damage — Missing / Corrupt LIF Utilities

The recovery shell can be used to repair many things on the boot disk.
Example: Repairing missing corrupt HPUX / ISL in the LIF area.



The procedure:

1. Boot from the support media.
2. From the Main Menu, select "Run a Recovery Shell"
3. Verify the hardware path to your boot disk
4. From the `Recovery` menu, select
   `a. Mount the root disk and exit to a shell only`
5. From the Recovery shell, execute the following commands:

```
# loadfile chroot
# chroot /ROOT /sbin/sh
# vgchange -a y /dev/vg00
# mountall
# mkboot /dev/rdsk/{bootdisk}
. . .
# reboot
```

## Student Notes

To this point, we have chosen the following selections:

1. Booted from the Install Media
2. Run a Recovery Shell
3. Recover an Unbootable System
4. Mount the Root Disk and Exit to a shell only

The next steps depend on what is wrong with our system.

If there are problems with our BDRA or LIF areas, we can easily fix them from here.

Be sure to execute the following commands as suggested:

```
The root disk has been mounted under /ROOT, if you want to chroot
to this directory type the commands:

# loadfile chroot
# chroot /ROOT /sbin/sh
#
```




© 2005 Hewlett-Packard Development Company, L.P.

```
# pwd
/
#
```

Remember, we need to **loadfile chroot** because the **chroot** command isn't currently in our 8MB ram-based file system.

Entering **chroot /ROOT /bin/sh makes** the **/ROOT** file system our root file system.

Let's activate the root volume group and mount our file systems.

```
# vgchange -a y vg00
# mountall
```

Now, depending on the problem, we'll obviously use different commands to resolve it.
If, for example, **HPUX** or **ISL** (in the LIF area) was missing or corrupt, we could do the following:

```
# mkboot /dev/rdsk/cxtydz
```

This will rebuild the LIF area with all new utilities.
Remember, the LIF contains: **HPUX**, **ISL**, **LABEL**, **AUTO**, and a few others.
The above command replaces these files, but what happens to the pointers in the **LABEL** file?
We'd better "refresh them".

```
# lvlnboot -b /dev/vg00/lvol1
# lvlnboot -s /dev/vg00/lvol2
# lvlnboot -r /dev/vg00/lvol3
# lvlnboot -d /dev/vg00/lvol2
```

Verifying:

```
# lvlnboot -v

Boot Definitions for Volume Group /dev/vg00:
Physical Volumes belonging in Root Volume Group:
        /dev/dsk/c1t15d0 (0/0/1/1.15.0) -- Boot Disk
Boot: lvol1     on:     /dev/dsk/c1t15d0
Root: lvol3     on:     /dev/dsk/c1t15d0
Swap: lvol2     on:     /dev/dsk/c1t15d0
Dump: lvol2     on:     /dev/dsk/c1t15d0, 0
```

It might not be a bad idea to check the contents of the **AUTO** file while we're here.

```
# lifcp /dev/rdsk/cxtydz:AUTO -

hpux
```

**hpux** is the default, but if that's not what you want, you can execute the following:

```
# mkboot -a "your_startup_string" /dev/rdsk/cxtydz
```


**© 2005 Hewlett-Packard Development Company, L.P.**

That should do it. Let's reboot and see if it comes back up

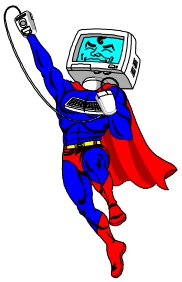
```
# reboot
```




© 2005 Hewlett-Packard Development Company, L.P.

## 4–19. SLIDE: Repairing Damage — Corrupt LABEL File

### Repairing Damage — Corrupt LABEL File



The recovery shell can be used to repair many things on the boot disk.
Example: Repairing a corrupted `LABEL` file or BDRA.

The procedure:

1. Boot from the support media.
2. From the Main Menu, select "Run a Recovery Shell"
3. Verify the hardware path to your boot disk
4. From the `Recovery` menu, select
   `a. Mount the root disk and exit to a shell only`
5. From the Recovery shell, execute the following commands:

```
# loadfile chroot
# chroot /ROOT /sbin/sh
# vgchange -a y /dev/vg00
# mountall
# lvlnboot -b /dev/vg00/lvol1
. . .
# reboot
```

## Student Notes

The steps to repair a corrupt `LABEL` file would be similar to the steps listed on the previous slide. The only difference: It might not be necessary to replace all the LIF utilities. You might be able to get away with the following steps:

After the Recovery Shell is loaded and you've selected:
`"Mount the root disk and exit to a shell only",`

```
# loadfile chroot
# chroot /ROOT /sbin/sh
#
# vgchange -a y vg00
# mountall
```

**`LABEL`** was corrupt , so all we need are the following:

```
# lvlnboot -b /dev/vg00/lvol1
# lvlnboot -s /dev/vg00/lvol2
# lvlnboot -r /dev/vg00/lvol3
# lvlnboot -d /dev/vg00/lvol2
```


**© 2005 Hewlett-Packard Development Company, L.P.**

Verifying:

```
# lvlnboot -v

Boot Definitions for Volume Group /dev/vg00:
Physical Volumes belonging in Root Volume Group:
        /dev/dsk/c1t15d0 (0/0/1/1.15.0) -- Boot Disk
Boot: lvol1     on:     /dev/dsk/c1t15d0
Root: lvol3     on:     /dev/dsk/c1t15d0
Swap: lvol2     on:     /dev/dsk/c1t15d0
Dump: lvol2     on:     /dev/dsk/c1t15d0, 0

# reboot
```




© 2005 Hewlett-Packard Development Company, L.P.

## 4–20. SLIDE: Repairing Damage — Missing Kernel

### Repairing Damage — Missing Kernel

In the event the kernel was accidentally deleted:

```
ISL booting hpux
Exec failed: cannot find /stand/vmunix or /vmunix

ISL> hpux ll

Ls
: disc(52.6.0;0)/.
dr-xr-xr-x      5     2     2     1024     ./
dr-xr-xr-x      5     2     2     1024     ../
drwxr-xr-x      2     0     0     8192     lost+found/
-rw-r--r--      1     0     3     2244     ioconfig
-rw-r--r--      1     0     3     19       bootconf
-rw-r--r--      1     0     3     961      system
drwxr-xr-x      4     0     3     1024     build/
drwxr-xr-x      5     0     3     1024     dlkm/
-r--r--r--      1     0     3     82       kernel
-rw-------      1     0     0     12       footconf

ISL>
```

## Student Notes

The **`hpux ll`** command shows that no bootable kernel (**`vmunix`**) or backup kernel (**`vmunix.prev`**) is present. A wise system administrator always maintains a bootable backup kernel.

If the **`hpux ll`** command returns an error, the most likely causes are:

- The LVM pointers to the boot volume are corrupt. This can be fixed by following the steps in the student notes under the previous slide (Repairing Damage – Corrupt **`LABEL`** File)
- The boot file system is corrupt. This may require reloading the boot file system, which in turn requires reinstalling the operating system.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–21. SLIDE: Repairing Damage — Rebuilding Kernel

### Repairing Damage — Rebuilding Kernel

**Boot from the support media.**

1. From the Main Menu, select "Run a Recovery Shell"
2. Verify the hardware path to your boot disk
3. From the `Recovery` menu, select
   `a. Mount the root disk and exit to a shell only`

From the Recovery shell, execute the following commands:

```
# loadfile chroot
# chroot /ROOT /sbin/sh
# vgchange -a y /dev/vg00
# fsck -P
# mountall
# cd /stand
# mk_kernel
# kmupdate
# /sbin/init.d/kmbuild stop
# reboot
```

## Student Notes

This example illustrates how to use the support shell to access your regular boot disk and to make your system look the way it does when you boot normally. The problem in this example is that there is no bootable kernel on the disk and it must be recompiled. You can solve many problems such as this, once your system is up and configured as if from a normal boot.

Here are the steps:

After the Recovery Shell is loaded and you've selected "Mount the root disk and exit to a shell only",

```
# loadfile chroot
# chroot /ROOT /sbin/sh

# vgchange -a y vg00
# fsck -P
# mountall
```




© 2005 Hewlett-Packard Development Company, L.P.

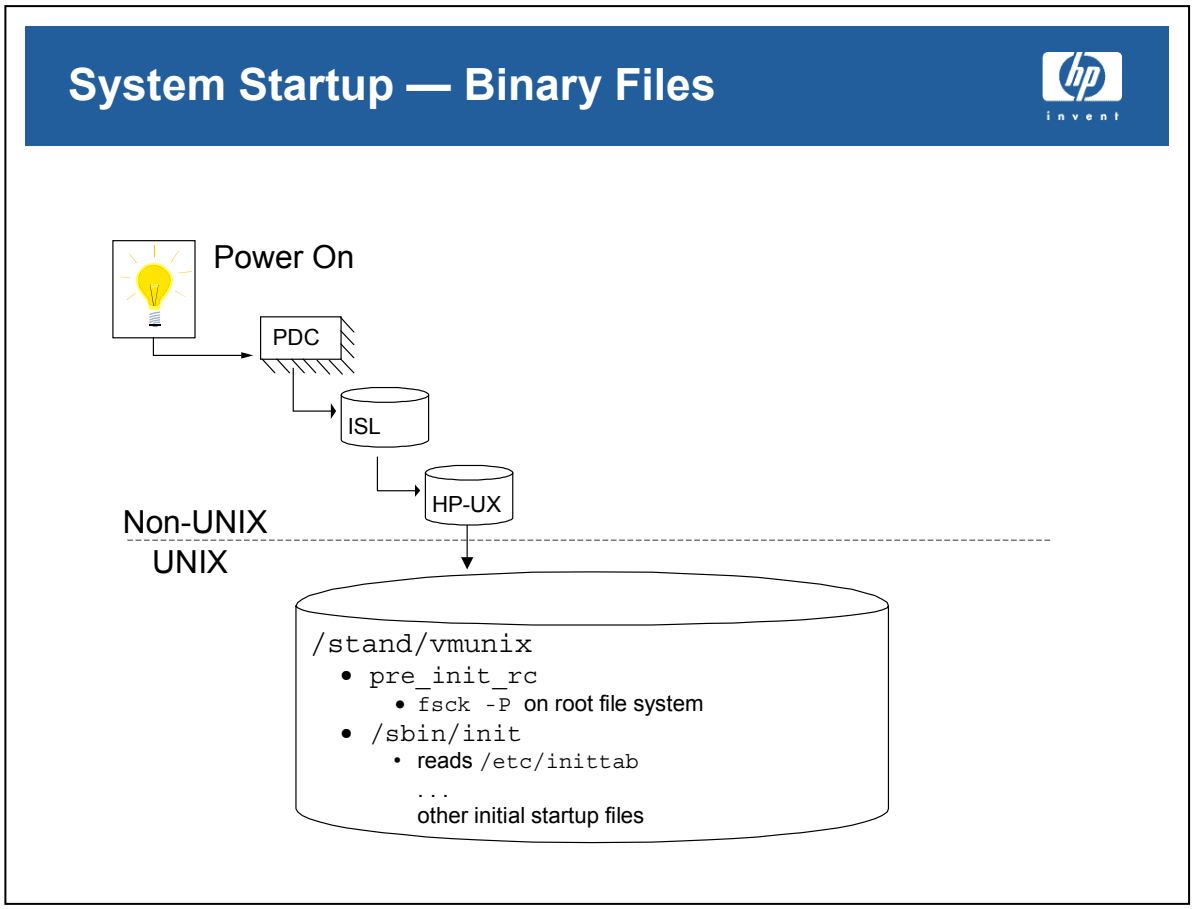
Now your system looks exactly as if you had just booted from your regular boot disk. You can fix all sorts of problems from here. For this example, we need to compile the kernel:

```
# cd /stand
# mk_kernel

Compiling /stand/build/conf.c ...
.
.
.

# kmupdate
# /sbin/init.d/kmbuild stop
# reboot
```

*NOTE:* Because of the **dlkm**'s in HP-UX 11, you should use **shutdown** after compiling the kernel rather than **reboot**. However, the sleight of hand we played earlier means that the system does not know what run level it is in, so most of the shutdown programs would fail. Since we manually execute **kmbuild**, we can get away with a reboot.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–22. SLIDE: System Startup — Binary Files




## Student Notes

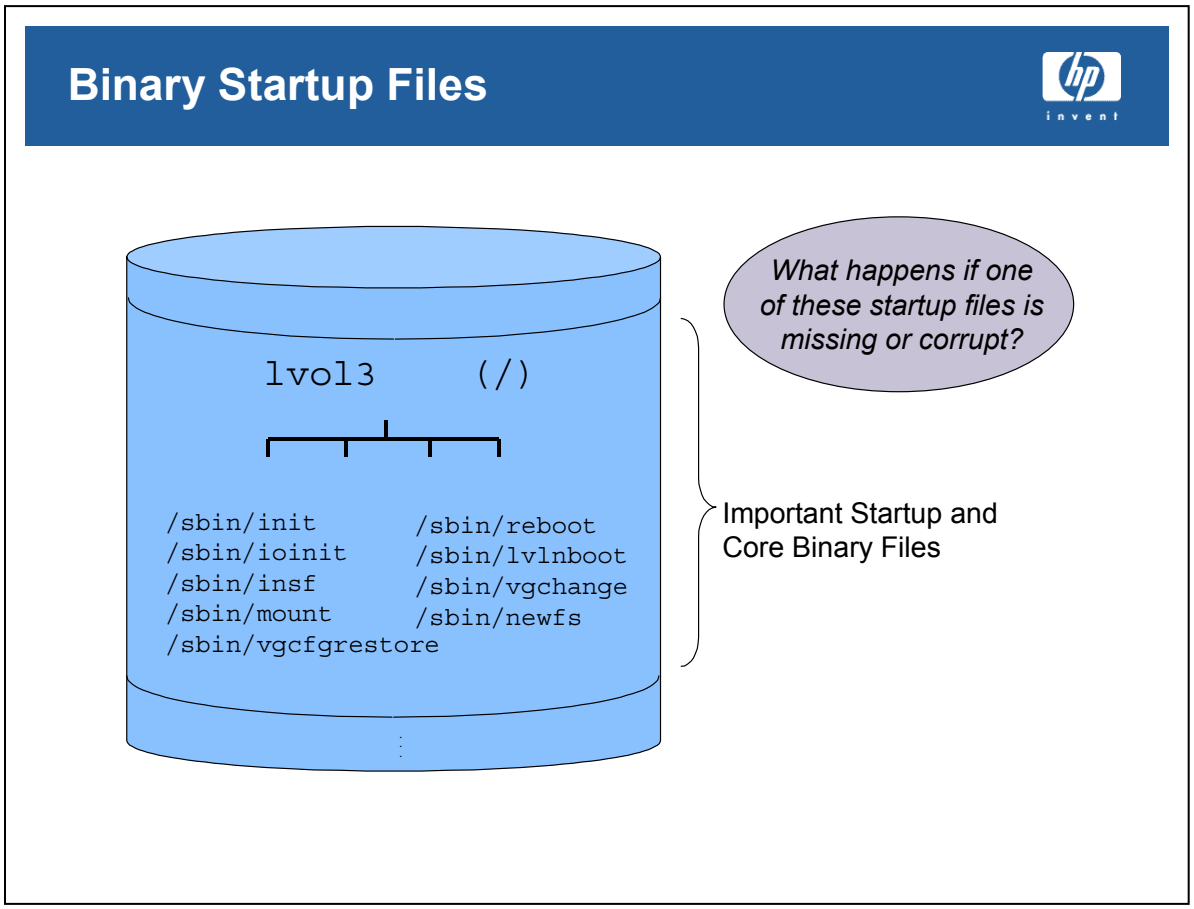
Once the operating system is loaded, there are a number of binary startup files that must execute for the boot process to complete. One critical binary file is **`/sbin/init`**. Its job is to start all other system and user processes that will ultimately bring the system to a multi-user state.

The **`pre_init_rc`** script performs a file system check on the root file system. Afterwards, the kernel mounts the root file system properly.




© 2005 Hewlett-Packard Development Company, L.P.

## 4–23. SLIDE: Binary Startup Files




## Student Notes

Sixty-one (61) startup and core command files are necessary to boot or fix your system. If certain binary startup files are corrupt or missing, the system will not fully boot.

One example of a critical binary startup file is **`/sbin/init`**. A missing or corrupt **`/sbin/init`** file can cause the system to fail to boot. If this file is corrupt, the system will PANIC.

These types of problems can be very tough to both diagnose and repair. Diligent monitoring of the output is the key. Also, always be mindful of the time it takes to resolve the problem. If it can be accomplished more quickly by re-Igniting the system, that may be the more prudent approach.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–24. SLIDE: Repairing Damage — Missing Binary File(s)

### Repairing Damage — Missing Binary File(s)

**Boot from the support media.**

1. From the Main Menu, select "Run a Recovery Shell"
2. Verify the hardware path to your boot disk
3. From the `Recovery` menu, select

   `a. Mount the root disk and exit to a shell only`

From the Recovery shell, execute the following commands:

```
# loadfile xxx
.....
```

## Student Notes

Here is an example where **`/sbin/init`** became corrupt:

```
Firmware Version  1.1

Duplex Console IO Dependent Code (IODC) revision 1

-------------------------------------------------------------------------
(c) Copyright 1995-1998, Hewlett-Packard Company, All rights reserved
-------------------------------------------------------------------------

Processor   Speed            State       Coprocessor State  I/D Cache
---------  --------   ------------------ -----------------  ---------
0          367 MHz    Active             Functional         512 KB/1 MB

  Central Bus Speed (in MHz) :        118

  Available memory (bytes)    : 268435456
  Good memory required (bytes):   21610496

  Primary boot path:    FWSCSI.6.0
  Alternate boot path:  SESCSI.0.0
```




© 2005 Hewlett-Packard Development Company, L.P.

```
  Console path:         SERIAL_1.9600.8.none
  Keyboard path:        PS2


Processor is starting auto boot process.

To discontinue, press any key within 10 seconds.

10 seconds expired.
Proceeding...

Trying Primary Boot Path
------------------------

Booting...
Boot IO Dependent Code (IODC) revision 1


HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL booting  hpux

Boot
: disk(8/0/19/0.6.0.0.0.0.0.0;0)/stand/vmunix
7503872 + 1757184 + 1261424 start 0x1e1f68
gate64: sysvec_vaddr = 0xc0002000 for 2 pages
NOTICE: nfs3_link(): File system was registered at index 4.
NOTICE: autofs_link(): File system was registered at index 6.
NOTICE: cachefs_link(): File system was registered at index 7.

    System Console is on the Built-In Serial Interface
Entering cifs_init...
Initialization finished successfully... slot is 9
Logical volume 64, 0x3 configured as ROOT
Logical volume 64, 0x2 configured as SWAP
WARNING: Logical volume for Dump expected but not found.
    Swap device table:  (start & size given in 512-byte blocks)
        entry 0 - major is 64, minor is 0x2; start = 0, size = 1048576
Starting the STREAMS daemons-phase 1
Checking root file system.
file system is clean - log replay is not required
Root check done.
Create STCP device files
         $Revision: vmunix:    vw: -proj    selectors: CUPI80_BL2000_1108
-c 'Vw for CUPI80_BL2000_1108 build' -- cupi80_bl2000_1108
'CUPI80_BL2000_1108'  Wed Nov  8 19:24:56 PST 2000 $
Memory Information:
    physical page size = 4096 bytes, logical page size = 4096 bytes
    Physical: 262144 Kbytes, lockable: 176148 Kbytes, available: 204588
Kbytes
```

*NOTE*: At this stage the kernel attempts to start **`init`** and fails! So the system panics.

```
proc_init icode: execve("/sbin/init") failed, errno 0x0000000d
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
proc_init icode: execve("/etc/init") failed, errno 0x0000000d



Stored message buffer up to panic:
gate64: sysvec_vaddr = 0xc0002000 for 2 pages
NOTICE: nfs3_link(): File system was registered at index 4.
NOTICE: autofs_link(): File system was registered at index 6.
NOTICE: cachefs_link(): File system was registered at index 7.
8 ccio
8/0 GSCtoPCI
8/0/19/0 c720
8/0/19/0.5 tgt
8/0/19/0.5.0 sdisk
8/0/19/0.6 tgt
8/0/19/0.6.0 sdisk
8/0/19/0.7 tgt
8/0/19/0.7.0 sctl
8/0/20/0 btlan
8/0/63 asio0
8/16 bus_adapter
8/16/4 asio0
8/16/5 c720
8/16/5.0 tgt
8/16/5.0.0 stape
8/16/5.7 tgt
8/16/5.7.0 sctl
8/16/0 CentIf
8/16/1 audio
8/16/7 ps2
10 ccio
10/0 GSCtoPCI
    Variable size pages used to map 1000 graf pages at ffffffff7000000
10/0/4/0 graph3
32 processor
49 memory
btlan: Initializing 10/100BASE-TX card at 8/0/20/0....

    System Console is on the Built-In Serial Interface
Entering cifs_init...
Initialization finished successfully... slot is 9
Logical volume 64, 0x3 configured as ROOT
Logical volume 64, 0x2 configured as SWAP
WARNING: Logical volume for Dump expected but not found.
    Swap device table:  (start & size given in 512-byte blocks)
        entry 0 - major is 64, minor is 0x2; start = 0, size = 1048576
    Dump device table:  (start & size given in 1-Kbyte blocks)
        entry 0000000000000000 - major is 31, minor is 0x6000; start =
117600, size = 524288
Starting the STREAMS daemons-phase 1
Create STCP device files
         $Revision: vmunix:    vw: -proj    selectors: CUPI80_BL2000_1108
-c 'Vw for CUPI80_BL2000_1108 build' -- cupi80_bl2000_1108
'CUPI80_BL2000_1108'  Wed Nov  8 19:24:56 PST 2000 $
Memory Information:
    physical page size = 4096 bytes, logical page size = 4096 bytes
```




© 2005 Hewlett-Packard Development Company, L.P.

```
    Physical: 262144 Kbytes, lockable: 176148 Kbytes, available: 204588
Kbytes
```

**init died with return value 256.**
**Please check for init's execute permission,**
**init's location and the root partition's location.**

```
System Panic:

linkstamp:          Thu Jul 5 05:36:48 EDT 2001
_release_version:   @(#)     $Revision: vmunix:    vw: -proj    selectors:
CUPI80_BL2000_1108 -c 'Vw for CUPI80_BL2000_1108 build' --
cupi80_bl2000_1108 'CUPI80_BL2000_1108'  Wed Nov  8 19:24:56 PST 2000 $
panic: init died

PC-Offset Stack Trace (read across, top of stack is 1st):
  0x001f3b88  0x0018b8c0  0x001b2414
  0x0013c1f4  0x00033f5c
End Of Stack

sync'ing disks (0 buffers to flush): (0 buffers to flush):
0 buffers not flushed
0 buffers still dirty



*** A system crash has occurred.  (See the above messages for details.)
*** The system is now preparing to dump physical memory to disk, for use
*** in debugging the crash.

*** The dump will be a SELECTIVE dump:  36 of 256 megabytes.
*** To change this dump type, press any key within 10 seconds.
*** Proceeding with selective dump.

*** The dump may be aborted at any time by pressing ESC.
*** Dumping:   0% complete (0 of 36 MB) (device 64:0x2)        ***
Dumping:   9% complete (4 of 36 MB) (device 64:0x2)        *** Dumping:
19% complete (8 of 36 MB) (device 64:0x2)        *** Dumping:  29%
complete (11 of 36 MB) (device 64:0x2)        *
** Dumping:  39% complete (15 of 36 MB) (device 64:0x2)        ***
Dumping:  49% complete (18 of 36 MB) (device 64:0x2)        *** Dumping:
59% complete (22 of 36 MB) (device 64:0x2)        *** Dumping:  69%
complete (25 of 36 MB) (device 64:0x2)
 *** Dumping:  79% complete (29 of 36 MB) (device 64:0x2)        ***
Dumping:  89% complete (32 of 36 MB) (device 64:0x2)        *** Dumping:
99% complete (36 of 36 MB) (device 64:0x2)        *** Dumping: 100%
complete (36 of 36 MB) (device 64:0x2)
   *** Dumping: 100% complete (36 of 36 MB).
*** System rebooting.
```

After the dump, the system tried to boot again, but we stopped it to boot to the Recovery Shell.

```
Firmware Version  1.1
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
Duplex Console IO Dependent Code (IODC) revision 1

-----------------------------------------------------------------------
(c) Copyright 1995-1998, Hewlett-Packard Company, All rights reserved
-----------------------------------------------------------------------

Processor   Speed            State         Coprocessor State  I/D Cache
---------   -----            -----         -----------------  ---------
   0        367 MHz          Active        Functional         512 KB/1 MB

  Central Bus Speed (in MHz) :        118

  Available memory (bytes)    : 268435456
  Good memory required (bytes):   21610496

  Primary boot path:    FWSCSI.6.0
  Alternate boot path:  SESCSI.0.0
  Console path:         SERIAL_1.9600.8.none
  Keyboard path:        PS2



Processor is starting auto boot process.

To discontinue, press any key within 10 seconds.
```

Interrupt the boot this time, so that we can boot to the recovery shell.

```
Boot terminated.


------- Main Menu -----------------------------------------------------

        Command                         Description
        -------                         -----------
        BOot [PRI|ALT|<path>]           Boot from specified path
        PAth [PRI|ALT|CON|KEY] [<path>] Display or modify a path
        SEArch [DIsplay|IPL] [<path>]   Search for boot devices

        COnfiguration [<command>]       Access Configuration menu/commands
        INformation [<command>]         Access Information menu/commands
        SERvice [<command>]             Access Service menu/commands

        DIsplay                         Redisplay the current menu
        HElp [<menu>|<command>]         Display help for menu or command
        RESET                           Restart the system
```

…
…
…

```
Boot from the Install Media (Recovery CD)


                 HP-UX CORE MEDIA SYSTEM RECOVERY
                            MAIN MENU
```




© 2005 Hewlett-Packard Development Company, L.P.

```
        s.  Search for a file
        b.  Reboot
        l.  Load a file
        r.  Recover an unbootable HP-UX system
        x.  Exit to shell


Select one of the above: r
Loading commands for recovery ....
-----------------------------------------------------------------


        DEVICE FILE VERIFICATION MENU

    This menu is used to specify the path of the root file system.
    When the information is correct, select 'a'.

        INFORMATION to verify:
          Device file used for '/'(ROOT) is c1t15d0
          The hardware path to disk is 0/0/1/1.15.0

        Select one of the following:
          a.  The above information is correct.
          b.  WRONG!! The device file used for '/'(ROOT) is incorrect.

          m.  Return to the 'HP-UX Recovery MENU.'
          x.  Exit to the shell.

     Selection: a
The disk is LVM
Setting boot and root device file for c1t15d0...


-----------------------------------------------------------------

                      HP-UX Recovery MENU

        Select one of the following:
        a.  Mount the root disk and exit to a shell only.
        b.  Recover the bootlif/os partitions.
        c.  Replace the kernel on the root file system.
        d.  Both Options: b and c
        v.  Read information about LVM Recovery (LVM.RECOVER)

        m.  Return to 'HP-UX Recovery Media Main Menu'.
        x.  Exit to the shell.

    Use this menu to select the level of recovery desired.

    Selection: a
Entering file system checking ...


-----------------------------------------------------------------
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
                FILE SYSTEM CHECK
                     MENU
       The file system check '/sbin/fs/vxfs/fsck -y
/dev/rdsk/c1t15d0s1lvm'
       will now be run.

       Select one of the following:
        a.  Run fsck -y.
        b.  Prompt for the fsck run string on c1t15d0s1lvm.
        m.  Return to the 'HP-UX Recovery MENU.'

        Selection: a



** /dev/rdsk/c1t15d0s1lvm
** Last Mounted on /ROOT/stand
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
50 files, 0 icont, 28995 used, 114578 free (114 frags, 14308 blocks)
Going to mount /dev/dsk/c1t15d0s1lvm to /ROOT/stand as read only
Umount /ROOT/stand
Doing fsck on /ROOT file system
/sbin/fs/vxfs/fsck -y /dev/rdsk/c1t15d0s2lvm
log replay in progress
replay complete - marking super-block as CLEAN
 Mount /ROOT file system
/sbin/fs/vxfs/mount /dev/dsk/c1t15d0s2lvm /ROOT
Mount /ROOT/stand again
/sbin/fs/hfs/mount  -F hfs /dev/dsk/c1t15d0s1lvm /ROOT/stand

==================================================================
The root disk has been mounted under /ROOT, if you want to chroot
to this directory type the commands:
     loadfile chroot
     chroot /ROOT /sbin/sh
```

We could execute the above 2 commands, but let's explore a little first.

```
# ls -l
total 114
-rw-------   1 root       sys        39026 Jan 16 14:06 .sh_history
drwxr-xr-x   2 root       sys           24 Jan 16 13:52 CDROM
drwxr-xr-x  17 root       sys         1024 Jan 16 14:03 ROOT
d---------   2 bin        bin           24 Jul 23 17:44 core
drwxr-xr-x   8 root       root        1024 Jan 16 13:52 dev
drwxr-xr-x   2 root       root        1024 Jan 16 13:52 dev.old
drwxr-xr-x   2 bin        bin           24 Jul 23 17:44 disc
dr-xr-xr-x   4 bin        bin         1024 Jan 16 13:52 duped_root
dr-xr-xr-x   4 bin        bin         1024 Jan 16 13:53 etc
drwxr-xr-x   2 root       root        4096 Jul 23 17:44 lost+found
drwxr-xr-x   3 bin        bin           40 Jul 23 17:44 opt
```




© 2005 Hewlett-Packard Development Company, L.P.

```
drwxr-xr-x   4 bin        bin         1024 Jan 16 13:58 sbin
drwxr-xr-x   2 bin        bin         1024 Jan 16 13:53 stand
drwxrwxrwx   2 bin        bin         1024 Jan 16 13:52 tmp
drwxr-xr-x   7 bin        bin           96 Jul 23 17:44 usr
drwxr-xr-x   5 bin        bin           60 Jul 23 17:44 var
```

Remember, the **/ROOT** directory that now resides in our / directory is our system boot disk. The other files, under **/**, are the files and directories in the 8MB RAM-based file system created by the Recovery CD.

Let's check our free space:

```
# bdf
Filesystem           kbytes    used   avail %cap iused   ifree iused
Mounted on
/                      8532    8175     357  96%   190    2690   7%   ?
```

Let's load **chroot**, as we might want it later

```
# loadfile chroot
# bdf
Filesystem           kbytes    used   avail %cap iused   ifree iused
Mounted on
/                      8532    8247     285  97%   192    2688   7%   ?
```

Looks like we are not going to be able to load many commands/files into the RAM-based file system at this rate.

We saw from the panic messages that "init died", and in looking at **/ROOT/sbin** (effectively the **/sbin** directory on our root disk) , we saw that its permissions did not allow it to execute. Furthermore, its size was zero.

So let's load init.

```
# loadfile init

/: write failed, file system is full
/: file system full
ERROR:   tar: ./sbin/init: HELP - extract write error: No space left on
device
         (errno = 28)
NOTE:    Retrying loadfile command...
```

Looks like we may have loaded too much already.
With the current Recovery CD, there is very little room to add any additional files into the nearly full 8MB RAM-based file system, so we need to carefully choose what we load.


**© 2005 Hewlett-Packard Development Company, L.P.**

Looking again at our **/** directory:

```
# ls -l
total 114
-rw-------   1 root       sys        39026 Jan 16 14:06 .sh_history
drwxr-xr-x   2 root       sys           24 Jan 16 13:52 CDROM
drwxr-xr-x  17 root       sys         1024 Jan 16 14:03 ROOT
d---------   2 bin        bin           24 Jul 23 17:44 core
drwxr-xr-x   8 root       root        1024 Jan 16 13:52 dev
drwxr-xr-x   2 root       root        1024 Jan 16 13:52 dev.old
drwxr-xr-x   2 bin        bin           24 Jul 23 17:44 disc
dr-xr-xr-x   4 bin        bin         1024 Jan 16 13:52 duped_root
dr-xr-xr-x   4 bin        bin         1024 Jan 16 13:53 etc
drwxr-xr-x   2 root       root        4096 Jul 23 17:44 lost+found
drwxr-xr-x   3 bin        bin           40 Jul 23 17:44 opt
drwxr-xr-x   4 bin        bin         1024 Jan 16 13:58 sbin
drwxr-xr-x   2 bin        bin         1024 Jan 16 13:53 stand
drwxrwxrwx   2 bin        bin         1024 Jan 16 13:52 tmp
drwxr-xr-x   7 bin        bin           96 Jul 23 17:44 usr
drwxr-xr-x   5 bin        bin           60 Jul 23 17:44 var
```

Let's remove some "unnecessary" files (these files might be necessarily under other circumstances, but we can get away with removing them here). Remember, we're not affecting the boot disk. These files are in RAM. If we go too far, we can always "**loadfile**" them back in.

```
# rm -rf /dev.old
# bdf
Filesystem           kbytes    used   avail %cap iused   ifree iused
Mounted on
/                      8532    8383     149  98%   188    2692   7%   ?
# rm -rf /lost+found
# bdf
Filesystem           kbytes    used   avail %cap iused   ifree iused
Mounted on
/                      8532    8379     153  98%   187    2693   6%   ?
```

Let's see if that was enough.

```
# loadfile init
```

Just for grins, let's see where **`init`** was placed

```
# find / -name init

/sbin/sh: find:  not found
# loadfile find

# find / -name init
/sbin/init
```

**`/sbin/init`** is in the RAM-based file system and we need to get it onto the root disk

```
# cp /sbin/init /ROOT/sbin/init
```




© 2005 Hewlett-Packard Development Company, L.P.

```
# chroot /ROOT /sbin/sh

# ls -l /sbin/init
-r-xr-xr-x   1 root       sys         299008 Jan 16 14:20 /sbin/init
```

**Success! Let's reboot.**

```
# reboot
Shutdown at 14:21 (in 0 minutes)
System shutdown time has arrived
```

And the system comes up normally.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–25. SLIDE: Corrupted PVRA/VGRA Structures on the Boot Disk

### Corrupted PVRA/VGRA Structures on the Boot Disk

If the PVRA or VGRA structures are corrupted, the volume group *cannot* be activated.

A special way to boot the system (LVM maintenance mode) can be used to bypass the normal LVM structures.

- LVM maintenance mode puts the system into single-user mode.
- LVM maintenance mode is used to repair damaged LVM data structures.

```
ISL> hpux -lm

ISL> hpux -lm (56/52.6.0;0)/vmunix
```

## Student Notes

Success of the maintenance mode boot depends on whether anything else on the disk was corrupted. For your boot to succeed, the LIF header and LIF volume, including the **LABEL** file, must not have been damaged, and the root file system must be intact.

To boot your system using a maintenance mode boot, interrupt the boot sequence and enter ISL:

```
ISL> hpux -lm
```

If you cannot boot your system using a maintenance mode boot, the LIF area, **LABEL** file, or the root file system is corrupt. You can try booting from a different disk if you have a mirror copy of the root file system somewhere. Otherwise, you will probably need to reinstall the boot information from the recovery shell, or in a worst case, by reinstalling HP-UX (hopefully from your **make_net/tape_recovery image**).




© 2005 Hewlett-Packard Development Company, L.P.

| | |
|---|---|
| ***WARNING:*** | **When you have booted your system in maintenance mode:**<br>• **You may need to activate the root volume group (for example, to repair the boot area), but use caution.**<br>• **Do not switch to multiuser mode (Do not use the `/sbin/init 2` command). You can corrupt your root file system if you do!** |

When the system is booted in maintenance mode, the system needs to locate the root file system. On a healthy system, the BDRA gives the details of the location of the root logical volume. In this case, the BDRA cannot be used, so an alternative must be found.

In HP-UX 11.0 and after, the file **`/stand/rootconf`** was added to give the details of the root logical volume. **`/stand`** is accessible to the HPUX secondary loader so the kernel can fetch files from here before mounting the root file system.

The **`rootconf`** file is maintained by the **`lvlnboot -c`** command, which is normally run at boot time by **`/sbin/ioinitrc`**.

During maintenance-mode boots, since the root volume group is not activated, **`lvlnboot -c`** does not update **`/stand/rootconf`**. For separate root/boot configurations, maintenance-mode boot will fail if **`/stand/rootconf`** does not already exist with the correct location of the root volume.

When a new volume group with separate boot and root volumes is created, the first boot must be a normal boot (versus maintenance-mode boot), so that **`/stand/rootconf`** is created.

Mirrored root logical volumes should start at the same offset on each physical volume so that the location stored in **`/stand/rootconf`** works for maintenance-mode boots from any mirror.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 4–26. SLIDE: LVM Maintenance Mode

### LVM Maintenance Mode

LVM Maintenance Mode

- The `/dev/vg00` volume group is *not* activated.
- The root and boot logical volumes are mounted, even though the `vg00` volume group is not active.
- This mode allows the LVM header information to be restored to the boot disk:

```
# vgcfgrestore -n /dev/vg00 /dev/rdsk/c0t6d0
# reboot -n
```

*CAUTION*: While in LVM maintenance mode, do not bring the system into multiuser mode.

## Student Notes

When the PVRA or VGRA structures on the boot disk are corrupt, boot your system to LVM maintenance mode. This mode is a special way of booting your system that bypasses the normal LVM structural checks.

Upon entering LVM maintenance mode, issue the **`vgcfgrestore`** command. This command restores the valid LVM configuration information to affected physical volume(s). In this example, root volume group information is being restored to the boot disk. Afterwards, the system is rebooted.




© 2005 Hewlett-Packard Development Company, L.P.

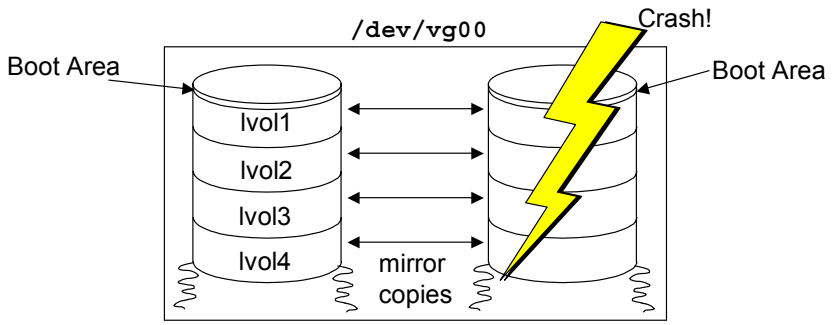
## 4–27. SLIDE: Booting without Quorum

### Booting without Quorum

Booting without quorum allows your system to come up when:

- The root volume group does not meet quorum requirement.
- The disk containing the root file system is functional.

```
ISL> hpux -is -lq
```




## Student Notes

The default LVM quorum requirement is that 51 percent of the disks that make up a volume group be present to activate the volume group. The volume group must be activated for you to have access to the logical volumes it contains. If your root volume group has more than one disk in it, failure to meet the quorum requirement will stop your system from booting normally. The boot will fail and an error will be displayed on the console.

The **-lq** option to the HP-UX utility will override the normal quorum requirement and allow the system to boot with less than 51% of the volume group. It is recommended that you also add the **-is** option to the HP-UX utility so your system comes up in single-user mode.

```
ISL> hpux -is -lq
```

If this boot also fails, your root disk has a corrupt **LABEL** file, boot area, or kernel file.

To improve availability, many system administrators keep a spare disk. In the example, the boot disk is mirrored, but one of the disks failed.


© 2005 Hewlett-Packard Development Company, L.P.

While disabling quorum allows the volume group to be activated even if less than half of the disk drives are available, the administrator should avoid making LVM configuration changes. It is possible that the next time the volume group is activated, the changes could be lost.




© 2005 Hewlett-Packard Development Company, L.P.

## 4–28. SLIDE: Recovering a Mirrored Boot Disk

### Recovering a Mirrored Boot Disk

1. If not already booted, boot from the remaining bootable mirror as described previously
2. Force HP-UX to recognize the replaced disk
   `# ioscan -C disk; insf -C disk`
3. Restore the LVM reserved areas
   `# vgcfgrestore -n vg00 /dev/rdsk/c0t1d0`
4. Repopulate the LIF area
   `# mkboot /dev/rdsk/c0t1d0`
5. Change the AUTO file contents to hpux –lq
   `# mkboot -a "hpux -lq" /dev/rdsk/c0t1d0`
6. Reactivate the VG to recognize the recovered disk
   `# vgchange -a y vg00`
7. Resynchronize mirrors
   `# vgsync vg00`
8. Verify the recovered mirror using the previously described verification procedures

Restore LVM structures with vgcfgrestore

Boot area must be manually restored

## Student Notes

What happens if a boot disk mirror fails? If the boot disk is mirrored, you should be able to boot your system using the remaining mirror. Once the system is booted, you can replace the failed disk and restore the necessary structures using the procedure below. **`vgcfgrestore`** only restores LVM structures! The boot LIF area and associated files must be manually recreated using **`mkboot`** and **`lvlnboot`**.

1. If not already booted, boot from the remaining bootable mirror as described on the previous slide.
2. Force HP-UX to recognize the replaced disk

   `# ioscan -C disk; insf -C disk`
3. Restore the LVM reserved areas

   `# vgcfgrestore -n vg00 /dev/rdsk/c0t1d0`
4. Repopulate the LIF area

   `# mkboot /dev/rdsk/c0t1d0`


© 2005 Hewlett-Packard Development Company, L.P.

5. Change the **AUTO** file contents to **hpux -lq**

   ```
   # mkboot -a "hpux -lq" /dev/rdsk/c0t1d0
   ```

6. Reactivate the VG to recognize the recovered disk

   ```
   # vgchange -a y vg00
   ```

7. Resynchronize mirrors

   ```
   # vgsync vg00
   ```




© 2005 Hewlett-Packard Development Company, L.P.

## 4–29. SLIDE: Module Objectives

### Module Objectives

You should now be able to do the following:

- Describe PA-RISC system startup
- Describe the boot disk structure
- Describe potential boot failures
- Understand the `LABEL` file and the `BDRA`
- Understand the operation of the `lvlnboot` command
- Describe the operation of the recovery shell and its capabilities
- Understand the implications of missing or corrupt binary startup files
- Describe the recovery process for a missing kernel

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 4–30. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. Write the command that lists the contents of the LIF area, after the system is fully up (at run level 2 or 3).

2. Write the command that displays the contents of the **`AUTO`** file, after the system is fully up (at run level 2 or 3).

3. From the ISL prompt, write the command that lists the contents of the **`AUTO`** file.

4. Write a single command that will restore a damaged LIF area when the system is fully up (at run level 2 or 3).

5. List 5 binary startup or core command files.

6. Which binary file ranks as one of the most important?




© 2005 Hewlett-Packard Development Company, L.P.

## 4–31. LAB: Booting: Corrupt Stable Storage and Bad LIF Area

**NOTE: The following labs should only be performed on a PA-RISC system**

## Part 1: Corrupt Stable Storage

This lab is designed to give you experience troubleshooting a corrupt stable storage.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.

2. Execute the **`mod4.lab1.sh`** script.

   ```
   # cd /home/labs
   # ./mod4.lab1.sh
   ```

   This will simulate the condition such that, when rebooting your machine, the boot and startup processes will fail, because stable storage is corrupt. Use the knowledge gained in the current module to fix the problem.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 2: Bad LIF Area

This lab is designed to give you experience troubleshooting a bad LIF area.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.

2. Execute the **`mod4.lab2.sh`** script

   ```
   # cd /home/labs
   # ./mod4.lab2.sh
   ```

   This script will simulate the condition such that, when rebooting your machine, the boot and startup processes will fail because of a bad LIF area. Using the knowledge gained in the current module, fix this problem.




© 2005 Hewlett-Packard Development Company, L.P.

# 4–32. LAB: Recovering an Unbootable System — Missing kernel

## Directions

This lab gives you experience troubleshooting a system when all kernel files have been accidentally removed or corrupted.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.
2. Execute the following sequence of commands:

```
# cd /stand
# rm vmunix*

# cd /
# shutdown -ry 0
```

   Using the knowledge gained in the current module, recover the system.


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 5 — Booting HP-UX 11i v2 via the EFI

## Objectives

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

- Describe the major steps in the HP-UX 11i v2 boot process
- Describe the major components of an HP-UX 11i v2 boot disk
- Initiate an HP-UX 11i v2 boot from a command prompt, the MP, or the EFI shell
- Navigate and manage EFI file systems
- Use the EFI **`help`** utility
- Boot from the default boot disk
- Boot from an alternate boot disk
- Boot from a DVD or CDROM
- Boot from an Ignite-UX install/recovery server
- Boot from an Ignite recovery tape
- Boot from an alternate kernel
- Boot to single-user, LVM maintenance, or tunable maintenance mode
- Configure a mirrored 11i v2 boot disk
- Recover a failed mirrored 11i v2 boot disk


© 2005 Hewlett-Packard Development Company, L.P.

## 5–1. SLIDE: Reviewing the Module Objectives

### Reviewing the Module Objectives

By the end of the module, you should be able to…

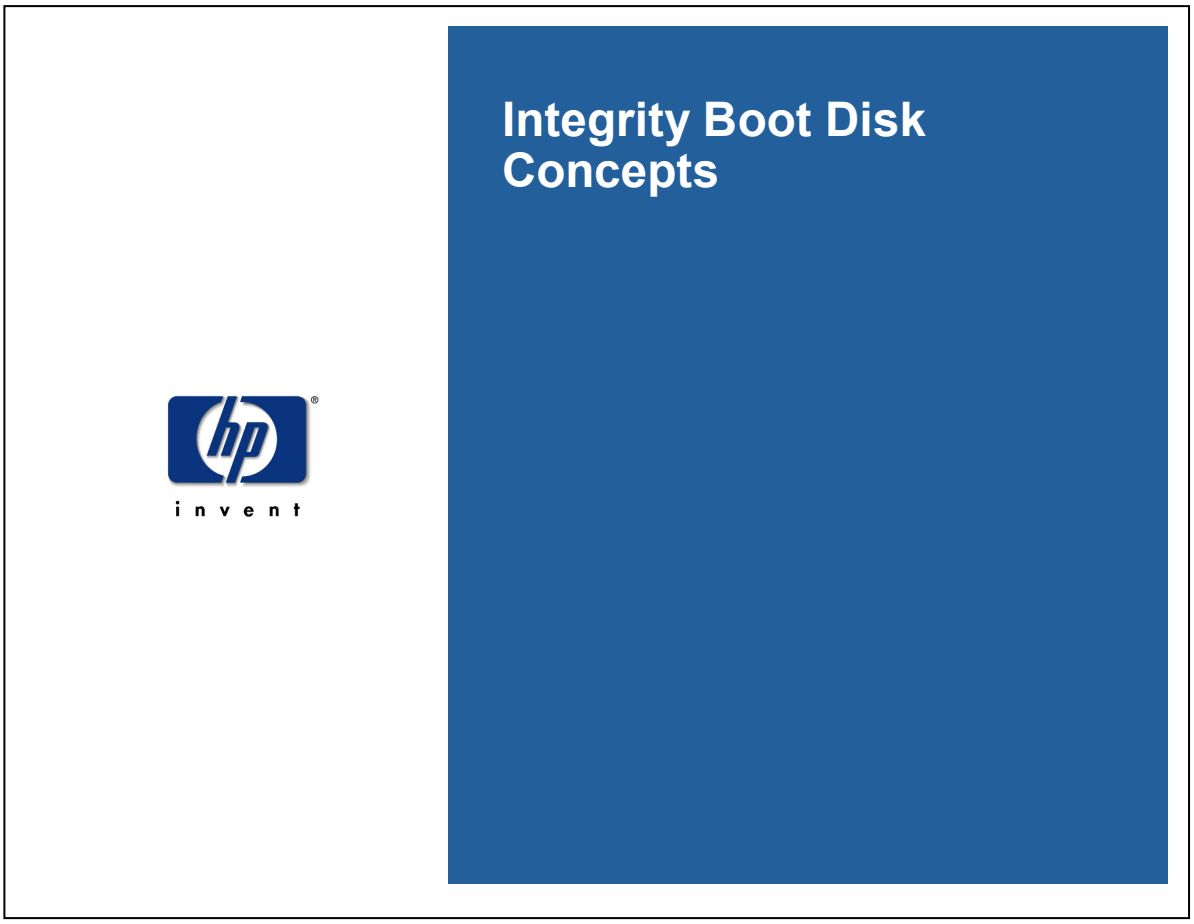
- Describe the major components of an HP-UX 11i v2 LVM boot disk
- Initiate an HP-UX 11i v2 boot from a command prompt, the MP, or the EFI shell
- Navigate and manage EFI devices and file systems
- Obtain system configuration information using EFI commands
- Manage the EFI boot menu using EFI commands
- Boot from the default boot disk
- Boot from an alternate boot disk
- Boot from a system recovery tape
- Boot from a DVD or CDROM
- Boot from an Ignite-UX install/recovery server
- Boot from an Ignite-UX recovery tape
- Boot from an alternate kernel
- Boot to single-user, LVM maintenance, or tunable maintenance mode
- Configure a mirrored 11i v2 boot disk
- Recover a failed mirrored 11i v2 boot disk

## Student Notes

The Integrity system boot process is significantly different from the boot process that PA-RISC system administrators are accustomed to. This chapter describes the major steps in the Integrity boot process, and provides cookbooks for many common boot procedures.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–2. SLIDE: Integrity/EFI Boot Concepts

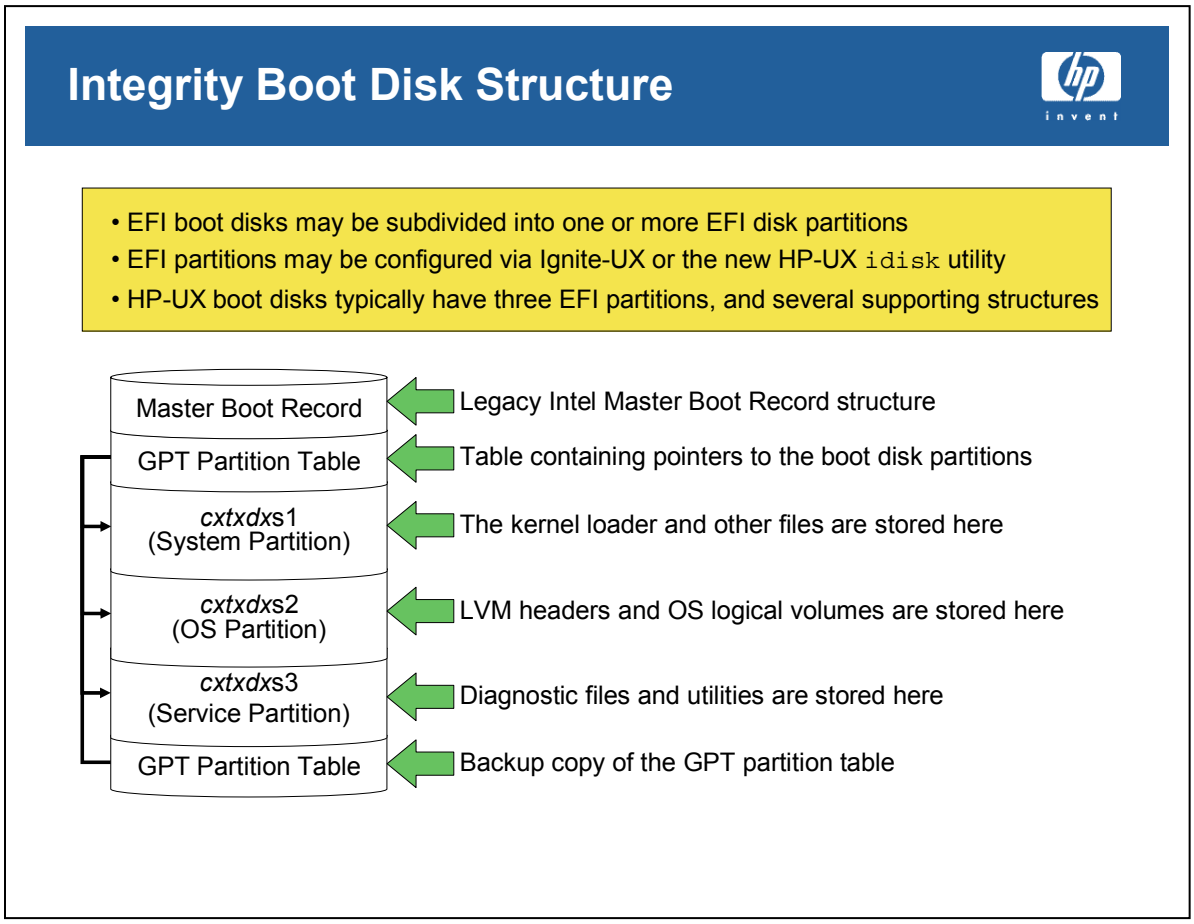



### Student Notes

Before we can discuss the commands required to interact with the Integrity boot process, you need to understand the role of the major players in the boot process.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–3. SLIDE: Integrity Boot Disk Structure




## Student Notes

Firmware components such as the PAL, SAL, and EFI reside in NVRAM. However, other important players in the boot process such as the **`hpux.efi`** OS loader and the **`/stand/vmunix`** kernel executable, reside on the boot disk. The next few slides discuss the layout and structure of the boot disk in detail.

Integrity boot disks are subdivided into EFI disk partitions. The EFI partitions on an HP-UX boot disk are initially created by Ignite-UX, but can also be manually viewed and managed via the new **`idisk`** utility, which will be described later in the chapter.

| | |
|---|---|
| *NOTE*: | EFI disk partitions are not related to HP-UX cell-based node partitions; nPars subdivide a system's resources into independent OS instances. An EFI partition is simply a portion of disk space. |


**© 2005 Hewlett-Packard Development Company, L.P.**

HP-UX boot disks have at least two and possibly three EFI partitions.

MBR: The Master Boot Record (MBR) is located at the beginning of the disk. This is a legacy Intel component that is ignored by the EFI firmware.

GPT: Every EFI partition is assigned a Globally Unique IDentifier (GUID). The partition GUIDs and locations are recorded in the EFI GUID Partition Table (GPT). This is a critical structure that is replicated at both the beginning and end of the boot disk.

EFI System Partition: At a minimum, the EFI system partition contains the OS loader that is responsible for loading the OS in memory during the boot process. On HP-UX boot disks, this OS loader is called **`\efi\hpux\hpux.efi`**. The **`\efi\hpux\auto`** file, which stores the system boot string, and several troubleshooting utilities are also stored in the system partition on HP-UX systems. Other OSes require different files in the system partition.

On HP-UX machines, the system partition is represented by device file `/dev/rdsk/`*`cxtxdx`***`s1`**.

The EFI partition uses a file system structure which is similar to the FAT32 file system structure traditionally used in the Microsoft Windows Operating System. Several new utilities have been added to HP-UX to allow administrators to view and manage files in these EFI file systems. These will be discussed later in the chapter.

OS Partition: The OS partition contains an encapsulated LVM boot physical volume, including all of the structures expected on an LVM boot disk: a PVRA, VGRA, BDRA, BBRA, and physical extents allocated to the logical volumes in vg00. In HP-UX, this partition is represented by device file `/dev/rdsk/`*`cxtxdx`***`s2`**.

The contents of this partition may be accessed using traditional LVM commands such as **`pvcreate`** and **`pvdisplay`**.

HPSP: The HP Service Partition (HPSP) is a FAT32 file system that may be used to store offline diagnostic utilities. In HP-UX, this partition is represented by device file `/dev/rdsk/`*`cxtxdx`***`s3`**.

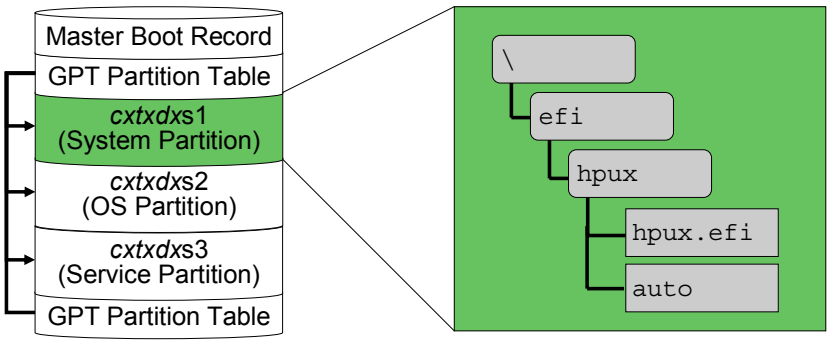
Since this partition contains an EFI file system, the partition must be accessed using the new HP-UX commands for accessing and managing EFI partitions and file systems.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–4. SLIDE: Integrity Boot Disk System Partition Structure

### Integrity Boot Disk System Partition Structure

- Every Integrity boot disk has an EFI system partition
- The system partition typically contains a FAT32 file system
- This file system contains two key files (and several other supporting files):
  - `\efi\hpux\auto`, a text file which determines the kernel boot string
  - `\efi\hpux\hpux.efi`, an executable which loads the kernel in memory




## Student Notes

Every Integrity boot disk contains an EFI system partition. The system partition (represented by the `/dev/rdsk/`*`cxtxdx`***`s1`**. device file in HP-UX) contains a variety of EFI executable files and utilities which are used to load the kernel, and facilitate troubleshooting and maintenance activities.

The EFI partition uses a file system structure which is identical to the file system structure traditionally used in the Microsoft Windows Operating System. Several new utilities have been added to HP-UX to allow administrators to view and manage files in these EFI file systems. These will be discussed later in the course.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Files and Tools in the HP-UX EFI System Partition

On an HP-UX boot disk, the most important files in the system partition are:

- **`\efi\hpux\hpux.efi`**, which is responsible for loading and running the kernel, and
- **`\efi\hpux\auto`**, which contains the kernel boot string that **`hpux.efi`** uses to determine which mode and kernel to boot.

The EFI is intended to be "extensible", so you may find other executables in the EFI system partition, too. Currently, HP includes a small suite of network troubleshooting tools in the **`\efi\hp\tools\network`** directory:

- **`ftp.efi`**, an FTP client
- **`ifconfig.efi`**, for enabling a LAN interface before booting HP-UX
- **`ping.efi`**, which is similar to the standard UNIX ping command
- **`route.efi`**, which is similar to the standard UNIX route command

For instructions on using these tools, run the **`\efi\hp\tools\network\inet.nsh`** EFI script from the EFI shell prompt.

You can write and compile your own EFI executable tools and scripts, too, using an EFI compiler from Intel. See the **`tools`** link on **`http://www.intel.com/technology/efi/`** for more information.

## Files and Tools in other EFI System Partitions

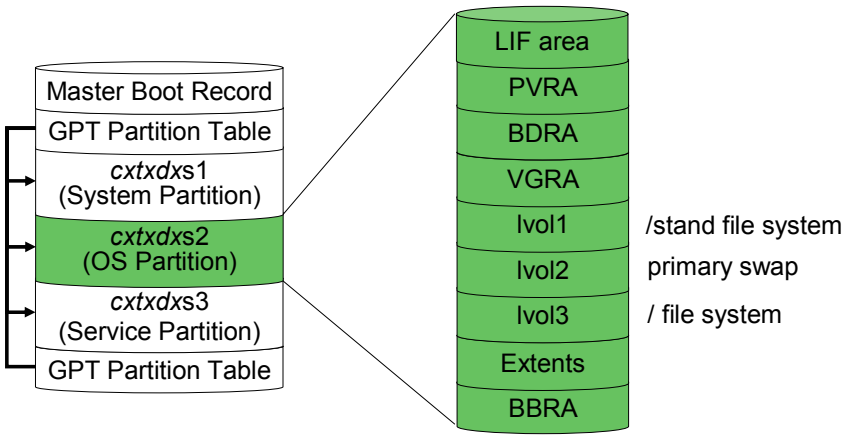
The EFI is also intended to be OS-neutral. Thus, whether you are running HP-UX, Windows, or Linux on your Integrity server, your boot disk will have an EFI system partition. The EFI executables and files within the partition vary from OS to OS. For instance, the system partition on a Linux Integrity boot disk contains the **`lilo.efi`** (LInux LOader) kernel loader rather than **`hpux.efi`**.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–5. SLIDE: Integrity Boot Disk OS Partition Structure

### Integrity Boot Disk OS Partition Structure

- Every Integrity boot disk has at least one operating system partition
- The HP-UX OS partition is similar to an LVM boot physical volume or VxVM rootdg




## Student Notes

The HP-UX OS partition contains an encapsulated LVM boot disk, including all the structures one would expect to see on a traditional Whole Disk, LVM, or VxVM boot disk. The system partition may be accessed using the `/dev/rdsk/`*`cxtxdx`*`s2` device file.

If you choose to configure your system using the Whole Disk option for the boot disk, the OS partition will contain a LIF area, a VxFS root file system, and primary swap.

If you choose to configure your system using LVM on the boot disk, the OS partition will contain a LIF area, PVRA, VGRA, BDRA, BBRA, and physical extents allocated to the **/** file system and primary swap logical volumes in vg00. You can view the contents of an LVM OS partition using the traditional LVM **`pvdisplay`** command.

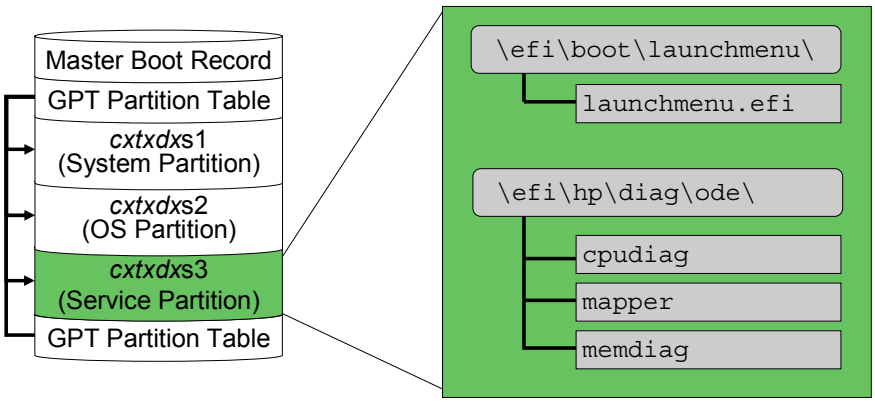
If you choose to configure your system using VxVM on the boot disk, the OS partition will contain a VxVM root disk group. You can view the contents of the VxVM OS partition using the traditional VxVM **`vxdisk`** and **`vxprint`** commands.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–6. SLIDE: Integrity Boot Disk HP Service Partition Structure

### Integrity Boot Disk HP Service Partition Structure

- An HP-UX boot disk may optionally contain a HP Service Partition (HPSP)
- The HPSP typically contains a 400MB FAT32 file system
- Diagnostic utilities may be stored in this partition




## Student Notes

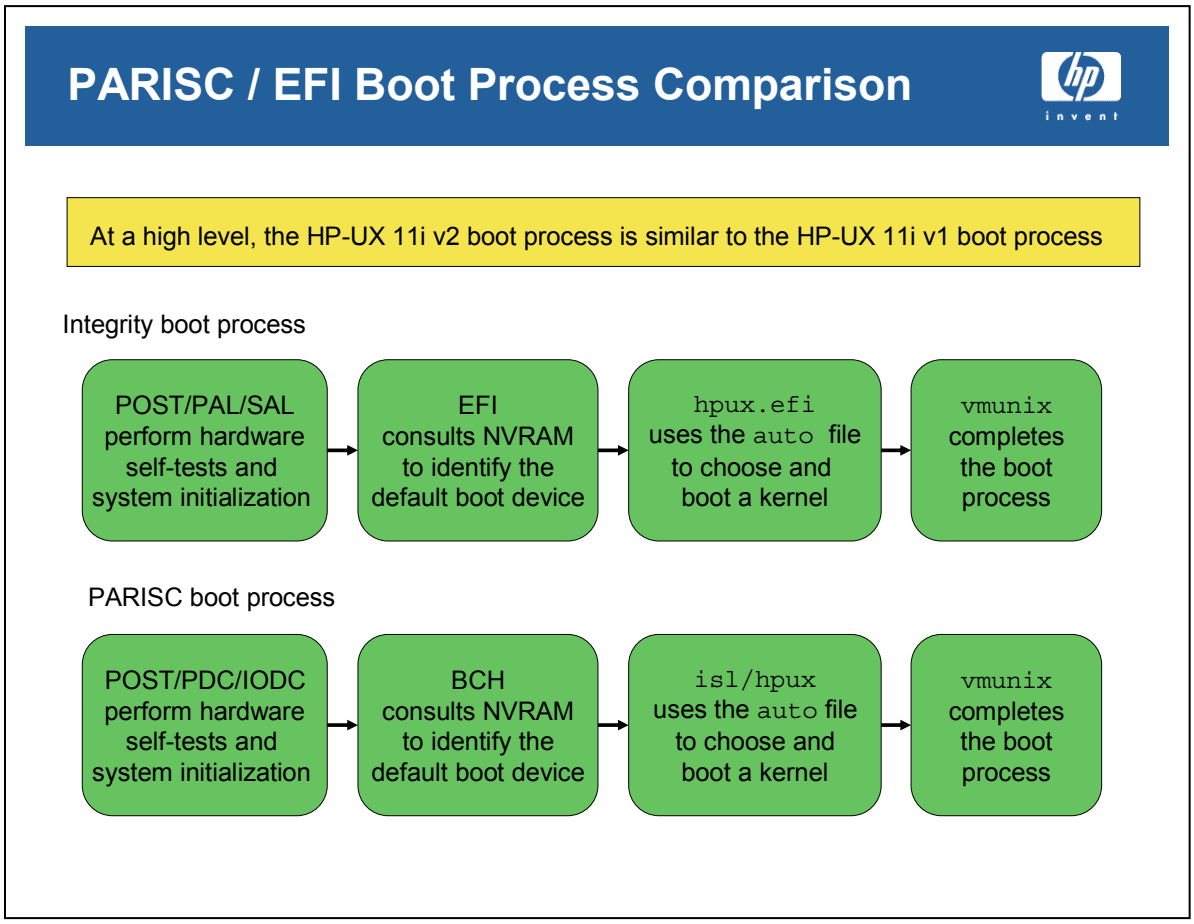
A new tool called e-Diagnostics ("e-Diag" for short) is now available for HP-UX 11i v2. The program allows you to troubleshoot hardware problems on a computer that cannot boot an operating system, and test the system hardware more thoroughly than online diagnostics and support tools can.

e-Diagnostics may be launched directly from the Offline Diagnostics and Utilities CDROM, which is included with every HP Integrity server, or from a 400MB, FAT32 Service Partition on the system boot disk. Running e-Diagnostics from the disk-based service partition provides much better performance than the CDROM-based solution.

The slide shows just a few of the many files that are typically found in the HPSP or on the Diagnostics and Utilities CDROM. The intuitive, menu-based interface for e-Diag may be executed by running **`\efi\boot\launchmenu\launchmenu.efi`** from either the Diagnostics and Utilities CD, or from the HPSP.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–7. SLIDE: PARISC / EFI Boot Process Comparison




## Student Notes

So far in this chapter we've discussed several firmware and disk-based components that play a part in the Integrity boot process. This slide shows how these players work together to boot an Integrity system to a fully functional state. For the sake of comparison, the slide also shows the corresponding steps in the PA-RISC boot process.

### PAL/SAL firmware perform hardware self-tests and system initialization

First, the PAL/SAL firmware performs a variety of processor and hardware self-test and initialization tasks.

On the PA-RISC platform, the Power On Self Test (POST), Processor Dependent Code (PDC), and I/O Dependent Code (IODC) perform these tasks.

### EFI boot manager consults NVRAM to identify the default boot device

Next, the EFI boot manager consults NVRAM, displays a menu of potential boot devices, and proceeds to load a boot loader from the default boot device if autoboot is enabled. Alternatively, the administrator can use the EFI boot menu to select an alternate boot source or use the EFI shell to interact with the system before the autoboot proceeds. On the PA-RISC platform, the Boot Console Handler (BCH) serves a similar role.


**© 2005 Hewlett-Packard Development Company, L.P.**

### `hpux.efi` uses the `auto` file to choose and load a kernel

On HP-UX Integrity systems, the **`hpux.efi`** kernel loader consults the **`\efi\hpux\auto`** file on the EFI system partition to determine which kernel and mode to boot. Typically, the **`auto`** file suggests booting **`vmunix`** to multi-user mode.

On PA-RISC systems, the Initial System Loader (**`ISL`**) uses the **`HPUX`** kernel loader and **`AUTO`** file in the `LIF` area to load the kernel in memory. The **`ISL`** and **`HPUX`** executables and the **`AUTO`** file are all stored in the boot area on the PA-RISC boot disk.

### `vmunix` Brings the System to a Fully Functional State

The kernel (typically **`/stand/vmunix`**) then scans the hardware, mounts the root file system, and starts the **`init`** daemon. The **`init`** daemon starts the daemons and services necessary to bring the system up to multi-user mode. This portion of the process is identical on PA-RISC and Integrity systems.

### Autoboot versus Manual Boot

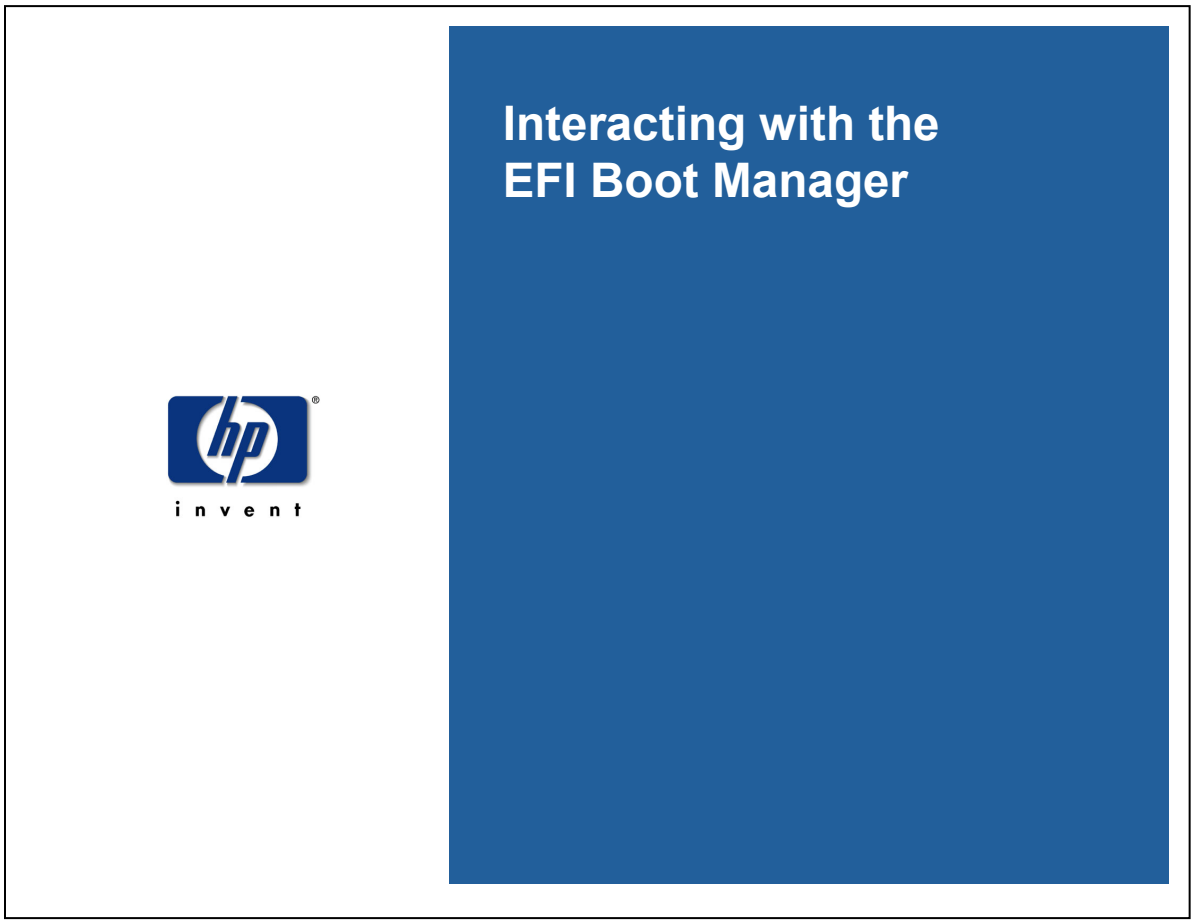
Usually, the boot procedure described on the above occurs without intervention from the System Administrator. The system boots from the primary boot disk to multi-user mode using the default kernel. This is known as an **autoboot**.

However, in some cases you may wish to override the autoboot process with a **manual boot**:

- To re-install the operating system from CD.
- To boot from an alternate boot disk if the primary boot disk is corrupt.
- To boot from a system recovery archive if no boot disk is available.
- To boot from an alternate named kernel configuration or to tunable maintenance mode.
- To boot to single-user mode to perform administrative tasks.
- To boot to single-user mode if the root password is lost.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–8. SLIDE: Interacting with the EFI Boot Manager




## Student Notes

Up to this point, this chapter has focused on EFI concepts and structures. The next few slides describe the practical steps required to boot an HP-UX system via the EFI boot manager.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–9. SLIDE: Initiating an Integrity System Boot

### Initiating an Integrity System Boot

The Integrity boot process may be initiated from a command prompt, the EFI shell, or the MP.

Initiate the boot process from multi-user mode:
```
# shutdown -r
```

Initiate the boot process from single-user mode:
```
# reboot
```

Initiate the boot process from the EFI:
```
Shell> reset or
HPUX> reset
```

Initiate the boot process from the MP:
```
MP:CM> rs   (system reset) or…
MP:CM> tc   (transfer of control)
```

Monitor the boot process from the MP:
```
MP> vfp     (monitor hardware self-checks and early boot messages)
MP> co      (monitor EFI and HPUX boot messages)
```

## Student Notes

There are several ways to initiate an Integrity system boot.

You can initiate the boot process from multi-user mode via the **`shutdown`** command.

```
# shutdown -r
```

You can initiate the boot process from single-user mode via the **`reboot`** command:

```
# reboot
```

You can initiate the boot process from the EFI shell via the **`reset`** command:

```
Shell> reset
```

The **`reset`** command is also recognized by the HPUX kernel loader:

```
HPUX> reset
```


© 2005 Hewlett-Packard Development Company, L.P.

| | |
|---|---|
| **Warning!** | Use these MP commands with care! Resetting a booted, running system will risks file system corruption. Whenever possible, use the **`shutdown`** command rather than these MP commands to shutdown your system. |

From the MP, you can initiate a system boot in either of two ways. The **`rs`** command does a system reset. The **`tc`** command does a reset, too, but first dumps the contents of physical memory to disk. On entry-level systems, these commands are available from the MP main menu. On cell-based systems, these commands are available from the command menu.

`MP:CM> rs` *(system reset) or…*
`MP:CM> tc` *(transfer of control)*

On cell-based systems there is an additional way to boot from the MP that can be used to boot a new or halted partition

```
MP:CM> bo (boot)
```

No matter how you initiate the boot process, you can monitor progress via the MP virtual front panel and console interface.

`MP> vfp` *(monitor hardware self-checks and early boot messages)*
`MP> co` *(monitor EFI and HPUX boot messages)*


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–10. SLIDE: Choosing a Boot Device

### Choosing a Boot Device

In the early stages of the boot process, the EFI boot manager presents a menu of potential boot sources.

- You may use the arrow keys to select a desired boot source, or …
- Allow the EFI to choose the first available source

```
EFI Boot Manager ver 1.10 [14.60]  Firmware ver 1.61 [4241]
Please select a boot option

    HP-UX Primary Boot Disk
    HP-UX HA Alternate Boot
    EFI Shell [Built-in]
    Boot option maintenance menu
    Security/Password Menu

    Use ^ and v to change option(s). Use Enter to select an option
    Default boot selection will be booted in 10 seconds
```

## Student Notes

After you initiate the system boot process, the system will run some preliminary system initialization functions, and eventually display an EFI boot manager menu screen similar to the menu on the slide above.

Uninterrupted, the EFI will automatically boot the first choice on the menu. If it fails, the EFI proceeds to the next choice on the list. If the second choice fails, the EFI proceeds to the third.

If you want to explicitly choose an alternate device, simply use the up/down arrow keys to select the desired device and press `Return` before the ten second timeout expires.

Note that the boot manager menu is OS-neutral. If your system contains an HP-UX boot disk, a Linux boot disk, and a Windows boot disk, the boot manager can be configured to display all three boot sources.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–11. SLIDE: Choosing an Alternate Kernel/Mode

### Choosing an Alternate Kernel/Mode

After you select a boot device, the EFI boot manager loads `\efi\hpux\hpux.efi`, which reads the `\efi\hpux\auto` file to determine which kernel and mode to boot. You can interrupt the autoboot process if you wish to select an alternate kernel/mode.

Interrupt the HPUX kernel loader when prompted

```
HP-UX Boot Loader for IA64  Revision 1.71
Press Any Key to interrupt Autoboot
\EFI\HPUX\AUTO ==> boot vmunix
Seconds left till autoboot - 10  [Escape]
  Type 'help' for help
```

Choose a kernel and mode, and boot

| | |
|---|---|
| `HPUX> ll` | *List contents of /stand* |
| `HPUX> showauto` | *View the contents of the auto file* |
| `HPUX> boot backup` | *Boot the backup kernel* |
| `HPUX> boot vmunix -v` | *Boot vmunix to multi-user mode* |
| `HPUX> boot vmunix -is` | *Boot to single-user mode* |
| `HPUX> boot vmunix -lq` | *Boot without checking LVM quorum* |
| `HPUX> boot vmunix -lm` | *Boot to LVM maintenance mode* |
| `HPUX> boot vmunix -tm` | *Boot to tunable maintenance mode* |
| `HPUX> exit` | *Return to the boot manager* |

## Student Notes

After you select an HP-UX boot device to boot, the EFI loads and runs the **`\efi\hpux\hpux.efi`** OS loader. If the selected boot device contains a **`\efi\hpux\auto`** file, **`hpux.efi`** reads this file to determine which kernel to load, and which mode to boot. You should see a message similar to the following:

```
HP-UX Boot Loader for IPF  --  Revision 1.73
Press Any Key to interrupt Autoboot
\EFI\HPUX\AUTO ==> boot vmunix
Seconds left till autoboot - 10
   Type 'help' for help
```

After 10 seconds, **`hpux.efi`** automatically boots the system based on the contents of the **`auto`** file.


**© 2005 Hewlett-Packard Development Company, L.P.**

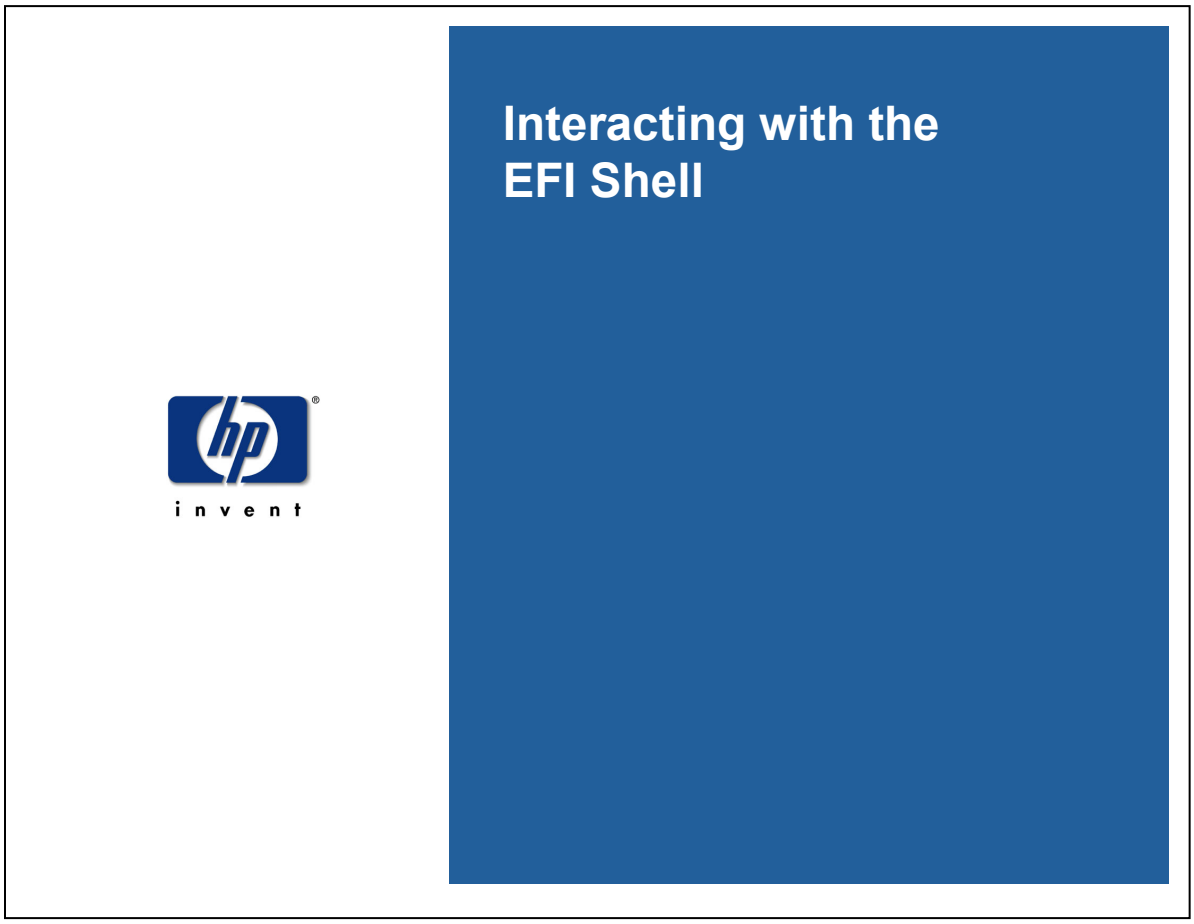
If there is no **auto** file, or if you interrupt the autoboot sequence by pressing **Escape**, you will find yourself at an **HPUX>** prompt. At this point, you can manually interact with the kernel loader. The commands available at the HP-UX OS loader prompt are similar to the options on the PA-RISC **hpux** command. A sampling of available commands are listed below. For more information, type **help** at the **HPUX>** prompt.

| | |
|---|---|
| `HPUX> ll` | *List contents of /stand* |
| `HPUX> showauto` | *View contents of the \efi\hpux\auto file* |
| `HPUX> boot backup` | *Boot the backup kernel* |
| `HPUX> boot vmunix -v` | *Boot vmunix to multi-user mode* |
| `HPUX> boot vmunix -is` | *Boot to single-user mode* |
| `HPUX> boot vmunix -lq` | *Boot without checking LVM quorum* |
| `HPUX> boot vmunix -lm` | *Boot to LVM maintenance mode* |
| `HPUX> boot vmunix -tm` | *Boot to tunable maintenance mode* |
| `HPUX> exit` | *Return to the EFI boot manager* |

*NOTE:* Tunable maintenance mode boots the kernel using system failsafe kernel parameters values. This will be described in greater detail in the kernel tuning module later in the course.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–12. SLIDE: Interacting with the EFI Shell




## Student Notes

The last few slides explained how to select a boot source via the menu-based EFI boot manager. You can also select an alternate boot source — and much more! — via the EFI shell. The next few slides introduce some of the important features available via the EFI shell.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–13. SLIDE: Accessing and Leaving the EFI Shell

### Accessing and Leaving the EFI Shell

- The EFI shell provides an opportunity to interact with and customize the boot process
- You can access the EFI shell via the boot manager
- You can return to the boot manager by typing `exit` at the EFI shell prompt

```
EFI Boot Manager ver 1.10 [14.60]  Firmware ver 1.61 [4241]

Please select a boot option

    HP-UX Primary Boot
    HP-UX HA Alternate Boot
    EFI Shell [Built-in]
    Boot option maintenance menu
    Security/Password Menu

Shell> exit        # to return to the boot manager
```

## Student Notes

The EFI shell provides a very flexible mechanism for interacting with the EFI firmware. From the shell you can:

- Boot from an arbitrary boot source, or even an alternate OS
- Customize the boot menu and autoboot settings
- Create customized batch scripts to execute at run time, and much more!

The EFI shell command set includes commands for managing files and directories in EFI partitions, managing EFI and boot manager settings, storing and accessing variables. It even provides looping and branching functionality! For a complete list of EFI shell commands, see the EFI shell command reference guide at **`http://www.intel.com/software/products/college/efishell/images/efi_shell_cmnd_1_1.pdf`**.

The built-in shell commands may be supplemented by custom EFI shell scripts, and binary executables built using Intel's EFI byte code compiler.


© 2005 Hewlett-Packard Development Company, L.P.

## Accessing and Leaving the EFI Shell

To access the EFI shell, interrupt the EFI autoboot process, and select the EFI shell option from the boot manager menu. The shell will automatically display a list of potential boot devices, run the **`startup.nsh`** shell startup script from the default boot disk, and display a **`Shell>`** prompt.

```
EFI Boot Manager ver 1.10 [14.60] Firmware ver 1.61 [4241]
Please select a boot option
    HP-UX Primary Boot
    HP-UX Alternate Boot
    EFI Shell [Built-in]
    Boot option maintenance menu
    Security/Password Menu

Loading.: Built-in EFI Shell
EFI Shell version 1.10 [14.60]
Device mapping table
fs0  : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB)
fs1  : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB)
blk0 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)
blk1 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)
blk2 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)
blk3 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB)
blk4 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part2,SigB)
blk5 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB8)
startup.nsh> echo -off
  setting hpux path(\EFI\HPUX)...
  type 'fs[x]:' where x is your bootdisk (0, 1, 2...)
  type 'hpux' to start hpux bootloader
Shell>
```

To return to the boot manager menu, simply type **`exit`** at the EFI shell prompt.

```
Shell> exit
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–14. SLIDE: EFI Shell Command Overview

### EFI Shell Command Overview

Several types of commands are available at the EFI shell prompt

EFI shell built-ins

Explanation: Core EFI shell commands specified by Intel
Location: Included in the EFI firmware on all vendors' Integrity systems
Examples: `cls`, `date`, `ls`, `cd`

Pre-OS System Startup Environment (POSSE) commands

Explanation: HP value-added EFI commands, designed to ease migration from PARISC
Location: Included in the EFI firmware on all HP Integrity servers
Examples: `info`, `fru`, `lanconfig`, `lanboot`

EFI Binaries

Explanation: C programs compiled using Intel's EFI compiler and libraries
Location: Usually stored as `.efi` files on the EFI system partition
Examples: `hpux.efi`, `install.efi`, `ping.efi`

EFI Shell Scripts

Explanation: Shell scripts calling other EFI, POSSE, and binary commands
Location: Usually stored as `.nsh` files on the EFI system partition
Examples: `startup.nsh`, `inet.nsh`

## Student Notes

The EFI supports four different classes of commands which may be used to interact with and customize the Integrity server boot process.

### EFI Shell Built-Ins

Intel specifies a core set of built-in EFI shell commands, which should be available on all vendors' Itanium 2-based systems. These commands are designed primarily to view, access, and manage EFI partitions, and load and run other EFI programs. The following list of built-in EFI shell is excerpted from Intel's EFI Shell Command Reference, which is available online at **`http://www.intel.com/software/products/college/efishell/images/efi_shell_cmnd_1_1.pdf`**. You can also type **`help`** ***`command`*** at the EFI shell prompt to view more detailed syntax descriptions.

| | |
|---|---|
| **`alias`** | Displays, creates, or deletes aliases in the EFI shell |
| **`attrib`** | Displays or changes the attributes of files or directories |
| **`bcfg`** | Displays/modifies the driver/boot configuration |
| **`break`** | Executes a debugger break point |
| **`cd`** | Displays or changes the current directory |
| **`child`** | Displays the device tree starting at a handle |


© 2005 Hewlett-Packard Development Company, L.P.

| | |
|---|---|
| `cls` | Clears the standard output with an optional background color |
| `comp` | Compares the contents of two files |
| `connect` | Binds an EFI driver to a device and starts the driver |
| `cp` | Copies one or more files/directories to another location |
| `date` | Displays the current date or sets the date in the system |
| `dblk` | Displays the contents of blocks from a block device |
| `dh` | Displays the handles in the EFI environment |
| `disconnect` | Disconnects one or more drivers from a device |
| `dmem` | Displays the contents of memory |
| `dmpstore` | Displays all NVRAM variables |
| `echo` | Displays messages or turns command echoing on or off |
| `edit` | Edits an ASCII or UNICODE file in full screen. |
| `EfiCompress` | Compress a file |
| `EfiDecompress` | Decompress a file |
| `err` | Displays or changes the error level |
| `exit` | Exits the EFI Shell |
| `for/endfor` | Executes commands for each item in a set of items |
| `getmtc` | Displays the current monotonic counter value |
| `goto` | Makes batch file execution jump to another location |
| `guid` | Displays all the GUIDs in the EFI environment |
| `help` | Displays commands list or verbose help of a command |
| `hexedit` | Edits with hex mode in full screen |
| `if/endif` | Executes commands in specified conditions |
| `load` | Loads EFI drivers |
| `LoadPciRom` | Loads a PCI Option ROM image from a file |
| `ls` | Displays a list of files and subdirectories in a directory |
| `map` | Displays or defines mappings |
| `memmap` | Displays the memory map |
| `mkdir` | Creates one or more directories |
| `mm` | Displays or modifies MEM/IO/PCI |
| `mode` | Displays or changes the mode of the console output device |
| `mount` | Mounts a file system on a block device |
| `mv` | Moves one or more files/directories to destination |
| `OpenInfo` | Displays the protocols on a handle and the agents |
| `pause` | Prints a message and suspends for keyboard input |
| `pci` | Displays PCI devices or PCI function configuration space |
| `reset` | Resets the system |
| `rm` | Deletes one or more files or directories |
| `set` | Displays, creates, changes or deletes EFI environment variables |
| `setsize` | Sets the size of a file |
| `stall` | Stalls the processor for some microseconds |
| `time` | Displays the current time or sets the time of the system |
| `touch` | Sets the time and date of a file to the current time and date |
| `type` | Displays the contents of a file |
| `unload` | Unloads a protocol image |
| `ver` | Displays the version information |


**© 2005 Hewlett-Packard Development Company, L.P.**

## Pre-OS System Startup Environment (POSSE) Commands

HP Integrity servers include a number of additional firmware-based EFI commands that approximate the functionality provided by the PA-RISC BCH menus and assist in troubleshooting situations. These commands are only available on HP's Integrity servers. The list below provides a brief description of the POSSE commands that are included in the initial EFI release; for a full description of each command, type **`help`** ***`command`*** at the EFI shell prompt.

| | |
|---|---|
| **`autoboot`** | View or set autoboot timeout variable |
| **`baud`** | Set serial port com settings |
| **`boottest`** | Set/View BootTest bits |
| **`cpuconfig`** | Deconfigure or reconfigure cpus |
| **`devices`** | Displays the list of devices being managed by EFI drivers |
| **`devtree`** | Displays the tree of devices that follow the EFI Driver Model |
| **`drivers`** | Displays the list of drivers that follow the EFI Driver Model |
| **`drvcfg`** | Invokes the Driver Configuration Protocol |
| **`drvdiag`** | Invokes the Driver Diagnostics Protocol |
| **`errdump`** | View/Clear logs |
| **`esiproc`** | Make an ESI call |
| **`hexedit`** | Edits with hex mode in full screen mode |
| **`info`** | Display hardware information |
| **`lanaddress`** | Display core I/O MAC address |
| **`lanboot`** | Performs boot over lan from EFI Shell |
| **`monarch`** | View or set the monarch processor |
| **`optload`** | Lists all optional ROM-based efi drivers and applications |
| **`PalProc`** | Make a PAL call. |
| **`pdt`** | View or set pdt |
| **`SalProc`** | Make a SAL call |
| **`tftp`** | TFTP to a bootp/dhcp enabled unix boot server |
| **`Verbose`** | Controls progress output on console |
| **`vol`** | Displays volume information of the file system |
| **`xchar`** | Turn on/off extended character features |

## EFI Binaries

Intel also sells a C compiler and library product that allows vendors and developers to write and compile their own EFI binaries. These binaries are typically stored in the EFI partition on disk and are identified by **`.efi`** file extensions. Several EFI binaries are included on HP's Integrity systems. Here are a few of the EFI programs you might encounter while using HP-UX:

| | |
|---|---|
| **`\efi\hpux\hpux.efi`** | The HP-UX kernel loader |
| **`\install.efi`** | The install menu utility on DVDs |
| **`\efi\boot\launchmenu\launchmenu.efi`** | e-Diagnostics menu utility |
| **`\efi\hp\tools\network\ifconfig.efi`** | Similar to standard UNIX **`ifconfig`** |
| **`\efi\hp\tools\network\ping.efi`** | Similar to standard UNIX **`ping`** |
| **`\efi\hp\tools\network\route.efi`** | Similar to standard UNIX **`route`** |
| **`\efi\hp\tools\network\tcpipv4.efi`** | TCP/IP drivers |


**© 2005 Hewlett-Packard Development Company, L.P.**

## EFI Shell Scripts

The EFI also supports a scripting language for batch mode processing. The EFI scripting language supports, loops, branches, variables, I/O redirection, and many other features found in other scripting languages. The Intel EFI reference guide at **`http://www.intel.com/software/products/college/efishell/images/efi_shell_cmnd_1_1.pdf`**. describes the script syntax and commands in detail.

EFI scripts always include a **`.nsh`** extension, and are usually stored in the EFI system partition on disk. There are two EFI scripts that are included on HP's Integrity servers.

**`\efi\hp\tools\network\inet.nsh`**
Script that may be modified to configure a LAN interface card's IP address and routing table without booting HP-UX

**`\startup.nsh`**
Script that executes automatically when the EFI shell is launched.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–15. SLIDE: Executing EFI Commands

### Executing EFI Commands

- In order to execute an EFI command, just type the command name
  - File extensions may be omitted
  - Full pathnames aren't required if the EFI path variable is set properly
  - Commands are not case-sensitive
  - Most internal commands support a `-b` option to view one screen at a time
- The EFI supports several UNIX-like shell features
  - Use the arrow keys for editing and command line recall
  - Use `*`, `?`, and `[]` for filename generation
  - Use `>`, `>>`, `2>`, and `2>>` for output and error redirection
- Use the `help` command to learn more about EFI commands
  - Use `help -a` to view a list of all commands
  - Use `help` *`command`* to learn more about a specific command
  - Use `help bch` to view EFI equivalents for familiar BCH commands

## Student Notes

The EFI shell provides a simple interface for executing commands.

- In order to execute an EFI command, just type the command name
  - File extensions may be omitted
  - Full pathnames aren't required if the EFI path variable is set properly
  - Commands are not case-sensitive
  - Most internal commands support a **`-b`** option to view one screen at a time
- The EFI supports several UNIX-like shell features
  - Use the arrow keys for editing and command line recall
  - Use **`*`**, **`?`**, and **`[]`** for filename generation
  - Use **`>`**, **`>>`**, **`2>`**, and **`2>>`** for output and error redirection
- Use the **`help`** command to learn more about EFI commands
  - Use **`help -a`** to view a list of all commands
  - Use **`help`** ***`command`*** to learn more about a specific command
  - Use **`help bch`** to view EFI equivalents for familiar BCH commands

### About the EFI Shell PATH Variable


© 2005 Hewlett-Packard Development Company, L.P.

Much like the UNIX shell, the EFI shell searches directories in a **`path`** variable to find executables. You can view or modify the contents of EFI shell variables (including **`path`**) via the EFI shell **`set`** command.

```
Shell> set path \efi\hpux
Shell> set
       path : .;\efi\hpux
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–16. SLIDE: Viewing System Information via the EFI

### Viewing System Information via the EFI

The EFI shell provides several commands for viewing your system configuration

| | |
|---|---|
| `Shell> info sys` | model string, product number, serial number, etc. |
| `Shell> info boot` | boot and self-test settings, MAC addresses |
| `Shell> info fw` | firmware revisions |
| `Shell> info spu` | CPU quantity and type |
| `Shell> info mem` | DIMM quantity and type |
| `Shell> info cache` | cache quantity and type |
| `Shell> info -b all` | all available information |
| `Shell> lanaddress` | MAC address information |
| `Shell> time` | system time |
| `Shell> date` | system date |

## Student Notes

The next few slides list some of the more frequently used EFI commands in a bit more detail. The commands shown on this slide may be used to view system configuration information.

| | |
|---|---|
| **`Shell> info sys`** | *model string, product number, serial number, etc.* |
| **`Shell> info boot`** | *boot and self-test settings, MAC addresses* |
| **`Shell> info fw`** | *firmware revisions* |
| **`Shell> info spu`** | *CPU quantity and type* |
| **`Shell> info mem`** | *DIMM quantity and type* |
| **`Shell> info cache`** | *cache quantity and type* |
| **`Shell> info -b all`** | *all available information* |
| **`Shell> lanaddress`** | *MAC address information* |
| **`Shell> time`** | *system time* |
| **`Shell> date`** | *system date* |

To learn more about viewing a list of EFI devices, see the next slide.


© 2005 Hewlett-Packard Development Company, L.P.

## Sample Output from the `info` Command

The **`info -b all`** command is perhaps the most useful of the commands listed above since it provides the most comprehensive output. Sample output is included below. The **`-b`** option displays the output one page at a time.

```
Shell> info -b all

SYSTEM INFORMATION

Date/Time:  Jan  6, 2004  08:56:31  (20:04:01:06:08:56:31)
Manufacturer:  hp
Product Name:  server rx2600
Product Number:  A6870A
Serial Number:  US21729721
UUID:  G0
System Bus Frequency:  200 MHz

PROCESSOR MODULE INFORMATION

      # of                L3      L4      Family/
CPU   Logical             Cache   Cache   Model          Processor
Slot  CPUs     Speed      Size    Size    (hex.)   Rev   State
----  -------  --------   ------  ------  -------  ---   ------------
0      1      900 MHz    3 MB    None    1F/00    B1    Active
1      1      900 MHz    3 MB    None    1F/00    B1    Active



MEMORY INFORMATION

     ---- DIMM A -----  ---- DIMM B -----
      DIMM   Current     DIMM   Current
---  ------ ----------  ------ ----------
 0    512MB     Active   512MB     Active
 1    512MB     Active   512MB     Active
 2    512MB     Active   512MB     Active
 3    512MB     Active   512MB     Active
 4    ----               ----
 5    ----               ----

   Active Memory    : 4096 MB
   Installed Memory : 4096 MB
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
I/O INFORMATION

BOOTABLE DEVICES

  Order  Media Type   Path
  -----  ----------   ---------------------------------------
     1        CDROM   Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)
     2    HARDDRIVE   Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB8)
     3    HARDDRIVE   Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB8)


 Seg  Bus  Dev  Fnc  Vendor  Device Slot
  #    #    #    #     ID      ID     #   Path
 ---  ---  ---  ---  ------  ------  ---  -----------
  00   00   01   00  0x1033  0x0035   XX  Acpi(HWP0002,0)/Pci(1|0)
  00   00   01   01  0x1033  0x0035   XX  Acpi(HWP0002,0)/Pci(1|1)
  00   00   01   02  0x1033  0x00E0   XX  Acpi(HWP0002,0)/Pci(1|2)
  00   00   02   00  0x1095  0x0649   XX  Acpi(HWP0002,0)/Pci(2|0)
  00   00   03   00  0x8086  0x1229   XX  Acpi(HWP0002,0)/Pci(3|0)
  00   20   01   00  0x1000  0x0030   XX  Acpi(HWP0002,100)/Pci(1|0)
  00   20   01   01  0x1000  0x0030   XX  Acpi(HWP0002,100)/Pci(1|1)
  00   20   02   00  0x14E4  0x1645   XX  Acpi(HWP0002,100)/Pci(2|0)
  00   40   01   00  0x103C  0x1029   03  Acpi(HWP0002,200)/Pci(1|0)
  00   60   01   00  0x1000  0x0021   02  Acpi(HWP0002,300)/Pci(1|0)
  00   80   01   00  0x1011  0x0019   01  Acpi(HWP0002,400)/Pci(1|0)
  00   E0   01   00  0x103C  0x1290   XX  Acpi(HWP0002,700)/Pci(1|0)
  00   E0   01   01  0x103C  0x1048   XX  Acpi(HWP0002,700)/Pci(1|1)
  00   E0   02   00  0x1002  0x5159   XX  Acpi(HWP0002,700)/Pci(2|0)


BOOT INFORMATION

   Monarch CPU:

   Current  Preferred
   Monarch   Monarch   Possible Warnings
   -------  ---------  -----------------
         0          0

   AutoBoot: ON - Timeout is : 7 sec

   Boottest:

   BOOTTEST Settings Default Variable

   OS is not speedy boot aware.

   Selftest     Setting
   ---------    --------------
   early_cpu    Run this test
   late_cpu     Run this test
   platform     Run this test
   chipset      Run this test
   io_hw        Run this test
   mem_init     Run this test
   mem_test     Run this test
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
   LAN Address Information:

   LAN Address        Path
   -----------------  ----------------------------------------
   Mac(00306E1E7EE0)  Acpi(HWP0002,0)/Pci(3|0)/Mac(00306E1E7EE0))
  *Mac(00306E1E9EA9)  Acpi(HWP0002,100)/Pci(2|0)/Mac(00306E1E9EA9))


FIRMWARE INFORMATION

   Firmware Revision: 2.21 [4334]

   PAL_A Revision: 7.31/7.31
   PAL_B Revision: 7.59

   SAL Spec Revision: 3.01
   SAL_A Revision: 2.00
   SAL_B Revision: 2.21

   EFI Spec Revision: 1.10
   EFI Intel Drop Revision: 14.61
   EFI Build Revision: 1.22

   POSSE Revision: 0.10

   ACPI Revision: 7.00

   BMC Revision 1.50
   IPMI Revision: 1.00
   SMBIOS Revision: 2.3.2a
   Management Processor Revision: kd.05.07


WARNING AND STOP BOOT INFORMATION


CHIP REVISION INFORMATION

   Chip                  Logical     Device       Chip
   Type                     ID         ID       Revision
   -------------------   -------     ------     --------
   Memory Controller         0       122b         0022
   Root Bridge               0       1229         0022
     Host Bridge          0000       122e         0022
     Host Bridge          0001       122e         0022
     Host Bridge          0002       122e         0022
     Host Bridge          0003       122e         0022
     Host Bridge          0004       122e         0022
     Host Bridge          0006       122e         0022
     Host Bridge          0007       122e         0022
     Other Bridge            0          0         0002
       Other Bridge          0          0         0004
         Baseboard MC        0          0         0150
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–17. SLIDE: Viewing EFI Devices via the EFI

### Viewing EFI Devices via the EFI

- Many EFI commands reference system devices
- You can view the EFI paths for these devices/partitions via the `map -r` command
- Each device/partition is mapped to a name such as blk1 or fs1
- A single device may appear multiple times in the `map` command output
  - blk0 through blk2 represent partitions on the CDROM
  - blk3 through blk6 represent partitions on a disk
  - fs0 through fs2 represent partitions containing FAT32 file systems

```
Shell> map -r
Device mapping table
blk0 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)
blk1 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)
blk2 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry1)
blk3 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)
blk4 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB8D)
blk5 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part2,SigB8D)
blk6 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB8D)
fs0  : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)
fs1  : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB8D)
fs2  : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB8D)
```

## Student Notes

Many EFI commands reference system devices or partitions. Some EFI commands identify devices by “device handle”, while others identify devices by “device name”.

### Device Handles and the `dh` Command

Every disk and disk partition on an Integrity server is automatically assigned a “device handle”. Some EFI commands use these device handles to identify and access devices on the system. The EFI **`dh`** command displays a list of device handles, and the hardware paths and protocol handlers associated with each path. Add the **`-b`** option to view one page at a time, and the **`-p`** option to limit the output to devices associated with a specific protocol handler. To ensure that each record fits on a single line, the **`dh`** command truncates the hardware paths. To view the complete hardware paths and other information, add the **`-v`** (verbose) option. The example below lists the device handles for mass storage devices, which use the **`blkio`** protocol handler.


© 2005 Hewlett-Packard Development Company, L.P.

```
Shell> dh -b -p blkio
Handle dump by protocol 'BlkIo'
 4E: DiskIo BlkIo DevPath (..Pci(2|0)/Ata(Primary,Master))
 8D: Fs DiskIo BlkIo DevPath (..Primary,Master)/CDROM(Entry0))
 50: DiskIo BlkIo DevPath (..00)/Pci(1|0)/Scsi(Pun0,Lun0))
 51: DiskIo BlkIo DevPath (..00)/Pci(1|0)/Scsi(Pun1,Lun0))
 8E: Fs DiskIo BlkIo ESP DevPath (..146C-11D8-8002-D6217B60E588))
 8F: DiskIo BlkIo DevPath (..146C-11D8-8003-D6217B60E588))
 90: Fs DiskIo BlkIo DevPath (..146C-11D8-8004-D6217B60E588))
```

## Device Names, and the `map` Command

Some EFI commands reference devices by device handle, but many of the commonly-used EFI commands reference devices by user-defined device names. The EFI automatically maps a name to each partition and disk device, but administrators can add their own mappings, too. Use the **`map`** command to view a list of device names and hardware paths. The **`-b`** option displays one page at a time, and the **`-r`** option regenerates the device name assignments in case a new DVD or floppy disk has been added. **`blk`** lines in the **`map`** output identify disks and partitions. Each partition that contains an EFI file system will also generate an **`fs`** entry.

```
Shell> map –b -r
Device mapping table
blk0 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)
blk1 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)
blk2 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry1)
blk3 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)
blk4 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB8D)
blk5 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part2,SigB8D)
blk6 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB8D)
fs0  : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)
fs1  : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB8D)
fs2  : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB8D)
```

## Adding and Removing User Defined Mappings

The **`map`** command may also be used to create temporary user-defined mappings. These mappings are lost every time the system reboots.

```
Shell> dh -p blkio      # find the device handle

Handle dump by protocol 'BlkIo'
 4E: DiskIo BlkIo DevPath (..Pci(2|0)/Ata(Primary,Master))
 8D: Fs DiskIo BlkIo DevPath (..rimary,Master)/CDROM(Entry0))
 50: DiskIo BlkIo DevPath (..00)/Pci(1|0)/Scsi(Pun0,Lun0))
 51: DiskIo BlkIo DevPath (..00)/Pci(1|0)/Scsi(Pun1,Lun0))
 8E: Fs DiskIo BlkIo ESP DevPath (..146C-11D8-8002-D6217B60E588))
 8F: DiskIo BlkIo DevPath (..146C-11D8-8003-D6217B60E588))
 90: Fs DiskIo BlkIo DevPath (..146C-11D8-8004-D6217B60E588))
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
Shell> map cdrom 8d     # create the mapping
Shell> map              # verify the mapping

Device mapping table
  fs0  : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)
  fs1  : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB8)
  fs2  : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB8)
  blk0 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)
  blk1 : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)
  blk2 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun0,Lun0)
  blk3 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)
  blk4 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB8)
  blk5 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part2,SigB8)
  blk6 : Acpi(HWP0002,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part3,SigB8)
  cdrom : Acpi(HWP0002,0)/Pci(2|0)/Ata(Primary,Master)/CDROM(Entry0)

Shell> map -d cdrom     # delete the mapping
Shell> map –r           # the –r option removes all user-defined mappings
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–18. SLIDE: Viewing EFI Devices via HP-UX

### Viewing EFI Devices via HP-UX

On a booted system, you can view selected HP-UX hardware paths and their associated EFI disk paths via `ioscan -e`.

```
# ioscan -eC disk
H/W Path       Class   Description
============================================================
0/0/2/0.0.0.0  disk    TEAC    DV-28E-A
Acpi(000222F0,0)/Pci(2|0)/Ata(Primary,Master)/
\EFI\HPUX\HPUX.EFI

0/1/1/0.0.0    disk    HP 73.4GST373405LC
Acpi(000222F0,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part1,SigF9)/
\EFI\HPUX\HPUX.EFI

0/1/1/0.1.0    disk    HP 73.4GST373405LC
Acpi(000222F0,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,SigB8)/
\EFI\HPUX\HPUX.EFI
```

## Student Notes

The hardware addressing chapter earlier in the course explained how to translate EFI hardware paths to HP-UX hardware paths and actual hardware locations on the system. The **`ioscan`** command supports a new **`-e`** option that simplifies the translation by appending the EFI hardware path to the description field for each disk device.

```
# ioscan -eC disk
H/W Path       Class   Description
============================================================
0/0/2/0.0.0.0  disk    TEAC    DV-28E-A
Acpi(000222F0,0)/Pci(2|0)/Ata(Primary,Master)/\EFI\HPUX\HPUX.EFI

0/1/1/0.0.0    disk    HP
73.4GST373405LCAcpi(000222F0,100)/Pci(1|0)/Scsi(Pun0,Lun0)/HD(Part1,
SigF9)/\EFI\HPUX\HPUX.EFI

0/1/1/0.1.0    disk    HP
73.4GST373405LCAcpi(000222F0,100)/Pci(1|0)/Scsi(Pun1,Lun0)/HD(Part1,
SigB8)/\EFI\HPUX\HPUX.EFI
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–19. SLIDE: Managing the EFI Boot Menu via the EFI

### Managing the EFI Boot Menu via the EFI

From the EFI shell, you can customize the EFI boot menu via the `bcfg` command

View available devices

```
Shell> map -b -r -fs
```

Define the primary, high availability alternate, and alternate boot devices

```
Shell> bcfg boot add 1 fsn:\efi\hpux\hpux.efi "HP-UX Primary Boot"
Shell> bcfg boot add 2 fsn:\efi\hpux\hpux.efi "HP-UX HA Alt Boot"
Shell> bcfg boot add 3 fsn:\install.efi "HP-UX Alternate"
```

Remove the third item from the boot menu

```
Shell> bcfg boot rm 3
```

View the resulting boot menu

```
Shell> bcfg boot dump
```

Enable or disable the EFI autoboot feature

```
Shell> autoboot on|off
Shell> autoboot time 10
```

## Student Notes

Managing the EFI boot menu items is one of the most common tasks performed from the EFI shell. During the boot process, the EFI uses the boot manager menu to identify the default boot source. If the first device on the boot menu fails to boot, the EFI attempts to boot each successive device included on the menu until it finds a viable boot source. If you configure a mirrored boot disk you will probably want to add the new mirror to the boot menu. You might also choose to add your DVD drive to the boot menu to simplify installation and recovery processes.

The steps below may be used to add three boot sources to the top of the EFI boot manager menu.

First, use the **`map`** command to view a list of available devices. **`-b`** displays one screen at a time. **`-r`** refreshes the list to ensure that we are viewing the most current data. The **`-fs`** option only lists partitions containing EFI file systems. EFI system partitions, which are typically included in the boot menu, always contain a EFI file system.

```
Shell> map -b -r -fs
```


© 2005 Hewlett-Packard Development Company, L.P.

Next, define the first (primary), second (high availability alternate), and third (alternate) boot sources. For each source, select a device and add it to the menu via the **bcfg** command.

- The first **boot** argument indicates that you wish to modify the boot menu. The **bcfg** command may also be used to add and remove drivers.
- The second **add** argument indicates that you wish to add an item to the boot menu. You can also remove (**rm**) a menu item, or rearrange/move menu items (**mv**).
- The third argument determines where each device appears in the boot menu. Existing menu items are demoted down the menu to make space for each new boot source.
- The fourth argument specifies the EFI system partition and default boot executable on each device. On HP-UX boot disks, the kernel loader is typically called **\efi\hpux\hpux.efi**. The install executable on the HPUX install DVDs is typically called **\install.efi**.
- The fifth argument specifies the text string to display in the boot menu.

```
Shell> bcfg boot add 1 fsn:\efi\hpux\hpux.efi "HP-UX Primary Boot"
Shell> bcfg boot add 2 fsn:\efi\hpux\hpux.efi "HP-UX HA Alt Boot"
Shell> bcfg boot add 3 fsn:\install "HP-UX Alternate"
```

You can also remove items from the boot menu via the **bcfg** command.

```
Shell> bcfg boot rm 3
```

To view the boot menu contents at any time, type **bcfg boot dump**

```
Shell> bcfg boot dump
```

During the boot process, the EFI firmware automatically displays a menu of boot devices recorded in NVRAM. By default, the boot manager waits ten seconds before automatically booting from the first choice in the menu. As noted earlier, you can press **Escape** during the 10-second boot delay period if you wish to choose an alternate boot device. The EFI **autoboot** command makes it possible to disable the autoboot sequence entirely, or change the timeout period. The autoboot command syntx varies somewhat from model to model; type **help autoboot** for more information.

```
Shell> autoboot off       # Turn off the autoboot feature
Shell> autoboot on        # Turn on the autoboot feature
Shell> autoboot time 10   # Change the default autoboot grace period
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–20. SLIDE: Managing the EFI Boot Menu via HP-UX

### Managing the EFI Boot Menu via HP-UX

You can customize the EFI boot menu via HP-UX `setboot` command, too.

View available devices

```
# ioscan -eC disk
```

Define the primary, high availability alternate, and alternate boot devices

```
# setboot -p 0/1/1/0.0.0
# setboot -h 0/1/1/0.1.0
# setboot -a 0/0/2/0.0.0.0
```

Enable, disable, or customize the EFI boot manager autoboot timeout period

```
# setboot -b on|off
```

Verify the configuration

```
# setboot
```

## Student Notes

If your system is already booted, you can make changes to the EFI boot menu via the HP-UX **`setboot`** command.

First, use the **`ioscan`** command to view the list of available disks.

```
# ioscan -eC disk
```

Next, use the **`setboot`** command to define the boot menu order. Use the **`-p`** (primary) option to define the first boot device, **`-h`** (high availability alternate) option to define the second device, and **`-a`** (alternate) to define the third device. Low end systems don't support the **`-h`** option.

```
# setboot -p 0/1/1/0.0.0
# setboot -h 0/1/1/0.1.0
# setboot -a 0/0/2/0.0.0.0
```


© 2005 Hewlett-Packard Development Company, L.P.

You can also enable, disable, or customize the EFI boot manager autoboot feature. During the boot process, if you turn autoboot off, the system will pause indefinitely at the boot manager menu screen. If you turn autoboot on, the system will pause several seconds at the boot menu, then autoboot the first device in the menu. The administrator can interrupt the autoboot process by pressing **Escape** at the boot manager menu.

```
# setboot -b on|off
```

Use the **setboot** command without any options to verify the resulting configuration.

```
# setboot
```

On some systems, the **setboot** command also supports **-t** and **-T** options, which may be used to enable or disable some of the self-tests that occur during the boot process. This is known as Speedy Boot. Disabling these self-tests can significantly improve system boot time on large-memory and multi-processor systems. The options aren't supported on entry-level systems. See the **setboot** man page for details.

No matter which options you specify, the **setboot** command automatically saves your changes in NVRAM.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–21. SLIDE: Managing EFI File Systems via the EFI

### Managing EFI File Systems via the EFI

The EFI provides a rich set of commands for managing files on the EFI system partition.

EFI file system commands are based on traditional UNIX and DOS file commands.

Aliases may be defined to simplify commands that you frequently forget.

Select an EFI device containing a FAT32 file system
```
Shell> map -r
Shell> fs1:
```

Use `cd` change to a different directory
```
fs1:\> cd \efi\hpux
```

Use `cd` without arguments to determine the current directory
```
fs1:\> cd
```

Use `ls` or `dir` to list the contents of a directory
```
fs1:\> ls -b
```

Use `type` or `cat` to view the contents of a file
```
fs1:\> type -b \efi\hpux\auto
```

Use `mkdir` or `md` to create directories
```
fs1:\> mkdir \efi\hpux\backup
```

Use `copy` or `cp` to copy a file
```
fs1:\> cp \efi\hpux\auto \efi\hpux\backup\
```

Use `edit` to edit a file
```
fs1:\> edit \efi\hpux\auto
```

Use `rm`, `rd`, or `del` to remove directories and/or files
```
fs1:\> rm \efi\hpux\auto\backup\
```

## Student Notes

You can also use the EFI shell to manage files on the EFI system partition, or any other EFI file system. This might be useful if you need to make changes to the **`auto`** file, which specifies your kernel boot string. Some administrators use these commands, too, to create EFI batch scripts.

The EFI system and service partitions typically contain EFI file systems, which are similar to the FAT32 format that is popular on Windows systems. Note that FAT32 file systems use backslashes ("****") rather than forward-slashes ("**/**") in directory paths.

The EFI commands used to access and manage EFI file systems are an odd amalgamation of traditional UNIX and DOS commands.

First, select a device containing an EFI file system. Simply type the partition name followed by a "**:**". After you select a file system, your prompt string changes to indicate the partition you've selected. Regardless of the prompt string, you can still execute any of the EFI shell commands we've discussed thus far.

```
Shell> map -r
```


© 2005 Hewlett-Packard Development Company, L.P.

```
Shell> fs1:
fs1:>
```

You can change directories within the selected file system via the **cd** command. Note that EFI path names use back-slashes rather than UNIX forward slashes. You can also use **..** to represent the parent directory.

```
fs1:> cd \efi\hpux
```

Use the **cd** command without any arguments to determine your present working directory. The EFI shell doesn't support a **pwd** command.

```
fs1:\> cd
```

Use the **ls** or **dir** command to list the contents of an EFI directory. If you don't specify a directory as an argument, you will see the contents of the current directory. On some models, the **-b** option displays one page at a time.

```
fs1:\> ls -b
```

Use the **type** command to view the contents of an ASCII file in an EFI file system. The **-b** option displays one page at a time. On some models, the UNIX-like **cat** command may work, too.

```
fs1:\> type -b \efi\hpux\auto
```

Use the UNIX-like **mkdir** command to create directories. Multiple arguments may be specified on a single command line. On most models, the DOS **md** command works, too.

```
fs1:\> mkdir \efi\hpux\backup
```

Use the UNIX-like **cp** command to copy files. Use **-r** (recursive) when copying directories. On most models, the DOS **copy** command works, too.

```
fs1:\> cp \efi\hpux\auto \efi\hpux\backup\
```

Use the **edit** command to edit files. Use the arrow keys to navigate the file contents, and backspace to delete text. Use **[Escape] [s]** to save, and **[Escape] [q]** to quit.

```
fs1:\> edit \efi\hpux\auto
```

Use the UNIX-like **rm** command to remove files. Add the **-r** option to recursively remove directories. On most models, the DOS **rd** and **del** commands work, too.

```
fs1:\> rm \efi\hpux\auto\backup\
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## Aliases

The EFI shell provides an **alias** command very similar to the POSIX **alias** command. To view a list of currently defined aliases, run the **alias** command without any arguments. The aliases shown below are predefined in HP's EFI implementation.

```
fs1:\> alias
    dir  : ls
    md   : mkdir
    rd   : rm
    del  : rm
    copy : cp
    find : ls -r
    ll   : ls
    cat  : type
```

You can also define additional aliases. The example below creates a **pwd** alias which displays the present working directory.

```
fs1:\> alias pwd cd
```

By default aliases persist across reboots. To define a temporary alias, add the **–v** (volatile) option.

```
fs:\> alias –v pwd cd
```

To delete an alias, use the **–d** option.

```
fs1:\> alias –d pwd
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–22. SLIDE: Managing EFI File Systems via HP-UX

### Managing EFI File Systems via HP-UX

You can also manage EFI files and file systems via HP-UX commands

Create and populate EFI partitions on disk

```
# vi /tmp/idf
  3
  EFI 500MB
  HPUX 100%
  HPSP 400MB
# idisk -wf /tmp/idf /dev/rdsk/c0t1d0
# mkboot -e -l /dev/rdsk/c0t1d0
```

List the contents of an EFI file system

```
# efi_ls -d /dev/rdsk/c0t0d0s1 /efi/hpux
```

Copy and edit an EFI file via an HP-UX temporary file

```
# efi_cp -d /dev/rdsk/c0t0d0s1 /efi/hpux/auto /tmp/auto
# vi /tmp/auto
# efi_cp -d /dev/rdsk/c0t0d0s1 -u /tmp/auto /efi/hpux/auto
```

Other useful commands: `efi_mkdir`, `efi_rmdir`, `efi_rm`

## Student Notes

EFI file systems aren't accessible via traditional UNIX file system commands. However, several EFI-specific utilities are now available to manage EFI partitions on a booted HP-UX system:

- **`idisk`** for creating EFI partitions
- **`efi_fsinit`** for creating an EFI file system on an EFI partition
- **`efi_ls`** for listing the contents of EFI file systems
- **`efi_cp`** for copying files to and from EFI file systems
- **`efi_mkdir`** for creating directories in a EFI file system
- **`efi_rmdir`** for removing directories from a EFI file system
- **`efi_rm`** for removing files from an EFI file system


**© 2005 Hewlett-Packard Development Company, L.P.**

The examples described below show some common uses for these commands.

- The **/tmp/idf** file describes the desired disk layout. Based on the sample file below, the EFI system partition will be 100MB. The HP Service Partition will be 400MB. All remaining space will be allocated to the HP-UX OS partition.

- The **idisk** command creates partitions based on the parameters in the file following the **-f** option (**/tmp/idf** in this case). The **-w** option is required if you wish to write changes to the GPT on the disk. **idisk** also automatically creates EFI file systems in the system and HPSP partitions, so there is no need to explicitly run the **efi_fsinit** command.

- The **mkboot -e** command copies **hpux.efi** and other EFI executables from **/usr/lib/efi** to the new system partition. The **-l** option is required if the disk will be configured using a volume manager such as LVM or VxVM. Use the **-h** option instead if you plan to use the "whole disk" approach.

```
# vi /tmp/idf
  3
  EFI 500MB
  HPUX 100%
  HPSP 400MB
# idisk -wf /tmp/idf /dev/rdsk/c0t1d0
# mkboot -e -l /dev/rdsk/c0t1d0
```

You can use the **efi_ls** command to list the contents of an EFI file system. The **-d** option specifies the device filename of the partition you wish to access. The example below displays the contents of the **/efi/hpux/** directory on the /dev/rdsk/c0t0d0**s1** EFI partition. Note that the HP-UX **efi*** commands require forward slashes in EFI pathnames, while the EFI shell utilities require backslashes.

```
# efi_ls -d /dev/rdsk/c0t0d0s1 /efi/hpux
```

You can edit files stored in an EFI file system, too. Use the **efi_cp** command to copy the desired file to a file in **/tmp**, edit the temporary file with your favorite editor, and return it to the EFI system partition with **efi_cp**.

```
# efi_cp -d /dev/rdsk/c0t0d0s1 /efi/hpux/auto /tmp/auto
# vi /tmp/auto
# efi_cp -d /dev/rdsk/c0t0d0s1 -u /tmp/auto /efi/hpux/auto
```

For more detailed information, see the **efi(4)** man page, or the man pages for any of these commands.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–23. SLIDE: Booting from an Alternate Disk via the EFI

### Booting from an Alternate Disk via the EFI

If you have multiple mirrored boot disks on an Itanium system, and the primary boot disk fails, you can select and alternate boot disk from either the EFI boot manager menu, or the EFI shell.

View available devices, and identify your desired boot disk

```
Shell> map -b -r -fs
```

Run the `hpux.efi` kernel loader from the desired device

```
Shell> fsn:\efi\hpux\hpux.efi
```

## Student Notes

If you have multiple mirrored boot disks on an Integrity server, and the primary boot disk fails, you can select an alternate boot disk from either the EFI boot manager menu, or the EFI shell.

View available devices, and identify the desired boot disk. The **-b** option displays one page at a time if you have many potential boot devices. The **-r** option refreshes the mappings to ensure that you are viewing the most current information. **-fs** only displays partitions containing EFI file systems.

```
Shell> map -b -r -fs
```

Run the **hpux.efi** kernel loader from the desired device.

```
Shell> fsn:\efi\hpux\hpux.efi
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–24. SLIDE: Booting Alternatives from the HP-UX Boot Loader

### Booting Alternatives from the HP-UX Boot Loader

If you have a backup kernel and the primary kernel fails, you can boot from an alternate kernel via the EFI.

View available devices, and identify your desired boot disk

```
Shell> map -b -r -fs
```

Run the `hpux.efi` kernel loader from the desired device, then interrupt the autoboot

```
Shell> fsn:\efi\hpux\hpux.efi
[Escape]
```

Choose a kernel and mode, and boot

| | |
|---|---|
| `HPUX> ll` | *List contents of /stand* |
| `HPUX> boot backup` | *Boot the backup kernel* |
| `HPUX> boot vmunix -v` | *Boot vmunix to multi-user mode* |
| `HPUX> boot vmunix -is` | *Boot to single-user mode* |
| `HPUX> boot vmunix -lq` | *Boot without checking LVM quorum* |
| `HPUX> boot vmunix -lm` | *Boot to LVM maintenance mode* |
| `HPUX> boot vmunix -tm` | *Boot to tunable maintenance mode* |
| `HPUX> exit` | *Return to the boot manager* |

## Student Notes

If you have a backup kernel and the primary kernel configuration fails, the EFI provides booting alternatives. HP-UX automatically maintains a backup kernel configuration called **`/stand/backup`**. You can also boot to one of the special recovery modes that were described earlier in the chapter.

View available devices, and identify your desired boot disk

```
Shell> map -b -r -fs
```

Run the **`hpux.efi`** kernel loader from the desired device, then interrupt the HPUX autoboot sequence.

```
Shell> fsn:\efi\hpux\hpux.efi
Escape
```


© 2005 Hewlett-Packard Development Company, L.P.

Choose a kernel configuration and mode, and boot

| | |
|---|---|
| `HPUX> ll` | *List contents of /stand* |
| `HPUX> boot backup` | *Boot the backup kernel* |
| `HPUX> boot vmunix -v` | *Boot vmunix to multi-user mode* |
| `HPUX> boot vmunix –is` | *Boot to single-user mode* |
| `HPUX> boot vmunix –lq` | *Boot without checking LVM quorum* |
| `HPUX> boot vmunix –lm` | *Boot to LVM maintenance mode* |
| `HPUX> boot vmunix –tm` | *Boot to tunable maintenance mode* |
| `HPUX> exit` | *Return to the boot manager* |


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–25. SLIDE: Booting from the Install DVD

### Booting from the Install DVD

In order to install HP-UX you may need to boot from the HP-UX 11i v2 install DVD.

View available devices, and look for the DVD/CDROM device

```
Shell> map -b -r -fs
```

To install HP-UX, run the `install.efi` program from the install DVD

```
Shell> fsn:\install.efi
```

## Student Notes

To install HP-UX you may need to boot from the HP-UX 11i v2 installation DVD.

View available devices, and look for the DVD/CDROM device. The description field will usually include “CDROM”.

```
Shell> map -b -r -fs
```

Run the **`install.efi`** program from the DVD device.

```
Shell> fsn:\install.efi
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–26. SLIDE: Booting from an Ignite-UX Server

### Booting from an Ignite-UX Server

- You can also boot from an Ignite-UX install/recovery server.
- Unlike the PA-RISC BCH, the EFI doesn't allow you to specify a server IP.
- Instead, the client uses a BOOTP broadcast to identify an Ignite server.

Execute the `lanboot` command, and choose a local interface to boot from

```
Shell> lanboot select
01  Acpi(HWP0002,0)/Pci(3|0)/Mac(00306E1E7EE0)
02  Acpi(HWP0002,100)/Pci(2|0)/Mac(00306E1E9EA9)
Select Desired LAN: 1
```

## Student Notes

You can also boot from an Ignite-UX install/recovery server. Note that unlike the PA-RISC BCH, the EFI doesn't allow you to specify which server IP to boot from; the EFI only supports the BOOTP mechanism for identifying a server. The Ignite server must have a hard-coded MAC<-> IP address entry for your client in the **`/etc/bootptab`** file. The client uses a BOOTP broadcast to identify a server, and boots from the first server that responds You do, however, have an opportunity to specify which interface card from which you wish to boot.

Execute the **`lanboot`** command, and choose a local interface to boot. Momentarily, the Ignite interface should appear.

```
Shell> lanboot select
  01  Acpi(HWP0002,0)/Pci(3|0)/Mac(00306E1E7EE0)
  02  Acpi(HWP0002,100)/Pci(2|0)/Mac(00306E1E9EA9)
  Select Desired LAN: 1
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–27. SLIDE: Booting from a make_tape_recovery Tape

### Booting from a make_tape_recovery Tape

- Currently, the EFI shell can't boot directly from a recovery tape
- Instead, boot from the install DVD, ...
- Then use the Ignite menus to select your tape drive as the install source

Find your DVD/CDROM device

```
Shell> map -b -r -fs
Shell> fsn:\install.efi
```

Use the Ignite menus to select the tape drive as the install media source

```
Media Installation Selection

This screen provides an option to switch the install source
from the default CD/DVD to a recovery tape.  This is helpful
for those systems and for tape devices which do not support
booting from a tape.
[    ] CD/DVD Installation
[  * ] Boot from CD/DVD, Recover from Tape
[ OK ]          [ Cancel ]       [ Help ]
```

## Student Notes

The EFI doesn't currently support booting directly from a tape drive. There is a workaround, however, that makes it possible to boot from a **`make_tape_recovery`** tape: boot from the install DVD first, then use the Ignite interface to select your tape drive as your install source. The notes below explain the process in detail.

First, insert the core install media in your DVD drive, boot to the EFI shell, and use the **`map`** command to identify your DVD device. Load the **`install.efi`** Ignite client from the DVD.

```
Shell> map -r
Shell> fsn:\install.efi
```


© 2005 Hewlett-Packard Development Company, L.P.

Select a keyboard language when prompted.

```
=====================================================================
In order to use a keyboard on this interface, you must specify
a language mapping which will be used by X windows and
the Internal Terminal Emulator (ITE).
The characters "1234567890" will appear as "!@#$^&*()"
on keyboards that use the shift key to type a number.
Your choice will be stored in the file /etc/kbdlang

 1) USB_PS2_DIN_Belgian          2) USB_PS2_DIN_Belgian_Euro
 3) USB_PS2_DIN_Danish           4) USB_PS2_DIN_Danish_Euro
 5) USB_PS2_DIN_Euro_Spanish     6) USB_PS2_DIN_Euro_Spanish_Euro
 7) USB_PS2_DIN_French           8) USB_PS2_DIN_French_Euro
 9) USB_PS2_DIN_German          10) USB_PS2_DIN_German_Euro
11) USB_PS2_DIN_Italian         12) USB_PS2_DIN_Italian_Euro
13) USB_PS2_DIN_JIS_109         14) USB_PS2_DIN_Korean
15) USB_PS2_DIN_Norwegian       16) USB_PS2_DIN_Norwegian_Euro
17) USB_PS2_DIN_S_Chinese       18) USB_PS2_DIN_Swedish
19) USB_PS2_DIN_Swedish_Euro    20) USB_PS2_DIN_Swiss_French2_Euro
21) USB_PS2_DIN_Swiss_German2   22) USB_PS2_DIN_Swiss_German2_Euro
23) USB_PS2_DIN_T_Chinese       24) USB_PS2_DIN_UK_English
25) USB_PS2_DIN_UK_English_Euro 26) USB_PS2_DIN_US_English
27) USB_PS2_DIN_US_English_Euro

Enter the number of the language you want: 26
Please confirm your choice by pressing RETURN or enter a new number:
=====================================================================
```

When you see the Ignite Welcome screen, click the [Install HP-UX] menu option.

```
=====================================================================
                        Welcome to Ignite-UX!

Use the <tab> key to navigate between fields, and the arrow keys
within fields.  Use the <return/enter> key to select an item.
Use the <return> or <space-bar> to pop-up a choices list. If the
menus are not clear, select the "Help" item for more information.

Hardware Summary:       System Model: 9000/712/80
+-----------------------------------------------+ [ Scan Again ]
| Disks: 4  (5.8GB)| Floppies: 1 | LAN cards: 2 |
| CDs:   1         | Tapes:    0 | Memory:  2Mb |
| Graphics Ports: 1| IO Buses: 2 |              | [ H/W Details
+-----------------------------------------------+
            [      Install HP-UX       ]
            [   Run a Recovery Shell   ]
            [    Advanced Options      ]
 [  Reboot  ]                          [  Help  ]
=====================================================================
```


**© 2005 Hewlett-Packard Development Company, L.P.**

On the Media Options screen, select **`Media only installation`**.

```
=====================================================================
                    User Interface and Media Options

This screen lets you pick from options that will determine if an
Ignite-UX server is used, and your user interface preference.

Source Location Options:
  [ * ] Media only installation
  [   ] Media with Network enabled (allows use of SD depots)
  [   ] Ignite-UX server based installation

User Interface Options:
  [   ] Guided Installation   (recommended for basic installs)
  [ * ] Advanced Installation (recommended for file system mgmt)
  [   ] Remote graphical interface running on the Ignite-UX server

Hint: If you need to make LVM size changes, or want to set the
      final networking parameters during the install, you will
      need to use the Advanced mode (or remote graphical interface).

   [   OK   ]                [ Cancel ]                  [  Help  ]

=====================================================================
```

The next screen allows you to specify which media you wish to install from. Select the **`Boot from CD/DVD, Recover from Tape`** option.

```
=====================================================================
              Media Installation Selection

This screen provides an option to switch the install source
from the default CD/DVD to a recovery tape.  This is helpful
for those systems and for tape devices which do not support
booting from a tape.

[   ] CD/DVD Installation
[ * ] Boot from CD/DVD, Recover from Tape

[ OK ]          [ Cancel ]       [ Help ]
=====================================================================
```


© 2005 Hewlett-Packard Development Company, L.P.

If you have multiple tape drives, you'll have to specify which tape you want to boot from on the Tape Drive Selection screen. Note that Ignite reports the tape drive paths using the traditional HP-UX hardware addressing scheme.

```
=====================================================================
Tape Drive Selection

There are one or more tape drives detected on the system.  Insert
your recovery tape into one of the drives then select that drive
from the list below.

Use the <tab> and/or arrow keys to move to the desired TAPE device,
then press <Return/Enter> to select.

   HW Path           Device File          Description
   ---------------------------------------------------------------
  [0/1/1/1.4.0       /dev/rmt/0m          COMPAQ_SDLT320          ]
=====================================================================
```

Finally, you should see the Ignite tabbed menu interface. Note, however, that the only Configuration available is **[Recovery Archive]**. If you wish you can use the **System** tab to change your network parameters, or the **File System** tab to change your LVM/file system layout. When you are ready to begin the recovery, press **[Go!]**.

```
+-------++----------++--------++-------------++----------+
| Basic || Software || System || File System || Advanced |
|       \-------------------------------------------------------+
|                                                               |
| Configurations:  [ Recovery Archive ->] [ Description...  ]   |
|                                                               |
| Environments:    [ Recovery Archive ->] (HP-UX B.11.23)       |
|                                                               |
| [ Root Disk...  ] HP_73.4GST373405LC, 0/1/1/0.1.0, 70007 M    |
|                                                               |
| File System:     [ HP-UX save_config layout ->]               |
|                                                               |
| [ Root Swap (MB)... ] 4096   Physical Memory (RAM) =  4088 MB |
|                                                               |
| [ Languages... ] English [ Keyboards... ] [ Additional...  ]  |
|                                                               |
+---------------------------------------------------------------+
|  [ Show Summary...  ]               [ Reset Configuration ]   |
+---------------------------------------------------------------+
|  [  Go!   ]              [ Cancel ]                [  Help  ] |
+-------------------------------------------------------------+
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–28. SLIDE: Booting from the HPSP or Diagnostics CD

### Booting from the HPSP or Diagnostics CD

HP's Integrity Diagnostic utilities provide comprehensive troubleshooting tools that may be used to troubleshoot system hardware problems. You can run the e-Diag utility from CDROM or from the HPSP.

View available devices, and look for the DVD/CDROM device or HPSP partition

```
Shell> map -b -r -fs
```

Run the diagnostics utility from the disk-based HPSP partition, or from the diagnostics CD

```
Shell> fsn:\efi\boot\launchmenu\launchmenu.efi
```

```
HP Itanium Processor Family Diagnostics and Utilities CD

a. Run e-Diagtools for IPF
b. Run the Off-line Diagnostic Environment (ODE)
c. Run CD Installer to install/update CD content to HPSP
d. View Release Notes and Documentation Menu
e. View I/O Cards FW Update and Configuration Utilities
f. View License and Warranty Agreements Notice
x. exit and reboot q. exit menu without reboot
```

## Student Notes

A new tool called e-Diagnostics ("e-Diag" for short) is now available for HP-UX 11i v2. The program allows you to troubleshoot hardware problems on a computer that cannot boot an operating system, and test the system hardware more thoroughly than online diagnostics and support tools can.

e-Diagnostics may be launched directly from the Offline Diagnostics and Utilities CDROM, which is included with every HP Integrity server, or from a 400MB, FAT32 Service Partition on the system boot disk. Running e-Diagnostics from the disk-based service partition provides better performance than the CDROM-based solution.

### Obtaining the Diagnostics and Utilities CDROM

All Integrity Hardware systems are shipped with an Offline Diagnostics and Utilities CDROM. The Integrity Offline Diagnostics and Utilities are updated periodically, to add new functionality, incorporate new hardware diagnostics, and improve performance.


© 2005 Hewlett-Packard Development Company, L.P.

You can order a free copy of the CDROM from:

**`http://partsurfer.hp.com/cgi-bin/spi/main`** or
**`http://www.software.hp.com/ER_products_list.html`**.

Alternatively, you can download a **`*.iso`** image file of the most current Integrity Offline Diagnostics and Utilities CD-ROM from

**`http://www.hp.com/support/itaniumservers`**.

From this **`*.iso`** file, you can burn your own bootable CDROM. A **`*.iso`** image is not a collection of compressed/uncompressed data files; nor is it a binary executable. Rather, a CD **`*.iso`** image is a large, single file, which must be copied exactly as is onto a CD. In effect, you will download the image from the download site, "burning" its contents onto your CD.

First, download the **`ipf_offline_xxxx.zip`** file from the site mentioned above to a PC. This is a large file (>70MB) and may take several minutes to download. It arrives as a compressed zip file, which then needs to be unzipped on your computer. The unzipped file should be a single file, approximately 175MB, with a **`*.iso`** extension.

Burn the full ISO image onto a blank CD-R using a CD burner and any major CD burning software such as Roxio (Adaptec) or Nero. Most common CD burn software such as Roxio (Adaptec) and Nero have the ability to burn an image to CD. See your CD burner software documentation for details. The instructions below explain how to burn a CD-R using the Roxio/Adaptec software.

1. Insert a blank CD-R into your CD burner.
2. From the File menu, select Create CD from CD Image.
3. Select the desired **`*.iso`** file.
4. Click Open. The CD Creation Setup dialog box should appear.
5. If necessary, change the CD Creation Setup settings.
6. Click OK.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Creating the HP Service Partition

Although you can run e-Diag from the Utilities CDROM, it runs faster when launched from the HP Service Partition.

First, space must be allocated for the HPSP partition on the boot disk. Ignite-UX automatically configures an HPSP partition in HP-UX 11i v2. The **[Additional]** button on the Ignite-UX basic tab provides an opportunity to increase the size of the HPSP partition at install time. By default, though, the partition size is 400MB.

```
Additional Configuration Controls
+-------------------------------------------------------+
|The miscellaneous controls listed are for two purposes:  |
|  - setting config file variables, and                  |
|  - shortcutting complex tasks                          |
+-------------------------------------------------------+
| Create /export volume           [ No   ->]             |
| Create separate volumes (/usr,  [ Yes  ->]             |
| [ EFI Boot Partition (KB)...  ] 512000                 |
| [HP Service Partition (KB)..  ] 409600                 |
| [Secondary Swap space (KB)..  ] 0                      |
| [  # of disks in root VG...   ] 1                      |
+-------------------------------------------------------+
```

If you are creating a mirrored boot disk, be sure to reserve space for the HPSP when you run the **idisk** command.

```
# vi /tmp/idf
  3
  EFI 500MB
  HPUX 100%
  HPSP 400MB
# idisk -wf /tmp/idf /dev/rdsk/c0t1d0
```

## Populating the HPSP

Assuming that you have an HPSP partition on your boot disk, you can use the **\efi\boot\launchmenu\launchmenu.efi** program on the Diagnostics and Utilities CDROM to populate the partition. Several steps are required.

1. Reboot the system and access the EFI shell.

2. View available devices, and look for the DVD/CDROM device.

   ```
   Shell> map -b -r -fs
   ```

3. Run the diagnostics utility from the DVD/CD.

   ```
   Shell> fsn:\efi\boot\launchmenu\launchmenu.efi
   ```

Follow the instructions that follow to copy the utilities to the service partition.


© 2005 Hewlett-Packard Development Company, L.P.

## Running e-Diag

The same **`launchmenu.efi`** utility that is used to populate the HPSP partition may also be used to run the diagnostic utilities.

1. Reboot the system and access the EFI shell.

2. View available devices, and look for the DVD/CDROM device, or the HPSP partition.

   ```
   Shell> map -b –r -fs
   ```

3. Run the diagnostics utility from the DVD/CD or HPSP partition.

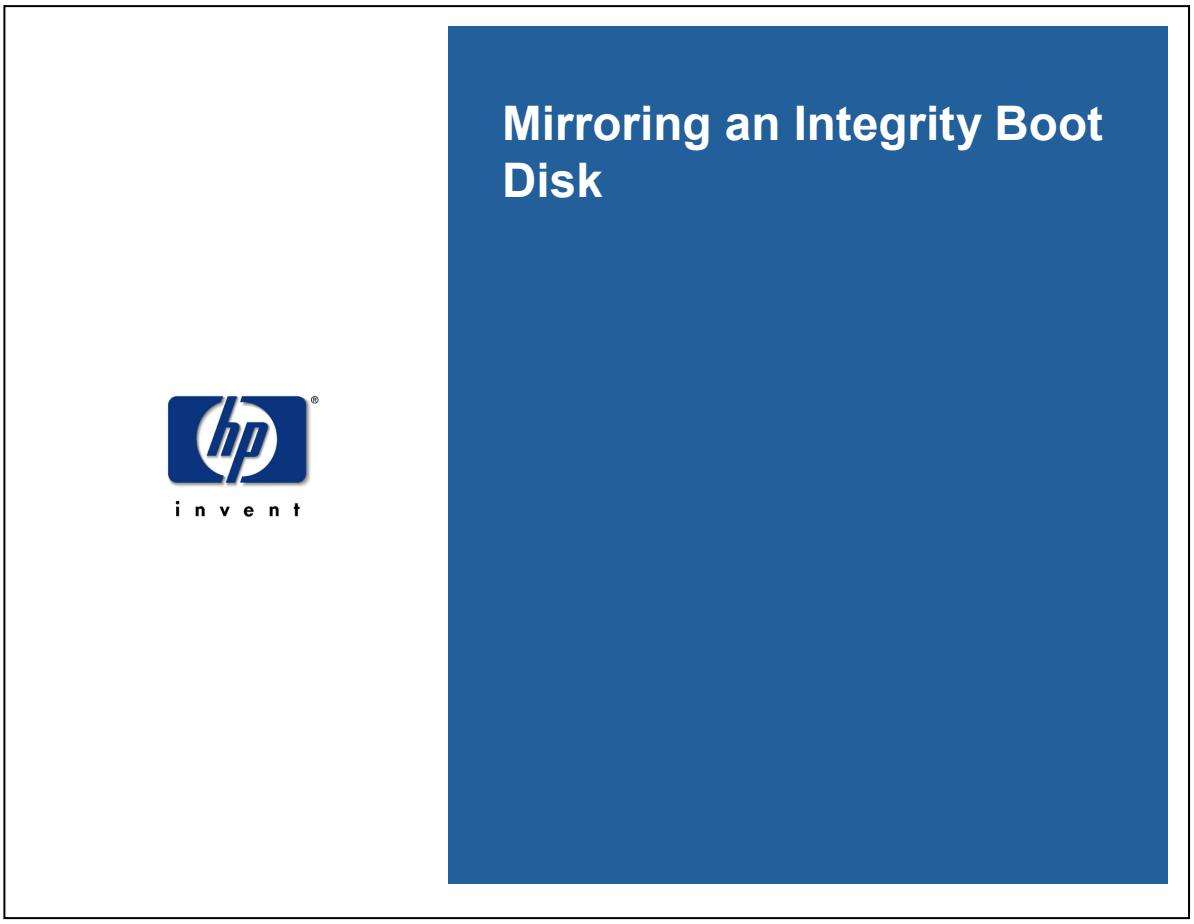
   ```
   Shell> fsn:\efi\boot\launchmenu\launchmenu.efi
   ```

## For More Information…

To learn more about e-Diag, visit **`http://docs.hp.com/hpux/onlinedocs/diag/ode/ode_ediag.htm`** or choose option **`[d]`** (View Release Notes and Documentation Menu) from the **`launchmenu.efi`** program.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–29. SLIDE: Mirroring an Integrity Boot Disk




### Student Notes

This next portion of the chapter discusses the process used to mirror, test, and restore mirrored Integrity boot disks. This task provides a practical application for the EFI commands discussed earlier in the chapter.


© 2005 Hewlett-Packard Development Company, L.P.

## 5–30. SLIDE: Configuring the Partition Table

### Configuring the Partition Table

1. Use `ioscan` and `diskinfo` to identify a disk large enough to mirror the existing boot disk
```
# ioscan -funC disk
# diskinfo /dev/rdsk/c0t1d0
```
2. Create the system, OS, and service partitions
```
# vi /tmp/idf
  3
  EFI 500MB
  HPUX 100%
  HPSP 400MB
# idisk -wf /tmp/idf /dev/rdsk/c0t1d0
```
3. Create device files for the new partitions
```
# insf -eC disk
```
4. Verify the partition table
```
# idisk /dev/rdsk/c0t1d0
```
5. Verify that the device files were created properly
```
# ioscan -funC disk
```

## Student Notes

Mirroring an Integrity boot disk is a multi-step process. First, you need to create three EFI partitions on the new boot mirror using the process described on the slide.

1. Use **`ioscan`** and **`diskinfo`** to identify a disk large enough to mirror the existing boot disk.

```
# ioscan -funC disk
# diskinfo /dev/rdsk/c0t1d0
```

2. Create the system, OS, and service partitions. Replace **`c0t1d0`** with your new mirror's block device file name.

```
# vi /tmp/idf
  3
  EFI 500MB
  HPUX 100%
  HPSP 400MB
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
# idisk -wf /tmp/idf /dev/rdsk/c0t1d0
```

3. Create device files for the new partitions.

```
# insf -eC disk
```

4. Verify the partition table. Near the end of the **idisk** output, you should see three partitions. Replace **c0t1d0** with your new mirror's raw device file name.

```
# idisk /dev/rdsk/c0t1d0
```

5. Verify that the device files were created properly. There should be block and raw ***cxtxdx*s1**, ***cxtxdx*s2**, and ***cxtxdx*s3** device files for the new boot mirror.

```
# ioscan -funC disk
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–31. SLIDE: Configuring the System Partition

### Configuring the System Partition

6. Populate the `/efi/hpux/` directory in the new EFI system partition
```
# mkboot -e -l /dev/rdsk/c0t1d0
```

7. Change the auto file on both boot disks so the host can boot without quorum
```
# vi /tmp/auto
  boot vmunix -lq
# efi_cp -d /dev/rdsk/c0t0d0s1 /tmp/auto /efi/hpux/auto
# efi_cp -d /dev/rdsk/c0t1d0s1 /tmp/auto /efi/hpux/auto
```

8. Verify the contents of the system partition
```
# efi_ls -d /dev/rdsk/c0t1d0s1
```

9. Verify the contents of the `auto` file in the system partition
```
# efi_cp -d /dev/rdsk/c0t1d0s1 -u /efi/hpux/auto /tmp/auto
# cat /tmp/auto
```

## Student Notes

After you create the partition table, you can initialize and populate the EFI system partition.

6. Populate the **`/efi/hpux/`** directory in the new EFI system partition. This command copies the **`hpux.efi`** boot loader and other files to the system partition.

```
# mkboot -e -l /dev/rdsk/c0t1d0
```

7. Change the **`auto`** file on both boot disks so the host can boot without quorum. Change the boot string in the file to **`boot vmunix -lq`**.

```
# vi /tmp/auto
  boot vmunix -lq
# efi_cp -d /dev/rdsk/c0t0d0s1 /tmp/auto /efi/hpux/auto
# efi_cp -d /dev/rdsk/c0t1d0s1 /tmp/auto /efi/hpux/auto
```


**© 2005 Hewlett-Packard Development Company, L.P.**

8. Verify the contents of the system partition.

```
# efi_ls -d /dev/rdsk/c0t1d0s1
```

9. Verify the contents of the **auto** file in the system partition.

```
# efi_cp -d /dev/rdsk/c0t1d0s1 -u /efi/hpux/auto /tmp/efi.auto
# cat /tmp/efi.auto
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–32. SLIDE: Configuring the OS Partition

### Configuring the OS Partition

10. Initialize and add the new OS partition to vg00 (be sure to specify `s2`!)
```
# pvcreate -fB /dev/rdsk/c0t1d0s2
# vgextend vg00 /dev/dsk/c0t1d0s2
```

11. Mirror *all* vg00 logical volumes
```
# for lv in /dev/vg00/lvol*
  do
     lvextend -m 1 $lv
  done
```

12. Add a line to `/stand/bootconf` so SDUX knows which disks are boot disks
```
# vi /stand/bootconf
    l /dev/dsk/c0t0d0s2
    l /dev/dsk/c0t1d0s2
```

13. Verify that the disk was added to vg00, and that the logical volumes are sync'ed
```
# vgdisplay -v vg00
```

14. Verify that the BDRA was updated properly
```
# lvlnboot -v
```

## Student Notes

Next, configure the OS partition. Recall that the OS partition is essentially an encapsulated LVM boot disk.

10. Initialize and add the new OS partition to vg00. Be sure to specify s2, the OS partition!

```
# pvcreate -fB /dev/rdsk/c0t1d0s2
# vgextend vg00 /dev/dsk/c0t1d0s2
```

11. Mirror *all* vg00 logical volumes. If your vg00 has non-default logical volume names, you may need to manually **`lvextend`** each logical volume.

```
# for lv in /dev/vg00/lvol*
  do
     lvextend -m 1 $lv
  done
```


**© 2005 Hewlett-Packard Development Company, L.P.**

12. Add a line to **/stand/bootconf** so SDUX knows which disks are boot disks. When patches are **swinstall**'ed for LIF area utilities, SDUX consults **/stand/bootconf** to determine which disks have LIF areas that might need to be patched. Be sure to specify s2, the OS partition!

```
# vi /stand/bootconf
    l /dev/dsk/c0t0d0s2
    l /dev/dsk/c0t1d0s2
```

13. Verify that the disk was added to vg00, and that the logical volumes are sync'ed. The **LV Status** field for each logical volume should report **syncd**.

```
# vgdisplay -v vg00
```

14. Verify that the BDRA was updated properly. You should see references to both disks in the boot, root, and swap portions of the output.

```
# lvlnboot -v
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–33. SLIDE: Configuring the EFI Boot Manager Menu

### Configuring the EFI Boot Manager Menu

15. Update the EFI boot manager primary, high availability, and alternate boot paths
```
# setboot -p 0/1/1/0.0.0
# setboot -h 0/1/1/0.1.0
# setboot -a 0/0/2/0.0.0.0
```
16. Verify that the boot paths are configured properly in NVRAM
```
# setboot
```

## Student Notes

Finally, add the new mirrored disk to the EFI boot manager menu. This ensures that the EFI will automatically attempt to boot the mirrored boot disk if the primary boot disk fails.

15. Update the EFI boot manager primary, high availability, and alternate boot paths.

```
# setboot -p 0/1/1/0.0.0
# setboot -h 0/1/1/0.1.0
# setboot -a 0/0/2/0.0.0.0
```

16. Verify that the boot paths are configured properly in NVRAM.

```
# setboot
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–34. SLIDE: Booting from the Alternate Mirror

### Booting from the Alternate Mirror

17. Initiate the boot sequence
```
# shutdown -ry 0
```

18. Interrupt the EFI boot manager autoboot
```
EFI Boot Manager ver 1.10 [14.60] Firmware ver 1.61 [4241]
[Escape]
```

19. Select your mirrored disk from the boot manager selection menu
```
EFI Boot Manager ver 1.10 [14.60] Firmware ver 1.61 [4241]
Please select a boot option
   HP-UX Primary Boot
   HP-UX Alternate Boot
   EFI Shell [Built-in]
```

20. Verify which disk/kernel you booted from
```
# grep "Boot device's HP-UX HW path" /var/adm/syslog/syslog.log
  vmunix: Boot device's HP-UX HW path is: 0.0.0.0.1.0
```

## Student Notes

After you complete the mirroring process, it is a good idea to verify your work by booting from the new mirror.

17. Initiate the boot sequence.

```
# shutdown -ry 0
```

18. Interrupt the EFI boot manager autoboot.

```
EFI Boot Manager ver 1.10 [14.60] Firmware ver 1.61 [4241]
[Escape]
```


© 2005 Hewlett-Packard Development Company, L.P.

19. Select your mirrored disk from the boot manager selection menu.

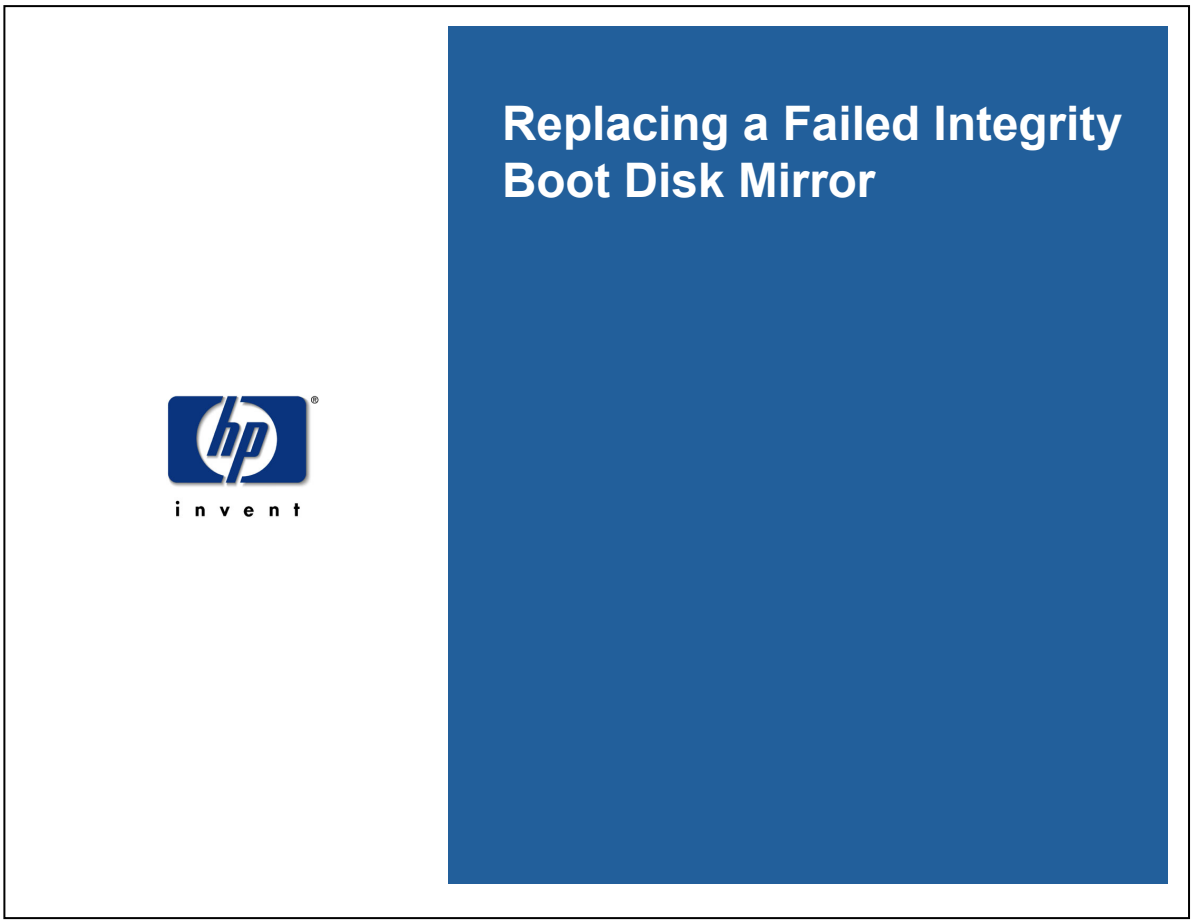
```
EFI Boot Manager ver 1.10 [14.60] Firmware ver 1.61 [4241]
Please select a boot option
   HP-UX Primary Boot
   HP-UX Alternate Boot
   EFI Shell [Built-in]
```

20. Verify which disk you booted from when the boot process completes.

```
# grep "Boot device's HP-UX HW path" /var/adm/syslog/syslog.log
  vmunix: Boot device's HP-UX HW path is: 0.0.0.0.1.0
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 5–35. SLIDE: Replacing a Failed Integrity Boot Disk Mirror




## Student Notes

The last issue we will consider is replacing a failed Integrity boot disk mirror. As you will see over the next few slides, the process is very similar to the process used to create the mirrored boot disk over the last few slides.


© 2005 Hewlett-Packard Development Company, L.P.

## 5-36. SLIDE: Rebuilding the Partition Table

### Rebuilding the Partition Table

1. Replace the failed disk and recreate the disk device files
```
# ioscan -fC disk
# insf -eC disk
```
2. Recreate the system, OS, and service partitions
```
# vi /tmp/idf
  3
  EFI 500MB
  HPUX 100%
  HPSP 400MB
# idisk -wf /tmp/idf /dev/rdsk/c0t1d0
```
3. Recreate and verify device files for the partitions
```
# insf -eC disk
```
4. Verify the partition table
```
# idisk /dev/rdsk/c0t1d0
```
5. Verify that the device files were created properly
```
# ioscan -funC disk
```

## Student Notes

After a mirrored boot disk is replaced, the GPT partition table must be rebuilt on the disk using the same procedure we used to configure the partition table initially.

1. Replace the failed disk and recreate the disk device files.

```
# ioscan -fC disk
# insf -eC disk
```

2. Recreate the system, OS, and service partitions.

```
# vi /tmp/idf
  3
  EFI 500MB
  HPUX 100%
  HPSP 400MB
# idisk -wf /tmp/idf /dev/rdsk/c0t1d0
```


**© 2005 Hewlett-Packard Development Company, L.P.**

3. Recreate and verify device files for the partitions.

```
# insf -eC disk
```

4. Verify the partition table.

```
# idisk /dev/rdsk/c0t1d0
```

5. Verify that the device files were created properly.

```
# ioscan -funC disk
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–37. SLIDE: Rebuilding the System Partition

### Rebuilding the System Partition

6. Populate the `/efi/hpux` directory in the EFI partition
```
# mkboot -e -l /dev/rdsk/c0t1d0
```

7. Change the AUTO file on both boot disks to `hpux -lq` so the host can boot without quorum
```
# vi /tmp/auto
  boot vmunix -lq
# efi_cp -d /dev/rdsk/c0t0d0s1 /tmp/auto /efi/hpux/auto
# efi_cp -d /dev/rdsk/c0t1d0s1 /tmp/auto /efi/hpux/auto
```

8. Verify the contents of the system partition
```
# efi_ls -d /dev/rdsk/c0t1d0s1
```

9. Verify the contents of the `auto` file in the system partition
```
# efi_cp -d /dev/rdsk/c0t1d0s1 -u /efi/hpux/auto /tmp/auto
# cat /tmp/auto
```

## Student Notes

After you recreate the partition table, repopulate the system partition using the same procedure you used earlier.

6. Populate the **`/efi/hpux/`** directory in the new EFI partition.

```
# mkboot -e -l /dev/rdsk/c0t1d0
```

7. Change the **`auto`** file on the replacement disk to **`hpux -lq`** so the host can boot without quorum.

```
# vi /tmp/auto
  boot vmunix -lq
# efi_cp -d /dev/rdsk/c0t1d0s1 /tmp/auto /efi/hpux/auto
```

8. Verify the contents of the system partition.

```
# efi_ls -d /dev/rdsk/c0t1d0s1
```


**© 2005 Hewlett-Packard Development Company, L.P.**

9. Verify the contents of the **auto** file in the system partition.

```
# efi_cp -d /dev/rdsk/c0t1d0s1 -u /efi/hpux/auto /tmp/efi.auto
# cat /tmp/efi.auto
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5–38. SLIDE: Rebuilding the OS Partition

### Rebuilding the OS Partition

10. Restore LVM headers to the disk
    `# vgcfgrestore -n vg00 /dev/rdsk/c0t1d0s2`

11. Reactivate the volume group to recognize the recovered disk
    `# vgchange -a y vg00`

12. Resynchronize mirrors
    `# vgsync vg00`

## Student Notes

Finally, rebuild the OS partition. Use **`vgcfgrestore`** to restore the LVM headers, then resynchronize the mirrored logical volumes from the remaining disk.

10. Restore LVM headers to the disk.

```
# vgcfgrestore -n vg00 /dev/rdsk/c0t1d0s2
```

11. Reactivate the volume group to recognize the recovered disk.

```
# vgchange -a y vg00
```

12. Resynchronize mirrors.

```
# vgsync vg00
```


© 2005 Hewlett-Packard Development Company, L.P.

## 5-39. SLIDE: Reviewing the Module Objectives

### Reviewing the Module Objectives

You should now be able to do the following:

- Describe the major components of an HP-UX 11i v2 LVM boot disk
- Initiate an HP-UX 11i v2 boot from a command prompt, the MP, or the EFI shell
- Navigate and manage EFI devices and file systems
- Obtain system configuration information using EFI commands
- Manage the EFI boot menu using EFI commands
- Boot from the default boot disk
- Boot from an alternate boot disk
- Boot from a system recovery tape
- Boot from a DVD or CDROM
- Boot from an Ignite-UX install/recovery server
- Boot from an Ignite-UX recovery tape
- Boot from an alternate kernel
- Boot to single-user, LVM maintenance, or tunable maintenance mode
- Configure a mirrored 11i v2 boot disk
- Recover a failed mirrored 11i v2 boot disk

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 5–40. SLIDE: For Further Study

### For Further Study

Intel offers a number of free and for-fee EFI related courses.
Look for the Extensible Firmware Interface course section on
`https://shale.intel.com/SoftwareCollege/CourseCatalog.asp`

Read the EFI shell documentation at
`http://www.intel.com/software/products/college/efishell/images/efi_shell_cmnd_1_1.pdf`

HP's *HP System Partitions Guide Administration for nPartitions* on
`http://docs.hp.com` includes a chapter on booting and resetting partitions on both HP-UX 11i v1 and HP-UX 11i v2

For HP-UX command syntax descriptions and options, see the HP-UX online man pages.

For EFI command syntax descriptions and options, access the EFI help screens by typing `help` at the `Shell>` prompt

To learn more about the HP Service Partition visit:
`http://docs.hp.com/hpux/onlinedocs/diag/ode/ode_ediag.htm`

## Student Notes

Several resources are available if you wish to learn more about the EFI.

Intel offers a number of free and for-fee EFI related courses and documentation resources on their website. Look for the Extensible Firmware Interface section on **`https://shale.intel.com/SoftwareCollege/CourseCatalog.asp`**.

To learn more about the EFI shell, read the EFI shell documentation at **`http://www.intel.com/software/products/college/efishell/images/efi_shell_cmnd_1_1.pdf`**.

HP's *HP System Partitions Guide Administration for nPartitions* on **`http://docs.hp.com`** includes a chapter on booting and resetting node partitions.


**© 2005 Hewlett-Packard Development Company, L.P.**

For HP-UX command syntax descriptions and options, see the HP-UX online man pages. The following man pages are particularly relevant:

```
boot(1m)
efi(4)
efi_cp(1m)
efi_fsinit(1m)
efi_ls(1m)
efi_mkdir(1m)
efi_rm(1m)
efi_rmdir(1m)
hpux.efi(1m)
idisk(1m)
ioscan(1m)
insf(1m)
lvm(7)
mkboot(1m)
setboot(1m)
```

For EFI and POSSE command syntax descriptions and options, access the EFI help screens by typing **`help`** at the **`Shell>`** prompt.

To learn more about the HP Service Partition visit:
**`http://docs.hp.com/hpux/onlinedocs/diag/ode/ode_ediag.htm`**


**© 2005 Hewlett-Packard Development Company, L.P.**

# 5–41. LAB: Booting HP-UX 11i v2 via the EFI

## Directions

Carefully follow the instructions below.

## Part 1: Interacting with the EFI Boot Manager and Shell

The first parts of the lab gave you an opportunity to interact with the EFI boot manager menu and some HP-UX EFI commands. This part of the lab provides an opportunity to explore some of the commands and utilities that are available at the EFI shell prompt.

1. Access the MP console interface, and login as user **`root`**.

2. Reboot your system, interrupt the autoboot process, and select the EFI shell option from the boot manager menu.

3. Use the EFI **`info -b all`** command to view your system configuration.

   a. What is your system model string?

   b. How many CPUs do you have?

   c. How much memory do you have?

   d. How many LAN interfaces do you have? What are their MAC addresses?

4. Use the **`map -r -fs`** command to view a list of potential boot devices. Record the **`fs`*`n`*`:`** names below, and see if you can determine which hardware path each **`fs`*`n`*`:`** name represents. You may need to refer back to the EFI hardware addressing slide earlier in the book to interpret the hardware paths.


**© 2005 Hewlett-Packard Development Company, L.P.**

5. Type **`fsn:`** to select the system partition on your primary boot disk, where **`fsn:`** is the drive name that you identified in the previous step.

6. Use the EFI file system navigation commands that you learned in the chapter to explore the file system and find the **`hpux.efi`** OS loader and the **`auto`** file. What are the full pathnames for these files?

7. Which EFI shell command can be used to view the contents of the **`auto`** file? Try it!

8. Use the appropriate **`bcfg`** arguments to dump the current boot menu to the screen. (Note: the **`bcfg`** dump command output may be greater than 80 characters wide. Adjust your terminal emulator settings to 132 columns to see the entire line)

9. Add a new entry to the boot menu. Since we don't have another actual bootable device available on the system, simply add a second a second boot menu option for your primary boot disk. Use "My primary boot disk", or perhaps a more creative label, for your new boot menu option.

10. Dump the EFI boot menu to the screen again to verify your work.

11. Since we don't really want two entries in the boot menu for the same disk, remove the boot menu option that you just added.


**© 2005 Hewlett-Packard Development Company, L.P.**

12. Dump the EFI boot menu to the screen again to verify your work.

13. Boot the system from the EFI shell. When given an opportunity to interrupt the HP-UX kernel loader autoboot, press `Escape`.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 2: Interacting with the HPUX Kernel Loader

In this part of the lab, you will have a chance to explore the HPUX kernel loader.

1. If you didn't interrupt the autoboot process in the previous part of the lab, reboot again, and press **Escape** when the HP-UX autoboot message appears.

```
HP-UX Boot Loader for IA64  Revision 1.71
Press Any Key to interrupt Autoboot
\EFI\HPUX\AUTO ==> boot vmunix
Seconds left till autoboot - 10  Escape
  Type 'help' for help
```

2. Type the **help** command to view a list of available commands.

3. Execute the command to list the kernels available in **/stand**.

4. Which command can you use to view the contents of the **auto** file? What is the purpose of the **auto** file?

5. Boot **vmunix** to single-user mode.

6. Once you reach single-user mode, use the HP-UX `init 3` command to transition to run-level 3.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 3: Mirroring the Boot Disk: Configuring Partitions

The next few parts of the lab walk you through the process required to mirror an HP-UX 11i v2 boot disk.

1. Use **`lvlnboot`** to determine the location of your current boot disk, then use **`ioscan`** and **`diskinfo`** to identify a disk large enough to serve as a mirror.

2. Use the **`idisk`** command to create a 500MB system partition, an OS partition, and a 400MB service partition on your new boot disk.

3. Create device files for the new partitions.

4. Use the **`idisk`** command to verify the partition table on the new boot disk.

5. Verify that the partition device files were created properly. There should be block and raw ***`cxtxdx`***`s1`, ***`cxtxdx`***`s2`, and ***`cxtxdx`***`s3` device files for the new boot mirror.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 4: Mirroring the Boot Disk: Configuring the System Partition

After you create the partition table, you can initialize and populate the EFI system partition.

1. Populate the **`/efi/hpux`** directory in the new EFI partition.

2. Change the **`auto`** file on both boot disks so the host can boot without quorum. Change the boot string in the file to **`boot vmunix -lq`**. (Hint: Be sure to specify the s1 partition!)

3. Verify the contents of the system partition. (Hint: Be sure to specify the **`s1`** partition.)

4. Verify the contents of the **`auto`** file in the system partition.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 5: Mirroring an 11i v2 Boot Disk: Configuring the OS Partition

Configure the OS partition, essentially an encapsulated LVM boot disk.

1. Initialize and add the new OS partition to vg00 (Hint: Be sure to specify s2, the OS partition!).

2. Mirror *all* of the logical volumes in vg00.

3. Add a line to **`/stand/bootconf`** so SDUX knows which disks are boot disks (Hint: Be sure to specify s2, the OS partition!).

```
# vi /stand/bootconf
    l /dev/dsk/c0t0d0s2
    l /dev/dsk/c0t1d0s2
```

4. Verify that the disk was added to vg00, and that the logical volumes are synchronized. The LV Status field for each logical volume should report **`syncd`**.

5. Verify that the BDRA was updated properly. You should see references to both disks in the boot, root, and swap portions of the output.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 6: Configuring an HP-UX 11i v2 Boot Disk: Configuring the EFI Boot Manager Menu

Add the new mirrored disk to the EFI boot manager menu. This ensures that the EFI will automatically attempt to boot the mirrored boot disk if the primary boot disk fails.

1. Ensure that your initial boot disk is configured as your primary boot source, and add your new boot disk to the EFI boot manager as the second (i.e. high availability alternate) boot source.

2. Verify that your new boot path was recorded properly in NVRAM.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 7: Mirroring the Boot Disk: Testing the New Mirror

Boot from the new mirror disk.

1. Use the **`shutdown`** command to initiate the reboot sequence. When given an opportunity to interrupt the EFI autoboot sequence by pressing the down arrow key.

2. Select the mirrored disk from the boot manager selection menu and proceed with the boot process.

```
EFI Boot Manager ver 1.10 [14.60] Firmware ver 1.61 [4241]
Please select a boot option
   HP-UX Primary Boot
   HP-UX Alternate Boot
   EFI Shell [Built-in]
```

3. When the boot process finishes, check the **`syslog.log`** file to verify which disk/kernel you booted from.

```
# grep "Boot device's HP-UX HW path" /var/adm/syslog/syslog.log
  vmunix: Boot device's HP-UX HW path is: 0.0.0.0.1.0
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 8: Replacing a Failed Mirror: Reconfiguring Partitions

The next three sections of the lab exercise walk through the process of replacing a failed HP-UX 11i v2 boot disk mirror.

1. Overwrite the first 1MB of your system and OS partitions on the primary boot disk to render it unbootable. Reboot the system to see what impact this has on your machine. Watch the console messages carefully.

```
# dd if=/dev/zero of=/dev/rdsk/cxtxdxs1 bs=1024 count=1024
# dd if=/dev/zero of=/dev/rdsk/cxtxdxs2 bs=1024 count=1024
# shutdown -ry 0
```

2. Once your system reboots from the remaining mirror, begin the recovery process. Type **`ioscan`** and **`insf`** to verify that the disk's device files are still in tact (they should be).

3. Recreate the system, OS, and service partitions.

4. Recreate and verify device files for the partitions if necessary.

5. Verify that the partition table was successfully recreated.

6. Verify that the partition device files were created properly.

```
# ioscan -funC disk
```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 9: Replacing a Failed Mirror: Reconfiguring the System Partition

After you recreate the partition table, repopulate the system partition using the same procedure you used earlier.

1. Populate the **`/efi/hpux/`** directory in the new EFI partition.

2. Change the **`auto`** file on the replacement disk to **`hpux -lq`** so the host can boot without quorum.

3. Verify the contents of the system partition.

4. Verify the contents of the **`auto`** file in the system partition.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 10: Replacing a Failed Mirror: Rebuilding the OS Partition

Rebuild the OS partition. Use **`vgcfgrestore`** to restore the LVM headers, then resynchronize the mirrored logical volumes from the remaining disk.

1. Restore LVM headers to the disk.

2. Reactivate the volume group to recognize the recovered disk.

3. Resynchronize mirrors.

## Part 11: Replacing a Failed Mirror: Rebuilding the EFI Boot Menu

Update the EFI boot menu so it references the GUID of the disk's new GUID rather than its old GUID.

1. Use the **`setboot`** command to re-define the primary boot path.

2. Run **`setboot`** without any options to verify your work.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 11: Cleaning Up

Before proceeding, remove any EFI boot manager menu items that you may have added during the lab.

1. Reboot your system, interrupt the EFI boot manager autoboot process, and access the EFI shell.

```
EFI Boot Manager ver 1.10 [14.60]  Firmware ver 1.61 [4241]
Please select a boot option

    HP-UX Primary Boot
    HP-UX HA Alternate Boot
    EFI Shell [Built-in]
    Boot option maintenance menu
    Security/Password Menu
```

2. Dump the contents of the EFI boot manager menu to the screen.

3. Use the **`bcfg`** command to remove any EFI boot menu items that you added over the course of the lab. Verify your work by dumping the remaining menu items to the screen.


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 6 — Booting: Missing or Corrupt ASCII Startup Files



### Objectives

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

- Describe the system startup process after the operating system is loaded and binary startup files are executed.
- Identify important system configuration files and start links.
- Identify function and importance of `/etc/rc.config.d`.
- Boot into single-user mode.
- Systematically identify, check, and fix corrupted system configuration files and missing start links.
- Explain the significance of an invalid or corrupted configuration file existing in the `/etc/rc.config.d`.


© 2005 Hewlett-Packard Development Company, L.P.

## 6–1. SLIDE: Module Objectives

### Module Objectives

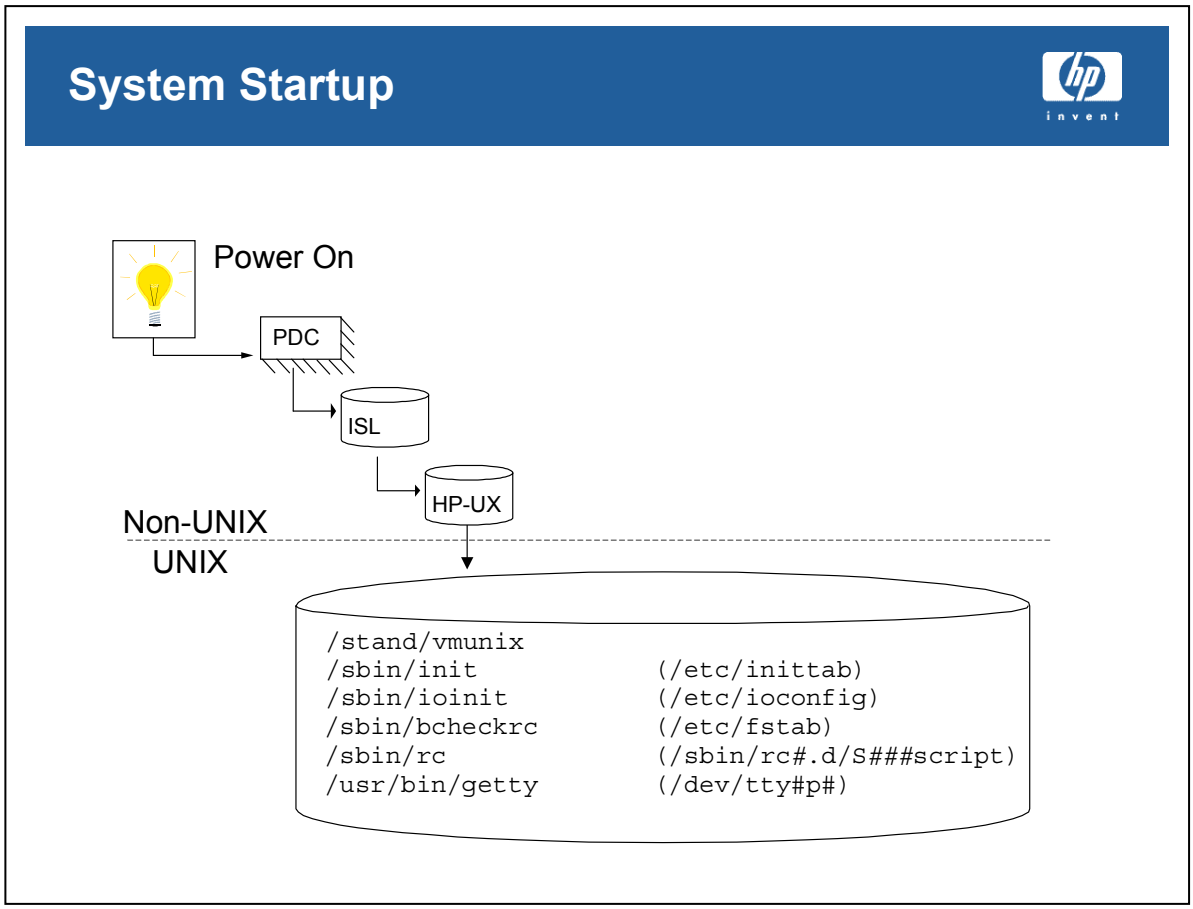
By the end of the module, you should be able to…

- Describe the system startup process after the operating system is loaded and binary startup files are executed.
- Identify important system configuration files and start links.
- Identify function and importance of `/etc/rc.config.d`.
- Boot into single-user mode.
- Systematically identify, check, and fix corrupted system configuration files and missing start links.
- Explain the significance of an invalid or corrupted configuration file existing in the `/etc/rc.config.d` directory.

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 6–2. SLIDE: System Startup





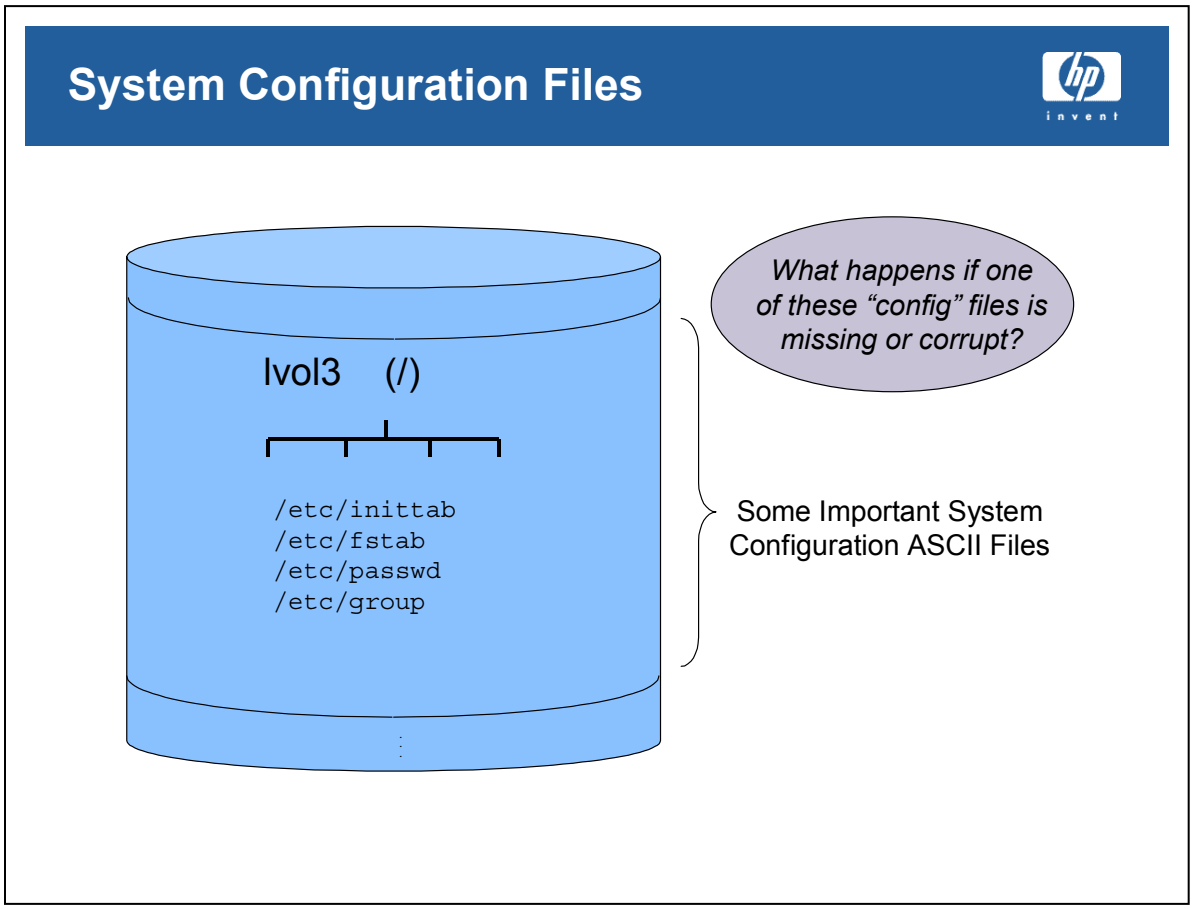

## Student Notes

Once the operating system has been loaded and binary startup files are executed, a number of system configuration files and start links are needed for system customization. Without some of these files, the system will still boot, but with problems. Some likely problems: processes that do not start and file systems that do not mount.

Two of the more important system configuration files are **`/etc/inittab`** and **`/etc/fstab`**. These files control which processes get started and which file systems are mounted, respectively.


© 2005 Hewlett-Packard Development Company, L.P.

## 6–3. SLIDE: System Configuration Files




## Student Notes

System configuration files define the customization of a system. Some of these customizations include establishing default run-level, processes to start, file systems to mount, valid user identifications, valid group declarations, and so on.

A corrupt or missing system configuration file can cause problems. Certain symptoms are present, depending on which file is bad. Below are system configuration files and symptoms present when the file is bad:

**`/etc/inittab`**

- As the system is booting, it prompts for a default run-level.
- The system checklist keeps reappearing.
- Processes spawn too rapidly.

**`/etc/fstab`**

- After booting, the file system is not mounted.
- Multiple "FAILS" are present in the system checklist.
- System boot may hang if fields are missing.


© 2005 Hewlett-Packard Development Company, L.P.

| | |
|---|---|
| **`/etc/passwd`** | • A user cannot log in.<br>• A message noting that root's shell should be **`/sbin/sh`**. |
| **`/etc/group`** | • Long listings display numbers (group IDs) in the group field instead of group names. |

In the event system configuration files are missing or corrupt, try the following:

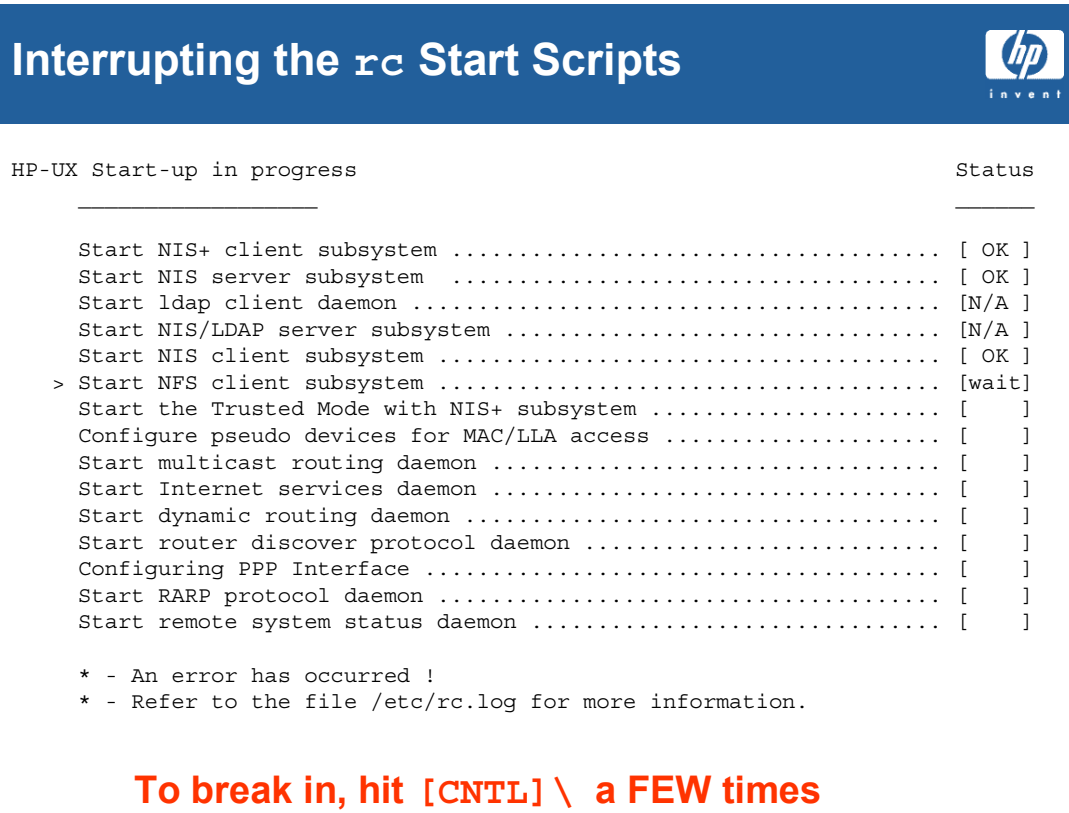
- Restore system configuration file(s) from backup media.
- Obtain a clean copy of system configuration file(s) from **`/usr/newconfig/etc`**.
- Manually fix affected system configuration file(s).




© 2005 Hewlett-Packard Development Company, L.P.

## 6–4. SLIDE: Interrupting the `rc` Start Scripts

### Interrupting the `rc` Start Scripts




## Student Notes

The main task of starting up the UNIX environment, once the kernel has loaded, is executed by **`/sbin/rc`**. This script is run whenever the run-level is changed (unless **`init s`** is used to drop straight to **`init`**'s single-user mode[1].)

**`/sbin/rc`** is responsible for running many other scripts. If one of these scripts hangs on startup, the startup process itself will hang. This can be observed by a flashing busy/wait on the screen. The key combination, **`<cntl> \`** , sends a **`QUIT`** signal to the **`/sbin/rc`** script, allowing you access to the system. Note that often this does not work immediately and may need to be repeated.

Other key sequences that may allow **`/sbin/rc`** script interruption are `<cntl> c` **or** `<cntl> <break>`.

[1] **`init`**'s run level **`s`** is a special case; **`init`** does not process the **`inittab`** file. When **`init`** changes run level, it looks at what it is supposed to start at the new run level, and what it started at the old run level. It then kills off the difference. However, since **`init`** itself starts very little, changing to run level "**`s`**" will result in very little being killed off. Usually **`inittab`** tells **`init`** to use **`rc`** to shutdown programs not expected at the new run level.


**© 2005 Hewlett-Packard Development Company, L.P.**

If all else fails, you will have to reboot the system into single-user mode. On all the supported workstations and servers running HP-UX 11i , the on/off switch has logic circuitry built-in to perform a **`reboot -h`** on the system. This should shut down the operating system cleanly, though it may not shut down your applications properly.

When interrupting the boot sequence, you should get a login prompt, but it will probably be mixed up in the middle of the boot menu. If you have an older style HP console, you can press the **Clear Display** key. On others consoles, it is usually best to hold down the **Enter** key until the login appears on clear screen space.

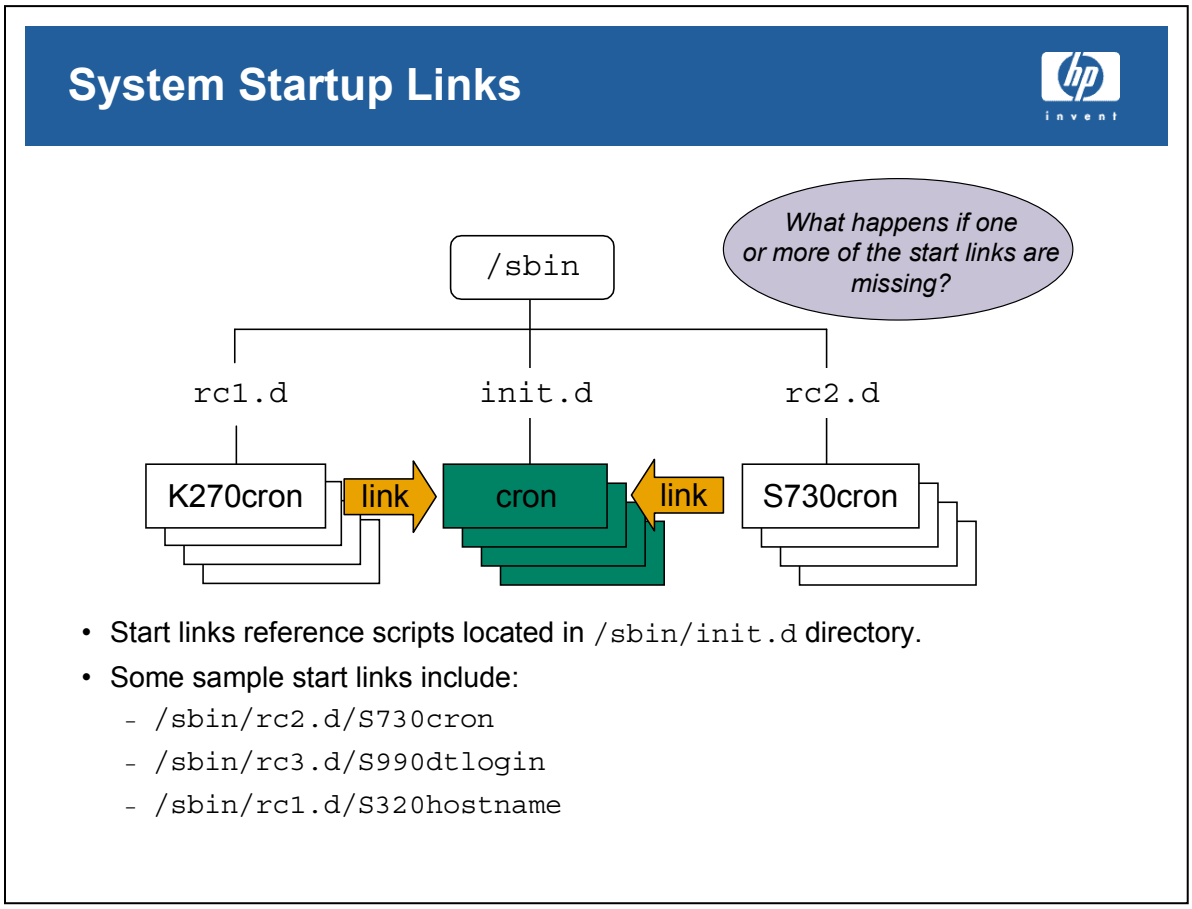


```
     HP-UX Start-up in progress                                        Status
     __________________________                                        ______

     Start NIS+ client subsystem ..................................... [ OK ]
     Start NIS server subsystem ...................................... [ OK ]
     Start ldap client daemon ........................................ [N/A ]
     Start NIS/LDAP server subsystem ................................. [N/A ]
     Start NIS client subsystem ...................................... [ OK ]
   NFS server (pid845@/net) not responding still trying............... [busy]
GenericSysName [HP Release B.11.11] (see /etc/issue)................... [    ]
Console Login: pseudo devices for MAC/LLA access ..................... [    ]
     Start multicast routing daemon .................................. [    ]
     Start Internet services daemon .................................. [    ]
     Start dynamic routing daemon .................................... [    ]
     Start router discover protocol daemon ........................... [    ]
     Configuring PPP Interface ....................................... [    ]
     Start RARP protocol daemon ...................................... [    ]
     Start remote system status daemon ............................... [    ]

     * - An error has occurred !
     * - Refer to the file /etc/rc.log for more information.
```

This system had misconfigured network settings, and had failed to start the networking; therefore, it hung when trying to start the automounter as part of NFS. Most LAN failures will result in the boot sequence hanging at this point.


© 2005 Hewlett-Packard Development Company, L.P.

## 6–5. SLIDE: System Startup Links




## Student Notes

Start links are symbolic links. These symbolic links reference startup scripts located in the **`/sbin/init.d`** directory. The scripts in this directory are responsible for starting many of the services that allow the system to be in a multiuser state.

If there are invalid, improperly named, or missing start links, those specific services will not start. Common start links worth noting that can have a negative effect on the system when missing are listed below:

**`/sbin/rc3.d/S990dtlogin.rc`** → **`/sbin/init.d/dtlogin.rc`**
Set up and configure CDE environment.

**`/sbin/rc2.d/S340net`** → **`/sbin/init.d/net`**
Set up and configure core-networking environment.

**`/sbin/rc1.d/S320hostname`** → **`/sbin/init.d/hostname`**
Set up and configure system's hostname.

If missing start links are discovered, they must be recreated with the **`ln -s`** command. An example of how to reestablish the CDE start link is documented below:


**© 2005 Hewlett-Packard Development Company, L.P.**

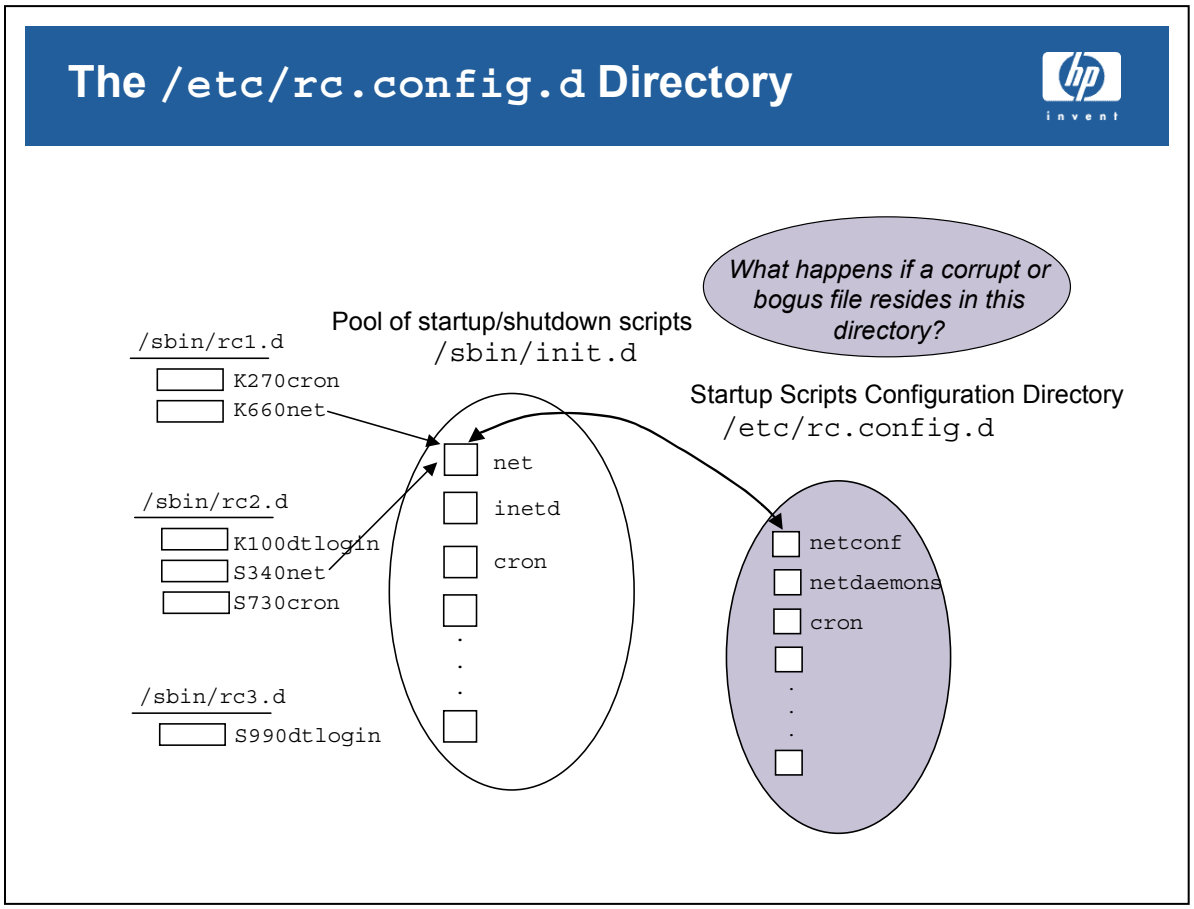
```
# ln -s /sbin/init.d/dtlogin.rc /sbin/rc3.d/S990dtlogin.rc
```

Error messages from the **rc** scripts are written to the **/etc/rc.log** file.




© 2005 Hewlett-Packard Development Company, L.P.

## 6–6. SLIDE: The `/etc/rc.config.d` Directory




## Student Notes

Many files exist within the **`/etc/rc.config.d`** directory. These files are configuration files used by startup scripts. Without these files, startup scripts do not have the necessary information needed to run properly.

If files are missing, corrupt, or contain erroneous information, problems will occur. Following are some of the problems that can occur and their solutions:

| | |
|---|---|
| **`Service(s) will not start.`** | Go to the affected configuration file and modify **`CONTROL_VARIABLE`**. The **`CONTROL_VARIABLE`** determines whether or not to start a specific service. A value of "1" starts the service. |
| **`Invalid configuration information is being used.`** | Go to the affected configuration file and modify specific configuration variable(s). In cases such as configuring an IP address and hostname, other variables are used. These variables must be updated with appropriate information. |


**© 2005 Hewlett-Packard Development Company, L.P.**

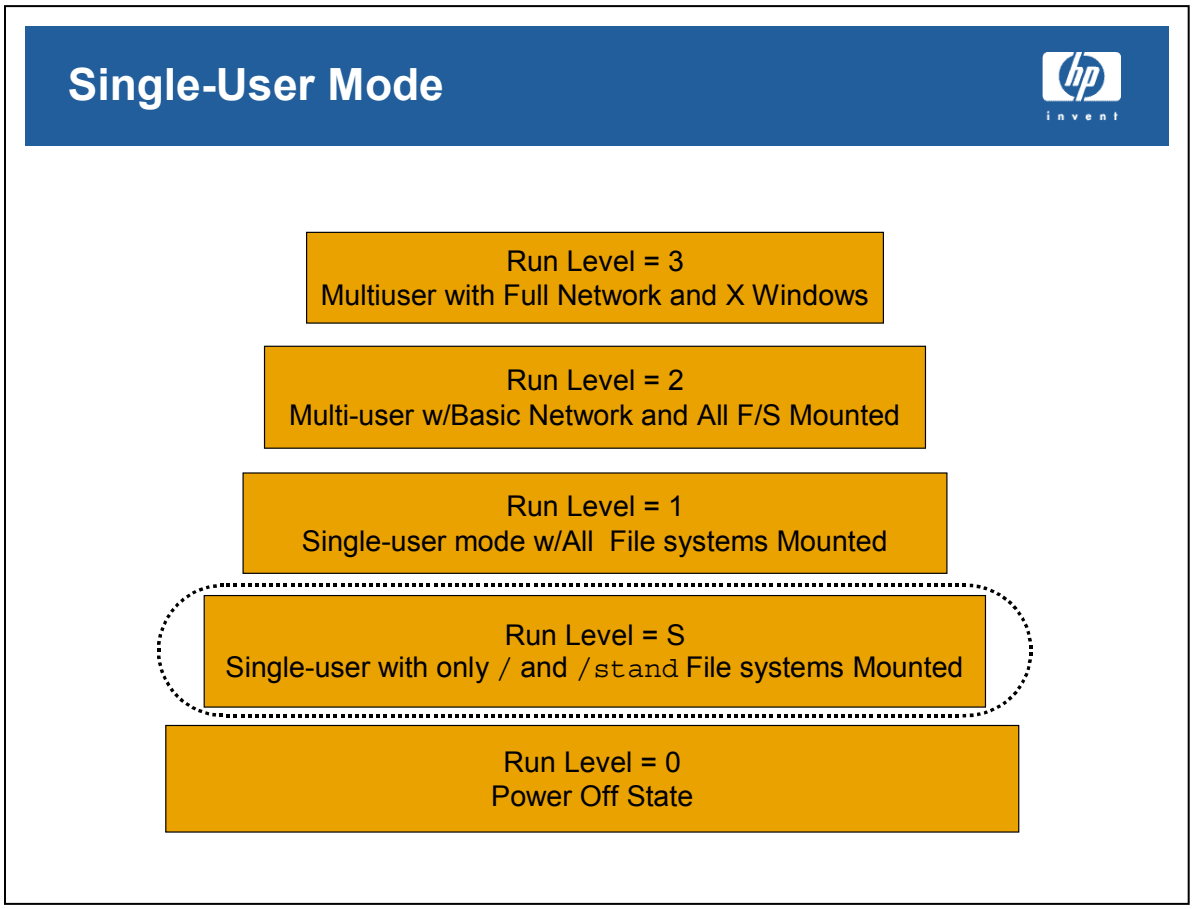
| | |
|---|---|
| **`Errors resulting in checklist.`** | Go to **`/etc/rc.config.d`** and locate the configuration file(s) containing syntax errors. Once located, fix or remove them. Errors are occurring in the checklist because a great number of configuration files in **`/etc/rc.config.d`** are being sourced into most startup scripts. |

Most of the startup scripts in **`/sbin/init.d`** call specific configuration scripts in **`/etc/rc.config.d`**. However, some will call **`/etc/rc.config`**, which then calls all the files in the **`/etc/rc.config.d`** directory. If any file in the **`/etc/rc.config.d`** directory is not a valid shell script, problems can occur at boot time. Do not store old versions of files in this directory, because they may be run after the current version and overwrite the new settings.




© 2005 Hewlett-Packard Development Company, L.P.

## 6–7. SLIDE: Single-User Mode




## Student Notes

Single user mode is a special run level where **`init`** does not process the **`/etc/inittab`** file except at boot time, where it processes the **`sysinit`** entries only. In this mode, **`init`** starts a root shell on the console by invoking the **`su`** command, if available (it probably isn't), or it runs **`/usr/bin/sh`** (probably not available either), and last, tries **`/sbin/sh`**.

Since none of the normal entries in **`/etc/inittab`** are processed, all the software they typically start will not be running.

Instead, only kernel processes, **`init,`** and the root shell, will be running. Only **`/`** and **`/stand`** will be mounted. (**`/sbin/ioinitrc`** needs to mount this file system to check **`/stand/ioconfig`**).

```
# ps -ef
sh: ps:  not found
# mount -a
mount: /dev/vg00/lvol1 is already mounted on /stand
vxfs mount: Cannot open /dev/vg01/lvol1: No such device or address
# ps -ef
     UID   PID  PPID  C    STIME TTY       TIME COMMAND
```


© 2005 Hewlett-Packard Development Company, L.P.

```
    root     0     0  0 11:26:11 ?         0:01 swapper
    root     8     0  0 11:26:11 ?         0:00 supsched
    root     9     0  0 11:26:11 ?         0:00 strmem
    root    10     0  0 11:26:11 ?         0:00 strweld
    root    11     0  0 11:26:11 ?         0:00 strfreebd
    root     2     0  0 11:26:11 ?         0:00 vhand
    root     3     0  0 11:26:11 ?         0:00 statdaemon
    root     4     0  0 11:26:11 ?         0:00 unhashdaemon
    root    12     0  0 11:26:11 ?         0:00 ttisr
    root    13     0  0 11:26:11 ?         0:00 ioconfigd
    root     1     0  0 11:26:12 ?         0:00 init s
    root    19     0  0 11:26:12 ?         0:00 lvmkd
    root    20     0  0 11:26:12 ?         0:00 lvmkd
    root    21     0  0 11:26:12 ?         0:00 lvmkd
    root    22     0  0 11:26:12 ?         0:00 lvmkd
    root    23     0  0 11:26:12 ?         0:00 lvmkd
    root    24     0  0 11:26:12 ?         0:00 lvmkd
    root    52     1  0 11:26:16 console   0:00 sh
    root    71    52  3 11:29:49 console   0:00 ps -ef
    root    30     0  0 11:26:13 ?         0:00 vxfsd
#
```

Note, the first **ps** failed, as its path is **/usr/bin/ps**, and **/usr** is not mounted. Here, **mount** failed to find file systems outside the root volume group. Normally, volume groups are activated from **/sbin/lvmrc**, which is invoked by **/etc/bcheckrc** which, of course, has not been run.

One very annoying feature of HP-UX 11.11 is that **init** has executed **/sbin/sh** with its name set to "**sh**", and not "**-sh**", as in earlier versions of HP-UX. The leading "**-**" in the name of a shell tells the shell that it is a login shell. It then runs **/etc/profile**, and **~/.profile** to set up your normal working environment. Under HP-UX 11.11, this does not happen. To resolve this, execute:

```
# . /etc/profile                # Note the leading dot space.
# . ~/.profile
```

Otherwise, you will have to remember to type things like **#** instead of backspace!

Once you have finished with single user mode, you can change to multi-user mode by using the **init** command to enter the appropriate run level, usually run level 3 on HP-UX.

```
# cat /etc/inittab
init:3:initdefault:
.
.
.
# init 3
```




© 2005 Hewlett-Packard Development Company, L.P.

## 6–8. SLIDE: Booting to Single-User Mode on PA-RISC

### Booting to Single-User Mode on PA-RISC

- Power up the system ( or <cntl> B and RS )
- Interrupt the boot
- At BCH prompt > `boot pri`
  Do you wish to interact with IPL? `y`
- `ISL > hpux -is`

  ...Booting to single user mode

## Student Notes

You can bring your system into single-user mode in one of two ways. You can use the **`shutdown`** command while the system is running in multiuser mode. The exact syntax is as follows:

```
# cd /
# shutdown
```

Notice, the **`shutdown`** command is issued with no specified options, which forces the **`shutdown`** command to take the system into single-user mode.

Another way to bring your system into single-user mode is to follow the steps below during bootup.

1. Power cycle the system.
2. Interrupt the boot sequence.
3. From the PDC prompt, enter **`boot pri`**.
4. From the ISL prompt, enter **`hpux -is`**.


© 2005 Hewlett-Packard Development Company, L.P.

After going through these steps, you will see a "**#**" prompt. This prompt is the default prompt for the root user.




**© 2005 Hewlett-Packard Development Company, L.P.**

## 6–9. SLIDE: Module Objectives

### Module Objectives

You should now be able to do the following:

- Describe the system startup process after the operating system is loaded and binary startup files are executed.
- Identify important system configuration files and start links.
- Identify function and importance of `/etc/rc.config.d`.
- Boot into single-user mode.
- Systematically identify, check, and repair corrupted system configuration files and missing start links.
- Explain the significance of an invalid or corrupted configuration file existing in the `/etc/rc.config.d` directory.

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 6–10. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. List five ASCII startup or start link files.

2. Which ASCII startup file ranks as one of the most important?

3. What is the purpose and function of `/etc/rc.config.d`?

4. What two ways are used to get into single-user mode?




© 2005 Hewlett-Packard Development Company, L.P.

# 6–11. LAB: Recovering an ASCII Startup File, Start Link, and Startup-Script Configuration File

## Lab Directions

Since the exercises in this lab will cause your system to reboot, be sure to login to your system via the MP. This way, you will see the entire shutdown and startup process.

## Part 1

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links and/or startup-script configuration files.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.

2. Execute the **`mod6.lab1.sh`** script

   ```
   # cd /home/labs
   # ./mod6.lab1.sh
   ```

   When your system reboots, the boot and startup processes will partially succeed. This occurs because one of the ASCII startup files is corrupt or missing. Use the knowledge gained in this module to fix this problem.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 2

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links and/or startup-script configuration files.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.

2. Execute the **`mod6.lab2.sh`** script

   ```
   # cd /home/labs
   # ./mod6.lab2.sh
   ```

   When your system reboots, the boot and startup processes will partially succeed. This occurs because one of the ASCII startup files is corrupt or missing. Use the knowledge gained in this module to fix this problem.




© 2005 Hewlett-Packard Development Company, L.P.

## Part 3

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links or startup-script configuration files.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.

2. Execute the **`mod6.lab3.sh`** script

   ```
   # cd /home/labs
   # ./mod6.lab3.sh
   ```

   When your system reboots, the boot and startup processes will partially succeed. This occurs because one of the startup-script configuration files is bogus, corrupt, or missing. Using the knowledge gained in this module, fix this problem.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 4 (Optional)

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links or startup-script configuration files.

1. If you are attending this course in the RAIL environment, please skip this lab.

   If you are accessing your system directly (not via RAIL or telnet), login at the CDE login screento the system console.

2. Execute the **`mod6.lab4.sh`** script

   ```
   # cd /home/labs
   # ./mod6.lab4.sh
   ```

   When your system reboots, the boot and startup processes will partially succeed. This occurs because one of the start links is corrupt or missing. Using the knowledge gained in this module, fix this problem.




© 2005 Hewlett-Packard Development Company, L.P.

## Part 5 (Optional)

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links or startup-script configuration files.

1. In this exercise, work with a lab partner who can telnet to your system after you have executed the script below.

2. Execute the **`mod6.lab5.sh`** script

   ```
   # cd /home/labs
   # ./mod6.lab5.sh
   ```

   Now have your lab partner attempt to login to your system. Then will not be successful because an important ASCII file is not present. Using the knowledge gained in this module, fix this problem.


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 7 — Troubleshooting Login Problems

## Objectives

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

- Describe the login process as it pertains to a **`getty`**, **`telnet`**, and the common desktop environment (CDE) session.
- Identify and fix login problems relating to a **`getty`**, **`telnet`**, and CDE session.
- Explain the benefits of password shadowing




© 2005 Hewlett-Packard Development Company, L.P.

## 7–1. SLIDE: Module Objectives

### Module Objectives

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

- Describe the login process as it pertains to a `getty`, `telnet`, and the common desktop environment (CDE) session.
- Identify and fix login problems relating to a `getty`, `telnet`, and CDE session.
- Explain the benefits of password shadowing

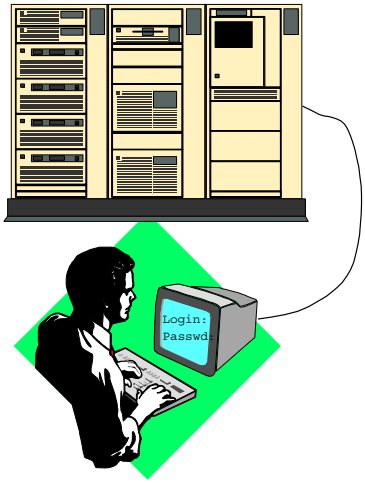
## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 7–2. SLIDE: Login Methods

### Login Methods

- ASCII Logins
  - Direct connect (for example, `getty`)
  - Network (for example, `telnet`)
- Graphical user interface (GUI)
  - CDE




## Student Notes

In today's UNIX environment, several methods can be used to gain access to a system. While the most popular access method is via a network using a terminal emulation program on a PC, the serial connection is still viable.

Connection methods include establishing a **`getty`**, **`telnet`**, and CDE sessions. A **`getty`** allows a user to gain access with a standard dumb terminal. CDE sessions allow users to gain access through a graphical environment using X11 libraries.

**`init`** is usually configured to start a **`getty`** on the console at run level 1; for other directly connected terminals, a **`getty`** is usually started at run levels 2 and above.

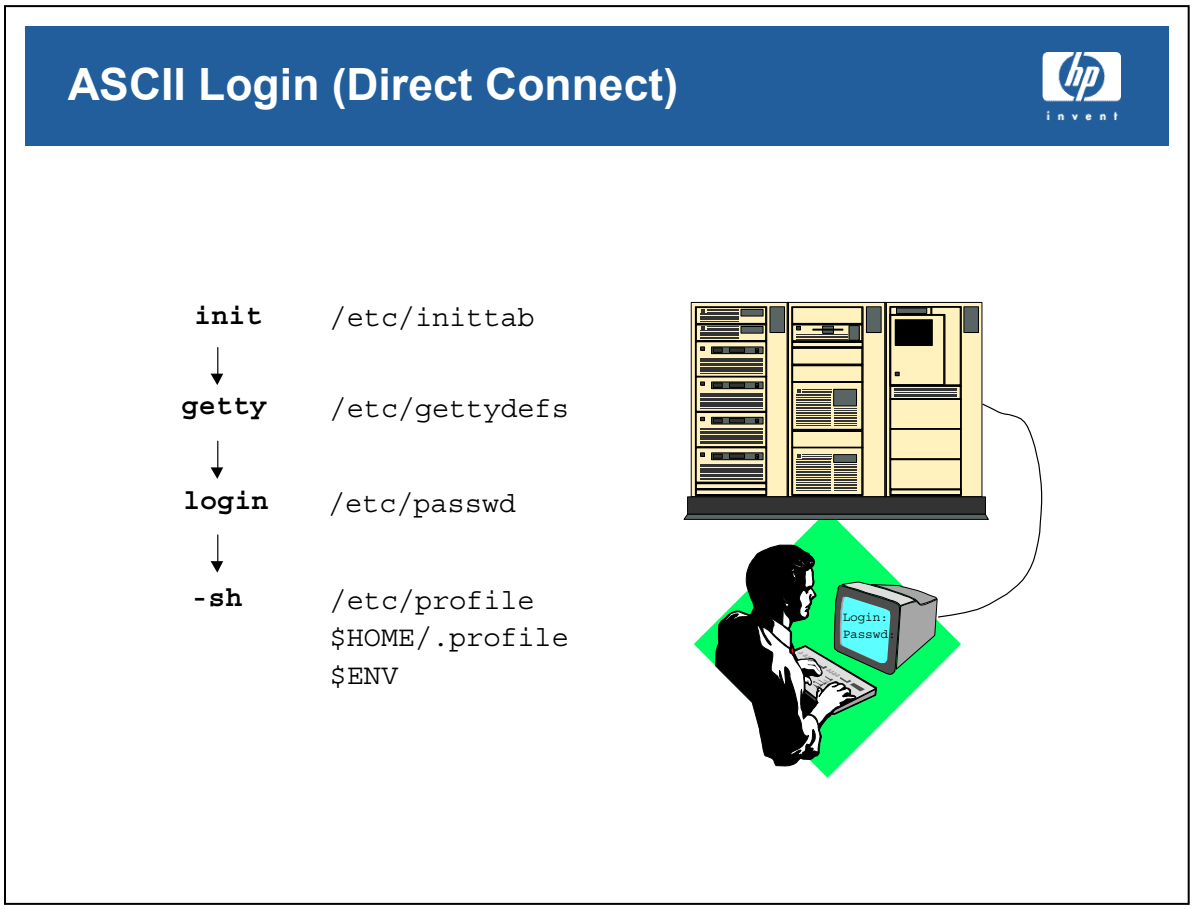
Network-based logins are started by default with **`/sbin/rc`** at run level 2. The X Windows/CDE login is usually started by **`/sbin/rc`** at run level 3.

Other configurations must be in place as well. These configurations will be discussed later in this module.




© 2005 Hewlett-Packard Development Company, L.P.

## 7–3. SLIDE: ASCII Login (Direct Connect)




## Student Notes

### Establishing a `getty` Session

- The **`init`** process:

  The **`init`** process is one of the first processes to start after bootup. This process references **`/etc/inittab`**. Within this file, there is an entry starting a **`getty`** process.

  Failure to get a login prompt on a directly connected terminal can be caused by many things.

  – No **`getty`** declaration in the **`/etc/inittab`** file
  – Incorrect run-level or action keyword declaration for **`getty`** entry
  – Errors in the **`gettydefs`** file
  – Cabling problems
  – Incorrectly configured serial line settings on the terminal
  – Break signals have been sent to the system. **`getty`** has the ability to cycle between different sets of serial line settings. Sending the break signal on the line triggers this.


© 2005 Hewlett-Packard Development Company, L.P.

Break is not a character; it is a long "*spike*" on the signal line. Various things, potentially even turning the terminal off and on, can cause these spikes! Choosing a **gettydef** entry that remains fixed is the best practice here. Unfortunately, the defaults do not.

- The **getty** process:
  - Configures serial port connecting terminal using **/etc/gettydefs**
  - Prints the **/etc/issue** file
  - Prompts and reads the user's login name
  - Reconfigures the serial port using **/etc/gettydefs**
  - Executes the login process

  Listed below are problems resulting from a corrupted or incorrect **/etc/gettydefs**:
  - Garbage displayed on screen
  - Slow transmission of text to and from the screen

  If you are sure that the terminal settings match the line settings requested in the **gettydefs** file, and you still are having these problems, check the settings that are actually in use on the line with:

```
stty -a < /dev/tty<<your port>>
```

  If they do not match the ones you are expecting, kill the **getty** and try again. You are probably experiencing a feature of **getty** that was very useful on prehistoric modems when the modem always communicated with the system at the speed of the remote user's connection.

  To cope with this situation, **getty** has the ability to cycle through various settings from the **gettydefs** file. The entries in the **gettydefs** file look like this:

```
..
..
..
1200      # B1200 HUPCL  IGNPAR PARENB ICRNL IXON OPOST ONLCR CS7 CREAD
                        ISIG ICANON ECHO ECHOK PARENB ISTRIP IXANY TAB3
          # B1200       SANE CS7 PARENB ISTRIP IXANY TAB3 HUPCL
                             #login: #2400

2400      # B2400 HUPCL  IGNPAR PARENB ICRNL IXON OPOST ONLCR CS7 CREAD
                        ISIG ICANON ECHO ECHOK PARENB ISTRIP IXANY TAB3
          # B2400       SANE CS7 PARENB ISTRIP IXANY TAB3 HUPCL
       #login: #4800

4800      # B4800 HUPCL  IGNPAR PARENB ICRNL IXON OPOST ONLCR CS7 CREAD
                        ISIG ICANON ECHO ECHOK PARENB ISTRIP IXANY TAB3
          # B4800       SANE CS7 PARENB ISTRIP IXANY TAB3 HUPCL
       #login: #9600
9600      # B9600 HUPCL  IGNPAR PARENB ICRNL IXON OPOST ONLCR CS7 CREAD
                        ISIG ICANON ECHO ECHOK PARENB ISTRIP IXANY TAB3
          # B9600       SANE CS7 PARENB ISTRIP IXANY TAB3 HUPCL
       #login: #300

38400   # B38400 HUPCL IGNPAR PARENB ICRNL IXON OPOST ONLCR CS7 CREAD
```




© 2005 Hewlett-Packard Development Company, L.P.

```
                    ISIG ICANON ECHO ECHOK PARENB ISTRIP IXANY TAB3
# B38400            SANE CS7 PARENB ISTRIP IXANY TAB3 HUPCL
#login: #38400
```

Each **/etc/gettydefs** entry consists of five fields, separated by a hash (**#**) character.

| | |
|---|---|
| Entry name | This field looks line a baud rate setting, but it is really just the name of the entry. This is the entry to which **getty** refers. |
| Initial line settings | The settings for the serial port before **getty** hands over to **login**. |
| Final line settings | After **getty** has read your user name, it changes the line settings. |
| The login prompt | "**login**:" need not be your prompt. This field can be customized for your needs |
| Next entry to try | If **getty** receives a break, it will try this next entry from the **gettydefs** file. |

If you do not need this feature (you almost certainly will not) then choose a **gettydef** entry that has the "next entry to try" field set to the same as the "name" field, such as the 38400 setting above (or "**h**" for hardwired), rather than entries like the 9600 one above.

- The **login** process

  The login validates the login and password combination entered by checking the **/etc/passwd** file. If it is not validated, the user will not be logged in. If validated, the user is logged in, and the shell process issued.

  Typically, problems do not arise from the login process. This process is literally validating information entered and executing the command for a user specified in **/etc/passwd**.

  If the username and password combination entered at login does not match the **/etc/passwd** entry, that user is not able to gain access to the system. One check for **/etc/passwd** is to execute the **pwck** command. This command scans the **/etc/passwd** file and reports any inconsistencies. Checks include validation of the number of fields, login name, user ID, group ID, and whether the login directory and optional program name exist.

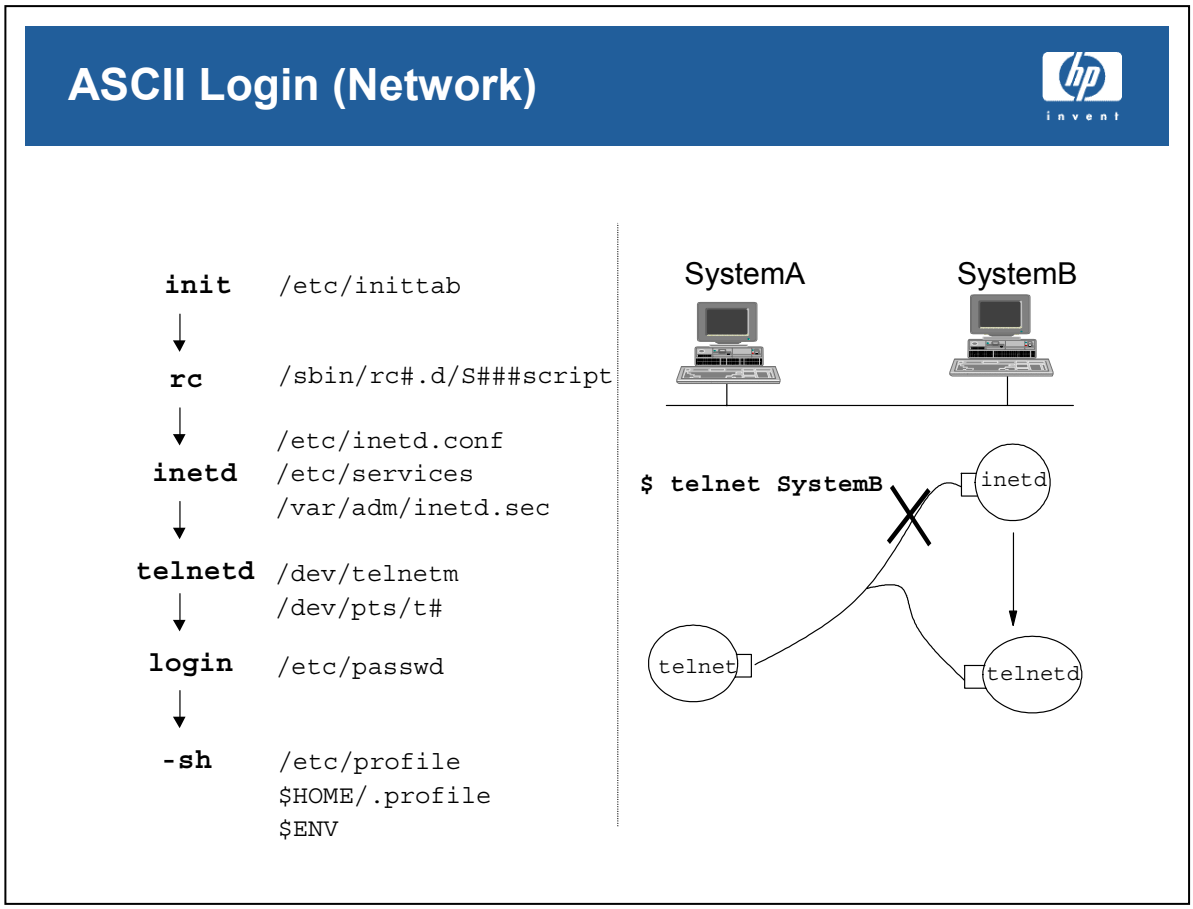
  For root, the access to logging in can be controlled on a terminal-by- terminal basis using the **/etc/securetty** file. (See **login(1)** for more information)

- The **sh** process:

  Shell processes allow users to perform work. When login invokes the shell, it is started with a hyphen prepended to its name (**-sh** for example). This tells the shell it is a login shell. Therefore, the startup scripts are executed. The startup scripts are **/etc/profile**, **$HOME/.profile** (optional) and **$ENV** (optional). If **$HOME/.profile** does not exist or the environmental variable **ENV** is unset, those specific startup scripts will not execute.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 7–4. SLIDE: ASCII Login (Network)




## Student Notes

### Establishing a `telnet` Session

- The **`init`** process

  The **`init`** process is one of the first processes to start after boot up. This process references **`/etc/inittab`**. Within this file, there is a record referencing **`/sbin/rc`**.

- The **`/sbin/rc`** script:

  This script executes start links located in **`/sbin/rc?.d`**. Start links play a major role in bringing the system to a multiuser state. The start link **`/sbin/init.d/inetd`** points to **`/sbin/rc2.d/S500inetd`**.




© 2005 Hewlett-Packard Development Company, L.P.

- The **/sbin/init.d/inetd** script:

  This script starts the **inetd** daemon. The **inetd** daemon references **/etc/inetd.conf** during startup. The **/etc/inetd.conf** file documents available network services. A **telnet** entry should exist if you want **telnet** server functionality. Further, /var/adm/inetd.sec controls access from other systems to yours and should be configured appropriately.

- The **/usr/lbin/telnetd** process

  **/usr/lbin/telnetd** is the telnet server process started by **inetd**. This process is started for each **telnet** client request. After each request, **telnetd** allocates a pseudo-terminal device.

  A telnet pseudo-terminal has a single character device file. On HP-UX, **telnetd** is usually implemented in kernel space. This makes handling **telnet** connections more efficient.

  Common problems experienced with **telnetd** are:

  – Inability to allocate a **telnet** device for use with login process.
    Either all **telnet** pseudo-terminals are in use, or **telnet** pseudo-terminals are not properly set up.

    If all pseudo-terminals are in use, increase the configurable **nstrtel** parameter.

    If **telnet** pseudo-terminal problems exist, recreate the device files. Examples are listed below:

  – Master device file

    ```
    insf -C pseudo
    or
    mknod /dev/telnetm c 72 0x000012
    ```

  – Slave device files

    insf -d tels –n <<the current value of the kernel parameter **nstrtel** >>

  – Fork: No more processes

    **telnetd** was unable to fork a process to handle the incoming connection. If this problem exists, wait for a time and try again. If this message persists, the server's host may have runaway processes that are using all the entries in the process table.

  – A mismatch between the kernel **telnet** patch level, and the patch level of the **telnetd** command.


**© 2005 Hewlett-Packard Development Company, L.P.**

- The **`login`** process:

  The responsibility of the login process is to validate the login and password combination entered by checking the **`/etc/passwd`** file. If not validated, the user will not be allowed to login. If validated, the user will be logged in and the shell process issued.

- The **`sh`** process:

  Shell processes allow users to perform work. When login invokes the shell, it is started with a hyphen prepended to its name (**`-sh`**, for example). This tells the shell it is a login shell, and the startup scripts are executed. The startup scripts are **`/etc/profile`**, **`$HOME/.profile`** (optional) and **`$ENV`** (optional). If **`$HOME/.profile`** does not exist, or the environmental variable **`ENV`** is unset, those specific startup scripts will not execute.




© 2005 Hewlett-Packard Development Company, L.P.

## 7–5. SLIDE: GUI Login (CDE Session)

### GUI Login (CDE Session)

## Student Notes

### Establishing a CDE Session

- The **`init`** process:

  The **`init`** process is one of the first processes to start after boot up. This process references **`/etc/inittab`**. Within this file, there is a line that calls the **`/sbin/rc`** script.

- The **`/sbin/rc`** script

  This script will execute start links located in the **`/sbin/rc?.d`** directories. Start links play a major role in bringing the system to a multiuser state. **`/sbin/rc3.d/S990dtlogin.rc`** is the start link that points to **`/sbin/init.d/dtlogin.rc`**.


© 2005 Hewlett-Packard Development Company, L.P.

- The **/sbin/init.d/dtlogin.rc** script

  This script will start processes needed to establish CDE environment.

  Several of the CDE processes use the **/tmp** directory as a work area. Since some of these processes are run using the user's ID, file permissions on **/tmp** should be set to 777 (**rwxrwxrwx**) or 1777 (**rwxrwxrwt**). If they are set differently, the creation of the CDE desktop session will fail. The user will be forced to establish an ASCII login session.

- The **dtgreet** process:
  to display the login screen, **dtlogin** executes **dtgreet**. There is one **dtgreet** process per display where a login screen is visible.

- The **dtwm** process:

  **dtwm** is a window manager. This window manager communicates with and facilitates access to other components in the environment, such as the Session Manager and the Style Manager.

  **dtwm** processes need to be able to resolve local hostnames into IP addresses. If this cannot be done, CDE sessions are not established. CDE exits the window manager and returns to the **dtgreet** screen. To fix this problem, simply use the Command Line or Fail Safe login and update the hostname database (**/etc/hosts**, **dns**, **nis**) with relevant hostname information.

- The **dtterm** process:

  **dtterm** is a terminal emulation program. Its primary use is within a CDE environment.

- The **sh** process:

  Shell processes allow users to perform work. When a shell is started, startup scripts are executed. The startup scripts are **/etc/profile**, **$HOME/.profile** (optional), and **$ENV** (optional).

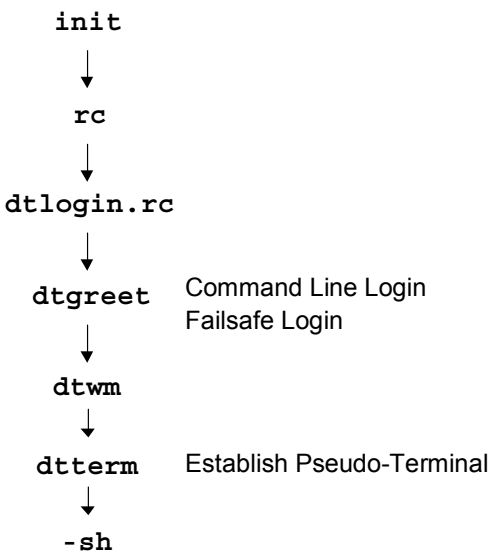
  If **$HOME/.profile** does not exist or the environmental variable **ENV** is unset, those specific startup scripts will not execute.

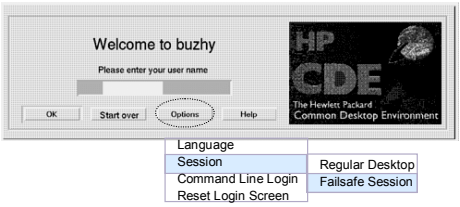



© 2005 Hewlett-Packard Development Company, L.P.

## 7–6. SLIDE: GUI Login (CDE Failsafe Login)

### GUI Login (CDE Failsafe Login)




## Student Notes

Other problems worth noting within CDE environments are listed below:

- The Command Line login option is grayed out.

  This problem occurs when a shell session is already established on the console. To fix this problem, the existing console shell session must be terminated. Below is an example of how to do this:

  – Select **`Options`**, **`Session`**, then **`Failsafe`** login.

  – **`kill -1 console-shell-PID`**

- You are unable to open a terminal emulation window.

  This problem occurs when there are missing or non-existent pseudo-terminal device files. To fix this problem, new pseudo-terminal device files must be created.

  **Example**

  – Select **`Fail Safe`** or **`Command Line`** login.

  – `insf -d pts -n 60 -e`

  – `insf -d pty -n 60 -e`


© 2005 Hewlett-Packard Development Company, L.P.

- Errors that occur during the initial CDE startup are logged in the **`/var/dt/Xerrors`** file.
- Messages that are sent during CDE sessions are logged in **`$HOME/.dt/startlog`**.
- Errors that occur during CDE sessions are logged in **`$HOME/.dt/errorlog`**.

Refer to these files when experiencing problems logging in via CDE.

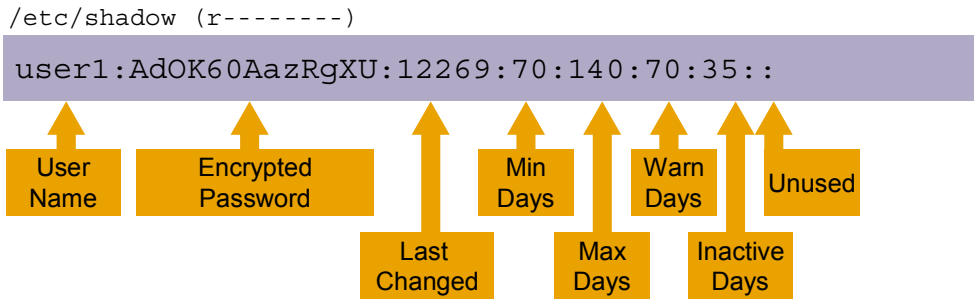



© 2005 Hewlett-Packard Development Company, L.P.

## 7–7. SLIDE: Protecting the System with `/etc/shadow`

### Protecting the System with `/etc/shadow`

- Passwords can optionally be stored in `/etc/shadow` rather than `/etc/passwd`
- `/etc/shadow` is more secure than `/etc/passwd`
- Install ShadowPassword from `http://software.hp.com`
- Use `/usr/sbin/pwck` to verify your current `/etc/passwd` file syntax
- Use `/usr/sbin/pwconv` to move passwords from `/etc/passwd` to `/etc/shadow`




## Student Notes

The default permissions on the **`/etc/passwd`** file are **`r--r--r--`**. Since the file is world-readable, anyone with a valid login can view the file and view encrypted passwords. Hackers sometimes exploit this fact to extract a list of encrypted passwords, run a password cracking utility, and potentially gain access to other users' accounts.

Unfortunately, removing world-read permission on the **`/etc/passwd`** file isn't a viable solution to this problem. Many commands, from **`login`**, to **`ps`**, to **`ll`** use the **`/etc/passwd`** file to convert UIDs to usernames, and vice versa. Changing the **`/etc/passwd`** file permissions to 400 would cause these commands to fail.

HP's shadow password product addresses this problem by moving encrypted passwords and other password information to the **`/etc/shadow`** file, which has 400 permissions to ensure that it is only readable by root. Other user account information (UIDs, GIDs, home directory paths, and startup shells) remain in the **`/etc/passwd`** file to ensure that **`login`**, **`ps`**, **`ll`**, and other commands can still convert UIDs to usernames.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Configuring Shadow Passwords

By default, the **/etc/shadow** file doesn't exist. Use the cookbook below to convert to a shadow password system:

1. Download and install the ShadowPassword product from **http://software.hp.com**. You can use the **swlist** command to determine if the product has already been installed on your system.

   ```
   # swlist ShadowPassword
   ```

2. Run **pwck** to verify that there aren't any syntax errors in your existing **/etc/passwd** file.

   ```
   # pwck
   ```

3. Use the **pwconv** command to move your passwords to the **/etc/shadow** file.

   ```
   # pwconv
   ```

4. Verify that the conversion succeeded. The **/etc/passwd** file should remain world-readable, but the **/etc/shadow** file should only be readable by root. The encrypted passwords in **/etc/passwd** should have been replaced by "**x**"s.

   ```
   # ll /etc/passwd /etc/shadow
    -r--r--r--   1 root   sys   914 May 18 14:35 /etc/passwd
    -r--------   1 root   sys   562 May 18 14:35 /etc/shadow
   ```

All of the standard password commands, including **passwd**, **useradd**, **usermod**, **userdel**, and **pwck** are shadow password aware.

You can revert to the traditional non-shadowed password functionality at any time via the **pwunconv** command.

```
# pwunconv
```

## Fields in /etc/shadow

The **/etc/shadow** file is an ASCII file consisting of any number of user entries separated by newlines. Each user entry line consists of the following fields separated by colons:

*login name* Each login name must match a login name in **/etc/passwd**.

*password* When you convert to a shadowed system, each password in **/etc/passwd** is replaced with an "**x**", and the encrypted passwords are copied to the second field in **/etc/shadow**. If the **/etc/shadow** password field is null, then there is no password and no password is demanded on login. Login can be prevented by entering a "*****" in the **/etc/shadow** password field.

*last changed* The number of days since January 1, 1970 that the password was last modified. This field is used by the password aging mechanism, which will be described later in the chapter.




© 2005 Hewlett-Packard Development Company, L.P.

*min weeks* The minimum number of weeks that a user must retain a password before it can be changed. This field is used by the password aging mechanism, which will be described later in the chapter.

*max weeks* The maximum number of weeks for which a password is valid. A user who attempts to login after his password has expired is forced to supply a new one. If min weeks and max weeks are both zero, the user is forced to change his password the next time he logs in. If min days is greater than max weeks, then the password cannot be changed. These restrictions do not apply to the superuser. This field is used by the password aging mechanism, which will be described later in the chapter.

*warn weeks* The number of weeks the user is warned before his password expires. This field is used by the password aging mechanism, which will be described later in the chapter.

*inactivity* The maximum number of days of inactivity allowed after a password has expired. The account is locked if the password is not changed within the specified number of days after the password expires. If this field is set to zero, then the user is required to change his password. This field is only used by HP-UX trusted systems, which aren't discussed in this course.

*expiration* The absolute number of days since Jan 1, 1970 after which the account is no longer valid. A value of zero in this field indicates that the account is locked. This field is only used by HP-UX trusted systems, which aren't discussed in this course.

*reserved* The reserved field is always null, and is reserved for future use.

## Editing `/etc/shadow`

Manually editing the **`/etc/shadow`** file isn't recommended. On a shadow password system, you should use the **`useradd`**, **`usermod`**, **`userdel`**, and **`passwd`** commands to manage user accounts in both **`/etc/passwd`** and **`/etc/shadow`**.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 7–8. SLIDE: Pluggable Authentication Modules

### Pluggable Authentication Modules

The traditional UNIX method to authenticate users is via the `/etc/passwd` file.

But, there are others:

- Trusted System mode adds the `/tcb/files/auth/?/…` files to manage passwords more securely.
- NIS and NIS+ have provided networked alternatives.
- PAM provides a framework to manage these and others, such as:
  - LDAP
  - DCE or Kerberos
  - NTLM



## Student Notes

PAM gives system administrators the flexibility of choosing any authentication service available on the system to perform authentication. The PAM framework also allows new authentication service modules to be plugged in and made available without modifying the applications.

For example, a system may use any user-authentication method, such as the **`/etc/passwd`** file, NIS, NIS+, or Trusted System. Programs requiring user authentication pass their requests to PAM, which determines the correct verification method and returns the appropriate response. The programs do not need to know what authentication method is being used.

Control is available on both a system-wide and an individual user basis.
The configuration files are:

| | |
|---|---|
| **`/etc/pam.conf`** | System-wide control file |
| **`/etc/pam_user.conf`** | Individual user control file |

The PAM services are provided by shared libraries installed or linked into the directory **`/usr/lib/security`**.


© 2005 Hewlett-Packard Development Company, L.P.

The main configuration file, **/etc/pam.conf**, is needed in order for logins to be allowed. If this file is missing, you will see errors like:

```
# telnet r206c41
Trying...
Connected to r206c41.
Escape character is '^]'.
Local flow control on
Telnet TERMINAL-SPEED option ON

HP-UX r206c41 B.11.11 U 9000/800 (ta)

System error
ERROR: Check /etc/pam.conf file permissions and ownership
       -r--r--r--   1 root       sys      /etc/pam.conf
Boot in single-user mode for missing /etc/pam.conf
Connection closed by foreign host.
#
```

The **/etc/pam_user.conf** file is optional, but can be used to configure services on a user by user basis.


© 2005 Hewlett-Packard Development Company, L.P.

## 7–9. SLIDE: Module Objectives

### Module Objectives

You should now be able to do the following:

- Describe the login process as it pertains to a `getty`, `telnet`, and the common desktop environment (CDE) session.
- Identify and fix login problems relating to a `getty`, `telnet`, and CDE session.
- Explain the benefits of password shadowing



## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 7–10. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What is the minimum run-level that allows users to log in?

2. What starts **`getty`** processes during system startup?

3. What starts **`telnetd`** servers upon **`telnet`** client requests?

4. What run-level does CDE start in?

5. List three files or directories on which CDE is dependent for proper operation.


**© 2005 Hewlett-Packard Development Company, L.P.**

# 7–11. LAB: Problems Logging On

## Part 1: Login Problems

This lab is designed to give you experience troubleshooting user login problems.

1. Use the Common Desktop Environment (CDE) for this lab. If using the RAIL environment, choose the ReflectionX option to access your system.

2. Execute the **`mod7.lab1.sh`** script:

   ```
   # cd /home/labs
   # ./mod7.lab1.sh
   ```

   After running this script, **`user1`** will have problems logging in. Using the knowledge gained in the current module, fix this problem such that **`user1`** can log into system. **`user1`**'s password is **`class1`**.




© 2005 Hewlett-Packard Development Company, L.P.

## Part 2: `getty` Problems

This lab is designed to give you experience troubleshooting **`getty`** process problems.

1. Use the Common Desktop Environment (CDE) for this lab. If using the RAIL environment, choose the ReflectionX option to access your system.

2. Execute the **`mod7.lab2.sh`** script:

   ```
   # cd /home/labs
   # ./mod7.lab2.sh
   ```

   After running this script, there will be a non-existent **`getty`** process to the console. Using the knowledge gained in the current module, fix this problem.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 3: Telnet Access Problems

This lab is designed to give you experience troubleshooting **`telnet`** client access problems.

1. Use the Common Desktop Environment (CDE) for this lab. If using the RAIL environment, choose the ReflectionX option to access your system.

2. Execute the **`mod7.lab3.sh`** script:

   ```
   # cd /home/labs
   # ./mod7.lab3.sh
   ```

   After running this script, **`telnet`** client requests will be denied. Using the knowledge gained in the current module, fix this problem such that **`telnet`** client requests are processed.




© 2005 Hewlett-Packard Development Company, L.P.

## Part 4: CDE Session Problems

This lab is designed to give you experience troubleshooting problems involving the establishment of CDE sessions.

1. Execute the **`mod7.lab4.sh`** script:

   ```
   # cd /home/labs
   # ./mod7.lab4.sh
   ```

   When the system reboots, no one will be able to establish a CDE session. Using the knowledge gained in the current module, fix this problem.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 5: Command-Line Login Option Not Available

This lab is designed to give you experience troubleshooting a grayed out Command-Line Login option from the **`dtgreet`** menus.

1. Execute the following commands:

```
# init 2

login: root
password: hp

# /sbin/init.d/dtlogin.rc start
```

   After issuing the above commands, the Command-Line Login option will be grayed out from the **`dtgreet`** menus. Using the knowledge gained in the current module, fix this problem.




© 2005 Hewlett-Packard Development Company, L.P.

## Part 6: No Pseudo-Terminals Available

This lab is designed to give you experience troubleshooting non-existent pseudo-terminals.

1. Execute the **`mod7.lab6.sh`** script:

   ```
   # cd /home/labs
   # ./mod7.lab6.sh
   ```

   After running this script, you will not be able to open another pseudo-terminal window. Using the knowledge gained in the current module, fix this problem.


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 8 — Troubleshooting File Systems

## Objectives

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

- State the purpose and describe the functionality of **`fsck`**.
- Determine what **`fsck`** is auditing and repairing during each pass of a file system (HFS/VxFS).
- Perform an **`fsck`** on a file system (HFS/VxFS).
- Perform a full **`fsck`** on a VxFS file system.
- Understand **`fsdb`** basics
- Describe the steps to defragment a VxFS file system




© 2005 Hewlett-Packard Development Company, L.P.

## 8–1. SLIDE: Module Objectives

### Module Objectives

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

- State the purpose and describe the functionality of `fsck`.
- Determine what `fsck` is auditing and repairing during each pass of a file system (HFS/VxFS).
- Perform an `fsck` on a file system (HFS/VxFS).
- Perform a full `fsck` on a VxFS file system.
- Understand `fsdb` basics
- Describe the steps to defragment a VxFS file system

## Student Notes

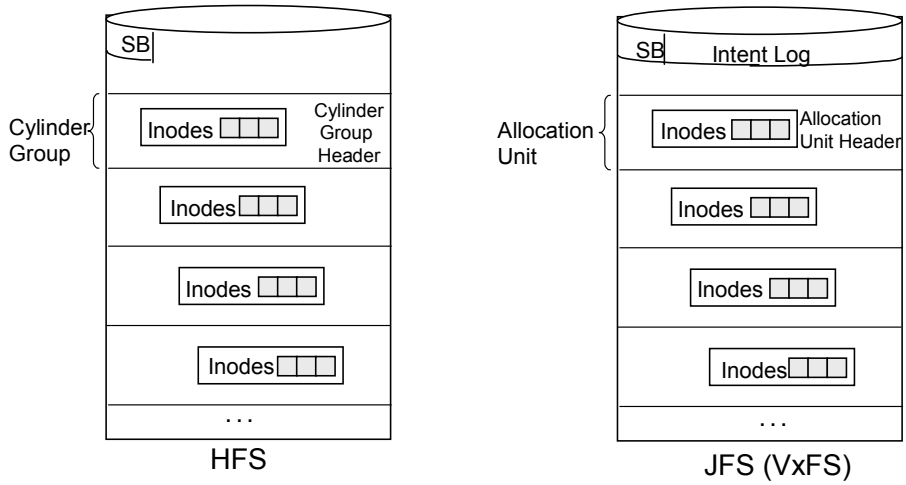

© 2005 Hewlett-Packard Development Company, L.P.

## 8–2. SLIDE: HP-UX File Systems

### HP-UX File Systems

Q: What is a file system?

A: A structure on a device that is used to manage files and directories.




## Student Notes

A file system is a structure on a device, such as a whole disk or logical volume, which is used to manage files and directories. Depending on the type of file system, specific structures are present.

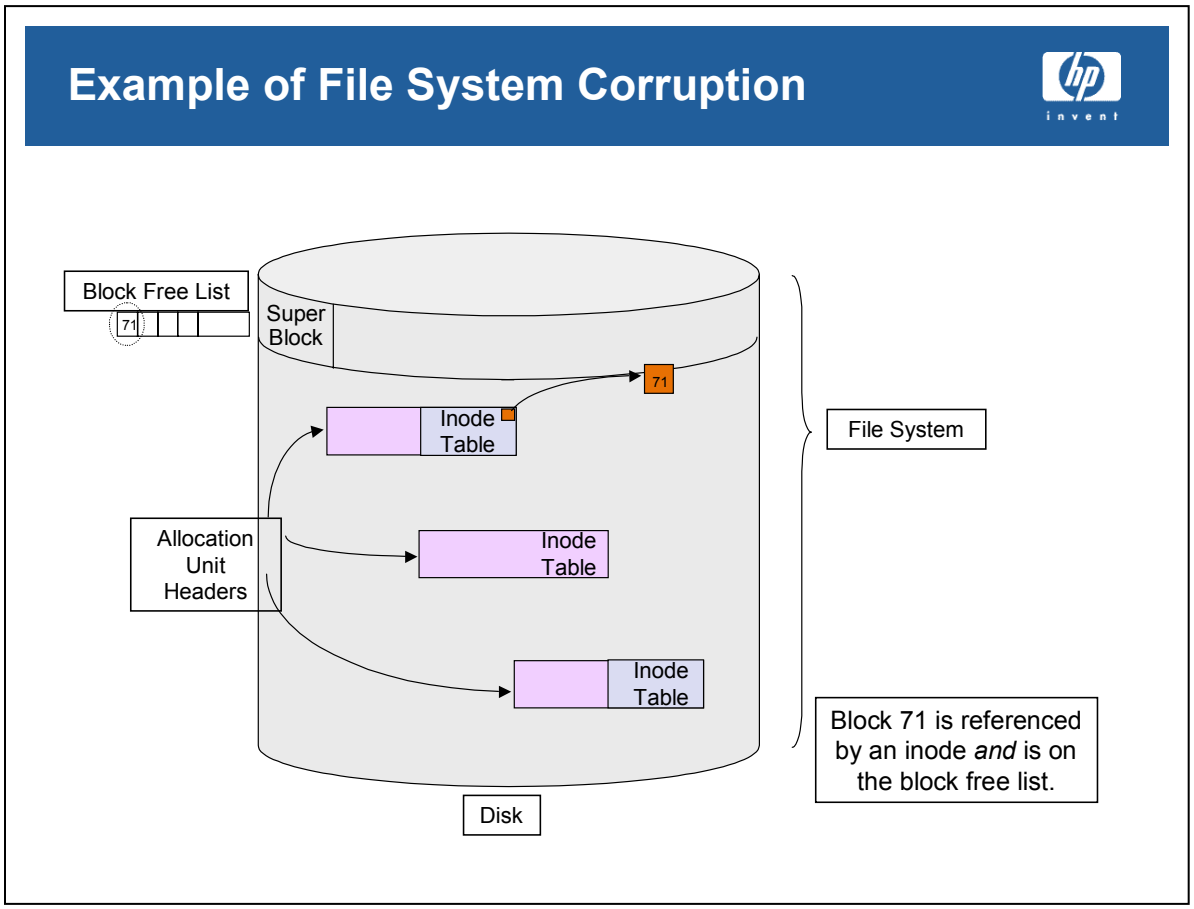
In the case of an HFS file system, major structures include the superblock and cylinder groups. Superblocks hold summary information about the file system. Cylinder groups allow file systems to be broken down into smaller, more manageable pieces. Each cylinder group makes up a portion of the entire file system.

In the case of a VxFS file system, major structures include the superblock, intent log, and allocation units. Superblocks hold summary information about the file system. Intent logs hold pending file system metadata updates. Allocation units allow file systems to be broken down into smaller pieces that are more manageable. Each allocation unit, except possibly the last one, is 32 MB and makes up a portion of the entire file system.




© 2005 Hewlett-Packard Development Company, L.P.

## 8–3. SLIDE: Example of File System Corruption




## Student Notes

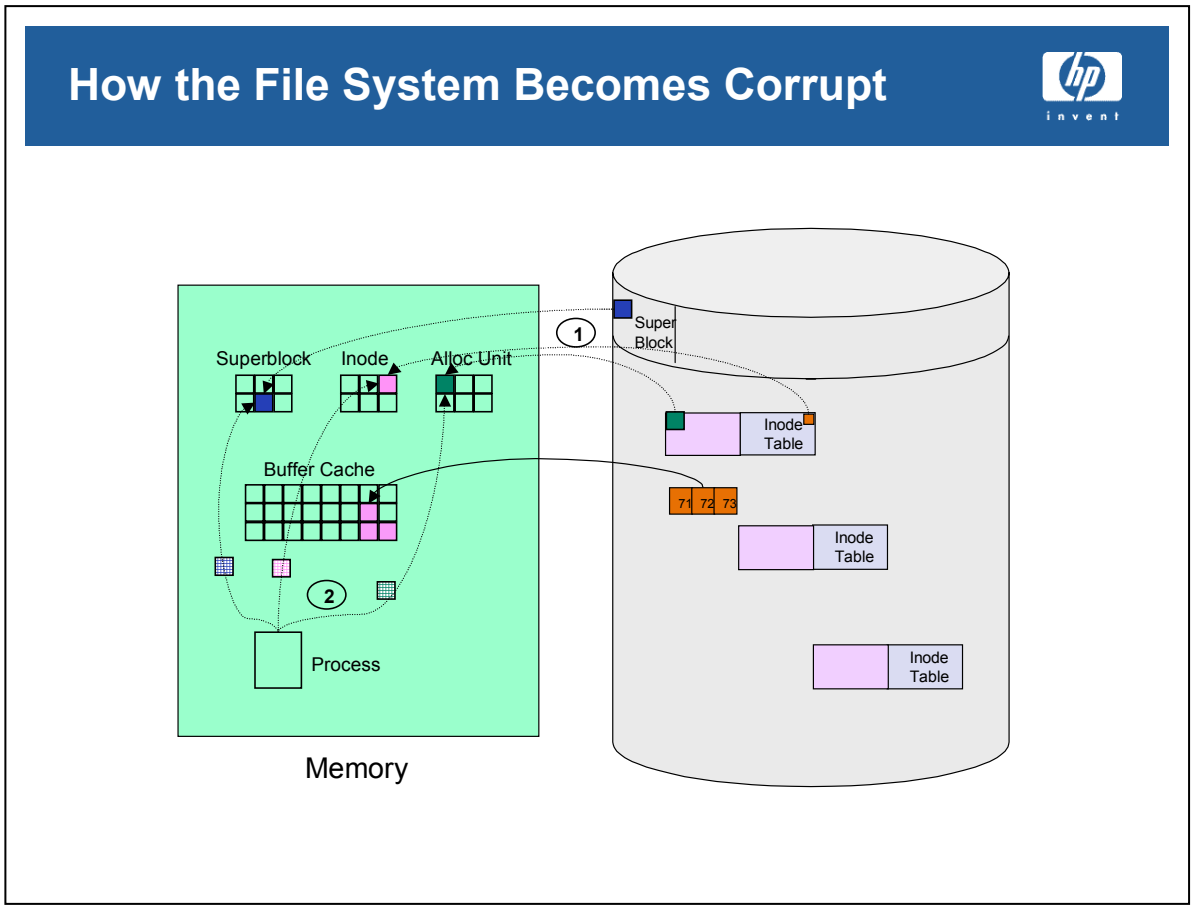
There are many instances of file system corruption. Over time, it is probable you will experience a few of them.

Each file system contains inode entries and a list of free data blocks available. An **inode** is a record that describes the contents of a file. It is broken up into an administrative area and data pointer area. The administrative area holds information regarding file types, permissions, link counts, owner, group, modification times, etc. The data pointer area contains pointers to data blocks that make up the contents of a file.

In the example above, an inode entry is pointing to data block #71. Data block #71 is also listed in the free list. In this case, file system corruption is present. It is incorrect to have an inode reference a data block that is also on the free map.


© 2005 Hewlett-Packard Development Company, L.P.

## 8–4. SLIDE: How the File System Becomes Corrupt




## Student Notes

The above slide illustrates the many ways a file system can become corrupt. The slide shows a process in memory reading in a file from the disk located at blocks 71 through 73.

When the new data is written to the end of a file, it causes the free maps to be updated. It also causes the file's inode and some other structures to be updated.

When these updated "header" structures are written back to the disk, they are updated at different times. There are windows of time when *some* of the memory header structures are updated to disk, while others are *not*. It is during these periods that the file system is vulnerable to corruption, should a system crash occur.

In addition to file system corruption being caused by improper system shutdowns, or system failures, corruption can also be caused by

- Hardware errors, such as failures of disk sectors.
- User errors, such as writing directly to disk device files.
- BUGs




© 2005 Hewlett-Packard Development Company, L.P.

When errors occur at run time, HFS typically PANICs the system. At the reboot, fsck will be run and the damage is normally repaired.

For JFS, rather than failing the whole system, it marks the file system as requiring a full **fsck** and stops using the area of the file system where the problem was detected. These conditions are written out as console errors and logged in **dmesg** and **syslog**.

```
# dmesg
.
.
.
msgcnt 2 vxfs: mesg 003: vx_mapbad - /dev/vg00/lvex14 file system
free extent bitmap in au 0 marked bad
#
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 8–5. SLIDE: `fsck`

### `fsck`

- `fsck` checks and repairs inconsistent conditions found within the file system structures.
- For HFS file systems, a full `fsck` is performed.
- For JFS file systems, an “intent log” replay is performed.

## Student Notes

For JFS, before the file system control structures (metadata) are updated on the disk, full details of the changes are logged to the intent log. If the operation involves multiple metadata updates, they are all written as a single transaction to the log.

In the event a Journaled file system was not shut down properly, **`fsck`** can just read the intent log. It will then know what operations were pending and exactly what to update.

For corruption to a JFS file system caused by factors other than improper shutdown, we cannot rely on the intent log.

On HFS, where there is no intent log, or when JFS cannot rely on its intent log, a full **`fsck`** must be performed.

**`fsck`** scans through all file system structures, comparing them. Where it finds inconsistencies, it attempts to repair them or suggests an action.

However, since **`fsck`** does not know what was happening, it does not know how to repair the structures. Instead, it relies on a series of rules or assumptions.



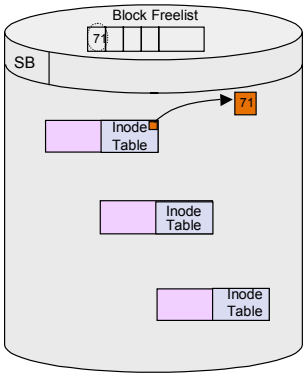

© 2005 Hewlett-Packard Development Company, L.P.

## 8–6. SLIDE: What Is `fsck` Checking?

### What Is `fsck` Checking?

Some common checks performed by `fsck` include:

- Unreferenced inodes
- Link counts in inodes too large
- Missing blocks in the freelist
- Blocks in freelist and also in files
- Incorrect counts in superblock




## Student Notes

### `fsck` Phases for HFS File Systems

- Checks blocks and sizes

  **Phase 1** builds a map of all allocated fragments within the file system and compares the fragments allocated to a file against the size information contained in the inode. If fragments are allocated to more than one file, it rescans the disk, looking for the first file that claims the fragments.

- Checks path names

  **Phase 2** checks that all directory entries are valid, and references valid inodes. If fragments from the first phase were reported as claimed by more than one inode, this phase will suggest removing the directory entries for all files that contain duplicate blocks.

- Checks connectivity

  **Phase 3** ensures directory structure is okay and all directories are linked into the file system.


© 2005 Hewlett-Packard Development Company, L.P.

- Checks reference counts

  **Phase 4** checks that all inodes have the correct number of links and that all inodes are actually referenced in the file system.

- Checks cylinder groups

  **Phase 5** checks that the cylinder group information tables are correct.

- Salvages Cylinder groups

  **Phase 6** is invoked to rebuild all the cylinder group information. All data held within the cylinder group information tables can be rebuilt from other data within the file system.

### `fsck` Phases for VxFS File Systems

- Checks structural files

  **Phase 0** checks the validity of structural files. Structural files include fileset headers, object location tables (OLT), inode list files, inode allocation summary files, attribute files, and attribute link counts.

- Checks inode sanity and blocks

  **Phase 1** checks for incorrect file size values and bad inode formats.

- Checks directory linkage

  **Phase 2** performs directory checks. This includes checking to see if any file is pointing to an unallocated inode, if any inode number is out of range, and if a subdirectory is not linked to a parent directory.

- Checks reference counts

  **Phase 3** ensures reference counts are correct.

- Checks resource maps

  **Phase 4** checks the accuracy of data block and inode bitmaps. These bitmaps track which data blocks and inodes are free or are in use.




© 2005 Hewlett-Packard Development Company, L.P.

## 8–7. SLIDE: `fsck` on HFS versus JFS

### `fsck` on HFS versus JFS

**`fsck` on HFS**

```
/usr/sbin/fsck [-F hfs] [-m] [special]
/usr/sbin/fsck [-f] [-F hfs] [-p|-P] [-V] [special...]
/usr/sbin/fsck [-b block#] [-f] [-F hfs] [-n|-N|-y|-Y] [-q] [-V] [special...]
```

- Always run `fsck` on unmounted file systems.
- You cannot unmount the root file system; therefore:
  - Use single-user mode.
  - Use sync on the disk first.
  - Use preening mode first, if it is an HFS file system.

**`fsck` on JFS**

```
/usr/sbin/fsck [-F vxfs] [-V] [-pmnNyY][-o full,nolog] [special . . .]
```

- Always run `fsck` on unmounted file systems.
- The `/stand` file system does not support JFS.
- Checking the raw device is almost always faster.

## Student Notes

The designs of HFS and JFS are very different. While JFS is a much newer file system with many advantages over HFS, it is optimized for different applications.

When **`fsck`** is being run against a file system, the file system should not be mounted. When a file system is mounted, it causes two problems for **`fsck`**.

First, if work is being done on the file system, **`fsck`** is looking at a constantly moving target. It was not designed to do this. The chances are that **`fsck`** will corrupt the file system rather than repair it, as it tries to make it look stationary.

Second, when a file system is mounted, many of its data structures are read into the kernels memory. Since **`fsck`** is only a "userland " [1] program, it has no way to work with the data in the kernel. It works only with the copy on the disk. If it repairs a disk bound structure that has been copied into memory, the working copy in memory will be out of date and inconsistent with the rest of the file system. Using it can result in data corruption. Therefore, you must perform a **`reboot -n`** after the **`fsck`** to prevent writing things back to the disk.

[1] Not a part of the UNIX kernel


© 2005 Hewlett-Packard Development Company, L.P.

There is one exception to this rule to **umount** file systems. You cannot **umount /**. You can boot off something else, such as the recovery shell, but you cannot **umount** it.

For JFS, there appears to be no way to **fsck** a mounted file system. If JFS has any reason to believe that it needs to be **fsck**'d, it sets a flag in its superblock to indicate that a full **fsck** is needed, which will then be performed at boot time.

For HFS, if you happen to have an HFS root file system, the **fsck** should be carried out while the system is as quiet as possible.

Start by going to single-user mode.

```
# shutdown -y now
.
.
.
Unmount file systems ..................................... OK
```

The **rc** shutdown script claims to be **umount**ing the file systems, but as we shall see, it does not **umount** all of them.

```
Transition to run-level 0 is complete.

Wait for transition to run-level S (allow 20 seconds).

INIT: New run level: S

INIT: SINGLE USER MODE

.
.
.
# mount
/ on /dev/vg00/lvol3 log on Fri Jul  6 10:50:05 2001
/var on /dev/vg00/lvol9 delaylog,nodatainlog on Fri Jul  6 10:50:16 2001
/usr on /dev/vg00/lvol8 delaylog,nodatainlog on Fri Jul  6 10:50:16 2001
#
# umount /var
umount: cannot unmount /var : Device busy
# ps -ef
     UID   PID  PPID  C    STIME TTY       TIME COMMAND
    root     0     0  0 10:50:04 ?         0:01 swapper
    root     8     0  0 10:50:04 ?         0:00 supsched
    root     9     0  0 10:50:04 ?         0:00 strmem
    root    10     0  0 10:50:04 ?         0:00 strweld
    root    11     0  0 10:50:04 ?         0:00 strfreebd
    root     2     0  0 10:50:04 ?         0:00 vhand
    root     3     0  0 10:50:04 ?         0:02 statdaemon
    root     4     0  0 10:50:04 ?         0:00 unhashdaemon
    root    12     0  0 10:50:04 ?         0:00 ttisr
    root    13     0  0 10:50:04 ?         0:00 ioconfigd
    root     1     0  0 10:50:05 ?         0:00 init
    root    19     0  0 10:50:05 ?         0:00 lvmkd
    root    20     0  0 10:50:05 ?         0:00 lvmkd
```




© 2005 Hewlett-Packard Development Company, L.P.

```
    root    21     0  0 10:50:05 ?         0:00 lvmkd
    root    22     0  0 10:50:05 ?         0:00 lvmkd
    root    23     0  0 10:50:05 ?         0:00 lvmkd
    root    24     0  0 10:50:05 ?         0:00 lvmkd
    root  4521  4492  1 14:32:33 console   0:00 ps -ef
    root    30     0  0 10:50:06 ?         0:02 vxfsd
    root   728     0  0 10:50:49 ?         0:00 nfskd
    root  4492     1  0 13:51:36 console   0:00 -sh
# fuser -c /var
/var:     4492o
```

As we can see, our shell has a file open in the **/var** file system; it is a temporary file:**/var/tmp/sh.<<PID>>.** There is no way to ask the shell to close this file. The only way to get rid of it is to get rid of the shell. Using **exec** to overlay our shell with the **umount** command, we can umount the file system. **init** will then see the process exit and give us another chance.

```
# exec umount /var

ENTER RUN LEVEL (0-6, s, or S): s

INIT: SINGLE USER MODE
.
.
.
# mount
/ on /dev/vg00/lvol3 log on Fri Jul  6 10:50:05 2001
/usr on /dev/vg00/lvol8 delaylog,nodatainlog on Fri Jul  6 10:50:16 2001
# umount /usr
#
# sync
sh: sync:  not found
```

**sync** is not in the root filesystem!

The last stage before **fsck**'ing the root file system is to close the shell history file, if you use one. For **root**, that is easy. Unset the **HISTFILE** variable, and use **exec** to restart the shell.

```
# unset HISTFILE
# exec sh
```

```
# fsck -p /dev/vg00/lvol3
.
.
.
```

Now execute **reboot -n**

```
# reboot -n
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 8–8. SLIDE: Introducing `fsdb`

### Introducing `fsdb`

`fsdb` – File System Debugger

CAUTION: If you don't know what you're doing, using this utility to do anything more than look around can be hazardous to the health of your system.

- `fsdb` allows viewing and manipulating inode information on files and directories.
- Can be useful in correcting file system corruption.

Determine the following first:

1. In which file system does the file in question reside?
   - `# bdf`
2. What is the file's inode number?
   - `# ls -i filename`

Once armed with the above, we can start investigating.

## Student Notes

When used by a knowledgeable administrator, the File System Debugger (**`fsdb`**) utility can reasonably quickly reign in file system corruption.

In this slide, we will demonstrate how you might look around and make some changes. HOWEVER, if you don't know what you are doing, this can quickly lead to file system corruption.

Before we can get any practical information out of **`fsdb`**, we need to first understand what we want. **`fsdb`** is generally used to view and manipulate file and directory inode information. Before we can look at inode information on a file, we need to know: 1) in which file system the file resides, and 2) the files' inode number.

Finding the file system is easy. Use **`bdf`** and locate the mount point and file system for your file (you may need to first **`find / -name <filename>`** if you don't know to which file system the file belongs).

Next, find the files' inode number: **`# ls -i <filename>`**

Once armed with this information, we can start **`fsdb`** and look around.




© 2005 Hewlett-Packard Development Company, L.P.

## 8–9. SLIDE: Using `fsdb`

### Using `fsdb`

`fsdb` Syntax

```
/usr/sbin/fsdb [-F FStype] /dev/vg01/rlvol2
```

```
# fsdb -F vxfs /dev/vg00/rlvol7
> 2562i
inode structure at 0x0001fa38.0200
type IFREG mode 102555  nlink 1  uid 0  gid 3  size 40960
atime 1073881066 900002  (Sun Jan 11 20:17:46 2004 PDT)
mtime 974188800 0  (Tue Nov 14 00:00:00 2000 PDT)
ctime 1073876666 700007  (Sun Jan 11 19:04:26 2004 PDT)
aflags 0 orgtype 1 eopflags 0 eopdata 0
fixextsize/fsindex 0  rdev/reserve/dotdot/matchino 0
blocks 40  gen 0  version 0 11  iattrino 0
de:  138368      0      0      0      0      0      0      0      0      0
des:     40      0      0      0      0      0      0      0      0      0
ie:       0      0
ies:      0
```

(the `/usr` file system)

## Student Notes

The **`fsdb`** syntax can be viewed in the **`fsdb`** man page. But for more file system specific information, look at **`fsdb_hfs(1m)`** and **`fsdb_vxfs(1m)`**.

Here is an example with notes:

Let's look at the **`diskinfo`** command.

```
# find / -name diskinfo
/etc/diskinfo
/usr/sbin/diskinfo

# ls -i /usr/sbin/diskinfo
    51 /usr/sbin/diskinfo
```

We now know that **`diskinfo`** resides in the **`/usr`** file system and its inode is 51.
Let's find out which file system is mounted at **`/usr`**.

```
# bdf
Filesystem          kbytes    used   avail %used Mounted on
/dev/vg00/lvol3     147456   77193   65912   54% /
```


© 2005 Hewlett-Packard Development Company, L.P.

```
/dev/vg00/lvol1     143573   28585  100630   22% /stand
/dev/vg00/lvol8    3072000 1361859 1604465   46% /var
/dev/vg00/lvol7    1007616  902239   98822   90% /usr
/dev/vg00/lvol6      49152    1203   45019    3% /tmp
/dev/vg00/lvol5     794624  711009   78416   90% /opt
/dev/vg00/lvol4      49152    2617   43677    6% /home
```

Now, we're ready to investigate:

```
# fsdb -F vxfs /dev/vg00/rlvol7
> 51i                    (51i instructs fsdb to dump inode #51)
inode structure at 0x000007c4.0300
type IFREG mode 100555  nlink 1  uid 2  gid 2  size 24576
atime 1073881060 350004  (Sun Jan 11 20:17:40 2004 PDT)
mtime 974188800 0  (Tue Nov 14 00:00:00 2000 PDT)
ctime 1073875933 730008  (Sun Jan 11 18:52:13 2004 PDT)
aflags 0 orgtype 1 eopflags 0 eopdata 0
fixextsize/fsindex 0  rdev/reserve/dotdot/matchino 0
blocks 24  gen 0  version 0 13  iattrino 0
de:  3053    0    0    0    0    0    0    0    0    0
des:   24    0    0    0    0    0    0    0    0    0
ie:     0    0
ies:    0
>
```

The above display shows the access time, the modification time, and the creation time for the diskinfo file.

Further down, note the last few lines:

```
de:  3053    0    0    0    0    0    0    0    0    0
des:   24    0    0    0    0    0    0    0    0    0
ie:     0    0
ies:    0
```

Since JFS file systems are made up of extents and blocks, we can see a data extent (de) address of 3053 – the location on the hard disc, and the data extent size (des) of 24 (1k blocks). If we were viewing a more fragmented file, we would see many data extents with an associated size for each.

Let's look at an HFS example:

```
# ls -i /stand/vmunix
  1238 /stand/vmunix

root@r206c44   [/usr]
# fsdb -F hfs /dev/vg00/rlvol1
file system size = 147456(frags)   isize/cyl group=48(Kbyte blocks)
primary block size=8192(bytes)
fragment size=1024
no. of cyl groups = 60
```

Note that when looking at an HFS file system, **fsdb** does not display a prompt.




© 2005 Hewlett-Packard Development Company, L.P.

**1238i** (1238*i instructs* **fsdb** *to dump inode #1238)*

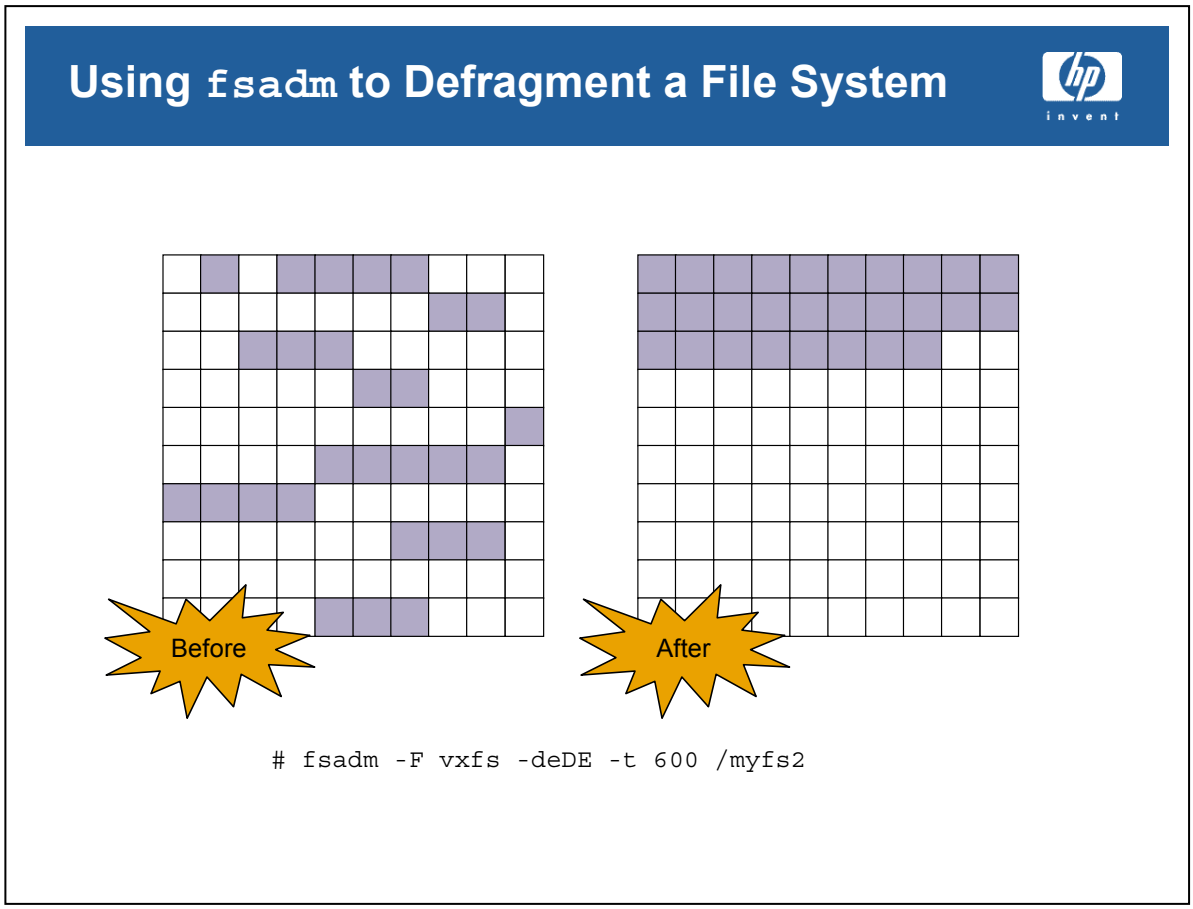
```
i#:1238  md: f---rwxr-xr-x ln:   1 uid:    0 gid:   3 sz: 27573440 ci:0
a0 : 15672  a1 : 15680  a2 : 15688  a3 : 15696  a4 : 15704  a5 : 15712
a6 : 15720  a7 : 15728  a8 : 15736  a9 : 15744  a10: 15752  a11: 15760
a12: 56704  a13:120856  a14:     0
at: Sun Jan 11 20:18:31 2004
mt: Sun Jan 11 20:02:16 2004
ct: Sun Jan 11 20:07:20 2004
```

When looking at the inode for **/stand/vmunix** (which resides in an HFS file system), the **fsdb** display is a little different – an HFS file system uses blocks (8kB default) and fragments (1k default). Above, we are seeing direct (8kB) block locations (a0 through a11). Block locations a12, a13, and a14 are pointers to other blocks than contain the rest of the file.

In the lab, we'll create a small (dispensible) file system and there, we'll have an opportunity to do more than just view.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 8–10. SLIDE: Using `fsadm` to Defragment a File System




## Student Notes

When a JFS file system is first created, the file system free space is consolidated in a small number of very large extents. However, as files and directories are created, extended, reduced, and removed, the file system gradually becomes "fragmented" into many smaller extents. In a fragmented file system, it is very difficult for JFS to allocate contiguous blocks when existing files are extended, and when large new files are created. Over time, this situation can seriously degrade system performance.

In order to ensure optimal file system performance, HP recommends "defragmenting" JFS file systems on a regular basis using the **`fsadm`** utility. **`fsadm`** attempts to reorganize and consolidate file extents, free extents, and directories to ensure the best possible file system performance.

### Defragmenting File Systems with `fsadm`

**`fsadm`** may be executed several different ways:

```
# fsadm -F vxfs -e /myfs2    (re-organize and consolidate extents)
# fsadm -F vxfs -d /myfs2    (re-organize and optimize directories)
# fsadm -F vxfs -de /myfs2   (re-organize and optimize directories and extents)
```




© 2005 Hewlett-Packard Development Company, L.P.

Note that the file system must be mounted while **fsadm** is running! Users may even be accessing the file system while **fsadm** runs, but performance will be somewhat degraded.

The amount of time required for the defragmentation process to complete varies, depending on the degree of fragmentation, disk speed, and the number of inodes in the file system. If your production schedule allows only a brief period of time for system maintenance activities, you can define a fixed period of time for **fsadm** to execute. **fsadm** then processes as many extents and directories as possible within the given interval, then exits gracefully. The example below defragments the file system for 600 seconds, then terminates:

```
# fsadm -F vxfs -de -t 600 /myfs2
```

To simplify administration and minimize the impact on users, many administrators choose to run **fsadm** automatically as a batch job after hours on a nightly or weekly basis.

## Reporting Fragmentation with `fsadm`

**fsadm** is able to report, as well as fix, file system fragmentation when executed with two additional options: **-D** reports directory fragmentation, and **-E** reports extent fragmentation. These options may be used independently, or concurrently:

```
# fsadm -F vxfs -E /myfs2      (report extent fragmentation)
# fsadm -F vxfs -D /myfs2      (report directory fragmentation)
# fsadm -F vxfs -DE /myfs2     (report directory and extent fragmentation)
```

Many administrators use **-DE** in conjunction with **-de**:

```
# fsadm -F vxfs -DEde /myfs2
```

This generates a pre-defrag fragmentation report, defragments the file system, then generates a post-defrag report, too. By comparing the pre- and post-defragmentation report values after a day of system activity, a week of system activity, and a month of system activity, you can determine how frequently **fsadm** needs to be executed. If you run **fsadm -DEde** after one week of system activity and see little difference between the pre- and post-values, you may be able to run the utility monthly rather than weekly.

Here's a sample fragmentation report:

```
# fsadm -F vxfs -DE /myfs2
Directory Fragmentation Report
       Dirs        Total      Immed    Immeds   Dirs to   Blocks to
       Searched    Blocks     Dirs     to Add   Reduce    Reduce
       345         812        210      3        6         13

Extent Fragmentation Report
       Total    Average      Average     Total
       Files    File Blks    # Extents   Free Blks
       216      30           1           3973

    blocks used for indirects: 12
    % Free blocks in extents smaller than 64 blks: 29.12
    % Free blocks in extents smaller than  8 blks: 1.33
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
      % blks allocated to extents 64 blks or larger: 30.05

   Free Extents By Size
      1:          3            2:          3            4:         11
      8:         24           16:         33           32:         12
     64:          8          128:          6          256:          2
    512:          0         1024:          1         2048:          0
   4096:          0         8192:          0        16384:          0
```

Here are a few of the key fields you should look at:

Immeds to Add: (Smaller is better.)
: Most directories have an inode that records the permissions on the directory and one or more data blocks that record the list of files that are included in the directory. If a directory contains very few files, JFS can actually store the file names in the directory's inode rather than a separate data block. The "Immeds to Add" field reports the number of directories that could be optimized in this manner if **`fsadm`** were executed.

Dirs to Reduce: (Smaller is better.)
: Directory defragmentation attempts to reclaim wasted space in directories. The "Dirs to Reduce" field indicates how many directories might be reduced by directory defragmentation.

Blocks to Reduce: (Smaller is better.)
: Indicates the number of blocks that may be reclaimed as a result of directory defragmentation.

Average # Extents: (Smaller is better, "1" is ideal.)
: Describes the average number of extents occupied by each file.

% Free blocks in extents smaller than 64 blks: (Smaller is better and should be less than 50%.)
: JFS prefers to have a few large extents of free blocks rather than many smaller extents of free blocks. Larger free-block extents make it easier for JFS to allocate space for new files.

% Free blocks in extents smaller than 8 blks: (Smaller is better.)
: JFS prefers to have a few large extents of free blocks rather than many smaller extents of free blocks. Larger free-block extents make it easier for JFS to allocate space for new files.

For a more detailed description of JFS fragmentation issues, and the **`fsadm`** command syntax, see the **`fsadm_vxfs(1m)`** man page.




© 2005 Hewlett-Packard Development Company, L.P.

## 8–11. SLIDE: Module Objectives

### Module Objectives

You should now be able to do the following:

- State the purpose and describe the functionality of `fsck`.
- Determine what `fsck` is auditing and repairing during each pass of a file system (HFS/VxFS).
- Perform an `fsck` on a file system (HFS/VxFS).
- Perform a full `fsck` on a VxFS file system.
- Understand `fsdb` basics
- Describe the steps to defragment a VxFS file system

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 8–12. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What is a file system?

2. What are the major components of an HFS and VxFS file system?

3. What major feature does a VxFS file system have that is not present with an HFS file system?

4. What is the purpose of the **`fsck`** command?

5. On what file systems can **`fsck`** be run?

6. What are the **`fsck`** phases for an HFS and VxFS file system?

7. Is it possible to run a full **`fsck`** on a VxFS file system? If so, how?




© 2005 Hewlett-Packard Development Company, L.P.

# 8–13. LAB: Recovering a Corrupt VxFS File System

## Directions

In this lab, your answers may differ from the answers in the book. Use your best judgment to determine the appropriate course of action.

1. Create a 20MB Logical Volume named **`test`**, build a JFS file system in it, create a mount point called **`/mod8`** and mount the file system to your new mount point.

   ```
   # lvcreate -L 20 –n test  /dev/vg01
   # newfs –F vxfs /dev/vg01/rtest
   # mkdir /mod8
   # mount /dev/vg01/test /mod8
   ```

2. Copy some scratch files into this new file system with the following command:

   ```
   # cd /sbin
   # cp vg* /mod8
   ```

3. Now, **`umount`** the file system.

   ```
   # umount /mod8
   ```

4. Extend this logical volume to the new size of 36 MB.

   ```
   # lvextend -L 36 /dev/vg01/test
   ```

   You should see something like the following:

   ```
   # lvextend -L 36 /dev/vg01/test
   Logical volume "/dev/vg01/lvol2" has been successfully extended.
   Volume Group configuration for /dev/vg01 has been saved in \
   /etc/lvmconf/vg01.conf
   ```

   Special Note: Be sure to read the entire question before you try to execute it.

   Next, we use the **`extendfs`** command. **Immediately** after you hit **Return** to begin execution of the **`extendfs`** command, hit **Ctrl**+**c** interrupt to abort the execution.

   ```
   # extendfs -F vxfs /dev/vg01/rtest
   <ctrl> C
   ```

5. What message(s) do you see on your screen?


**© 2005 Hewlett-Packard Development Company, L.P.**

6. Now, run a normal **fsck** on the VxFS file system.

   ```
   # fsck -F vxfs /dev/vg01/rtest
   ```

   What does the screen output say?

7. Yet the previous error message (from question 6 above) says that there is a problem with the structural integrity of the file system within **/dev/vg01/test**.

   So, we will run a full **fsck**.

   ```
   # fsck -F vxfs -o full /dev/vg01/rtest
   ```

   What does the screen output now look like?

8. Now, we again mount the file system.

   ```
   # mount /dev/vg01/test /mod8
   ```

   Are all the **vg** files still present?

9. Remove the **/mod8** file system and mount point.




© 2005 Hewlett-Packard Development Company, L.P.

# 8–14. LAB: Using `fsdb`

## Directions

In this lab, your answers may differ from the answers in the book. Use your best judgment to determine the appropriate course of action.

In this lab, we explore the **fsdb** tool. We will be changing data pointers within inodes, but we'll also be working with an expendable file system.

1. We first set up an HFS file system for this lab:

   a. Create a logical volume named **`disk5`** in volume group **`vg01`**:

   ```
   # lvcreate -L 100 -n disk5 /dev/vg01
   ```

   b. Next, build a file system within the logical volume:

   ```
   # newfs -F hfs /dev/vg01/rdisk5
   ```

   c. Create a mount point for the new file system:

   ```
   # mkdir /disk5
   ```

   d. Last, mount the file system:

   ```
   # mount /dev/vg01/disk5 /disk5
   ```

2. Copy some files into **`/disk5`** and then use **`fsdb`** to cause corruption. Use the following commands:

   a. First, copy some files into the **`disk5`** file system:

   ```
   # cd /disk5
   # cp /usr/sbin/ioscan .
   # cp /usr/sbin/lanscan .
   # cp /usr/sbin/diskinfo .
   ```

   b. Now, check the inode numbers for the files — you will need them when you run **`fsdb.`**

   ```
   # ls -i
   ```

   You should see something like the following:

   ```
   6  diskinfo      4  ioscan     5  lanscan   3 lost+found
   ```

   c. Next, **`umount /disk5`**.

   ```
   # cd /
   # umount /disk5
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

or

```
# cd /
# umount /dev/vg01/disk5
```

d. Run the **fsdb** tool on this file system.

```
# fsdb -F hfs /dev/vg01/rdisk5
```

You should see something like the following:

```
File system size = 1024000(frags) isize/cyl group=48(Kbyte blocks)
primary block size=8192(bytes)
fragment size=1024
no. of cyl groups = 42
```

e. Notice that **fsdb** has no prompt. (You are supposed to know what you are doing.☺ ) So, while in **fsdb**, look inside inode 5:

**5i** This command will show inode information for **lanscan**.

You should see something like the following:

```
i#:5  md:  fu--r-xr-xr-x  ln:  1  uid:  0  gid:  3  sz: 40960  ci:  0
a0:    128   a1:   136   a2:    240   a3:    248   a4:  256   a5:   0
a6:      0   a7:     0   a8:      0   a9:      0   a10:   0
a11:     0  a12:     0  a13:      0  a14:      0
at:  Thu  Aug 30  13:16:31 2001
mt:  Thu  Aug 30  13:16:31 2001
ct:  Thu  Aug 30  13:16:31 2001
```

f. Now, let us look inside of inode 6.

**6i** This will show inode information for **diskinfo.**

```
i#:6  md:  f---r-xr-xr-x  ln:  1  uid:  0  gid:  3  sz: 24576  ci:  0
a0:    264   a1:   272   a2:    280   a3:    0   a4:    0   a5:   0
a6:      0   a7:     0   a8:      0   a9:    0  a10:    0  a11:   0
a12:     0   a13:    0  a14:      0
at:  Thu  Aug 30  13:16:31 2001
mt:  Thu  Aug 30  13:16:31 2001
ct:  Thu  Aug 30  13:16:31 2001
```

g. Next, we deliberately cross a few pointers by pointing inode 6 pointers to inode 5 data blocks.

```
a0=128   ( where 128 is the value of a0 from inode #5 )
a1=136   ( where 136 is the value of a1 from inode #5 )
a2=240   ( where 240 is the value of a2 from inode #5 )
a3=248   ( where 248 is the value of a3 from inode #5 )
a4=256   ( where 256 is the value of a4 from inode #5 )
```




© 2005 Hewlett-Packard Development Company, L.P.

h. Now, look at inode 6 again to see the changes.

**6i** This will show inode information for **diskinfo.**

i. Last, we exit **fsdb**.

```
< ctrl-d > to exit fsdb
```

3. Now, with the **/disk5** file system still unmounted, run **fsck** on **/disk5** file system. Answer *n* (no) to all questions.

```
# fsck -F hfs /dev/vg01/rdisk5
```

You might see something like the following:

```
** /dev/vg01/rdisk5
** Last Mounted on /disk5
** Phase 1 - Check Blocks and Sizes
128 DUP I=6
129 DUP I=6
130 DUP I=6
131 DUP I=6
132 DUP I=6
133 DUP I=6
...
...
...
247 DUP I=5
248 DUP I=5
249 DUP I=5
250 DUP I=5
251 DUP I=5
** Phase 2 - Check Pathnames
DUP/BAD  I=5
OWNER=root MODE=104555
SIZE=40960 MTIME=Aug 26 13:16 2001
FILE=/lanscan

REMOVE? n

DUP/BAD  I=6
OWNER=root MODE=100555
SIZE=24576 MTIME=Aug 26 13:16 2001
FILE=/diskinfo

REMOVE? n

** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
28 BLK(S) MISSING
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
BAD CYLINDER GROUPS
FIX? n

5 files, 0 icont, 177 used, 99464 free (8 frags, 12432 blocks)
```

4. What type of problems do you encounter?

5. Which files were reported to have problems?

6. Mount the **/disk5** file system and execute the files found to have duplicate blocks. Which command does not work properly?

```
# mount /dev/vg01/disk5 /disk5
```

The **/disk5/lanscan** file and the **/disk5/diskinfo** file should now be executed.

```
# cd /disk5
# ./lanscan
# ./diskinfo
```

**Answer:**

```
# cd /disk5

# ll
total 408
-r-xr-xr-x   1 root       sys          28672 Mar 26 13:16 diskinfo
-r-xr-xr-x   1 root       sys         126976 Mar 26 13:16 ioscan
-r-sr-xr-x   1 root       sys          32768 Mar 26 13:16 lanscan
drwxr-xr-x   2 root       root          8192 Mar 26 13:15 lost+found

# ./lanscan
Hardware Station         Crd Hardware Net-Interface   NM  MAC     ...
Path     Address         In# State    NameUnit State  ID  Type    ...
8/16/6   0x0060B0179863  0   UP       lan0     UP     4   ETHER   ...
```

Note: To save space in this book, the above screen output was truncated and is incomplete. But, as you can see, the **lanscan** command does indeed execute.

```
# ./diskinfo
sh: ./diskinfo: Execute permission denied.
```

Conclusion: The **diskinfo** file does not work properly. The **diskinfo** command fails to execute and generates a **permission denied** message, even though the permission bits are set appropriately. This is due to the duplicate blocks with **lanscan**. This is an example of the problems that can result from a corrupted file system, problems that can be detected and corrected by **fsck**.




© 2005 Hewlett-Packard Development Company, L.P.

7. Umount the **disk5** file system and run **fsck** again. Remove only the **diskinfo** file, for example, answer *y* (yes) to any question that refers to **diskinfo** (**FILE=diskinfo**), or the inode for **diskinfo** (probably inode 6).

   Also, answer **y** (yes) to any question concerning the superblock or a cylinder group.

```
# cd /
# umount /disk5
# fsck -F hfs /dev/vg01/rdisk5
```

8. Was the **diskinfo** file placed in the **lost+found** directory?

```
# mount /dev/vg01/disk5 /disk5
# ls /disk5/lost+found
```

In summary, while we didn't use **fsdb** to correct problems (which material will be covered in a more advanced course), we did get a chance to see how it could be used to change pointers in an inode.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 8–15. LAB: Using fsadm

### Directions

1. Run a directory and extent fragmentation report on the **`/var`** file system. At this point, don't do anything to repair the fragmentation. Record the following values:

   a. Directories to reduce:
   b. Average number of extents per file:

2. Defragment the **`/var`** file system.

   How successful was the defragmentation utility? Run a second fragmentation report and compare the results to the values you recorded above in question 1. Do you see any improvement?




© 2005 Hewlett-Packard Development Company, L.P.

**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 9 — Troubleshooting Process Problems

## Objectives

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

- Define a process.
- Display a user's process hierarchy.
- Use the `kill` command.
- Identify and clear zombie processes.
- Troubleshoot CPU-intensive processes.




© 2005 Hewlett-Packard Development Company, L.P.

## 9–1. SLIDE: Module Objectives

### Module Objectives

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

- Define a process.
- Display a user's process hierarchy.
- Use the `kill` command.
- Identify and clear zombie processes.
- Troubleshoot CPU-intensive processes.

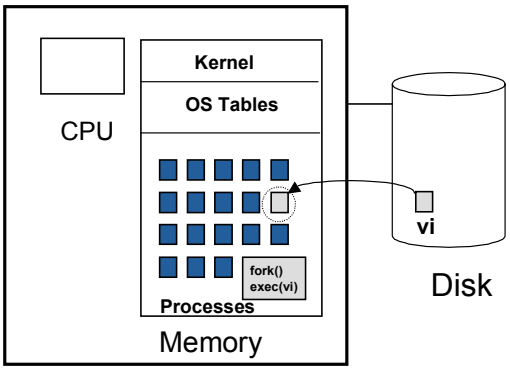
## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 9–2. SLIDE: What Is a Process?

### What Is a Process?

- A non-terminated instance of a program
- Created by the `fork()` and `exec()` system calls
- The configurable parameter `nproc` determines the maximum number of current processes.




## Student Notes

A process is a non-terminated instance of a program. Every command on a system is executed in a process. Since HP-UX 11.00, it is possible for a program to multitask within a single process. This is achieved with threads. A process now acts as a container for a series of threads. All processes contain at least one thread, but can potentially contain many. When the system is running, most problems encountered are related either to the file system (for example, running out of disk space) or to processes.

### HP-UX (UNIX) Kernel

UNIX executes as a series of threads, all of which may compete to share the system's resources, such as the CPU, memory, the file system, and I/O devices. At the heart of UNIX is the kernel. The kernel acts as the manager of the available resources. It controls sharing of and access to these resources.

The threads of a process share a common address space where they store all of their instructions and any data they are currently using. The kernel controls access to everything else, such as files, devices, shared resources, and other processes. A thread uses system calls to ask the kernel to perform actions on these on its behalf.




© 2005 Hewlett-Packard Development Company, L.P.

Most of the activity on a system occurs because of a system call. Understanding these calls helps to understand UNIX.

## Parent/Child Processes

A process is created when some other process in the system invokes a system call, known as a **`fork`**. The new process is the **`child`**; the process that called the **`fork`** is the **`parent`**. When **`fork`** is called successfully, it returns a process identifier (PID).

Parent and child processes have significant differences:

- They have different process IDs.
  Every process on the system has a unique ID, which is how the kernel recognizes it. (To communicate with a process, you need to know its PID.) The parent process is told the ID of its new child because of the **`fork`** system call.

- The child does not inherit any file locks from the parent.

- The child will have only one kernel thread.
  The first thread the process has will be a copy of the one that made the **`fork`** system call. To learn the PID of the parent, the child can use the **`getppid`** system call. To find its own PID, the process can use **`getpid`**.

Parent and child processes have significant similarities:

- Though parent and child processes have different process IDs, file locks and threads are similar between them

- Some resources are copied from parent to child.
  The data that the process was using is copied from the parent to the child. The only difference in the data held by the two processes is that returned by **`fork`**; otherwise, it is an exact copy.

- Some resources are actually shared.
  The files the parent has open are shared between the parent and child, not just access to the files, but actual pointers into the files. If the parent moves the file pointer to the beginning of the file and the child writes to the file, then the child will write at the beginning. Access to any communication tools, such as shared memory, messages, and so on, is also shared with the new process.

## Process Relationships

The kernel manages processes using a series of **`proc`** structures. Before HP-UX 11.11, these were held in a table (and array) having **`nproc`** slots. From HP-UX 11.0 forward, the approach of using a table has been abandoned, because the table wasted space if all the **`nproc`** slots were not occupied. In order to reduce this waste, the value of **`nproc`** needed to be carefully controlled when generating the kernel. For HP-UX 11.11, the system does not allocate the space for **`nproc`** processes unless they are needed. This allows the system administrator to choose a larger value for **`nproc`** without risking wasted space. This reduces the number of times the system needs to be rebooted to resize the process table.


**© 2005 Hewlett-Packard Development Company, L.P.**

A process can change the program it is running with an **`exec`** system call. When **`exec`** is called, the current set of instructions and all current data is discarded. A new set of instructions is loaded from a file on the system. If the new program needs data from the original program, it can be copied, using command-line arguments or environment variables.

On **`exec`**, the new program inherits the following from the original program:

- Process ID
- Slot in the process table
- Most of the open files

  As with **`fork`**, actual file pointers are retained. (An option can be set up for a file to prevent the new program from inheriting it.)
- Access to some communication tools, such as sockets and pipes
- Environment variables

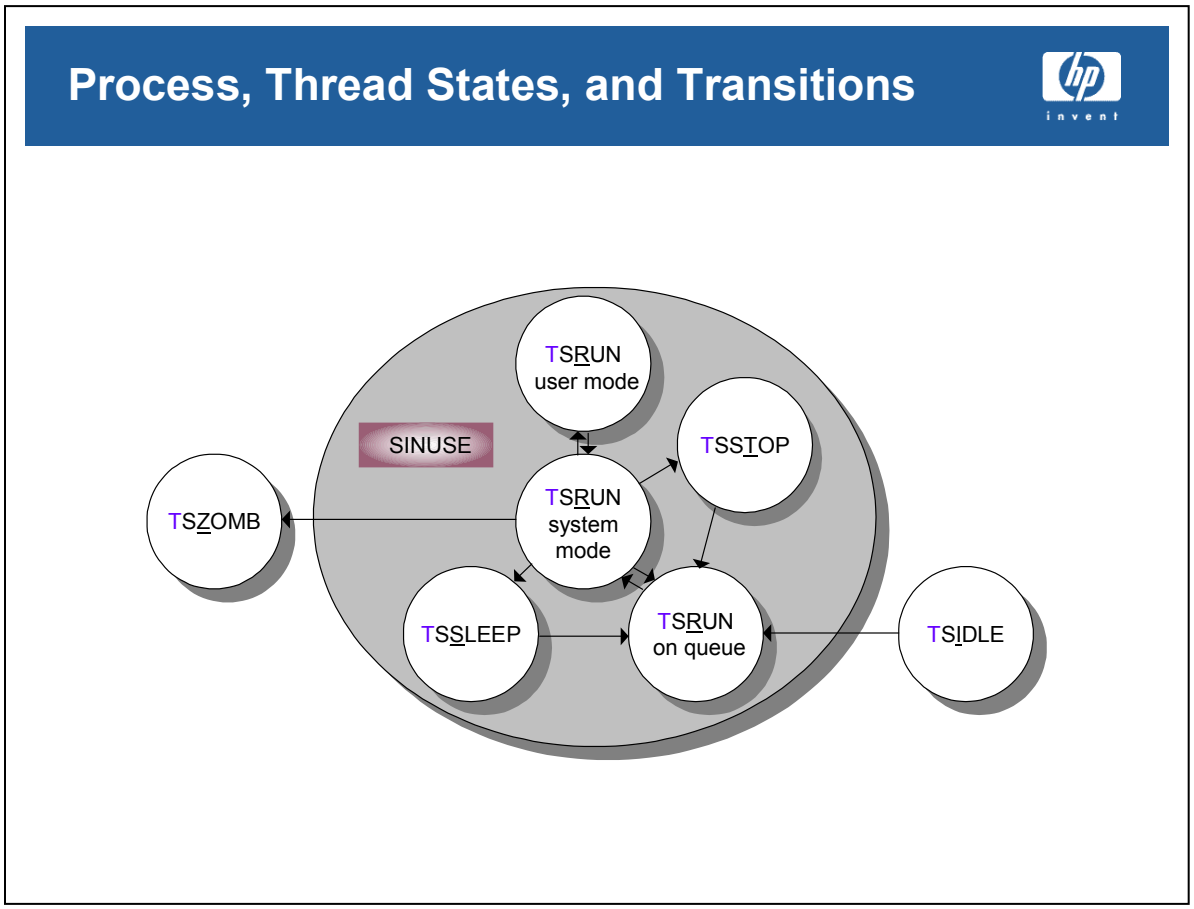
  Environment variables normally flow down from parent to child and through all execs. However, when using an exec system call, a program can choose to send a new set of environment variables. Typically, they are set up by a user's login shell. They describe the environment in which the user is working, such as terminal type, time zone, language, and preferred shell and editor.
- Command-line arguments

  Command-line arguments are the primary means of communication between the old program and the new one. Typically, they are used to tell the new program what to work on and any options to use.




© 2005 Hewlett-Packard Development Company, L.P.

## 9–3. SLIDE: Processes, Thread States, and Transitions




## Student Notes

A process comes into existence as the result of a **`fork`** system call. A fork is invoked by another process on the system. From its inception, the process spends some time executing user and system instructions, and more time waiting. Processes spend time queuing for a turn on the CPU. At other times, they wait for other resources or for data that is not yet available, such as input from the keyboard.

Most systems have only one processor. But no matter what the number of processors, there are always more processes than there are processors to run them. Processes must share CPU resources. Moreover, because every process shares the CPU, it is critical that a process does not tie up the CPU when waiting for resources or data. Instead, UNIX puts processes to *sleep* when they are waiting.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Process States

The following are the states in which a thread can exist (the names in parentheses are those used in the **`kthread`** table):

Idle (TSIDL)

A process enters this state briefly during its creation. It is a transitory state. At this point, the process has an entry in the process table and an ID, and its parent is being copied to it.

Running (TSRUN)

A process has two running states. First, it can run in user mode. In this mode, it is executing instructions from its own program, either from the executable file or from a shared library. Second, it can run in system mode. When a process needs to perform a task inside a system call that is a kernel function, it temporarily becomes the kernel. There is no HP-UX executing on the system as such.

If you have ever executed the **`time`** command to find out how long something takes to execute, you have seen that three different times are reported. One figure is the elapsed or real time between when the command starts and when it completes. The other two figures are the amount of time spent in the running state in user mode and in system mode.

As the process starts the transition from running in kernel mode to running in user mode, it checks to see if there are any signals waiting to be processed. This is where signals are handled. If the result of the signal is to *die*, then the exit routine in the kernel is called, and the process moves to the zombie state.

Sleeping (TSSLEEP)

If a system call or other action cannot be completed immediately, the process is put to sleep until a time when it can be completed. While sleeping, a process is given a special priority.

There are two main groups of sleeping priorities:

- A priority greater than PZERO — (a smaller number) is uninterruptible, and the process will continue to wait until the event it is waiting for happens.

  This priority is used when the sleep is expected to be very short. For instance, when we are talking to a disk drive, the request will likely take only a few milliseconds. Occasionally, one of these high-priority sleeps will not complete immediately. In this case, we have an uninterruptible process that we cannot kill until the event it is waiting for has occurred. One example of this is talking to a tape drive. Even if you use the **`kill -9`** command on a backup program, it will not terminate until it has rewound the tape. A second




© 2005 Hewlett-Packard Development Company, L.P.

and more difficult case is that of files being accessed via NFS, and the server is not responding.

- A priority lower than PZERO — allows sleep to be interrupted (so you can **`kill`** the process).

  With lower priority sleeps, the kernel expects that the sleep will be for a longer period and allows it to be interrupted. A system call that results in a long sleep must free up any kernel resources that it may use and must set up a cleanup routine to be called if it is killed.

Waiting to execute (TSRUN)

Processes waiting to be scheduled are also marked with a state of TSRUN, indicating that the process is in a position to execute when its turn comes. The system keeps a queue of these runnable processes. The length of the queue is used to work out the system load levels. Various programs use these load levels; **`xload`** is an example. The more processes marked as runnable, the harder the system is working. Not only runnable processes are counted to calculate the load average. Some of the situations that put processes to sleep at high priorities also cause the processes to count against the load averages.

Stopped (TSSTOP)

The stopped state is used when running under the control of the debugger or when job control is used. While the debugger is not allowing the program to execute, the process is held in the stopped state.

This state is also entered in response to the job control signals and left in response to the continue signal SIGCONT, 26.

Four different stop signals can be sent to a process:

| | |
|---|---|
| SIGSTOP, 24 | Cannot be caught or ignored by a process. |
| SIGTSTP, 25 | A stop signal is sent to the process by the terminal device driver when a suspend character has been entered. |
| SIGTTIN, 27 | A stop signal is sent to a background job, which attempts to read from its controlling terminal. If a background process can read from the terminal, a situation exists in which several processes try to read from the same terminal at the same time. Some input goes to one process and some to another. Because there is no control over distribution of data, a job control signal prevents reading from the terminal by background processes. |


**© 2005 Hewlett-Packard Development Company, L.P.**

| | |
|---|---|
| SIGTTOU, 28 | This stop signal is similar to the previous one, except that it applies to output rather that input from background processes. This background output control is optional and can be set using **`stty totstop`**. |

Zombie (TSZOMB)

When a process terminates by running out of code to execute, by calling **`exit`** or **`_exit`**, or by being killed, it returns a status code to its parent process. This status code consists of two bytes: the *return/exit* code and the reason for exiting. This status code tells the parent what happened to its child. Did it exit? Was it killed? Did it encounter an error, and so on. The parent process collects this return code by calling a **`wait`** system call. When the process is waiting to pass this code back to its parent, it becomes a zombie. In this state, it has freed up all of its resources, except for its slot in the process table, one slot in the thread table, and an area of memory in its kernel stack and its user area.

As part of freeing up all its other resources, it closes any open files. Closing a file is a simple action, but closing a device file may be more complex. If the process holds the last pointer to the device file, a special closing routine is called in the device driver. This closing routine clears up any device-specific actions. In the example of the tape drive, we talk to a tape drive through a rewind device file. When the file is closed, the tape rewinds.

The close function is occurring while the process is in the zombie state. If the close function is unable to complete, we end up with a zombie process on the system that we cannot kill, because a process in the zombie state cannot receive signals; it is already dead.

If a parent exits before the child does, the **`init`** process (process ID 1) inherits the child. **`init`** always executes a wait system call to allow the zombie to terminate completely.

## The `ps` Command

The **`ps`** command indicates what state a process is in and its current priority. From this information, you can determine if the process is interruptible. For sleeping processes, **`ps`** can also give us the wait channel, WCHAN, which is the kernel address associated with the event that sent the process to sleep.

The **`ps`** command, with the **`-l`** (ell) option, indicates the status of any executing process. The **`S`** field gives the process state, which can take one of the following values.

| | |
|---|---|
| **`O`** | Nonexistent (You should never see this.) |
| **`S`** | Sleeping |
| **`W`** | Waiting |




© 2005 Hewlett-Packard Development Company, L.P.

| | |
|---|---|
| `R` | Running |
| `I` | Intermediate; SIDL |
| `Z` | Terminated; TSZOMB |
| `T` | Stopped or Suspended |

*NOTE:* The command `ps` is shows processes, and `ps` does not understand threads. Any multi-threaded process will show up as being in the Run state at all times.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 9–4. SLIDE: Looking at Processes with `ps`

### Looking at Processes with `ps`

`ps` (Process Show) gives the following information about a process:

| | |
|---|---|
| `F` | Flags |
| `UID` | User |
| `PID` | Process number |
| `PRI` | Priority |
| `ADDR` | Address |
| `WCHAN` | Wait channel |
| `TIME` | CPU used |
| `COMD` | Command |
| `S` | Process states |
| `C` | CPU usage counter |
| `PPID` | Parent process's ID |
| `NI` | Nice value |
| `SZ` | Size of process |
| `STIME` | Start time |
| `TTY` | Terminal running on |

## Student Notes

Most HP-UX users use the **`ps`** command frequently. It reports the status of processes. The **`ps`** command provides many different output formats; short, long, and full are the main ones. Each format displays different amounts of information about the process.

### Short Form of the `ps` Command

Used without any options, **`ps`** gives a short output of all the processes associated with the current terminals. Usually, these are only the processes belonging to your session, but occasionally, other processes are reported.

The short form of **`ps`** gives the process ID of the process, (which is useful if you need to kill it), the terminal, the amount of CPU time utilized, and the name of the command executed.

```
$ ps

PID   TTY       TIME  COMMAND
2742  ttyp8     0:00  sh
2736  ttyp8     0:00  sh
2745  ttyp8     0:00  ps
```




© 2005 Hewlett-Packard Development Company, L.P.

## Long Form of the `ps` Command

In the long form, much more information is given. This can be the most useful form for troubleshooting.

```
$ ps -l

F S   UID   PID  PPID  C PRI NI     ADDR   SZ   WCHAN  TTY       TIME   COMD
1 S   201  2742  2736  1 158 20  222b8c0   71  4f13a8  ttyp8     0:00   sh
1 S   201  2736  2735  0 158 20  222b980   71  4f19d4  ttyp8     0:00   sh
1 R   201  2746  2742  6 179 20  222ba00   17          ttyp8     0:00   ps
```

## Full Format of the `ps` Command

The full format displays fewer fields than the long form, but gives the start time for the process and some of the parameters assigned to it.

```
$ ps -f

UID   PID  PPID  C    STIME       TTY     TIME   COMMAND
ken  2742  2736  0   12:07:42    ttyp8    0:00     sh
ken  2736  2735  0   11:58:14    ttyp8    0:00     sh
ken  2747  2742  7   12:08:01    ttyp8    0:00     ps -f
```

The full and long formats can be combined, but the output will exceed 80 columns, making it difficult to read unless your terminal supports wide output.

```
$ ps -lf

F S  UID   PID  PPID  C PRI NI     ADDR  SZ   WCHAN     STIME   TTY  TIME  COMD
1 S  ken  2742  2736  0 158 20  222b8c0  72  4f13a8  12:07:42 ttyp8  0:00  ksh
0 S  ken  2736  2735  0 158 20  222b980  71  4f19d4  11:58:14 ttyp8  0:00  ksh
1 R  ken  2754  2742  7 179 20  222ba00  17          12:19:03 ttyp8  0:00  ps -
lf
```

## Fields

F — Flags/process sub states. A number of conditions can be indicated in this field. Values shown here result from `OR`ing the actual flags together.

| | |
|---|---|
| 0 | Swapped |
| 1 | In core |
| 2 | System process |
| 4 | Locked in core (for example, for physical I/O) |
| 10 | Being traced by another process |
| 20 | Another tracing flag; actually a stopped process |

A system process is one for which there is no file in the system, such as swapper. System processes have PPIDs of 0. They are part of the kernel and cannot be killed. Even when the system is in single-user mode, these processes still run.

Certain I/O operations carried out by the kernel copy data directly from a device into the address space of a running process. In order for this to work, the kernel locks the part of the process to which it will write into physical memory. This


**© 2005 Hewlett-Packard Development Company, L.P.**

field is not used to show processes that have been locked into memory using the **`plock`** system call.

The two flags used for tracing and stopped process (10 and 20) indicate if the process is stopped, and if it is being traced by another process, such as a debugger.

If just the 10 flag is set, the process is executing actively or waiting on the run queue and is being traced by another process.

If the 20 flag is set, the process is stopped by a signal or another process that is tracing it has stopped it.

UID — The user ID number of the effective process owner.

PID — The process identifier is the handle the system uses to identify a process. All commands that directly access a process need to be given this value.

PRI — The current priority that the process has, which can be broken down into five ranges.

–VE and 0–127 — Real-time processes.
The VE values indicate POSIX real-time functions, and are the highest priorities. The processes in the 0-127 range are HP-UX real-time processes.
Real-time processes are provided for control applications. On most systems, it is unlikely that any application will be running at these priorities. A real-time process runs at a higher priority than any non-real-time priority process. If it is able to make use of the CPU, it will acquire the CPU at the next opportunity. A compute-intensive, real-time process will monopolize the CPU until it has completed and will not allow any activity on the system by normal timeshare processes. When troubleshooting problems involving real-time processes, it is often necessary to have a higher priority real-time shell. If you are running over a network or under X Windows, any process used to connect to the shell must also be set to a real-time priority. It is preferable to use a hard-wired terminal.

128–153 — Sleeping, waiting for an external event to occur. In this range, a process should be waiting for an I/O event to occur. The UNIX kernel uses this priority range for waits that are expected to be very short.
When a process has a priority in this range, it will not receive signals sent to it. They will be stored until the wait is complete and will be processed when the process returns to running in user mode. The system is capable of remembering only one of any type of signal for a process.
This behavior does not usually cause problems, as the waits are normally short, with a couple of exceptions.




© 2005 Hewlett-Packard Development Company, L.P.

- First, the tape device driver is considered a fast driver and makes use of these priorities. However, some tape operations, most noticeably rewinding, are not quick. While the tape is rewinding, the process driver cannot be killed. Unless the tape drive was accessed through a no-rewind device file, this mode is entered either when the process closes the tape at completion or when the process is killed.

- The other exception is NFS. A file access made through NFS uses a waiting priority of 148 (as do other file accesses). If the NFS server does not respond, long uninterruptible waits can result. Because these waits may be indefinite and are likely, the NFS code normally allows interrupts, even at this priority.

If a process is sleeping in this priority range and is not remaining in this state, it is sometimes possible to determine what sort of event it is waiting for by looking for the priorities in the header files under **/usr/include/sys**. However, not all wait states are listed, and the names that are given are not always clear.

154–177 Waiting for an event. Here UNIX expects that the waits may be lengthy and includes code to allow the sleep to be interrupted by a signal. When this happens, if the signal does not terminate the process, whatever system call is in progress fails and returns an error to indicate the interruption.
On HP-UX, this behavior can be defeated, allowing the system call to complete.
Most processes spend most of their time sleeping at one of these priorities. It is sometimes possible to find out what a process is waiting for, either from the priority itself or from the WCHAN.

178–255 Process either executing or on the run queue. The higher the number here, the lower the priority. Generally, high values in this range are accompanied by large values in the C field. Large values here indicate that the process has used significant amounts of CPU time recently.

ADDR Gives the kernel address of the process's virtual address space.

This value is of no use to you except in confirming that a parent and child are related and that the child is the result of a **vfork** rather than a standard one. In this situation, the parent and child share the same virtual address space. Another way to determine this is to check the priority. The parent of a **vfork**'d process will be sleeping at 152.

WCHAN When a thread is sleeping, it is waiting for some event. This field tells us an address associated with the event for which the thread is waiting. With some processes it is possible to use either **adb** or **nm** to find out from **/stand/vmunix** what this **wchan** means. This information can be used to figure out how to unlock


**© 2005 Hewlett-Packard Development Company, L.P.**

a sleeping process. Unfortunately, most waiting addresses can no longer be found within the kernel file.

TIME — The total amount of CPU time consumed by the process, including both user and system time. When looking at performance problems, looking first for processes that have consumed large quantities of CPU time can often speed the search for any problems.

COMD — The command name. The output in this column may contain spaces. The default heading for this column is COMMAND if **-o** is specified, and COMD otherwise. The short command name displayed is limited to 14 characters and the long version to 60. Note that programs can override the command name displayed. A good example of this is **sendmail**, which changes the long command display to say whether it is currently accepting connections.

**S** — Process States

- **0** — Non-existent.
  This process does not exist. **ps** does not report it.
- **S** — Sleeping.
  Waiting for an event. See **wchan** and priority.
- **W** — Waiting.
  This process state is no longer used in HP-UX.
- **R** — Running.
  Process executing or on the run queue, or is multi-threaded.
- **I** — Intermediate/IDLE.
  Process being created.
- **Z** — Terminated/Zombie.
  Process exiting waiting for parent to collect return status.
- **T** — Stopped or Suspended.
  Process was being traced and has been stopped by tracing processes, or the process has been stopped by one of the job control signals.

Most processes are usually sleeping. If a large number of processes are marked as runnable, the system may start to show performance degradation.

Terminated or zombie processes should not remain on the system for very long. If the priority is that of a sleeping process, it is in the cleanup phase, for example, closing device files. If its priority is in the normal user range, it is waiting for its parent process to execute a wait system call to collect the return status. If the process remains as a zombie for long, use its PPID to examine the parent.




© 2005 Hewlett-Packard Development Company, L.P.

**C** CPU usage counter

This CPU usage value was discussed in the section on process priorities. For interactive processes, this value will likely be 0, or very low. For CPU-intensive processes, the value will be higher.

This value has a range of 0 to 255.

If several processes display a large value in this field, it is an indication that the CPU loading is high. If this condition is causing performance problems, it may be worth killing or suspending some of these processes, or increasing their nice value. Remember that the scheduler favors interactive processes. Even when several CPU-intensive processes are all fighting for CPU time, interactive processes may be running effectively.

**PPID** The process ID of the parent of this process

The parent will either be the process that created it (using **fork** or **init**). If the parent process exits before the child process does, **init** adopts the process and collects its return code when it exits. If a process is sitting in the zombie state and appears to have completed, it has a priority figure in the normal running range. This field can be used to find the parent and find out why the parent has not collected the return status of its child. Send a **kill -18** to the parent. This signal notifies the parent that a child has exited and is awaiting the collection of its return status.

**NI** Nice Value

The scheduler makes use of the nice value associated with a process to calculate the rate at which a process loses or gains priority. A user can choose to set a nice value for a process using the **nice** command. Available values are the range 0-19. These are reported by **ps** as 20-39. A superuser can select negative nice values. These are reported in the range 0-19.

An increased nice value increases the rate at which a process loses priority and decreases the rate at which it gains priority, hence making the process nicer to the system.

HP-UX automatically increases the nice value of any background process by 4. CPU-intensive processes running on systems that are primarily used for interactive work can make use of the nice feature.

When looking for causes of performance issues, it is a good idea to look at processes with large priority numbers and at large CPU time values with low nice (NI) values. When troubleshooting a very busy system, as superuser, run a shell at a negative nice value to allow enough priority to execute your commands in a timely manner. If this fails, try using a real-time shell. (If you have a networked or X Windows session, remember to raise those values to real time as well.)


**© 2005 Hewlett-Packard Development Company, L.P.**

**SZ** The size in physical pages of the core image of the process, including text, data, and stack space. Physical page size is defined by _SC_PAGE_SIZE in the header file **unistd.h**, and for HP-UX is defined as 4 KB.

**STIME** The start time of a process is displayed either as a date or as a time, depending on how long the process has been running. Look for shells, interactive processes, or user commands started some time ago. This field can also be used in conjunction with others to see how the process is using the system. For example, if a process has used almost 60 minutes of CPU time and was started an hour ago, this indicates that the program is very CPU intensive. If performance problems are suspected with this process, the probability is that the CPU is not powerful enough, or that the program is inefficient or is malfunctioning.

**TTY** The controlling terminal for the process group to which this process belongs. If this field contains "?", the process is not attached to any terminal, or it has failed to open the terminal to which it wishes to be attached. Otherwise, it allows users to identify from where the program is being run. Daemon processes normally disassociate themselves from a terminal. As they have no input or output, they do not require a terminal. Processes attempting to talk to a modem will not succeed in opening it until all modem lines are correctly set. A **getty** attached to a modem sits with a "**?**" in this field, waiting for an incoming call.

If the environment variable **UNIX95** is set to true, then **ps** will offer a number of other options, mostly associated with formatting the output.




© 2005 Hewlett-Packard Development Company, L.P.

## 9–5. SLIDE: Identifying the Hierarchy of a Process

### Identifying the Hierarchy of a Process

```
# UNIX95=true ps -eH
  PID TTY          TIME CMD
    0 ?        00:00:00 swapper
    2 ?        00:00:02 vhand
    1 ?        00:00:00 init
 1249 ?        00:00:00   nfsd
 1250 ?        00:00:00     nfsd
 1254 ?        00:00:00     nfsd
 1256 ?        00:00:00     nfsd
  325 ?        00:00:14   syncer
 1307 ?        00:00:00   dtrc
 1324 ?        00:00:00     dtlogin
 1334 ?        00:00:00       dtlogin
 1353 ?        00:01:11         dtgreet
 1333 ?        00:00:00       dtlogin
16449 ?        00:00:02         dtsession
16488 ?        00:00:03           dtwm
17516 ?        00:00:00             dtexec
17517 ?        00:00:05               dtterm
17519 pts/0    00:00:00                 -ksh
18237 pts/0    00:15:55                   netscape
18238 pts/0    00:00:02                     (dns
18423 ?        00:00:00   dtterm
18425 pts/1    00:00:00     -ksh
18451 pts/1    00:00:00       ps
#=>
```

## Student Notes

In many situations, it becomes necessary to identify a process's hierarchy. In the past, this was done manually by referencing process identifications with the **ps** command, and checking for process relationships. This way of seeking out process relationships is tedious and inefficient.

Another way to accomplish this is:

```
#  UNIX95=true ps -H
```

An added extension to the **ps** option set is defined by the UNIX95 standards. Defining an environmental variable, UNIX95 accesses this extension. By defining this variable, the **-H** (hierarchy) option becomes available to the **ps** command.


© 2005 Hewlett-Packard Development Company, L.P.

## 9–6. SLIDE: The `kill` Command

### The `kill` Command

- `kill` sends signals to processes.
- Not all signals kill processes.

```
INT       Control-c
QUIT            Control-\
KILL            kill!
TERM            default-kill
```

Example

```
# ps -ef
  root 18292     1  0  Mar 31  console   0:00 -sh
  root  2637  2636  0  Mar 30  pts/t0    0:00 -sh
  zman 18669 18668  1 18:50:40 pts/t1    0:00 -csh
# kill [pid]
# ps -fp [pid]   to verify whether [pid] died
```

## Student Notes

### Signals

Signals are used to interact between UNIX and a process. A signal can be sent to a process at any time during its life. The target process does not have to be listening for the signal. As communication tools, signals are very limited because they contain no data.

The kernel, or software using the **`kill`** system call, can generate signals to indicate a problem, like a math error or process notification of a completed event, like an alarm signal.

Signals can tell a process to:
- Terminate immediately.
- Terminate after core.
- Dump — do nothing (ignores a signal).
- Suspend execution and enter the run state, **`SSTOP`**.

The process can call one of its own functions to handle the signal.




© 2005 Hewlett-Packard Development Company, L.P.

Some signals cannot be caught or ignored. These are signals **9**, **SIGKILL**, and **24**, **SIGSTOP**.

*NOTE:* Processes sleeping in the priority range 128 – 153 are *not* signal-able and cannot respond to signals from the **kill** command. These include processes waiting for fast I/O. If a process is stuck in this condition, for example an I/O error, it cannot be killed.

**1** **SIGHUP**, hang up signal
This signal is usually generated by the terminal device driver and is sent to all processes connected to a terminal to indicate that its connection has been lost. This loss of connection can be caused by a modem line or network connection being lost or by a user logging out while background processes are still running. The default response to this signal is for the process to terminate. The **nohup** command works by telling the program it is given to ignore the **SIGHUP** signal.

**2** **SIGINT**, interrupt signal
This signal is usually generated by the terminal device driver and is sent to all foreground processes running on the terminal in response to the user using the interrupt key (usually **Del** or **Ctrl**+**c**). The default response to this signal is process termination.

**3** **SIGQUIT**, the quit signal
This signal is usually generated by the terminal device driver and is sent to all foreground processes running on the terminal in response to the user using the quit key. This key is not normally defined, but can be set up using **stty quit ^y**, for instance, to define **Ctrl** + **y** as the quit key. The default response to this signal is that the process dumps a core file and terminates. This signal can be very useful when a program appears to lock up. The core file allows the programmer who developed the application to see what was happening at the time in the program.

**4** **SIGILL**, illegal instruction
This signal is usually generated by the kernel in response to the CPU generating an illegal instruction error. The default response to this signal is for the process to dump a core file and terminate. If processes die due to this signal, it indicates either the executable program file has become corrupted or a bug exists in the software.

**5** **SIGTRAP**, trace trap
This signal is provided for use by debuggers, who use it to control the execution of the program they are testing. The default response to this signal is the process dumping a core file and terminating.

**6** **SIGIOT** or **SIGABRT**, abort signal
This signal is given two names. Its main use is in the **abort(3)** function, where a program can use this signal to stop itself. The default response to this signal is the process dumping a core file and terminating.

**7** **SIGEMT**, emulation trap
This signal is documented as being a software generated signal, like **SIGABRT**. Both the GPIO and HP-IB device drivers also use it to indicate that a device is requesting


**© 2005 Hewlett-Packard Development Company, L.P.**

attention. The default response to this signal is the process dumping a core file and terminating.

8 **SIGFPE**, floating-point exception
This signal is generated by the kernel to indicate that a floating-point error has occurred. The default response is for the process is to dump a core file and terminate.

9 **SIGKILL**, the kill signal
This signal is generated by software. A process cannot catch or ignore this signal. It always causes a process to exit, once the process has received the signal. Remember that signals are received only by a process in the transition between running in kernel mode and running in user mode, or while sleeping at a lower priority than PZERO. As it is impossible to catch this signal, processes are unable to perform any cleanup functions before exiting. The kernel will clean up resources that the process was using, close all files, clean up semaphores, and so on, but if the process was part of or communicating with a larger application, any application-specific cleanup will not be performed. This can cause failure or corruption of the applications! Failing to clean up properly can cause files and IPC facilities to be left on the system and occupy space that is no longer required.

**SIGKILL** should be used only as a last resort. If it necessary to kill a process, you should attempt to use other signals first. The **shutdown** command sends a signal **15** and two signal **1**'s before resorting to using signal **9**.

10 **SIGBUS**, BUS error
This signal is sent by the kernel to a process that has either attempted to access an incorrectly aligned address (on a PA system, a read of a long integer from an address that is not divisible by 4, for instance) or has attempted unauthorized access to a memory address (a write to a read-only address like the text area, for instance). The default response to this signal is for the process to dump a core file and terminate.

11 **SIGSEGV**, segmentation violation
This signal is generated by the kernel when a process attempts to access a memory address that is not associated with it. This can happen if a pointer becomes corrupt due to a bug in the software, a process overflows its stack space (raising *maxtsiz* might cure the problem), or the stack is unable to grow due to a lack of swap space. The default response to this signal is for the process to dump a core file and terminate.

12 **SIGSYS**, bad argument to a system call
The kernel generates this signal when a process passes a bad value through to a system call. In most cases, the kernel is able to trap these when starting the system call and just returns an error to the calling process. If this does not happen, the kernel responds with this signal. The default response to this signal is for the process to dump a core file and terminate. POSIX defines this signal as being the result of an illegal value for the **whence** parameter of **lseek**.

13 **SIGPIPE**, broken pipe signal
The kernel sends this signal to a process that writes to a pipe when there is no longer a process reading from the other end. A frequent example of this is quitting from




© 2005 Hewlett-Packard Development Company, L.P.

**more**, when it is being used to control the output from a pipeline of commands. The default action for this signal is for the process to terminate.

**14** **SIGALRM**, the alarm clock signal
The kernel generates this signal at the appropriate time when a process requests an alarm. For instance, the **sleep** command requests that the kernel send it an alarm signal at the end of its sleep. It then enters run state SSLEEP and waits for the signal to arrive. The process consumes no CPU time while waiting. The default action for this signal is for the process to terminate.

**15** **SIGTERM**, the default kill signal
The **kill** command generates this signal if no other signal is requested. The default action for this signal is for the process to terminate.

**16** **SIGUSR1**, user signal 1

**17** **SIGUSR2**, user signal 2
These signals are provided for applications' developers to use for their functions. The default action for these signals is for the process to terminate.

**18** **SIGCHLD** or **SIGCLD**, death of child signal
The kernel sends this signal to a parent process when the child process exits. It warns the parent process that its child has become a zombie; the parent should execute a wait function to collect its return code and allow the child to finish dying. If zombie processes have been left on the system, to clear them, send this signal to the parent. However, the default action for this signal is to ignore it, so it does not always work. Also, if the parent has explicitly asked to ignore this signal, the wait system call will return only when all the children of the process have terminated.

**19** **SIGPWR**, power failure signal
This signal is sent to all processes when power is returned to the system after a power failure. Applications that want to note this occurrence must catch this signal. The default action for **SIGPWR** is to ignore it. Only the HP 9000 servers fitted with the power failure option generate this signal.

**20** **SIGVTALRM**

**21** **SIGPROF**
Both of these signals provide alternate alarm clock functions based on the Berkeley timer functions. The default action for these signals is for the process to terminate.

**22** **SIGIO**
Software can use this signal to provide asynchronous I/O functions. The process can ask the kernel to deliver this signal when it is possible to perform some I/O operation. It allows the process to do other work while waiting for an event. The default action for this signal is for the process to ignore it.

**23** **SIGWINCH**, window changed
This signal is used in a Windows environment to inform applications that the terminal window in which they are executing has changed size. The default action for this signal is for the process to ignore it.


**© 2005 Hewlett-Packard Development Company, L.P.**

**24** **SIGSTOP**, stop signal
This signal causes the process to suspend execution. It enters run state SSTOP. This signal cannot be caught or ignored. If the system is becoming overloaded with work and you want to reduce the number of processes currently executing, stopping processes rather than killing them is a better idea. You can restart the processes later without having to repeat work that is already done.

**25** **SIGTSTP**, suspend signal
This signal is generated by the terminal driver in response to the suspend key (typically CTRL+z ). The default action is to stop the process.

**26** **SIGCONT**, continue
This signal causes stopped processes to continue executing.

**27** **SIGTTIN**, stop for terminal input
This signal is generated by the terminal device driver while a background process attempts to read from the terminal. If a background process reads from the terminal when a foreground process is also attempting to read from the terminal, it is not possible for the user to control where input went. The default action for this signal is to stop the process.

**28** **SIGTTOU**, stop for terminal output
This signal is sent by the terminal device driver when a background process attempts to write to a terminal. This operation is controlled by the **stty to stop** command. The default action is to stop the process.

**29** **SIGURG**, urgent I/O signal
This signal is generated to warn a process that an urgent or out-of-bound message has arrived on a socket (network) connection. The default action is to ignore the signal.

**30** **SIGLOST**, file lock lost signal
This signal is sent to processes that lock files over an NFS connection, if it is not possible to regain the file lock after an NFS server crash. The default action for this signal is process termination.

**31** **RESERVED**

**32** **SIGDIL,** Device IO Library

**33** **SIGXCPU,** CPU time exceeded, set with **setrlimit**

**34** **SIGXFSZ,** File system exceeded, set with **setrlimit**

**35** **SIGCANCEL,** used for PTHREAD cancel

**36** **SIGGFAULT,** Grahocs frame buffer error

**37** **SIGRTMIN,** POSIX real-time signals

**38** **SIGRTMIN+1**




© 2005 Hewlett-Packard Development Company, L.P.

```
39  SIGRTMIN+2

40  SIGRTMIN+3

41  SIGRTMAX-3

42  SIGRTMAX-2

43  SIGRTMAX-1

44  SIGRTMAX
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 9–7. SLIDE: Zombie Processes

### Zombie Processes

```
# ps -f

UID   PID   PPID   C      STIME   TTY      TIME COMMAND
john 1102   1101   1   12:43:04   ttyp3    0:00 sh
john 1361   1360   1   16:19:42   ttyp3    0:00 <defunct>
john 1362   1360   1   16:19:42   ttyp3    0:00 <defunct>
john 1360   1102   93  16:19:42   ttyp3    0:02 ./f1
john 1363   1102   6   16:19:45   ttyp3    0:00 ps -f
```

## Student Notes

Zombies are defunct processes. They are dead, but they have not been cleared away. A zombie merely occupies a slot in each of the process and thread tables. It has a couple of small, dynamically allocated structures in the kernel. It has no other allocated space in user or kernel space. If large numbers of zombies are left on the system, they can cause problems by overflowing the system process table.

### Kernel-Related Cleanup

When a process exits, it performs several kernel-related cleanup functions:

- All open file descriptors in the calling process are closed.
- If the parent process of the calling process is executing one of the **`wait()`** system calls, it is notified of the termination of the calling process. The low-order eight bits of the exit code are made available to it (see **`wait(2)`**).
- If the parent process of the calling process is not executing one of the **`wait()`** system calls and does not have **`SIGCLD`** set to **`SIG_IGN`**, the calling process is transformed into a zombie process.




© 2005 Hewlett-Packard Development Company, L.P.

- Time accounting information is recorded for use by **`times()`** (see **`times(2)`**).
- The parent process ID is set to 1 for all of the calling process's existing child and zombie processes, which means that the initialization process (process ID 1) inherits each of these processes.
- Each attached shared memory segment is detached and the value of **`shm_nattach`** in the data structure associated with its shared memory identifier is decremented by 1 (see **`shmop(2)`**).
- For each semaphore for which the calling process has set a **`semadj`** value (see **`semop(2)`**), that **`semadj`** value is added to the **`semval`** of the specified semaphore.
- If the process has a process, text, or data lock, an **`unlock()`** is performed, (see **`plock(2)`**).
- An accounting record is written on the accounting file if the system's accounting routine is enabled (see **`acct(2)`**).
- A **`SIGCHLD`** signal is sent to the parent process.
- If the calling process is a controlling process, the **`SIGHUP`** signal is sent to each process in the foreground process group of the controlling terminal belonging to the calling process. The controlling terminal associated with the session is disassociated from the session, allowing it to be acquired by a new controlling process.
- If the exit of the calling process causes a process group to become orphaned and if any member of the newly orphaned process group is stopped, all processes in the newly orphaned process group are sent **`SIGHUP`** and **`SIGCONT`** signals.
- If the current process has any child processes that are being traced, they are sent a **`SIGKILL`** signal.
- When closing all open files, the kernel checks any device files to see if this was the only process that has the device open. If so, it calls the device driver's close function (if it has one). This can result in the process entering a high priority sleep during the exit.
- If any of the files that are being closed are not referenced in the file system and are not in use by any other process, they are deleted.
- If the parent process does not execute a **`wait`** function, the process will remain on the system as a zombie or defunct process.

## Clearing Zombie Processes

1. Use **`ps -l`** to determine if the process really has become a zombie. Its state will be Z; its size 0; its priority is the same as when it was killed.

   If the priority is in the range 128-153, it is in a high-priority sleep and should not yet be in the zombie state.


**© 2005 Hewlett-Packard Development Company, L.P.**

2. What is its parent process? Check to see if the parent is running.

3. Try **`kill -18`** on the parent process. Does that clear the problem?

4. If the parent has priority 158, it is possible it is waiting for another of its children to die. So, if other children of this process are not needed, try killing them.

5. If the parent process is not needed, kill it. If it dies correctly, **`init`** process ID 1 will inherit the zombie. **`init`** will execute a **`wait`** function for any process.

6. If this fails to clear a zombie process, determine what the zombie is waiting for.

### Limits on Processes

The maximum number of processes that can be active on a system at one time is limited to the maximum number of processes a user can execute and the maximum number of processes a non-superuser can execute.

- The kernel stores information about processes in the process table. The size of this table is controlled by the kernel parameter **`nproc`**. If the process table becomes full, a new process cannot start. Messages are printed on the console warning that the proc table is full.

- The **`kill`** command is built into the POSIX shell. So it is possible to use the **`kill`** command even if the process table is full. In the Bourne shell and **`csh, kill`** runs as a separate process, and thus needs a separate slot in the proc table.

- Using **`exec`** allows your current process to run a new program.

  In a worst-case scenario, **`exec /etc/reboot`** may be the only way to shut down the system (the **`shutdown`** command uses a large number of processes).

- The kernel limits the number of processes that a normal user can run with the **`maxuprc`** kernel parameter. When a user reaches this limit, additional attempts to start processes fail. The shells report the error, **`no more processes`**. On workstations, users can reasonably execute a very large number of processes.

- The superuser is not limited by **`maxuprc`**, only by the size of the process table.

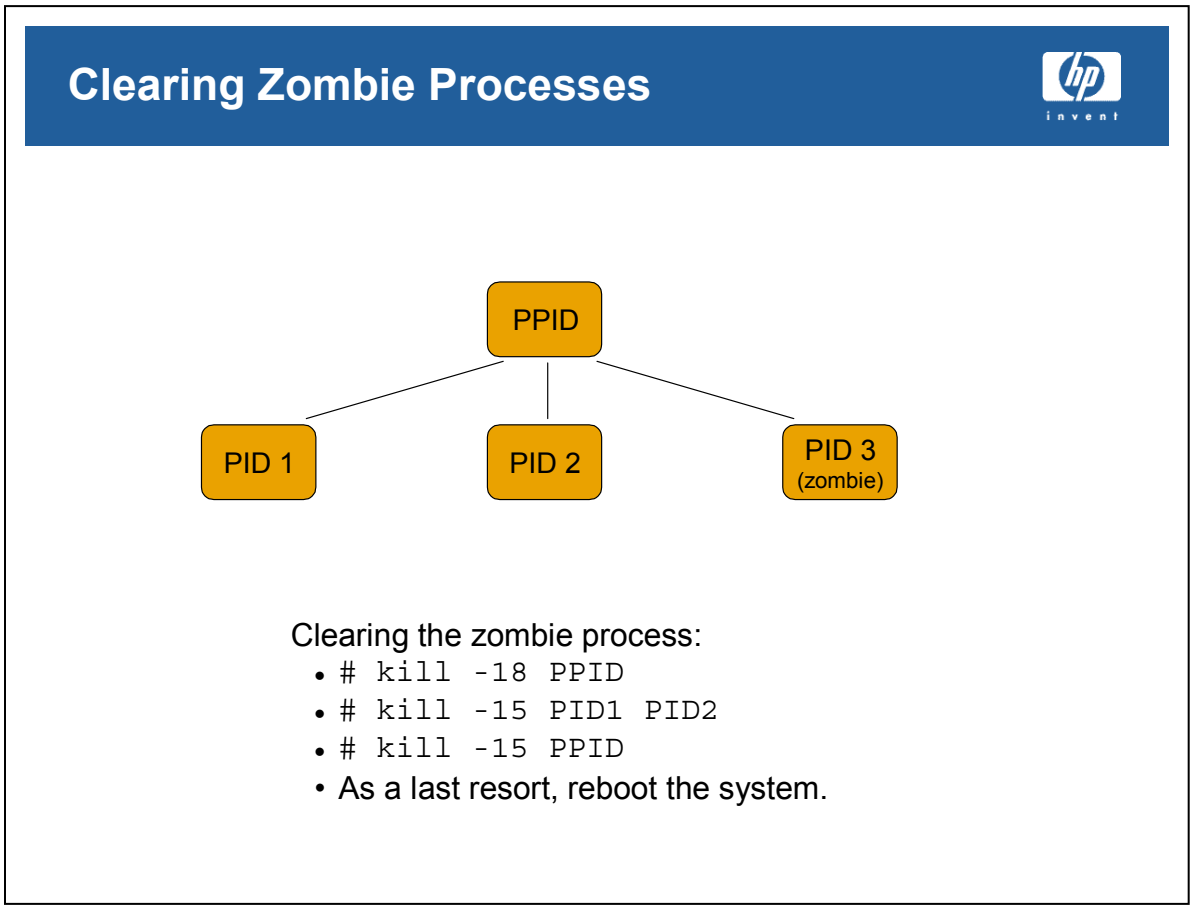
  In addition to the process-related limits, HP-UX also has limits on the number of threads that can be run on the system as a whole and the number of threads each process is allowed to execute.

If an application starts child processes, but isn't interested in their return status, then by explicitly ignoring SIGCHLD (use **`sigaction`**) then the child processes will not become zombies on exiting.




© 2005 Hewlett-Packard Development Company, L.P.

## 9–8. SLIDE: Clearing Zombie Processes




## Student Notes

Clearing zombie processes may require going through several steps. These steps are listed below:

1. Try **`kill -18`** on the parent process.

   Entering **`kill -18 PPID`** forces the parent process to issue wait system calls to the zombie child processes. Upon the zombie child processes receiving the wait system calls, they are removed from the process table.

2. Try **`kill -15`** on other children processes of parent.

   Entering **`kill -15 PID1 PID2`** terminates and changes the state of other children processes to zombie. This is important because in some instances a parent will not issue a wait system call to any zombie child process unless all of its children processes are zombies.

3. Try **`kill -15`** on the parent process.

   Entering **`kill -15`** terminates the parent process. Upon this happening, any zombie process left behind by the parent is inherited by the **`init`** process. The **`init`** process then issues the wait system call against any inherited zombie processes.


**© 2005 Hewlett-Packard Development Company, L.P.**

4. As a last resort, reboot the system.

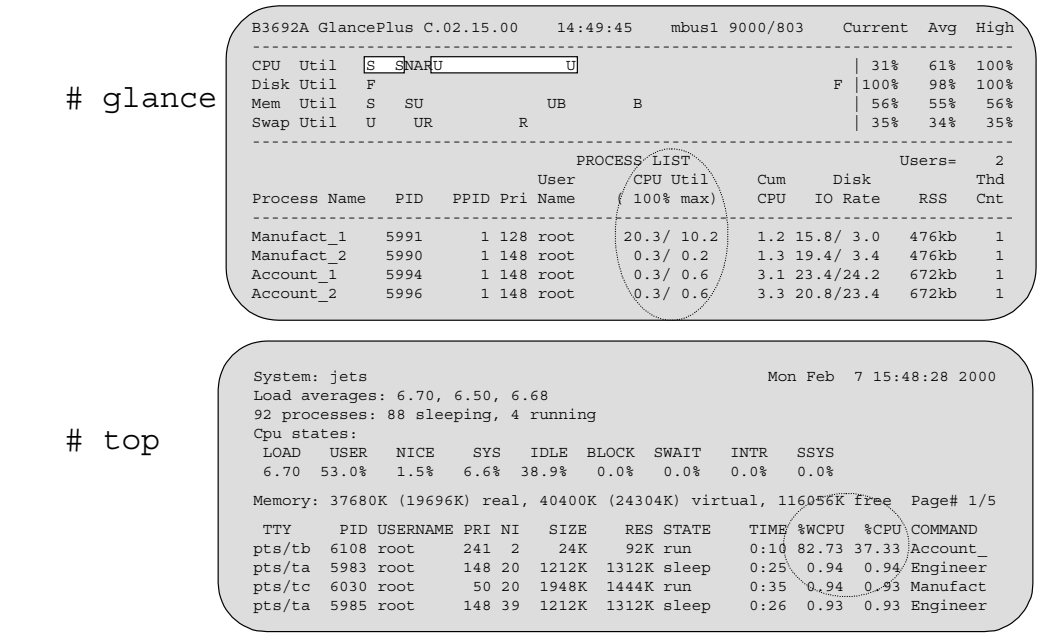
   If, for whatever reason, zombie processes cannot be cleared, a system shutdown and reboot may be necessary.




© 2005 Hewlett-Packard Development Company, L.P.

## 9–9. SLIDE: Identifying CPU-Intensive Processes

### Identifying CPU-Intensive Processes




## Student Notes

There are several ways to identify **CPU-intensive** processes. Some common ways include obtaining this information from purchasable products. One purchasable product, glance, shows this information in an easy to read format.

These purchasable products may not be available in all environments. Therefore, it is good to know about "right out of the box" commands that give similar data. The **`top`** command is an example of a "right out of the box" command that can be used to point in the direction of CPU-intensive processes.

**`glance/gpm`** tool — The **`glance`** command provides real-time monitoring capabilities specific to the HP-UX operating system. The tool provides access to a wide variety of performance data, including which processes are using the majority of the CPU's time.

**`top`** command — The **`top`** command displays the top processes on the system. It periodically updates the information. The raw CPU percentage is used to rank the processes.


© 2005 Hewlett-Packard Development Company, L.P.

## 9–10. SLIDE: Troubleshooting CPU-Intensive Processes

### Troubleshooting CPU-Intensive Processes

- Renice a process.
  ```
  # renice -n [offset_from_20] [PID]
  ```
- Suspend a process temporarily; resume later.
  ```
  # kill -25 [PID]
  # kill -26 [PID]
  ```
- Use Process Resource Manager (PRM) to load balance the processes.
  ```
  # vi /etc/prmconf
  # prmconfig -ek
  ```

## Student Notes

After identifying a CPU-intensive process, what is the next step? Do you allow the process to continue? Do you reprioritize the process? Do you suspend and resume the process? Do you kill the process? Answers to these questions depend on the environment.

In the above example, reprioritize and suspend or resume a process is being performed. You do that with the **`renice`** and **`kill`** commands.

- The **`renice`** command

  The **`renice`** command alters the nice value of one or more running processes. A nice value determines the rate at which a process loses priority while accessing the CPU, and the rate at which a process gains priority while waiting to access the CPU. A higher nice value makes a process less CPU-intensive. A lower nice value makes a process more CPU- intensive. Nice values range from 0 through 39, with 20 being the default nice value assigned to user processes.

- The **`kill`** command

  The **`kill`** command is a signal sender. Once a process receives a signal, it performs some action. Most of the time, it terminates itself. If the intent is to suspend then later resume a process, the **`kill`** command can be issued using signals “25” and “26.” Signal 25 is used to




© 2005 Hewlett-Packard Development Company, L.P.

suspend a process. Signal 26 is used to resume a process. Processes currently suspended can be seen by issuing the `jobs` command.

Be aware that stopping some jobs from an application can have a detrimental affect on the application as a whole. Suppose the job we chose to stop was holding some crucial file lock, then whole large parts of the application in effect also could be stopped.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 9–11. SLIDE: Module Objectives

### Module Objectives

You should now be able to do the following:

- Define a process.
- Display a user's process hierarchy.
- Use the `kill` command.
- Identify and clear zombie processes.
- Troubleshoot CPU-intensive processes.

## Student Notes




© 2005 Hewlett-Packard Development Company, L.P.

## 9–12. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What is a process?

2. Explain the purpose of the **`fork()`** and **`exec()`** system calls.

3. List valid process and thread states.

   Process states:

   Threads states:

4. What action does the **`kill`** command perform?

5. What is a zombie process?

6. What steps should you take to clear a zombie process?


**© 2005 Hewlett-Packard Development Company, L.P.**

7. What "right out the box" commands can help identify CPU-intensive processes?




© 2005 Hewlett-Packard Development Company, L.P.

# 9–13. LAB: Process Hierarchy

## Purpose

This lab is designed to give you hands on experience in determining the process hierarchy for any given user.

## Directions

Read and answer the following questions.

1. Define the appropriate environmental variable such that the **`-H`** (Hierarchy) option becomes available to the **`ps`** command and use the **`ps`** command to generate a report of every process on the system in a process hierarchical format.

2. Use the **`ps`** command to generate a report of every process belonging to root in a process hierarchical format.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 9–14. LAB: Zombie Processes

### Directions

This lab provides you with an opportunity to create and review zombie processes. In the standard **`ps -ef`** command, these processes are seen as defunct processes.

1. Enter these commands to view the source code for this lab:

   ```
   # cd /home/labs
   # more mod9.lab3.c.
   ```

2. Since this file, **`mod9.lab3.c`** is raw source code, we need to compile it into a binary executable that we will name **`mod9`**. To do so, enter these commands:

   ```
   # cc mod9.lab3.c -o mod9
   ```

3. Once you have the **#** prompt back, verify that you have a file named **`mod9`** in your current directory.

4. Now we are ready to execute the program that will create 10 zombie processes. When executing the binary, make it run in the background.

   ```
   # ./mod9 &
   ```

   Your # prompt should return almost immediately.

5. To see the 10 zombie (defunct) processes, use the standard **`ps-ef`** command, and **`grep`** for **`defunct`**:

   ```
   # ps -ef | grep defunct
   ```

6. For a more complete listing, pipe the **`ps -ef`** to more:

   ```
   ps -ef | more
   ```

7. Use the "gentle kill" command on one of the zombie processes: **`kill 3488`** (or whatever the PID number is your system). Did the process die?

8. Next, use an "absolute kill" command on one of the zombie processes: **`kill -9 3488`** (or whatever the PID number is your system). Did the process die?




© 2005 Hewlett-Packard Development Company, L.P.

9. Now kill the parent process (a "normal" `kill` will be OK).

10. Check whether there are zombies left on the system.

11. Did the zombie processes die?


**© 2005 Hewlett-Packard Development Company, L.P.**

# 9–15. LAB: CPU-Intensive Processes

## Purpose

This lab is designed to give you experience troubleshooting CPU-intensive processes.

## Directions

Read and answer the following questions.

1. Execute the following commands.

```
#  cd /tmp
#  vi cpuhog

   #!/sbin/sh
   while true
   do
      :
   done

#  chmod +x cpuhog
#  ./cpuhog
```

2. Open up another pseudo-terminal window.

3. Identify and suspend this CPU-intensive process.

   Issue the **`top`** command.

```
# top
# kill -25 PID
```

4. Issue the appropriate commands ensuring that the process was suspended.

```
# ps -lp PID
```

   The state column (**S**) should be a "T," which means stopped/suspended.

5. **`renice`** the CPU-intensive process using an offset of 12.

```
# renice -n 12 PID
# kill -26 PID (resumes process)
```

6. When finished, kill the **`cpuhog`** PID.




© 2005 Hewlett-Packard Development Company, L.P.

**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 10 — Troubleshooting an HP-UX 11i v2 Kernel

## Objectives

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

- Describe the major steps in recovering from a bad kernel
- Describe how to access **`kcweb`**.
- Describe how to view kernel changes via **`kcweb`** `and` **`kclog`**
- Describe other useful logfiles for troubleshooting kernel problems
- Boot from an alternate kernel
- Boot to tunable maintenance mode


© 2005 Hewlett-Packard Development Company, L.P.

## 10-1. SLIDE: Module Objectives

### Module Objectives

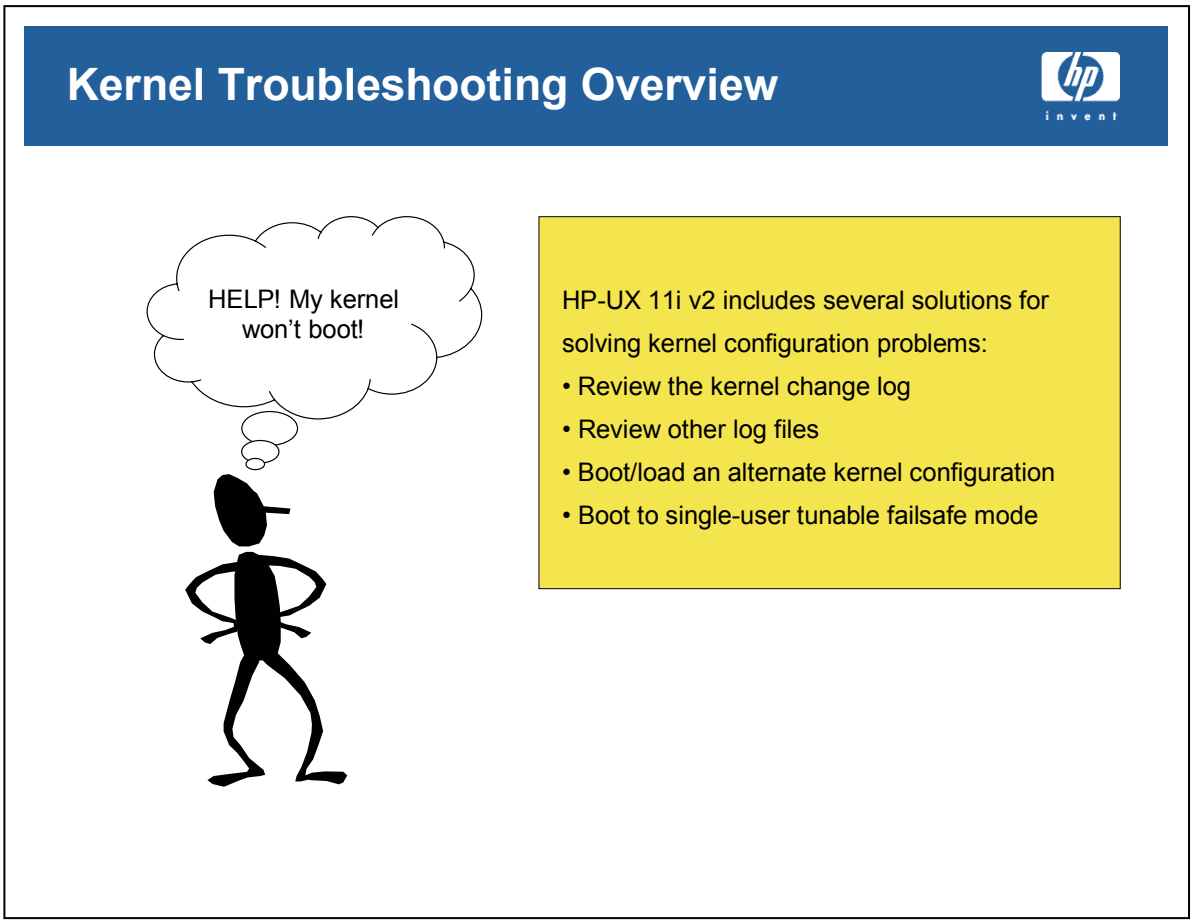
Upon completion of this module, you should be able to:

- Describe the major steps in recovering from a bad kernel
- Describe how to access `kcweb`
- Describe how to view kernel changes via `kcweb` and `kclog`
- Describe other useful logfiles for troubleshooting kernel problems
- Boot from an alternate kernel
- Boot to tunable maintenance mode

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 10-2. SLIDE: Kernel Troubleshooting Overview




## Student Notes

Although the HP-UX 11i v2 **`kc*`** tools automatically verify that tunable parameters fall within allowed boundaries and attempt to verify dependencies, inappropriately chosen kernel parameters may still cause an application – or the entire system – to crash or be unbootable. Fortunately, HP-UX 11i v2 also provides a number of tools to help troubleshoot these situations.

- The new kernel change log records all kernel configuration changes.
- Other log files record actions and errors logged by **`kcweb`**.
- HP-UX 11i v2 makes it very easy to boot or load a backup or alternate kernel if the current kernel configuration becomes unstable.
- If all else fails, every HP-UX 11i v2 tunable parameter has a hard-coded "failsafe" value. Booting to tunable maintenance mode boots the system to single-user mode using these "failsafe" parameters.

The next few slides discuss each of these tools in detail.


© 2005 Hewlett-Packard Development Company, L.P.

## 10-3. SLIDE: Kernel Recovery Roadmap

### Kernel Recovery Roadmap

| System State | Configuration State | Action |
|---|---|---|
| System is up | backup is good | Load the backup kernel configuration, or another good kernel configuration using `kconfig -l` |
| | backup is questionable | Find previous changes in the kernel change log and reverse them using `kctune` and `kcmodule` |
| System is down | backup is good | Boot a known good configuration |
| | backup is questionable | Boot to failsafe mode, find previous changes in the kernel change log, and reverse them using `kctune` and `kcmodule`. Alternative: boot from a system recovery tape |

## Student Notes

The diagram on the slide describes how the tools we've discussed might be used if your kernel configuration becomes unstable.

### System Up, Backup Good

If your system is up, and your **`/stand/backup/`** configuration is good, simply load the backup configuration or another named configuration. If the configuration requires changes to static modules or kernel parameters, it may be necessary to reboot.

```
# kconfig -l backup
# shutdown -ry 0
```


© 2005 Hewlett-Packard Development Company, L.P.

### System is Up, Backup is Questionable

If the system is up, but both **`/stand/current/`** and **`/stand/backup/`** are questionable, review recent changes in the kernel log and reverse them using **`kctune`** and **`kcmodule`**.

### System is Down, Backup is Good

If the system is down, but **`/stand/backup/`** or another configuration is good, use the kernel loader to boot the alternate configuration.

### System is Down, Backup is Questionable

If the system is down and **`/stand/backup/`** configuration is questionable, boot to failsafe / tunable maintenance mode, find previous changes in the kernel change log, and reverse them using **`kctune`** and **`kcmodule`**.

With an Ignite **`make_tape_recovery`** tape, it might be easier to reinstall the system from the recovery tape.

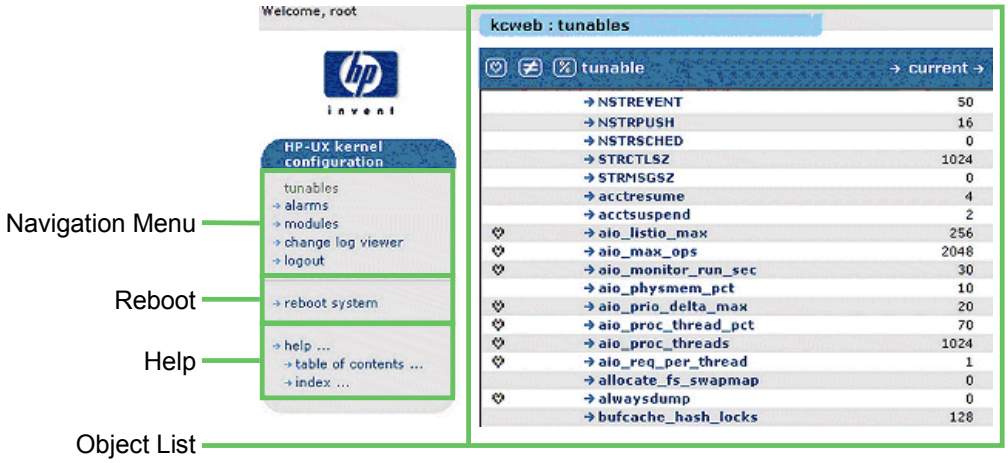

© 2005 Hewlett-Packard Development Company, L.P.

## 10-4. SLIDE: Using the kcweb GUI Interface

### Using the kcweb GUI Interface

If you have HTTPS or X-Window connectivity to your HP-UX 11i v2 server, `kcweb` may be accessed via an intuitive web GUI, too.

```
# echo $DISPLAY
# kcweb [-F] or sam->Kernel Configuration
```




## Student Notes

If you have HTTPS or X-Window connectivity to your HP-UX server, you can access **`kcweb`** via a web GUI interface, too.

### Launching the `kcweb` GUI Interface

The **`kcweb`** GUI is launched from the Kernel Configuration functional area in SAM. If you have a locally attached VGA monitor on your server, and have a **`DISPLAY`** variable set, type **`kcweb`** `at the shell command line prompt`. Running **`kcweb`** via a remote web browser will be discussed later in the chapter.

### Navigating the `kcweb` GUI

After launching **`kcweb`**, you should see an interface similar to the screenshot on the slide.

The navigation menu at top left provides links to the **`kcweb`** functional areas. From this menu, you can choose to manage tunable parameters, kernel resource alarms, kernel modules, or the kernel log,


**© 2005 Hewlett-Packard Development Company, L.P.**

The **`reboot`** button at middle left allows you to reboot the system without leaving the web interface. This is particularly convenient when you are tuning the kernel remotely via your web browser.

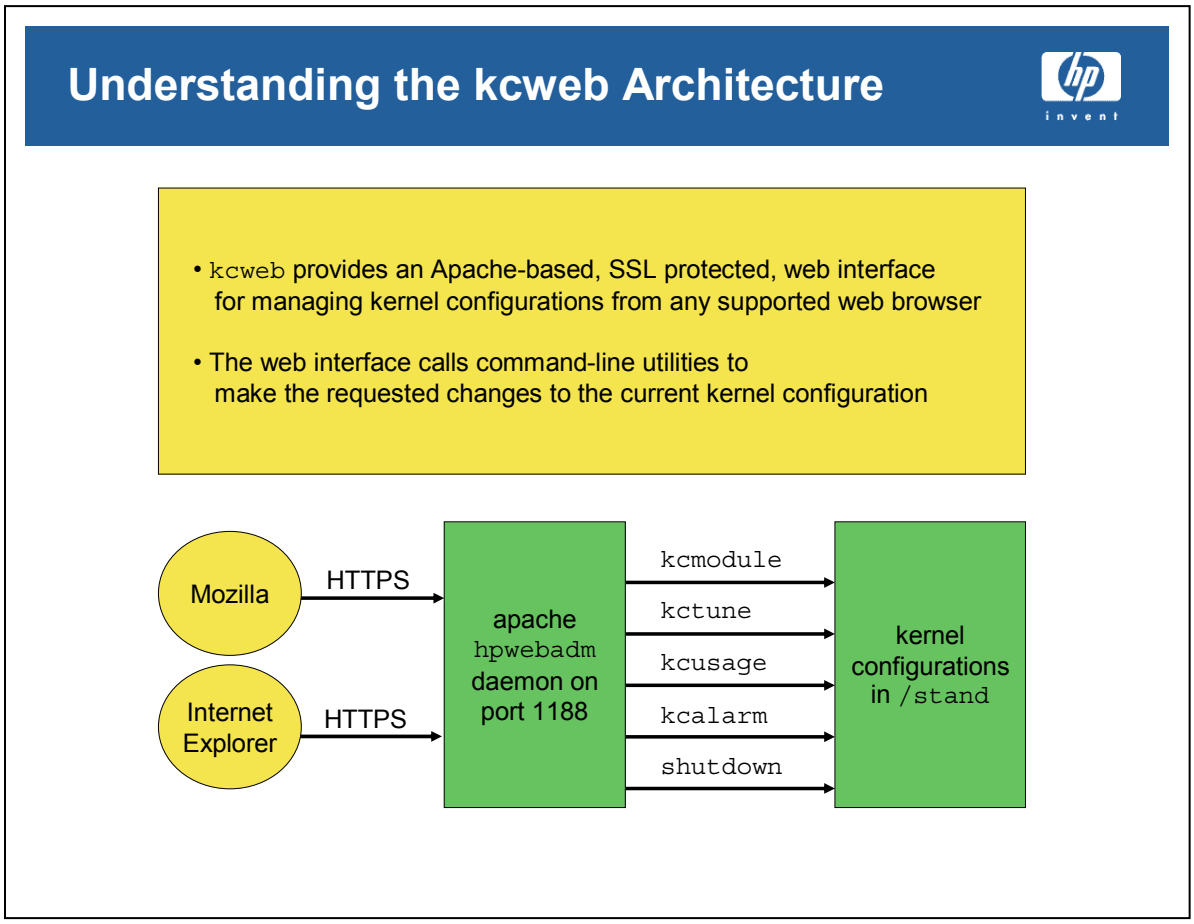
The **`help`** area provides a searchable index of help topics, including detailed information about each tunable kernel parameter.

When you select a functional area from the navigation menu, **`kcweb`** displays an object list in a pane on the right. If you select an object from the object list, **`kcweb`** displays a detail pane (not shown) which displays more detailed information about the selected tunable, module, or alarm.

Later in the chapter you will have an opportunity to explore the object and detail panes in much greater detail.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 10-5. SLIDE: Understanding the kcweb Architecture




## Student Notes

The **kcweb** web GUI interface is provided by a dedicated Apache/hpwebadm server daemon, which runs on port #1188. Administrators can access the **kcweb** interface via a Mozilla 1.2 or Netscape 4.0 or 6.0 web browser running locally on the HP-UX server, or via Internet Explorer 5.0 or 6.0 running on a PC (Note that Netscape 7.0 is not supported). You must have cookies enabled on your browser to use **kcweb**. Cookies are used to track user sessions.

**kcweb** uses Secure Socket Layer (SSL) technology to secure traffic passing between the Apache/hpwebadm daemon and the administrator's browser. Apache/hpwebadm is the same daemon that provides the **pdweb** web interface for managing peripheral devices and interface cards.

When **kcweb** receives a request to modify the kernel configuration, it uses standard command line utilities such as **kctune** and **kcmodule** to make the changes. These commands are logged in **/var/sam/log/samlog**, which may be viewed by typing **/usr/sam/bin/samlog_viewer**.


© 2005 Hewlett-Packard Development Company, L.P.

## 10-6. SLIDE: Launching kcweb via a Local Browser

### Launching kcweb via a Local Browser

To manage kernel configurations on your localhost,
simply set your DISPLAY and BROWSER variables, and run `kcweb`.
`kcweb` will:

- Create SSL server certificates if needed
- Launch the Apache/administration server if it isn't already running
- Launch a web browser

Verify that your `DISPLAY` variable is set

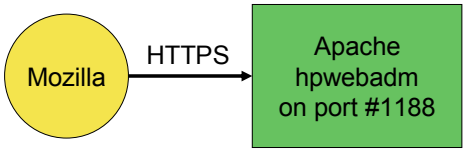
```
# echo $DISPLAY
```

Verify that your `BROWSER` variable is set

```
# export BROWSER=/opt/mozilla/mozilla
```

Run `kcweb` and enter the root username and password when prompted

```
# kcweb &
```




## Student Notes

Administrators can access the **kcweb** interface from a browser running locally on the target server, or from a browser running remotely on another UNIX system or PC.

Several steps are required if you wish to access the **kcweb** interface on the localhost:

1. Verify that your **DISPLAY** variable is set. Running **kcweb** via a remote X-windows session or X-windows emulator such as ReflectionsX is risky since X-windows traffic isn't secure. In fact, **kcweb** automatically verifies that the **kcweb** server's hostname matches the hostname in the **DISPLAY** variable. If the hostnames don't match, **kcweb** won't start unless you add the **-F** (force) option. If you need to configure a kernel on a remote target server, consider using the more secure technique described on the next slide.

   ```
   # echo $DISPLAY
   ```

2. Verify that your **BROWSER** variable is set. If the **BROWSER** variable isn't defined, **kcweb** launches **/opt/mozilla/mozilla**, the default browser in 11i v2.

   ```
   # export BROWSER=/opt/mozilla/mozilla
   ```


© 2005 Hewlett-Packard Development Company, L.P.

3. Run **kcweb** and enter the root username and password when prompted. If **kcweb** is executed without any options, the server daemon will stop automatically after a period of inactivity. If you want the server daemon to continue running until the next reboot, add the **-s startssl** option. If you are accessing the target system via a remote X-windows session, add the **-F** (force) option, too.

   ```
   # kcweb [-F] &
   ```

   **kcweb** will:

- Create SSL certificates for the server if they haven't already been created.
- Launch the Apache/**hpwebadm** server daemon if it isn't already running.
- Launch the web browser defined in your **BROWSER** environment variable.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 10-7. SLIDE: Using kcweb via a Remote Browser

### Using kcweb via a Remote Browser

It is also possible to access the Apache/hpwebadm daemon from a remote web browser on a PC or another UNIX system

Remotely login on the target server and start Apache/hpwebadm daemon

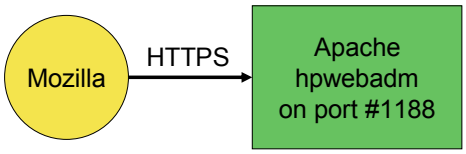
```
server# kcweb -s startssl
server# kcweb -s status
```

Point your local browser to the kcweb URL and enter the root password when prompted

```
client$ mozilla https://server:1188/casey/top.cgi
```

Remotely login on the target server and shutdown the Apache/hpwebadm server (optional)

```
server# kcweb -s stop
server# kcweb -s status
```




## Student Notes

It is also possible to access the Apache/`hpwebadm` daemon from a remote web browser on a PC or another UNIX system.

1. Remotely login on the target server and start the Apache/`hpwebadm` daemon

   ```
   server# kcweb -s startssl
   server# kcweb -s status
   ```

2. Point your local browser to the **`kcweb`** URL and enter the root password when prompted. Be sure to type **`https`** rather than **`http`**! If this doesn't work, verify that your firewall allows access from your local desktop to port 1188 on the target server.

   ```
   client$ mozilla https://server:1188/casey/top.cgi
   ```


© 2005 Hewlett-Packard Development Company, L.P.

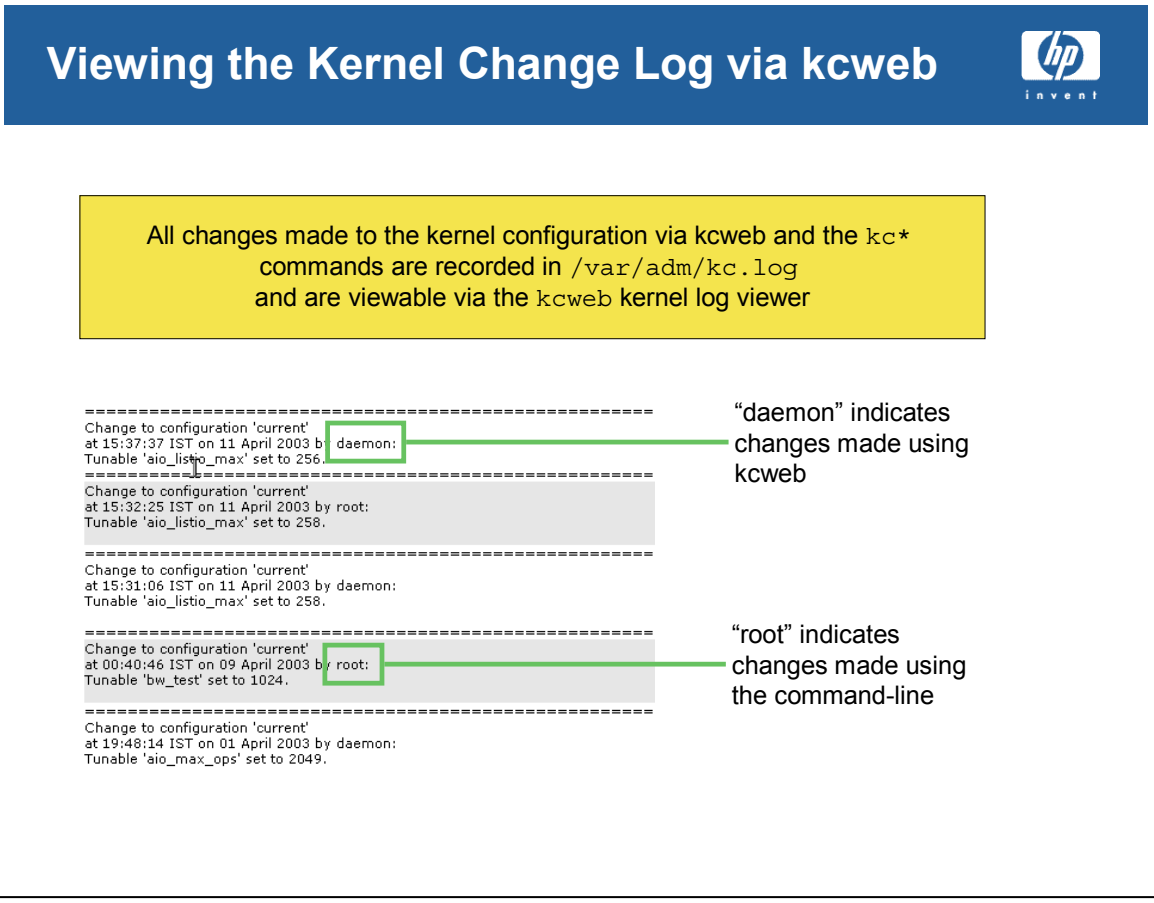
3. After several minutes of inactivity, the Apache/`hpwebadm` daemon will automatically die. If you prefer to stop the daemon explicitly, remotely login on the target server and shut down the Apache/`hpwebadm` server.

```
server# kcweb -s stop
server# kcweb -s status
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 10-8. SLIDE: Viewing the Kernel Change Log via kcweb

### Viewing the Kernel Change Log via kcweb




## Student Notes

It is often useful to know what changes have been made on your system. In HP-UX 11i v2, all kernel configuration changes made via **`kcweb`** and the **`kc*`** commands are automatically recorded in **`/var/adm/kc.log`**. The commands automatically log what is changed, by whom, and when. Administrators can optionally add additional comments to the log file via the **`-C`** option on **`kcmodule`**, **`kconfig`**, and **`kctune`**, or the **`comment`** field on the **`kcweb modules`** and **`tunables`** screens. HP recommends including the pre-change value in the comments.

**`kc.log`** is an ASCII text file that can be viewed via the **`more`** command, **`kcweb`**, or a dedicated log viewer called **`kclog`**. To view the log file from **`kcweb`**, select the **`change log viewer`** link from the navigation menu.

When a problem is encountered with a kernel configuration, review the kernel change log for recent changes, and consider using the **`kcmodule`** and **`kctune`** commands to reverse the changes until you find a stable configuration.


© 2005 Hewlett-Packard Development Company, L.P.

## 10-9. SLIDE: Viewing the Kernel Change Log via kclog

### Viewing the Kernel Change Log via kclog

Using the `kclog` command, you can view the contents of the kernel log from the command line, and filter log entries by search key, module, or tunable

View the most recent log entry for the current kernel

```
# kclog
```

View the last 5 entries for a specific named kernel configuration

```
# kclog -c weekendConfig 5
```

Search for entries containing a specified string

```
# kclog -f "Oracle"
```

View entries for a specific tunable or module

```
# kclog -n "maxdsiz"
```

View entries for all tunables

```
# kclog -t tunable
```

Add a general comment in the kernel log

```
# kclog -C "Add this comment to kernel config log"
```

Include a comment when configuring a kernel module/tunable

```
# kcmodule -C "comment" …
# kctune -C "comment" …
```

## Student Notes

Using the **`kclog`** command, kernel configuration events in the kernel log can be viewed from the command line, and may be filtered by search key, module, or tunable. For example:

View the most recent log entry for the current kernel.

```
# kclog
```

View the last 5 entries for a specific named kernel configuration.

```
# kclog -c weekendConfig 5
```

Search for entries containing a specified string.

```
# kclog -f "Oracle"
```

View entries for a specific tunable or module.

```
# kclog -n "maxdsiz"
```


**© 2005 Hewlett-Packard Development Company, L.P.**

View entries for all tunables.

```
# kclog -t tunable
```

Add a general comment in the kernel log.

```
# kclog -C "Add this comment to kernel config log"
```

Include a comment when configuring a kernel module/tunable.

```
# kcmodule -C "comment" …
# kctune -C "comment" …
```


© 2005 Hewlett-Packard Development Company, L.P.

## 10-10. SLIDE: Monitoring kcweb Logs

### Monitoring kcweb Logs

`kcweb` log messages are recorded in a number log files

| Log File | Purpose |
|---|---|
| `/var/sam/log/samlog` | hpwebadm log messages |
| `/var/opt/hpwebadmin/logs/error_log` | Apache error messages |
| `/var/opt/hpwebadmin/logs/access_log` | Apache access log |
| `/var/adm/kc.log` | `kc*` command log messages |
| `/var/adm/syslog/syslog.log` | System log messages |

## Student Notes

Troubleshoot **kcweb** problems by reviewing several different log files.

| Log File | Purpose |
|---|---|
| **/var/sam/log/samlog** | hpwebadm log messages; records **kc*** commands executed by the Apache/hpwebadm daemon. This file must be viewed using the **samlog_viewer** utility. |
| **/var/opt/hpwebadmin/logs/error_log** | Apache error messages; records failed attempts to access hpwebadm via Apache. |
| **/var/opt/hpwebadmin/logs/access_log** | Apache access log; records successful attempts to access hpwebadm via Apache. |
| **/var/adm/kc.log** | **kc*** command log messages; records all **kc*** commands executed from either kcweb or the command line. |
| **/var/adm/syslog/syslog.log** | System log messages |


**© 2005 Hewlett-Packard Development Company, L.P.**

## 10-11. SLIDE: Booting from an Alternate Kernel

### Booting from an Alternate Kernel

If the default `vmunix` kernel configuration won't boot,
boot from the `backup` kernel configuration, or any other named kernel configuration.

Booting the backup configuration on PA-RISC:

```
Processor is booting from first available device.
To discontinue, press any key within 10 seconds.  [Escape]
Main Menu: boot pri isl
Interact with IPL? y
ISL> hpux ls
ISL> hpux /stand/backup/vmunix
```

Booting the backup configuration on Integrity:

```
HP-UX Boot Loader for IA64  Revision 1.71
Press Any Key to interrupt Autoboot
\EFI\HPUX\AUTO ==> boot vmunix        [Escape]
HPUX> ll
HPUX> boot backup
```

## Student Notes

If the default **`vmunix`** kernel configuration won't boot, boot from the **`/stand/backup/`** kernel configuration, or any other named kernel configuration.

On PA-RISC servers, interrupt the BCH autoboot sequence, boot from the primary boot disk to the ISL, list the contents of **`/stand`**, then boot the desired kernel. Note that on PA-RISC you must specify the full path to the static kernel executable.

```
Processor is booting from first available device.
To discontinue, press any key within 10 seconds.
Main Menu: boot pri isl
Interact with IPL? y
ISL> hpux ls
ISL> hpux /stand/backup/vmunix
```


© 2005 Hewlett-Packard Development Company, L.P.

On Integrity servers, interrupt the **`hpux.efi`** autoboot sequence, list the contents of **`/stand`**, and boot the desired kernel. On Integrity, you only need to specify the kernel configuration name rather than the static kernel executable path.

```
HP-UX Boot Loader for IA64  Revision 1.71
Press Any Key to interrupt Autoboot
\EFI\HPUX\AUTO ==> boot vmunix
HPUX> ll
HPUX> boot backup
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 10-12. SLIDE: Booting to Tunable Maintenance Mode

### Booting to Tunable Maintenance Mode

Booting to tunable maintenance mode boots to single-user mode using failsafe tunable parameters hard-coded in the kernel modules. After booting to tunable maintenance mode, review the kernel log, modify the kernel as necessary, and reboot.

Booting to tunable maintenance mode on PA-RISC:

```
Processor is booting from first available device.
To discontinue, press any key within 10 seconds.  Escape
Main Menu: boot pri isl
Interact with IPL? y
ISL> hpux -tm
```

Booting to tunable maintenance mode on Integrity:

```
HP-UX Boot Loader for IA64  Revision 1.71
Press Any Key to interrupt Autoboot
\EFI\HPUX\AUTO ==> boot vmunix        Escape
HPUX> boot vmunix -tm
```

## Student Notes

Booting to tunable maintenance mode boots to single-user mode using failsafe tunable parameters hard-coded in the kernel modules. After booting to tunable maintenance mode, review the kernel configuration log, modify the kernel as necessary, and reboot.

On PA-RISC servers, interrupt the BCH autoboot sequence, boot from the primary boot disk to the ISL, and use **`hpux -tm`** to boot to tunable maintenance mode.

```
Processor is booting from first available device.
To discontinue, press any key within 10 seconds.
Main Menu: boot pri isl
Interact with IPL? y
ISL> hpux -tm
```


© 2005 Hewlett-Packard Development Company, L.P.

On Integrity servers, interrupt the **`hpux.efi`** autoboot sequence, and boot to tunable maintenance mode.

```
HP-UX Boot Loader for IA64  Revision 1.71
Press Any Key to interrupt Autoboot
\EFI\HPUX\AUTO ==> boot vmunix
HPUX> ll
HPUX> boot vmunix -tm
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 10-13. SLIDE: Module Objectives

### Module Objectives

You should now be able to:

- Describe the major steps in recovering from a bad kernel
- Describe how to access `kcweb`
- Describe how to view kernel changes via `kcweb` and `kclog`
- Describe other useful logfiles for troubleshooting kernel problems
- Boot from an alternate kernel
- Boot to tunable maintenance mode

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

# 10-14. LAB: Troubleshooting the HP-UX 11i v2 Kernel

## Directions

Carefully follow the instructions below.

## Preliminary Steps

1. Portions of this lab require console access. Login as root via your Integrity server's MP console interface.

2. Portions of this lab require access to a web browser, too. If you are accessing your Integrity server via HP's remote lab infrastructure, launch Internet Explorer from your portal server. If you are accessing your Integrity server directly via a PC desktop, simply launch Internet Explorer from your Start menu.

3. If your instructor tells you to disable your browser's proxy settings, do so now. Ask your instructor for details.

## Part 1: Configuring and using `kcweb`

Our goal in this portion of the lab is to configure, launch, and navigate the **`kcweb`** interface.

1. Point your web browser to the Apache/`hpwebadm` daemon at **`https://`*`server`*`:1188/casey/top.cgi`**. Don't forget the "**`s`**" on "**`https`**". Also, it may be necessary to fully qualify your server's hostname. What happens?

   **Answer:**

   This should fail since the daemon isn't running yet.

2. In your console window, configure SSL certificates for your server and manually start the Apache/`hpwebadm` daemon. Which **`kcweb`** option can you use to verify that the daemon is running?

   **Answer:**

   ```
   # kcweb -s startssl
   # kcweb -s status
   ```

3. Click the **`Reload`** or **`Refresh`** button in your web browser to attempt to contact the Apache/`hpwebadm` server again. What happens?

   **Answer:**

   It should successfully connect this time!

4. You may see a dialog box noting that the server is using a certificate issued by "a company you haven't chosen to trust". Click "yes" when asked if you wish to proceed.


**© 2005 Hewlett-Packard Development Company, L.P.**

5. When prompted for your username and password, enter the root username and password.

6. Explore the **kcweb** interface to answer the following questions:
   a. What is the current value of the **nproc** kernel parameter?
   b. If you changed the value of the **nproc** parameter would a reboot be required?
   c. Is the **fddi4** module currently linked into the kernel?
   d. If you changed the state of the **fddi4** module would a reboot be required
   e. Are any alarms defined on your system?

   **Answer:**

   The value of **nproc** and the state of the **fddi4** module may vary from system to system. **nproc** is dynamically tunable, but **fddi4** is a static kernel module. By default, three alarms are preconfigured. When any monitored kernel resource reaches 90%, **kcmond** sends a message to the console, root's email, and the **syslog.log** file.

7. Exit out of the **kcweb** interface.

8. Shutdown the Apache/hpwebadm server daemon.

   **Answer:**

```
# kcweb -s stop
```

9. To streamline future attempts to access **kcweb**, configure the **kcweb autowa** service in **/etc/inetd.conf** and **/etc/services**. After making the change, use the **netstat** command to verify that **inetd** is listening on port 1110.

   **Answer:**

```
# waconf -a on
# netstat -an | grep 1110
        tcp 0 0 *.1110 *.* LISTEN
```

Reconnect to the server via the autostart functionality. Point your browser to **http://server:1110/casey/top.cgi**. (Note: Be sure to use **http** rather than **http**s and port 1110 rather than 1188).

## Part 2: Logging and Troubleshooting

1. Reboot your system. When you get to the HPUX OS loader stage, terminate the autoboot sequence.

   **Answer:**

```
# cd /
# shutdown -ry 0
```


© 2005 Hewlett-Packard Development Company, L.P.

```
HP-UX Boot Loader for IA64  Revision 1.71
Press Any Key to interrupt Autoboot
 \EFI\HPUX\AUTO ==> boot vmunix
 Seconds left till autoboot - 10  Escape
 Type 'help' for help
HPUX>
```

2. Boot to tunable maintenance mode. What distinguishes tunable maintenance mode from a normal boot?

   **Answer:**

   ```
   HPUX> boot vmunix -tm
   ```

   When booting to tunable maintenance mode, all tunable parameters are set to the failsafe values defined by the module developers. These failsafe parameters are designed to ensure a successful boot to single-user mode so the administrator can troubleshoot kernel problems.

3. Once you reach tunable maintenance mode, mount all of your file systems. Then use the **`more`** command to review recent kernel changes that were recorded in the kernel change log.

   **Answer:**

   ```
   # mount -a
   # more /var/adm/kc.log
   ```

   At this point, if you needed to, you could reverse the changes found in `/var/adm/kc.log` to bring you back to a working kernel.

4. Reboot your system again and allow it to come up automatically with **`vmunix.`**

   **Answer:**

   ```
   # reboot
   HP-UX Boot Loader for IA64  Revision 1.71
   Press Any Key to interrupt Autoboot
    \EFI\HPUX\AUTO ==> boot vmunix
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 11 — Troubleshooting a Kernel Crash Dump

## Objectives

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

- Describe the purpose of kernel crash dumps
- Configure dump devices
- Configure selective crash dumps
- Configure compressed crash dumps
- Configure **`savecrash`** functionality
- Describe the crash dump directory structure
- Uncompress and view a crash dump via **`crashutil`** and `q4`




© 2005 Hewlett-Packard Development Company, L.P.

## 11–1. SLIDE: Module Objectives

### Module Objectives

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

- Describe the purpose of kernel crash dumps
- Configure dump devices
- Configure selective crash dumps
- Configure compressed crash dumps
- Configure `savecrash` functionality
- Describe the crash dump directory structure
- Uncompress and view a crash dump via `crashutil` and `q4`

## Student Notes

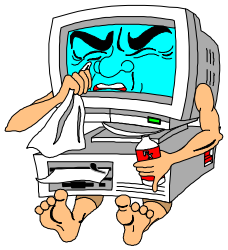

© 2005 Hewlett-Packard Development Company, L.P.

## 11–2. SLIDE: What is a Kernel Crash?

### What is a Kernel Crash?

There are 3 types of Kernel Crashes

- HPMC – High Priority Machine Check
  - Generally hardware caused
- ToC – Transfer of Control
  - Can be manually initiated (usually when system is hung)
  - Can be initiated by ServiceGuard
- Panic
  - Software event, driven by the O/S





## Student Notes

There are three types of kernel crashes:

- HPMC
- ToC
- Panic

### HPMC

A High Priority Machine Check (HPMC) is a special interrupt in PA-RISC and Integrity processors, which most often indicates that the system hardware has had a problem that the operating system software cannot handle. Another category, Low Priority Machine Check (LPMC), is used when software can deal with the problems.

If your system suffers from an HPMC, it is unlikely you will be able to work on the problem yourself. Contact your HP support representative instead.


© 2005 Hewlett-Packard Development Company, L.P.

If you have system diagnostics installed on your machine, in addition to having a crash dump in the **`/var/adm/crash`** directory, you will also have an entry in **`/var/tombstones`**. This has the low level diagnostic information from the CPU's stable storage. The HP Response Center can direct you on the proper steps related to the `/var/tombstones` entries.

Software induced HPMCs are most commonly caused by data corruption within the page directory.
When either user programs or parts of the kernel are running using virtual address mode, the CPU needs to translate the virtual addresses into physical addresses in order to be able to access the RAM on the system. This translation process is carried out within an area of the CPU known as the Translation Lookaside Buffer (TLB). However, the TLB is only able to hold a finite number of translations. If the address accessed is not held within one of the pages currently described in the TLB, a TLB miss fault occurs and a software data structure within the kernel is searched. The translation is then loaded from this structure, the page directory, into the TLB.

If the translation that was loaded points to a physical memory address that does not exist within the machine, an HPMC will result when the system attempts to access the physical address.

## Transfer of Control

Transfer of Control (ToC) can be caused either by manual intervention or by MC/ServiceGuard (as a means to prevent corruption of data).

The administrator usually manually performs a ToC when the system has hung. The method of initiating a ToC varies. On Series 800 systems with access ports or MP/SP/GSP, the ToC can be initiated from the console. Where no access port or MP/SP/GSP is available, a switch or button is used to initiate the ToC.

The primary reason an administrator would invoke a ToC would be in the case of a system/partition hang; That is, it is not responding to any input. Performing a ToC is a last resort. The shutdown is not clean at the operating system or application levels. The file systems will often require repair, and the applications will need to recover from the failure.

There are *different kinds* of hangs. Although, as a user, they all look the same…the system does not respond.

- The system can be busy processing real-time applications. Particularly when the **`rtprio`** or **`rtsched`** commands are used inappropriately within programs.
- The system can be very heavily overloaded.
- A *real* kernel hang may have occurred.

When the problem is a busy system, local action is required to alleviate the problem. In the event of a real kernel hang, you should contact your HP support representative.

In some cases, it may be possible to tell which type of hang has occurred; in other cases, it may not.

Some systems will respond to pings, even when hung, as their LAN cards process these requests.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Initiating a ToC

A transfer of control (ToC) resets the system. It causes a crash dump to be captured (if your system is so configured). The ToC physical mechanism — button, key switch, etc. — varies on different systems.)

- On workstations and some servers, a ToC button is located on the rear or side of the computer.

- On servers with either an access port or an MP/SP/GSP, this function can be initiated by entering console mode using **Ctrl** + **b** , and issuing the **TC** command. The actual key sequence differs between some classes of systems. For example, on a Superdome, N-Class, or L-Class system:

These systems have an MP and can ToC the system from the console

1. Enter **Ctrl** + **b** at the system console to gain access to the MP/SP/GSP, then type **CM.**

2. At the **CM** prompt, enter **TC**.

## ToC with ServiceGuard

ServiceGuard uses the ToC as the fastest way to guarantee that the system halts without communicating with any of the file system disks. It does this to guarantee data integrity only when it believes that it can no longer ensure that only one system may try to communicate with the shared drives.

## Panic

A panic is a software event. When the HP-UX kernel detects some form of internal inconsistency or error, it stops processing, as continuing could make the situation worse, possibly leading to data corruption. Data corruption needs to be avoided at all costs, so the system panics and stops processing immediately.




© 2005 Hewlett-Packard Development Company, L.P.

## 11–3. SLIDE: Configuring Selective and Compressed Kernel Dumps

## Student Notes

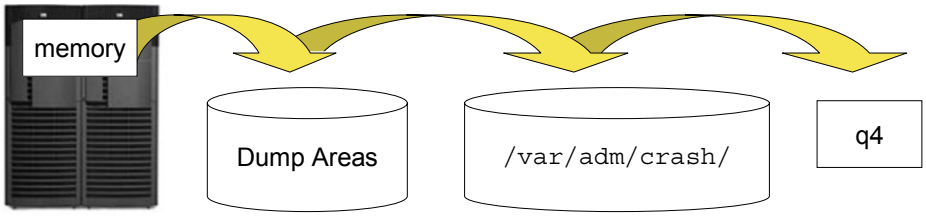
In the next several slides, we will discuss the specifics in capturing a kernel crash dump. The kernel crash dump is a means by which we (or the HP Response Center) can determine what caused the system hang or crash.


© 2005 Hewlett-Packard Development Company, L.P.

## 11–4. SLIDE: Kernel Crash Dump Overview

### Kernel Crash Dump Overview

- When HP-UX encounters an unrecoverable error, a system “panic” results
- When a panic occurs, the kernel can dump the contents of physical memory to disk
- Administrators can also manually initiate the dump process via the MP tc command
- After rebooting, HP-UX may move the data from dump areas to `/var/adm/crash/`
- HP dump analysts analyze the resulting files via a debugger






## Student Notes

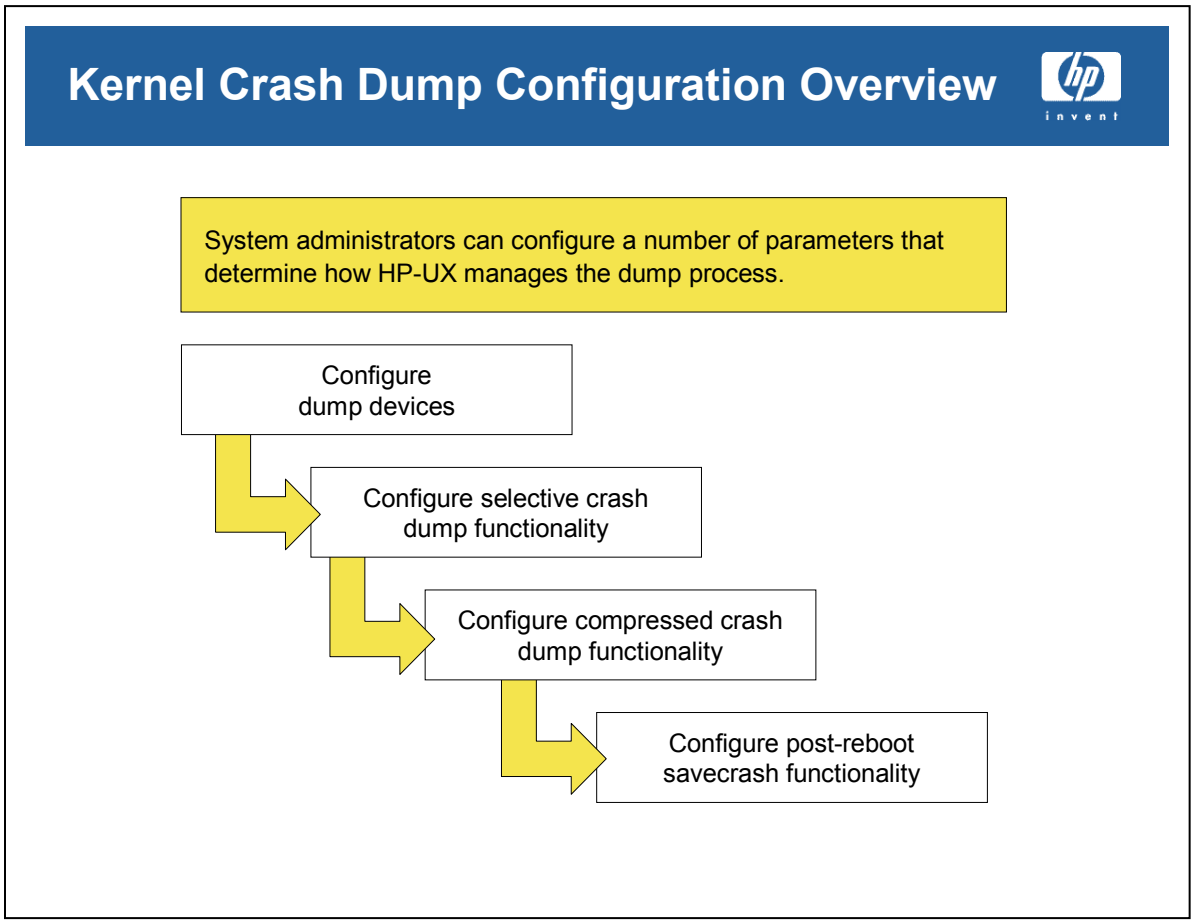
When HP-UX encounters an unrecoverable error, a “panic” results. “Panic” is a section of code in the kernel that causes the system to halt all processing, and may also cause a snapshot to be taken of key sections of memory. This snapshot can then be used to diagnose the cause of the unrecoverable error.

When a panic occurs, the kernel can dump the contents of physical memory to disk. Administrators can also manually initiate the dump process via the MP **`tc`** command.

After the dump is performed, and as HP-UX reboots, the dump is ordinarily moved from dump areas to the **`/var/adm/crash/`** directory.


© 2005 Hewlett-Packard Development Company, L.P.

## 11–5. SLIDE: Kernel Crash Dump Configuration Overview




## Student Notes

There are a number of kernel crash dump related parameters that can be configured in order to capture a crash dump.

First, an inventory of existing dump devices must be taken in order to determine whether more dump space is required.

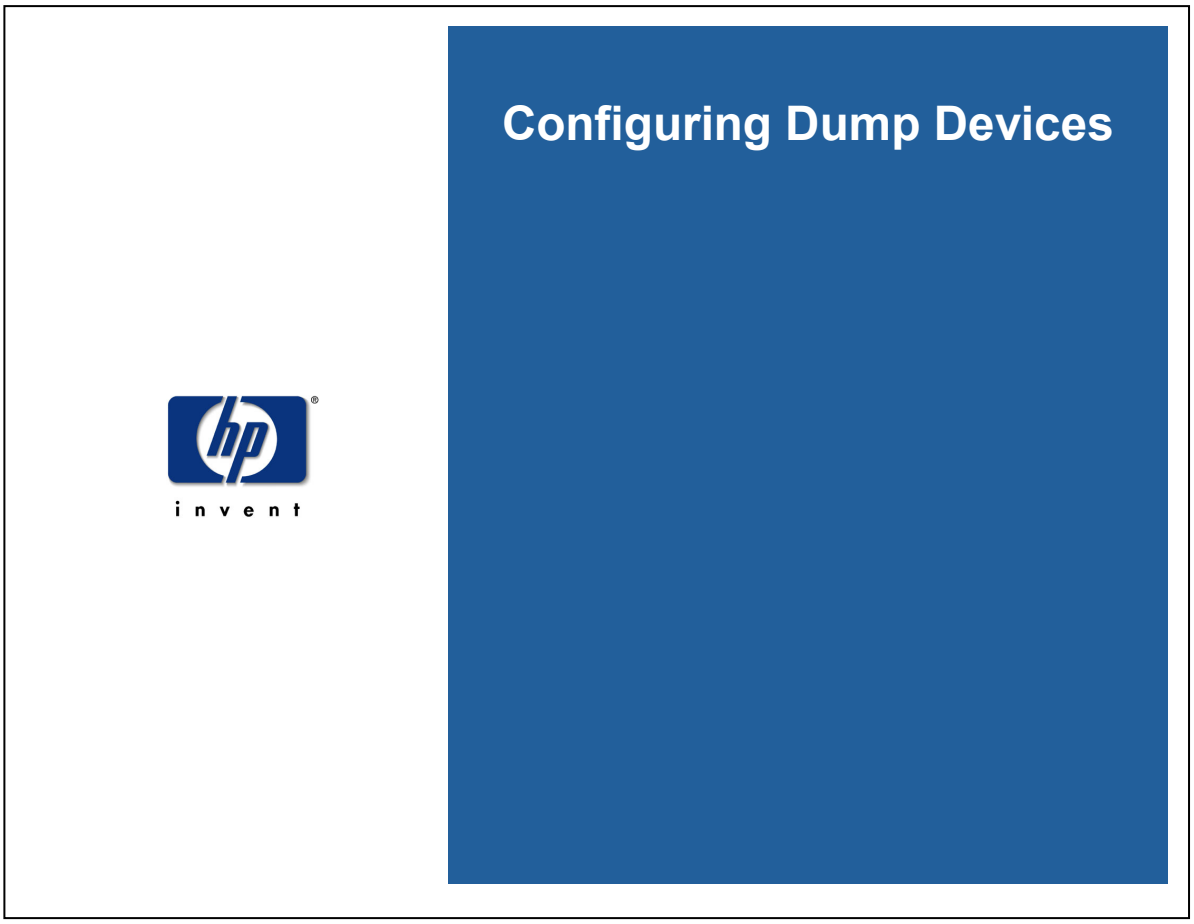
The actual amount of dump space is related to the amount of memory installed in the system. The old-school approach required configuring dump space at a 1:1 ratio with installed memory. However, with large-memory systems, this is most often a wasteful approach.

In addition, some systems allow dump compression, so the old-school approach is rarely practiced anymore.

Finally, some consideration should be given to post-reboot **`savecrash`** behavior.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 11–6. SLIDE: Configuring Dump Devices




### Student Notes

In the next few slides, we will discuss configuring dump devices.

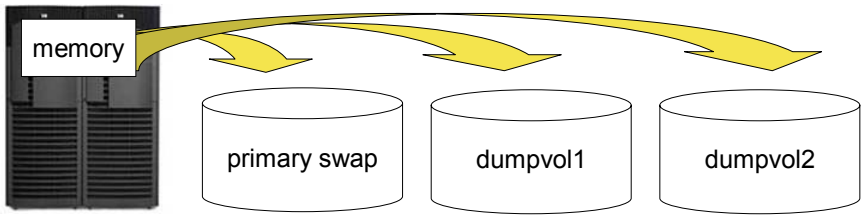



© 2005 Hewlett-Packard Development Company, L.P.

## 11–7. SLIDE: Dump Device Concepts

### Dump Device Concepts

- Ignite-UX automatically configures primary swap as the first or “primary” dump device
- The administrator can optionally specify additional disk or logical volume dump devices
- During the post-crashdump boot process, HP-UX attempts to move the dump data to `/var/adm/crash/` so combination swap/dump devices can be used for swap




## Student Notes

When Ignite installs a system, **`/dev/vg00/lvol2`** is configured as both the Primary Swap and Dump.

In some cases, **`/dev/vg00/lvol2`** may not be sufficiently sized to handle a kernel crash dump, necessitating the configuration of additional dump devices.

During the post-crashdump boot process, information that was stored in the dump areas needs to be moved to the **`/var/adm/crash/`** directory. This is especially important for combination swap/dump devices like **`/dev/vg00/lvol2.`**


**© 2005 Hewlett-Packard Development Company, L.P.**

## 11–8. SLIDE: How Much Dump Space is Required?

### How Much Dump Space is Required?

- Use the `crashconf` command to determine how much dump space is required
- Selective and compressed crashdump features may reduce the amount of space required
- Required dump space may vary as applications are added or removed
- Conservative approach: configure 1MB of dump for every 1MB of RAM

```
# crashconf
…
Total pages on system:          131072
Total pages included in dump:    41911
```

41911 pages x 4k per page = 167644kB of dump space required ???



## Student Notes

Kernel crashes cause downtime. Downtime is expensive, so downtime needs to be minimized, both in frequency and duration. A balance must be struck between minimizing the time required to reboot from a kernel crash and recovering as much information as possible to identify the cause of the crash.

When choosing to preserve a crashed kernel for analysis, two forms of disk space are required - space for the crash dump itself and file system space (typically in **`/var`**) for the preserved dump during the reboot. The amount of memory included in the dump varies based on the uses of the nine classes[1] of memory, and which classes the administrator chooses to include or exclude from the dump. The more inclusive the dump, the more disk space and time is required to preserve it, both after the crash and during the reboot.

In addition to configuring a selective dump, another dump-space consideration is whether or not we can enable dump compression.

[1] user pages, user stack, kernel dynamic data, kernel static data, kernel code, buffer cache, file system metadata, kernel super pages and unused pages


© 2005 Hewlett-Packard Development Company, L.P.

On the slide, without any other consideration, and after executing **`crashconf`** to display the number of pages that would be included in a dump, it would appear that there would only be a need to configure about 42000 pages (at 4kB per page), or approximately 168MB of dump space (on this 512MB system). But we need to understand more before we can come to this conclusion.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 11–9. SLIDE: Configuring Dump Devices via `crashconf`

### Configuring Dump Devices via `crashconf`

- After run level 1, additional dump devices in other VGs may be configured via `crashconf`
- To make these dump areas permanent, add them to `/etc/fstab`
- At run level 1, `/sbin/init.d/crashconf` enables `/etc/fstab` dump areas

View all dump devices
```
# crashconf -v
```

Create a new dump device
```
# lvcreate -n dumpvol2 -L 512 -C y -r n vg01
```

Add the new dump device to `/etc/fstab`
```
# vi /etc/fstab
  /dev/vg01/dumpvol2 ... dump defaults 0 0
```

Add dump devices listed in `/etc/fstab`
```
# crashconf [-r] -a
```

Configure `/sbin/init.d/crashconf` to read `/etc/fstab` at boot time
```
# vi /etc/rc.config.d/crashconf
  CRASHCONF_ENABLED=1
  CRASHCONF_READ_FSTAB=1
  CRASHCONF_REPLACE=1 (optional)
```

Update BDRA pointers: `# lvlnboot -d /dev/vg01/dumpvol2`

## Student Notes

The following steps will enable us to configure dump devices.

To start, view all existing dump devices:

```
# crashconf -v
```

In the next few slides, we will discuss how much additional dump space to configure, but for the moment, consider the steps to simply create a dump device (irrespective of size). Adding a new dump device:

```
# lvcreate -n dumpvol2 -L 512 -C y -r n vg01
```

Requirements for dump devices include the following:
- Must be contiguous (`-C y`)
- Must have Bad Block Relocation turned off (`-r n`)




© 2005 Hewlett-Packard Development Company, L.P.

There is no requirement that additional dump devices be outside of vg00, but it would ordinarily be advisable not to create a logical volume that might end up on the same physical disk as **/dev/vg00/lvol2**. (This is a much more serious concern if the new dump area doubles as swap).

To ensure the system sees this device as a dump logical volume at each boot, add the new dump device to **/etc/fstab**.

```
# vi /etc/fstab
  /dev/vg01/dumpvol2 ... dump defaults 0 0
```

The following command then reads **/etc/fstab** and updates the list of devices that can be used for dump:

**# crashconf [-r] -a** (where **-a** *adds* devices that are listed in **/etc/fstab** and -**r** *replaces* with devices listed in **/etc/fstab**)

Next, configure the startup script, **/sbin/init.d/crashconf**, to read **/etc/fstab** at boot time.

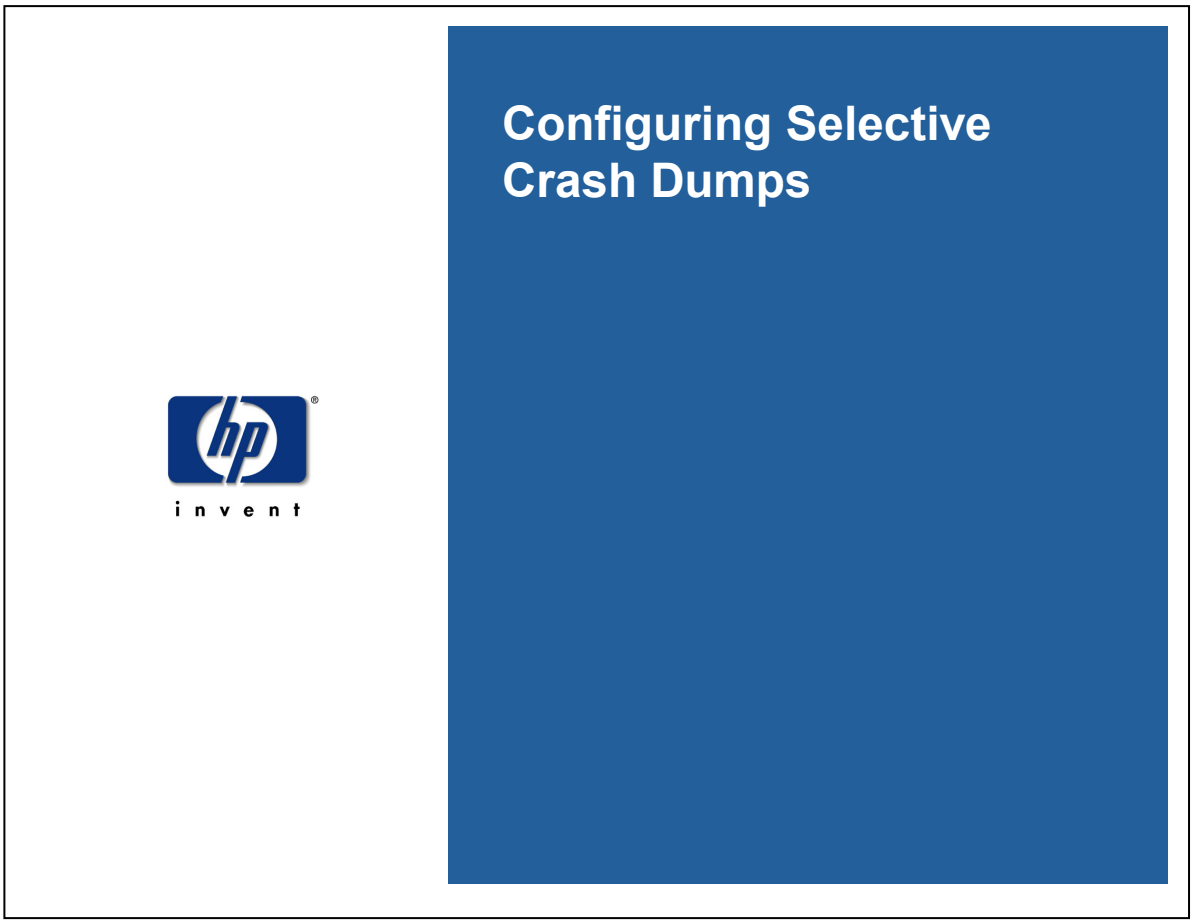
```
# vi /etc/rc.config.d/crashconf
  CRASHCONF_ENABLED=1
  CRASHCONF_READ_FSTAB=1
  CRASHCONF_REPLACE=1 (optional)
```

Finally, update the pointers in the BDRA to reflect the new dump device

```
# lvlnboot -d /dev/vg01/dumpvol2
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 11–10. SLIDE: Configuring Selective Crash Dumps






## Student Notes

One question that should be asked when determining the amount of dump space to configure: What classes of memory shall be dumped in the event of a kernel crash?


© 2005 Hewlett-Packard Development Company, L.P.

## 11–11. SLIDE: Selective Crash Dump Concepts

### Selective Crash Dump Concepts

- On large memory systems, dumping the full contents of memory becomes impractical
- Solution: Only dump the most "useful" memory classes
- Resulting dumps are typically significantly smaller than complete dumps
- By default, the kernel decides which memory classes to dump at dump time

```
# crashconf -v
CLASS          PAGES  INCLUDED IN DUMP  DESCRIPTION
--------  ----------  ----------------  --------------------------
UNUSED        753071  no,  by default   unused pages
USERPG         55255  no,  by default   user process pages
BCACHE         95339  no,  by default   buffer cache pages
KCODE           6462  no,  by default   kernel code pages
USTACK           513  yes, by default   user process stacks
FSDATA             2  yes, by default   file system metadata
KDDATA        118268  yes, by default   kernel dynamic data
KSDATA          2858  yes, by default   kernel static data
SUPERPG        12564  no,  by default   unused kernel super pages
Total pages on system:         131072
Total pages included in dump:   41911
```

Note the dump size difference!

## Student Notes

It is no longer practical to subscribe to the old-school approach of a 1:1 ratio of installed memory to configured dump space. Instead, a better philosophy is to dump selected classes of memory, or areas of memory that will be most useful to diagnose the cause of the kernel crash.

Resulting dumps can be considerably smaller in size. Two advantages to configuring selective crash dumps are: 1) less dump space required, and 2) less space used in **/var** when the dump is moved there post-reboot.

Another objective of both selective and compressed dumps, and likely more important, is to minimize crash downtime. A corollary objective is to minimize the amount of disk space and I/O time required to preserve crash-time memory for analysis. To capture enough memory for analysis without saving unnecessary data is the administrator's balance decision.


**© 2005 Hewlett-Packard Development Company, L.P.**

Selection and compression are separate configurations that can be used together or separately. In reality, all dumps are selective, in that the administrator or system defaults choose which of the nine memory classes will be included or excluded, should the kernel crash. The slide shows the HP-UX11iv2 default selections, which the administrator can modify with the **`crashconf`** command.

## How much Dump Space to Configure?

A good rule of thumb for configuring the proper amount of dump space follows:
On the slide, at the bottom, note the total number of pages included in dump. Ordinarily, we could simply multiply this number by 4kB to provide a rough estimate of required dump space.

One thing to consider: this **`crashconf`** listing is a snapshot at a moment in time. If the **`crashconf`** numbers are calculated when few users or applications are running, they could conceivably be low, which might prove problematic should you have sized your dump space based on this number, and should a dump be performed.

A more conservative approach: execute **`crashconf`** and capture the dump size when the system is busy. And for good measure, consider adding another 25% to the **`crashconf`** -generated "Total pages in dump" number. Make sure your total configured dump areas match this new number.




© 2005 Hewlett-Packard Development Company, L.P.

## 11–12. SLIDE: Configuring Selective Dumps via `crashconf`

### Configuring Selective Dumps via `crashconf`

- Selective dumps may be configured via `crashconf`
- Temporary changes are simply recorded in memory
- Permanent changes are recorded via `alwaysdump` & `dontdump` kernel parameters,

View current selective dump settings

```
# crashconf -v
```

Temporarily modify the existing dump settings ( `-i` = include, `-e` = exclude, `-r` = replace)

```
# crashconf [-r] -i FSDATA -e UNUSED
```

Permanently change selective dump settings

```
# crashconf -t [-r] -i FSDATA -e UNUSED
```

Verify permanent selective dump settings as recorded by kernel parameters

```
# kctune alwaysdump dontdump
```

## Student Notes

Selective dumps may be configured via **`crashconf`**.

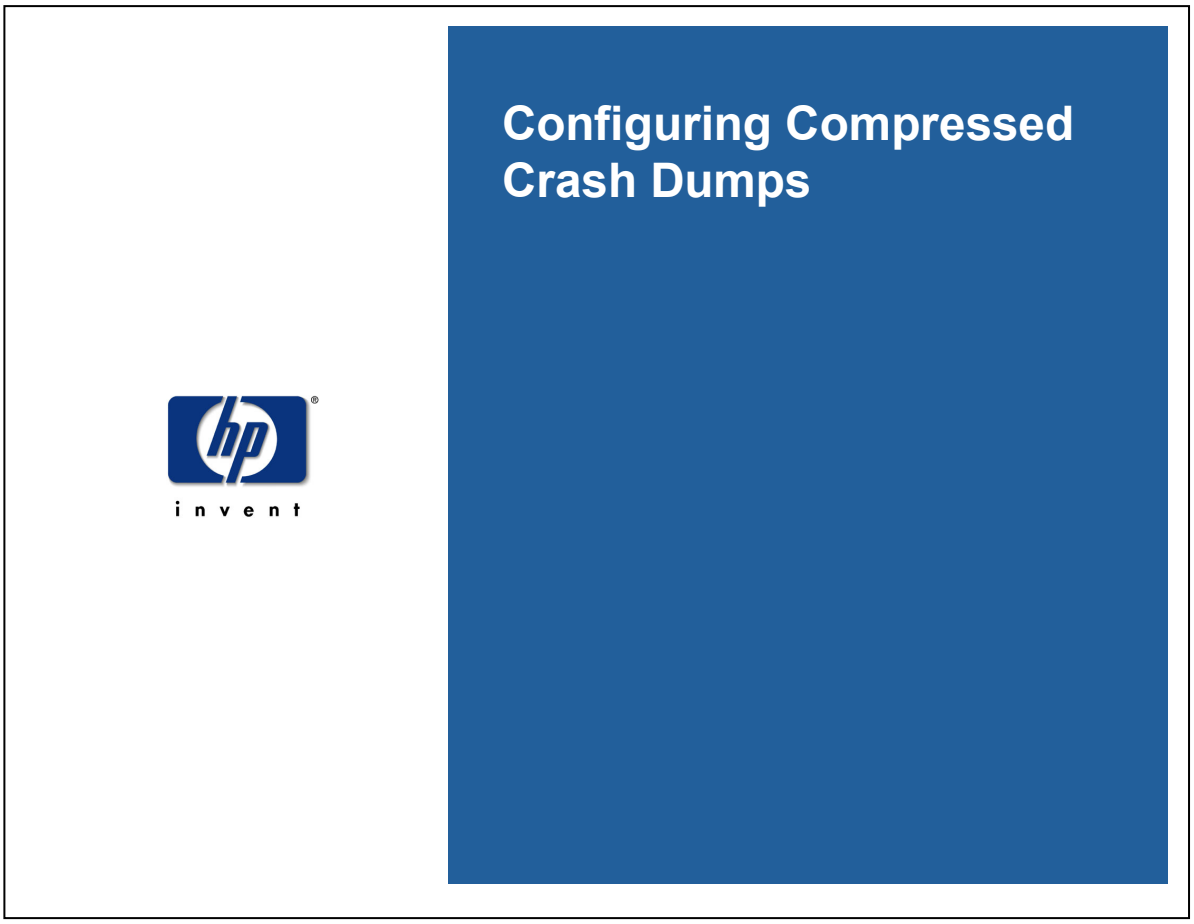
To make temporary changes, use **`crashconf -i`** (to include a memory class), **`-e`** (to exclude a memory class), or **`-r`** (to replace).

To make changes permanent, use **`crashconf -t`**, as illustrated on the slide.

To verify permanent selective dump settings, use **`kctune`** with the appropriate parameters.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 11–13. SLIDE: Configuring Compressed Dumps






### Student Notes

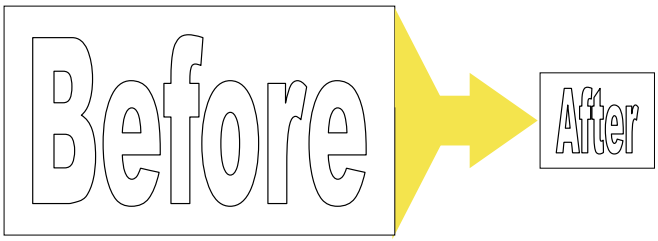
Another method for minimizing dump size, and thereby minimizing reboot time after a crash, is dump compression.


© 2005 Hewlett-Packard Development Company, L.P.

## 11–14. SLIDE: Compressed Crash Dump Concepts

### Compressed Crash Dump Concepts

- You can decrease the size of a crash dump even further by enabling dump compression
  - On HP-UX 11i v1, install CDUMP11i from the software pack CD
  - On HP-UX 11i v2, compressed dump functionality is included in the OE
- Requirements:
  - Five or more CPUs
- Advantages:
  - Faster dumps
  - Less dump space required
  - Less space required in `/var/adm/crash/`




## Student Notes

Dump compression is included with HP-UX11iv2 and can be patched into HP-UX11iv1[2]. Designed for large systems, both in memory and CPUs, it requires five CPUs - four to apply the compression algorithm and evaluate the results and one to control the dump output[3].

Compression is most effective on repetitive data, and therefore, is not guaranteed to create a smaller quantity of data to dump than the original, known as a compression unit.
Since compression units are isolated from each other, a crash dump may contain some compressed and some uncompressed images. If the compressed data is actually larger (due to its randomness) than the original uncompressed data, the uncompressed data will be dumped, and its subindex file is set accordingly. In other words, although dump compression is commanded, some memory compression units may be saved uncompressed while others in the same crash are saved compressed. The result is smaller chunks of memory to dump, taking less time and disk space, and less dump data to preserve, taking less time and file system space on the reboot. Compression can be enabled persistently or only for the next crash.

[2] http://software.hp.com/portal/swdepot/displayProductInfo.do?productNumber=CDUMP11i
[3] PA-RISC compression also requires >=2GBytes of memory and a 64-bit CPU


**© 2005 Hewlett-Packard Development Company, L.P.**

To determine whether your system can enable dump compression, use the command **`mpsched -s`** to determine the number of CPUs available to the kernel and use the **`dmesg`** command to determine the amount of installed memory..




© 2005 Hewlett-Packard Development Company, L.P.

## 11–15. SLIDE: Configuring Compressed Dumps via `crashconf`

### Configuring Compressed Dumps via `crashconf`

- Compressed dump functionality may be configured via `crashconf`
- Temporary changes are simply recorded in memory
- Permanent changes are recorded via the `dump_compress_on` kernel parameter

View current compressed dump setting

```
# crashconf -v
```

Temporarily modify the existing compressed dump setting

```
# crashconf –c ON|OFF
```

Permanently change the compressed dump setting

```
# crashconf -t –c ON|OFF
```

Verify permanent compressed dump settings as recorded by the kernel parameter

```
# kctune dump_compress_on  # 1=on, 0=off
```

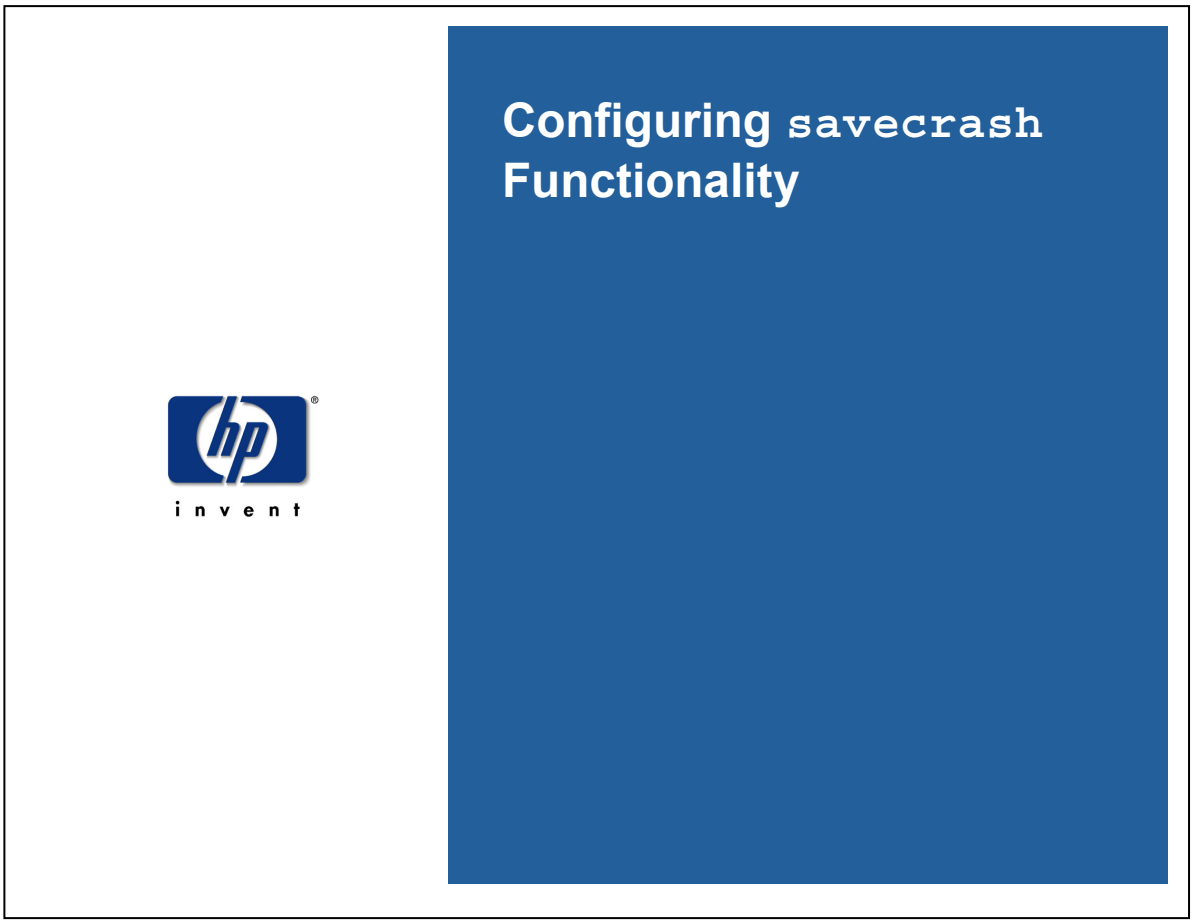
## Student Notes

The HP-UX11iv2 kernel enables crash dump compression by default. The administrator can use the **`kctune`** command to display or enable it (**`-s dump_compress_on=1`**) or disable it (**`-s dump_compress_on=0`**). Notice in the display on the slide that **`dump_compress_on`** is dynamic and changes dynamically, without a reboot.

**`crashconf -c on`** enables compression or **`-c off`** disables it. Without the **`-t`** option, the enable is temporary, and will not survive a reboot. An example of this is setting the boot time parameter **`CRASHCONF_COMPRESSED_DUMP`** in **`/etc/rc.config.d/crashconf`** to either enable or disable compression when entering run level 1. The **`crashconf`** option, **`-c,`** does not assume **`-t`**. With the **`-t`** option, it changes the kernel tunable parameter **`dump_compress_on`**, so the change is persistent. The **`-v`** option includes the current state of the compression enable in its display.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 11–16. SLIDE: Configuring `savecrash` Functionality






### Student Notes

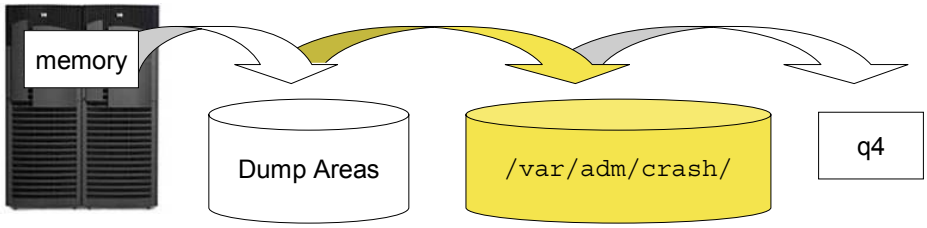
**`savecrash`** is ultimately responsible for managing the dump data and moving it from the dump areas to the designated file system (usually under **`/var`**).


© 2005 Hewlett-Packard Development Company, L.P.

## 11–17. SLIDE: `savecrash` Concepts

### `savecrash` Concepts

- Dump devices often also serve as swap devices
- To prevent `vhand` from overwriting the dump data, …
- `/sbin/init.d/savecrash` attempts to copy dump data to `/var/adm/crash/`
- As each swap/dump area is saved, `savecrash` enables the device for swap
- By default, `savecrash` runs in the background to ensure a speedy boot




## Student Notes

A delicate sequence of steps must take place after the crash dump is saved, post-reboot . Because primary swap also doubles as dump space, the dump data must be "drained" out of dump area (**`/dev/vg00/lvol2`**) and placed in **`/var`**) at the same time **`vhand`** is populating primary swap (also **`/dev/vg00/lvol2`**).

**`savecrash`** facilitates this delicate operation.

The **`savecrash`** command copies dumped crash data to file system files for crash analysis automatically post-reboot. For each crash, **`savecrash`** references **`/var/adm/crash/bounds`** for the next sequence number and creates a subdirectory, **`/var/adm/crash/crash.#,`** for the new files.

If multiple kernel crash dumps are to be archived, it is important to manage available disk space in **`/var`** accordingly. Otherwise, a full **`/var`** filesystem could impact unrelated logging (destined for logfiles in **`/var`**).


**© 2005 Hewlett-Packard Development Company, L.P.**

When **`savecrash`** is run from the command line, the crash files may be written directly to a secondary medium; this takes longer, but avoids the 2-step process of writing to disk then transferring the disk files to the secondary medium. If there is not enough space in the crash save directory to preserve the dump at the next reboot, **`savecrash`** writes an error message to **`/dev/console`** and **`/etc/rc.log`**, bypassing the preservation. The administrator can manually recover the crash data by directing it to a file system with enough free space, enlarging the requested file system, or sending it directly to secondary media, which will be written in `tar` format.

Finally, after moving the crash dump to **`/var/adm/crash/crash.#`**, the **`q4`** utility can be used to extract critical crash-related information from the raw dump.




© 2005 Hewlett-Packard Development Company, L.P.

## 11–18. SLIDE: Configuring Post-dump Behavior via `savecrash`

### Configuring Post-Dump Behavior via `savecrash`

- `/sbin/rc` automatically runs `/sbin/init.d/savecrash` at run level 1
- You can customize `savecrash` behavior via `/etc/rc.config.d/savecrash`

```
# vi /etc/rc.config.d/savecrash

  # If none of your dump devices are swap devices, set SAVECRASH=0 to
  # skip savecrash and ensure a speedy post-crash reboot.
  # when convenient, use crashutil to save the crash to a tape or file.
  SAVECRASH=1

  # If SAVE_PART=1, savecrash will only save dump data in dump+swap areas.
  # Assumes that you will manually save other dump area contents later.
  # Advantage: faster post-crash reboots!
  SAVE_PART=1

  # Set FOREGRD=1 if the boot process should wait for savecrash to finish.
  # Might be useful if starting applications without auxiliary swap is problematic,
  # but delays the post-crash reboot.
  FOREGRD=0
```

## Student Notes

**`savecrash`** is run by default at run level 1 on a boot. However, if customization of **`savecrash`** behavior is desired, **`/etc/rc.config.d/savecrash`** can be modified.

If none of your dump devices double as swap devices, setting **`SAVECRASH=0`** in **`/etc/rc.config.d/savecrash`** will speed up the reboot process because there is no need to transfer the dump information to **`/var`** during the startup process. However, you would need to manually transfer the dump data from the dump areas to **`/var`** later.

If you have a combination of dump and dump/swap areas, setting **`SAVE_PART=1`** in **`/etc/rc.config.d/savecrash`** will only save the dump/swap areas to **`/var`**. Again, if you wish to retain the dump information, you would need to manually transfer the dump data from the dump areas to **`/var`** later.

Finally, setting **`FOREGRD=1`** in **`/etc/rc.config.d/savecrash`** tells the boot process to wait for the **`savecrash`** process to complete before allocating swap space. Remember, dump areas are ordinarily transferred to **`/var`** at the same time as swap is being initialized in the same area.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 11–19. SLIDE: Configuring Post-dump Behavior via `savecrash` (Continued)

### Configuring Post-Dump Behavior with `savecrash` (Continued)

```
# vi /etc/rc.config.d/savecrash

  # Optionally specify a different savecrash destination directory.
  SAVECRASH_DIR=/var/adm/crash/

  # How much free space should savecrash preserve in the file system?
  # if savecrash doesn't have enough space, it simply saves as much as it can,
  # then aborts.
  MIN_FREE=0g

  # If crashconf compression was disabled, savecrash can compress the dump to
  # save space in /var/adm/crash. "0" = don't compress. "1" = always compress.
  # "2" = let the kernel decide.
  COMPRESS=2

  # Define the chunk size if your debugger doesn't support the 64M default.
  CHUNK_SIZE=64m
```



## Student Notes

By default, **`savecrash`** transfers the dump area information to **`/var/adm/crash/`** at boot. If you wish to change this default directory location, the **`SAVECRASH_DIR`** entry in **`/etc/rc.config.d/savecrash`** can be used to specify an alternate location. Be sure the alternate location has sufficient disk space allocated.

The **`MIN_FREE`** parameter in **`/etc/rc.config.d/savecrash`** can be used to configure the amount of free space that should remain in **`/var`** for a non-root user after **`savecrash`** completes. If this parameter is specified, and if **`savecrash`** cannot save the entire dump in the amount of space available, it simply aborts, saving what it has already, but leaving the **`MIN_FREE`** amount available.

The **`MIN_FREE`** parameter can be specified in bytes (b), kilobytes (k), megabytes (m), or gigabytes (g). Default value is 0, and the default unit is kilobytes.

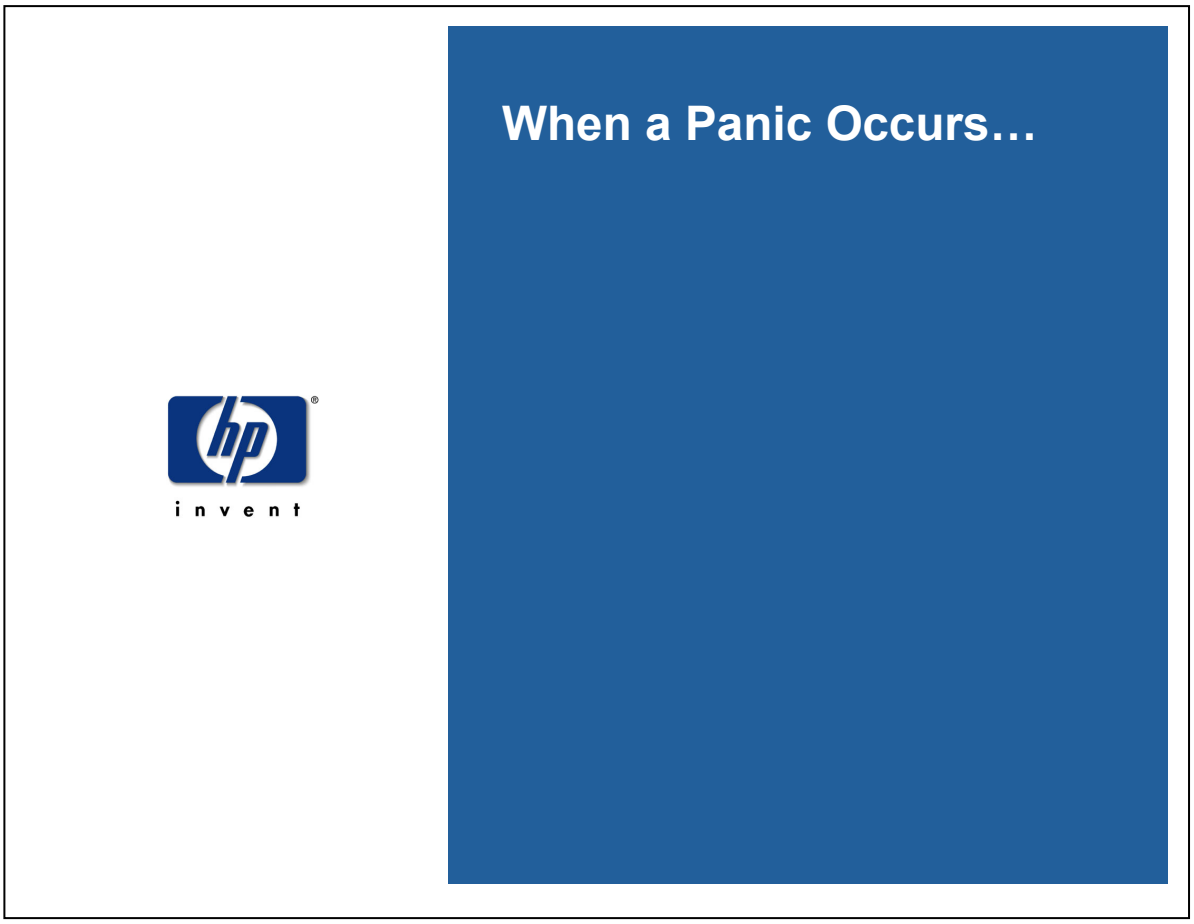

© 2005 Hewlett-Packard Development Company, L.P.

If **CRASHCONF** compression was disabled, **savecrash** can compress the dump to save space in **/var/adm/crash/**. The **COMPRESS** options follow:
0 - don't compress
1 - compress
2 - don't care (default)
If defaulted, **savecrash** will choose based on the physical memory size and available disk space. Note: Compressing will take longer for **savecrash** to complete.

Finally, if your debugger doesn't support the default chunk size of 64M (a parameter that specifies the maximum size of a crash image file), a chunk size supported by your debugger can be configured with the **CHUNK_SIZE** parameter. Restricting the image file size might be necessary for some custom analysis tools.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 11–20. SLIDE: When a Panic Occurs...




### Student Notes

In subsequent slides, the steps to execute after a panic will be discussed.




© 2005 Hewlett-Packard Development Company, L.P.

## 11–21. SLIDE: Manually Initiating a Dump

### Manually Initiating a Dump

If your system hangs and can't be recovered via any other mechanism, you can manually initiate a dump via the MP "transfer of control" command

```
MP> cm
MP:CM> tc
```

## Student Notes

If your system hangs, capturing a crash dump may be advised.

The most convenient method is to initiate a Transfer-of-Control via the MP/SP/GSP prompt.

From an **`MP>`** prompt, type **`CM`**. At the resulting **`MP:CM>`** prompt, type **`tc`** as follows:

```
MP> CM
MP:CM> tc
```

An alternate method, if access to the MP/SP/GSP is unavailable: On non-partitioned systems, there is usually a push-button ToC switch on the rear of the cabinet.


© 2005 Hewlett-Packard Development Company, L.P.

## 11–22. SLIDE: Enabling Selective Dump at Dump Time

### Enabling Selective Dump at Dump Time

When a panic occurs, the operator has an opportunity to override the existing selective dump settings, too.

```
*** The dump will be a SELECTIVE dump:  1240 of 16352 megabytes.
*** To change this dump type, press any key within 10 seconds.
```

Escape

```
*** Select one of the following dump types,
*** by pressing the corresponding key:
S) The dump will be a SELECTIVE dump:  1240 of 16352 megabytes.
P) The dump will be a PARTIAL dump:  6138 of 16352 megabytes.
F) The dump will be a FULL dump of 16352 megabytes.
*** Enter your selection now. S
*** Proceeding with selective dump.
```



## Student Notes

When a panic occurs, if you have access to the system console and have an opportunity to interrupt the automatic dump process within 10 seconds of start, you have the ability to "customize" the dump as Selective, Partial, or Full. The options:

S) The dump will be a SELECTIVE dump
P) The dump will be a PARTIAL dump
F) The dump will be a FULL dump

A selective dump is one where the administrator has specified classes of memory to dump. A partial dump is really an attempt to capture a full dump, but where it is recognized that there won't be enough file system space in the dump directory (**/var/adm/crash**, by default) to capture a full dump. However, the system gives priority to pages which are considered most useful.

A full dump is a dump of the entire memory contents!


© 2005 Hewlett-Packard Development Company, L.P.

## 11–23. SLIDE: Enabling Compression at Dump Time

### Enabling Compression at Dump Time

When a panic occurs, the operator has an opportunity to override the existing dump compression settings, too

```
*** The dump will be COMPRESSED.***
*** To change this dump type, press any key within 10 seconds.
```

Escape

```
*** Select one of the following dump types
*** by pressing the corresponding key:
C) The dump will be compressed.
S) The dump will be without compression.
N) There will be NO DUMP performed
*** Enter your selection now. C
*** Proceeding with compressed dump.
```

## Student Notes

Also at this time, you have an opportunity to control dump compression.

The options on the slide allow control of whether or not the dump will be compressed. In fact, if you change your mind, you can choose not to perform the dump at all.


© 2005 Hewlett-Packard Development Company, L.P.

## 11–24. SLIDE: Terminating a Dump

### Terminating a Dump

After the dump process begins, the administrator can terminate the dump at any time by pressing Escape

```
*** The dump may be aborted at any time by pressing ESC.
```



## Student Notes

If you change your mind about performing the dump after it has begun, press <esc>. There may be a delay in response to your **<esc>** entry, but this will abort the dump.


© 2005 Hewlett-Packard Development Company, L.P.

## 11–25. SLIDE: Converting the Dump with `crashutil`

### Converting the Dump with `crashutil`

- HP-UX has used a variety of dump formats over the years:
  - COREFILE 10.01
  - COREDIR 10.10, 10.20, 10.30
  - CRASHDIR 11.00
  - PARDIR 11.11 and beyond
- HP's `q4` and `adb` debuggers can read all of these formats
- If you use an older debugger, use `crashutil` to convert your dump to an older format

Convert a dump to the CRASHDIR format in place:

```
# crashutil -v CRASHDIR /var/adm/crash/crash.0/
```

Convert a dump to the CRASHDIR format in a new location:

```
# crashutil -v CRASHDIR /var/adm/crash/crash.0/ /tmp/crash/
```

## Student Notes

The q4 debugger expects the crash dump to be formatted before it can be read.

If the dump has been preserved as version 5, it will be incompatible with older debuggers and analyzers. HP-UX11iv2 versions of the tools will be compatible with compressed and uncompressed dumps. The **`crashutil`** command can be used to convert a version 5 crash dump to an earlier version compatible with the older tools.

On the slide, the first example uses **`crashutil`** to format an 11.0 dump in place. That is, the dump is formatted in the **`/var/adm/crash/crash.x`** directory.

In the second example, **`crashutil`** is used to format the dump, but places the formatted files in an alternate directory, **`/tmp/crash`** in this case.


**© 2005 Hewlett-Packard Development Company, L.P.**

The preserved dump is divided into image files, roughly equivalent to compression units, each described by a subindex file with the same ***.a.b*** extension. The dump version can be grep'ped from the **INDEX** file, and the kernel boot image (for symbol references) is included in **vmunix**. If the version reports as version 5 (PARDIR), compression was at least attempted. With dump compression disabled or unavailable, the **INDEX** file records version 2 (CRASHDIR).




© 2005 Hewlett-Packard Development Company, L.P.

## 11–26. SLIDE: Viewing the Dump Contents with q4

### Viewing the Dump Contents with q4

Crash dump contents can be viewed with the q4 debugger.

Launch q4

```
# cd /var/adm/crash/crash.0
# q4
```

View the dump's console message buffer

```
q4> ex &msgbuf+8 using s | more
```

View the system's uname information

```
q4> load struct utsname from &utsname
q4> print -tx
```

View the panic string

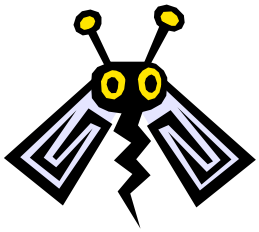
```
q4> ex &panicstr using s
```

View the boot device path

```
q4> ex &boot_string using s
```

Exit the debugger

```
q4> quit
```



## Student Notes

The local system administrator can query a preserved crash dump for basic system information recorded at the time of the crash. The slide shows using the **q4** debugger to retrieve some of this basic information.

When in the directory where the crash resides (typically **/var/adm/crash/crash.#**), **q4** uses the files it finds in the **PWD**. Once in the debugger (**q4>** is the default prompt), the console message buffer is displayed by the **ex**amine command, piped through **more** for paging. This information is exactly what is displayed from a live kernel with the **dmesg** command.

The second 2-step example displays system and HP-UX basic information, such as hostname, HP-UX version and revision, serial number and last recorded date/time stamps.
Most causes of kernel crash dumps will record a short string to the console indicating the cause of the crash. Sifting through the console message buffer from the first example yields this message, or the string itself can be displayed by **ex**amining the **panicstr** symbol.


© 2005 Hewlett-Packard Development Company, L.P.

One other interesting piece of information that can be extracted from the crash dump is the device and kernel that was used at the last boot before the crash. This can be done by **ex**amining the **boot_string** symbol.

The **quit** command exits the debugger. When the dump includes the proper pages and classes of memory, much more information can be extracted.

For further information on **q4** commands, view the **q4** help facility.



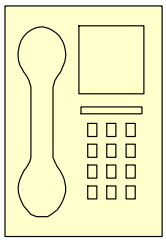

© 2005 Hewlett-Packard Development Company, L.P.

## 11-27. SLIDE: Working with the Response Center

### Working with the Response Center

After the crash dump completes:

1. Follow the instructions in the q4 document
2. Contact the HP Response Center to open a case
3. Email the appropriate q4 files to the Response Center



## Student Notes

After the kernel crash dump is finished, you can do a few things to prepare it for inspection by the Response Center.

The Response Center doesn't need to look at the entire kernel crash dump. Instead, they only need to view a few key files, and these files are usually small enough to be emailed.

A living document for preparing your system for kernel crash dump analysis resides on the ITRC. It is included here, but should you need to use it, it would be advisable to download the latest revision. The document ID never changes, so you can search for Doc ID OZBEKRRC00000611.


© 2005 Hewlett-Packard Development Company, L.P.

**How do I use q4 to pre-process a dump that HP needs to read**
**DocId:** OZBEKBRC00000611 **Updated:** 10/11/04 11:46:00 AM

## PROBLEM

Prior to Opening a Call with HP, I'd like to know the best way to pre-process my crash dump so HP can troubleshoot it.

## RESOLUTION

```
          ITRC DOCUMENT ID: OZBEKBRC00000611

          USING Q4 TO ANALYZE SYSTEM DUMP FILES
              (For HPUX 10.10-11.23 systems)

              [NOTE: These guidelines are for North American
              customers.  Other locales may use a different
              procedure for preprocessing dump files]

==================================================================
                          WHAT IS Q4 ?
==================================================================
If HP-UX crashes, system firmware saves critical O/S state info in
RAM to the swap LVOL or a dump device, then reboots the system
where the O/S copies the dump to a file system directory.

The /usr/contrib/bin/q4 utility is used to investigate the dump.
"q4" usage methods vary depending on the versions of both the O/S
and q4.

Please process the dump using the Q4 steps that follow.  Step 6
helps the user interpret the Q4 results and what to do next.

================================================================
STEP 1  =============  WHERE IS THE DUMP?  ===================
================================================================
1.1 /var/adm/crash/ is the default destination directory
    for dumps.  If a directory is not specified in one of the
    following boot-time parameters, do so now.

    11.X: /etc/rc.config.d/savecrash - SAVECRASH_DIR=
    10.X: /etc/rc.config.d/savecore  - SAVECORE_DIR=

    NOTE:  If "/var:  file system full" occurs during the dump
    save, uncompress or q4 processes in step 4 or 5, use a file
    system that can accommodate 2x the sum of the GZIPed dump
    files.  Update /etc/rc.config.d/save*

1.2 Determine if a recent crash.N (11.X) or core.N (10.X)
    directory exists in the dump directory:
      # ll /var/adm/crash/c*   (dump directory)

NOTE:  "N" increments with each new dump.

1.3 If the system dump is not at the expected path, attempt to
    save it using one of these commands:
      11.X:   # savecrash -vr
      10.X:   # savecore -vr

  If the command results in "invalid dump header", a valid dump
```




© 2005 Hewlett-Packard Development Company, L.P.

```
  does not exist in the swap/dump device.  (Swapping may have
  occured)

1.4 /etc/shutdownlog and /var/adm/crash/c*/INDEX contain a useful
    crash "panic" statement.  If it does not exist:
      # touch /etc/shutdownlog

================================================================
STEP 2  =====  WHAT VERSION OF Q4 IS LOADED?  ================
================================================================
2.1 Determine which version of q4 is loaded:
      # swlist -l product | grep -i q4

    The O/S comes with one of the following versions:
  OS-Core.Q4    B.10.20   HP-UX Crash Dump Debugger for PA-RISC systems
  OS-Core.Q4    B.11.00   HP-UX Crash Dump Debugger for PA-RISC systems

    If one of the following patched versions are listed, proceed
    to step 3:
    10.20                11.00                  11.11
      PHCO_28068(latest)   PHCO_28069(latest)    PHCO_28067(latest)
      PHCO_20261           PHCO_26075            PHCO_25723
                               PHCO_20262

2.2 If q4 is not loaded, or the dump was forced after a system
    hang, swinstall the appropriate patched version.  Loading a
    patched version will NOT cause a system reboot.  Installation
    instructions accompany the patch.

Links to each O/S version of "Patched Q4" (login required):
Link to the 11.22 patch for Q4
Link to the 11.11 patch for Q4
Link to the 11.00 patch for Q4
Link to the 11.20 patch for Q4
  NOTE: the webpage shows the most recent version.

2.3 If q4 is not loaded and a patch version cannot be downloaded,
    load the original version from the CD media...
    Mount the INSTALL media and verify a matching version of
    Q4 is available:
      # swlist -l product -s / | grep Q4

  OS-Core.Q4    B.10.10     HP-UX Crash Dump Debugger for PA-RISC systems
                  ^^^^^ -matches the O/S
    Install it:
      # swinstall -vs / OS-Core.Q4


================================================================
STEP 3  =====  CD TO THE DUMP DIRECTORY  =====================
================================================================
NOTE: csh (c-shell) will cause errors with q4. Use sh-posix.
IMPORTANT!  cd to the dump directory and then skip to step 5.
Example:
      # cd /var/adm/crash/crash.0   OR   /var/adm/crash/core.0
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
================================================================
STEP 4  =========  IF USING UNPATCHED Q4  ====================
================================================================
4.1  If vmunix is still compressed (vmunix.gz), unpack it:
      # /usr/contrib/bin/gunzip vmunix.gz

4.2 Prepare the dump tools ...
    For 10.20 through 11.11, type:
      # /usr/contrib/bin/q4prep -p

   For 11.20 and newer, type:
      # /usr/contrib/Q4/bin/q4prep -p

   For 10.10, type the following commands:
      # uncompress /usr/contrib/lib/Q4Lib.tar.Z
        (ignore the error if this was done previously)

      # tar -xf /usr/contrib/lib/Q4Lib.tar
        (output placed in the current directory)

      # cp q4lib/sample.q4rc.pl ~/.q4rc.pl
                                \ \        \
        Note the use of a tilde and letter "l" (not digit 1)

      # /usr/contrib/bin/q4pxdb vmunix
        (ignore a "already preprocessed" complaint)

4.3 Type:
      # q4 -p  .     (note the "dot" at the end of the command)
    Then:
      q4> trace event 0 > trace.out
      q4> include analyze.pl
            (last char. is letter "l", not the digit 1)
      q4> run Analyze AU >> ana.out
            (ctrl-c allows interrupt)
      q4> include whathappened.pl

      q4> run WhatHappened -HANG > what.out
            (ctrl-c allows interrupt)
      q4> exit

    If a "file system full" message occurs, follow NOTE in step
    1.1 and then redo the failed command and proceed.

    Skip to step 6


================================================================
STEP 5  ============ IF USING PATCHED Q4 =====================
================================================================
5.1 Type:
      # . /usr/contrib/Q4/bin/set_env
    Note the 'dot' at the beginning of the command.

5.2 Type:
      # ll vmunix.gz
```




© 2005 Hewlett-Packard Development Company, L.P.

```
    If the file is not found, go to step 5.3
    If the file is found, type:
      # gunzip vmunix.gz

5.3 Type:
      # /usr/contrib/Q4/bin/q4pxdb vmunix
        (Ignore "unnecessary" message)

      # /usr/contrib/Q4/bin/q4 -p .
         (note the "dot" at the end of the command)

If a "file system full" message occurs, follow NOTE in step 1.1
and redo the failed command and proceed.

5.4 At the q4> prompt, type these 'run' commands:
      q4> trace event 0 > trace.out
      q4> run Analyze AU > ana.out
      q4> run WhatHappened -HANG > what.out
    NOTE:  The last two commands may run for several minutes.
           CTRL-c can interrupt them.

5.5 Type:
      q4> exit


================================================================
STEP 6  ===========  REVIEW AND SEND DATA  ====================
================================================================
6.1 HPUX uses the acronym HPMC (High Priority Machine Check) to
    denote a hardware failure.  Determine if the crash was due to
    an HPMC ...

    Type:
      # grep HPMC ana.out trace.out

    If either of the following lines appear, open a hardware
    repair case for the system:

      "crash event was an HPMC"
      "Crash Event 0 (HPMC, struct crash_event_table_struct..."

    The OnlineDiag software bundle captures HPMC chassis codes in
    /var/tombstones/ts* files.  If available, email the 'ts' file
    created after the "dumptime" listed in the dump INDEX file
    (eg. ts99).

    If an HPMC did not occur, proceed to the next step.

6.2 Skip the remainder of this step if the following lines are not
    found in ana.out:

      MC/ServiceGuard: Unable to maintain contact with cmcld daemon.
      Performing TOC to ensure data integrity.

    If these statements occur, determine the NODE_TIMEOUT value:

      # cmviewconf | grep "node timeout"
```


**© 2005 Hewlett-Packard Development Company, L.P.**

If the value returned is 2 seconds, it's a good bet the value is too low, causing the crash. When the kernel is too busy to send a Serviceguard heartbeat packet to the other nodes within the NODE_TIMEOUT period, the other nodes reformed a cluster and 'orphaned' this node - causing a TOC/reboot.

The fix is to update the cluster ASCII file - NODE_TIMEOUT to 8 seconds and with the cluster down, 'cmapplyconf -C' the file. See document UXSGLVKBAN00000010 for details of this issue. If this is the case - stop here.

6.3 Generate a patch list:

```
# /usr/sbin/swlist -l product PH* > patchlist.out
```

6.4 Zip and Email the following files:

**ana.out**
**patchlist.out**
**trace.out**
**what.out**
**/etc/shutdownlog**
**/var/tombstones/ts99**
(if an HPMC was detected)
**/var/adm/syslog/OLDsyslog.log**
(if the dump was due to a hang)

To: **hpcu@atl.hp.com**
Cc: **support@atl.hp.com**
Subject:

EMAIL RECOMMENDATIONS:
- DO NOT send this data to the engineer's personal address.
- Send files as attachments when possible.
- Send fresh messages, not replies.
- Mail size must be < 2MB. Anything greater will be denied.

After emailing the data, please notify HP that dump email has been sent for action (via callback or ITRC note).

If e-mail or ftp is not available, tar up the dump (relative pathing please) and send the tape to:

Barry Britt
M/S ALF01-3T20
5555 Windward Parkway
Alpharetta
GA 30004

Acceptable tape formats:
DDS1 - DDS3
CD-media (containing dump files)




© 2005 Hewlett-Packard Development Company, L.P.

## 11–28. SLIDE: Removing the Dump

### Removing the Dump

When you are done reviewing the dump,

Copy the dump to tape

```
# tar -cv /var/adm/crash/crash.0
```

Remove the dump directory

```
# rm -rf /var/adm/crash/crash.0
```

## Student Notes

After the dump has been analyzed, unless there is a compelling reason to keep it on your system, it should be removed.

If keeping the dump is desired, it may be archived to tape, freeing up disk space.


© 2005 Hewlett-Packard Development Company, L.P.

## 11–29. SLIDE: Review Module Objectives

### Review Module Objectives

Your should now be able to:

- Describe the purpose of kernel crash dumps
- Configure dump devices
- Configure selective crash dumps
- Configure compressed crash dumps
- Configure `savecrash` functionality
- Describe the crash dump directory structure
- Uncompress and view a crash dump via `crashutil` and `q4`



## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 11–30. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What is a system panic?

2. What is a kernel crash dump?

3. Which command reports configured-memory classes and dump declarations on a system?

4. What startup scripts are used for system dump configuration during boot? List the startup configuration files.


© 2005 Hewlett-Packard Development Company, L.P.

# 11–31. LAB: Kernel Crash Dump Configuration

## Purpose

This lab is designed to give you hands on experience configuring memory classes and dump devices.

## Directions

Read and answer the following questions.

1. Use the **crashconf** command to view the current system crash dump configurations.

```
# crashconf -v
```

2. Use the **crashconf** command to include the memory classes "***KCODE***" and "***USERPG***" within the system crash dump configuration.

```
# crashconf –i kcode –i userpg
```

3. Use the **crashconf** command to exclude the "***KDDATA***" memory class from the system crash dump configuration.

```
# crashconf –e kddata
```

4. Create a 128-MB logical volume to use as a dump device. Afterwards, dynamically declare that logical volume as additional dump space using the **crashconf** command.

```
# lvcreate -L 128 -n dumplvol -C y -r n /dev/vg00
# crashconf /dev/vg00/dumplvol
```




© 2005 Hewlett-Packard Development Company, L.P.

5. Configure your system so that the above system crash dump configurations occur automatically during boot. Note: On Integrity systems, the following **`CRASH_INCLUDED_PAGES`** and **`CRASH_EXCLUDED_PAGES`** entries do not apply. Instead, the administrator would issue **`crashconf -e`** ***`class`*** or **`crashconf -i`** ***`class`*** commands. To make the changes permanent, issue the **`-t`** option in the above commands.

```
# vi /etc/rc.config.d/crashconf
  CRASH_INCLUDED_PAGES="kcode userpg"
  CRASH_EXCLUDED_PAGES="kddata"
  CRASHCONF_READ_FSTAB=1
```

**NOTE**: For purposes of this lab, we are excluding kernel dynamic data (**`kddata`**), which provides an example of how to exclude a memory class. However, in real life, excluding "kddata" would render a crash dump virtually useless.

```
# vi /etc/fstab
  /dev/vg00/dumplvol /       dump  defaults   0 0
```


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 12 — Troubleshooting Patch Problems

## Objectives

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

- List four new patch management attributes introduced with HP-UX 11.00.
- List three additional patch management tools that can be added to an HP-UX 11.x system through a patch.
- Describe the procedure for committing a patch.
- Describe options when installing patches.
- Describe the use and purpose of **`security_check_patch`** tool.
- Describe the purpose of various SD-UX logfiles.
- Understand the purpose of the Installed Products Database (IPD).




© 2005 Hewlett-Packard Development Company, L.P.

## 12–1. SLIDE: Module Objectives

### Module Objectives

By the end of the module, you should be able to:

- List four new patch management attributes introduced with HP-UX 11.00.
- List three additional patch management tools that can be added to an HP-UX 11.x system through a patch.
- Describe the procedure for committing a patch.
- Describe options when installing patches.
- Describe the use and purpose of `security_check_patch` tool.
- Describe the purpose of various SD-UX logfiles.
- Understand the purpose of the Installed Products Database (IPD).

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 12–2. SLIDE: Managing Patches

### Managing Patches

- Beginning with HP-UX 11.0, management of patches is accomplished through Software Distributor, *not* via scripts.
- Patch information is kept in the Installed Products Database (IPD).
- New attributes related to patch management at HP-UX 11.00:
  - `is_patch`
  - `patch_state`
  - `superseded_by`
  - `supersedes`



## Student Notes

Software Distributor (SD) is much more "patch aware" at HP-UX 11.00. At HP-UX 10.20, patch management was done through SD scripts: For example:

- **`checkinstall`**, **`preinstall`**, and **`postinstall`** were used to manage patch installation.
- **`checkremove`**, **`preremove`**, and **`postremove`** were used to manage patch removal.

As of HP-UX 11.00, SD does real patch management; it is not done with scripts. SD can now:

- Differentiate products that are software patches from those products that are not software patches.
- Commit existing patches to the system, freeing up disk space used by "saved" files. By committing a patch, an agreement is made that the patch will *never* be removed; therefore, the old saved files are removed.
- For new patches, SD can list the existing patch or patches the new patch supersedes.
- List the patches by which an older patch has been *superseded*.


© 2005 Hewlett-Packard Development Company, L.P.

## 12–3. SLIDE: New Attribute: `is_patch`

### New Attribute: `is_patch`

- A new attribute included when defining an SD product at HP-UX 11.x (in the product specification file): `is_patch`
- If `is_patch` is set to true, SD knows it is a patch.

```
# swlist -l product -a is_patch

  Accounting              false
  AudioSubsystem          false
  CDE                     false
  CPS                     false
  Curses-Color            false
  DCE-Core                false
  ...
  PHCO_13316              true
  PHCO_13363              true
  PHCO_14177              true
  PHCO_15768              true
  PHCO_17321              true
  ...
```

## Student Notes

All information about an SD product is contained in a product specification file (PSF). A new PSF field, called **`is_patch`**, allows SD to differentiate an SD patch product from other SD products.

To list which SD products are patches and which are not, enter:

```
# swlist -l product -a is_patch
```


© 2005 Hewlett-Packard Development Company, L.P.

## 12–4. SLIDE: New Attribute: `patch_state`

### New Attribute: `patch_state`

The `patch_state` attribute can be:

- `applied`
- `committed`
- `superseded`

```
# swlist -l patch -a patch_state \*.\*,c=patch

  # PHCO_13316.CORE-SHLIBS        applied
  # PHCO_13363.UX-CORE            superseded
  # PHCO_14177.SYS-ADMIN          committed
  # PHCO_14177.UX-CORE            applied
  # PHCO_15768.C-MIN              superseded
  # PHCO_15768.CORE-SHLIBS        superseded
  # PHCO_15768.PROG-AUX           superseded
  # PHCO_15768.PROG-MIN           superseded
  # PHCO_17321.UX-CORE            applied
  ...
```

Notice that PHCO_13363 has been superseded.

Q: Which patch superseded PHCO_13363?

## Student Notes

The new **`patch_state`** attribute lists the status of specified patches.

The possible values for the **`patch_state`** variable are:

| | |
|---|---|
| APPLIED | In this state, the patch is installed, is a current patch (i.e. it has NOT been superseded), and saved files are available if the patch needs to be rolled off the system. |
| SUPERSEDED | In this state, the patch has been made “obsolete” by a newer patch. The patch cannot be removed from the system until the newer patch is first removed. Saved files are still available for patches in the SUPERSEDED state. |
| COMMITTED | In this state, the saved files for the patch have been removed. The patch *cannot* be removed from the system, because there is no known previous set of files to restore. |

More information about the patch_state attribute can be found in `sd(5)`




© 2005 Hewlett-Packard Development Company, L.P.

## 12–5. SLIDE: New Attributes: `supersedes`, `superseded_by`

### New Attributes: `supersedes`, `superseded_by`

- These two attributes now make it possible to view which patches supersede which other patches, and vice-versa.
- To answer the question, "Which patch superseded PHCO_13363?" the `-a superseded_by` can be used.

```
# swlist -l patch -a superseded_by   PHCO_13363

      # PHCO_13363
      # PHCO_13363.UX-CORE     PHCO_17321.UX-CORE,l=/,r=1.0,a=HPUX_B.11.00_32/64,
                v=HP,fa=HP-UX_B.11.00_32/64
```

- The patches that PHCO_17321 supersedes can be viewed with the `-a supersedes` option.

```
# swlist -l patch -a supersedes   PHCO_17321

# PHCO_17321.UX-CORE             PHCO_13363.UX-CORE,fr=*
```

## Student Notes

Beginning with HP-UX 11.0, the attributes, **`supersedes`** and **`superseded_by`**, display patch history information for a given patch.

- To list which newer patch has superseded an older patch, the command is:

  `# swlist -l patch -a superseded_by <old_patch_name>`

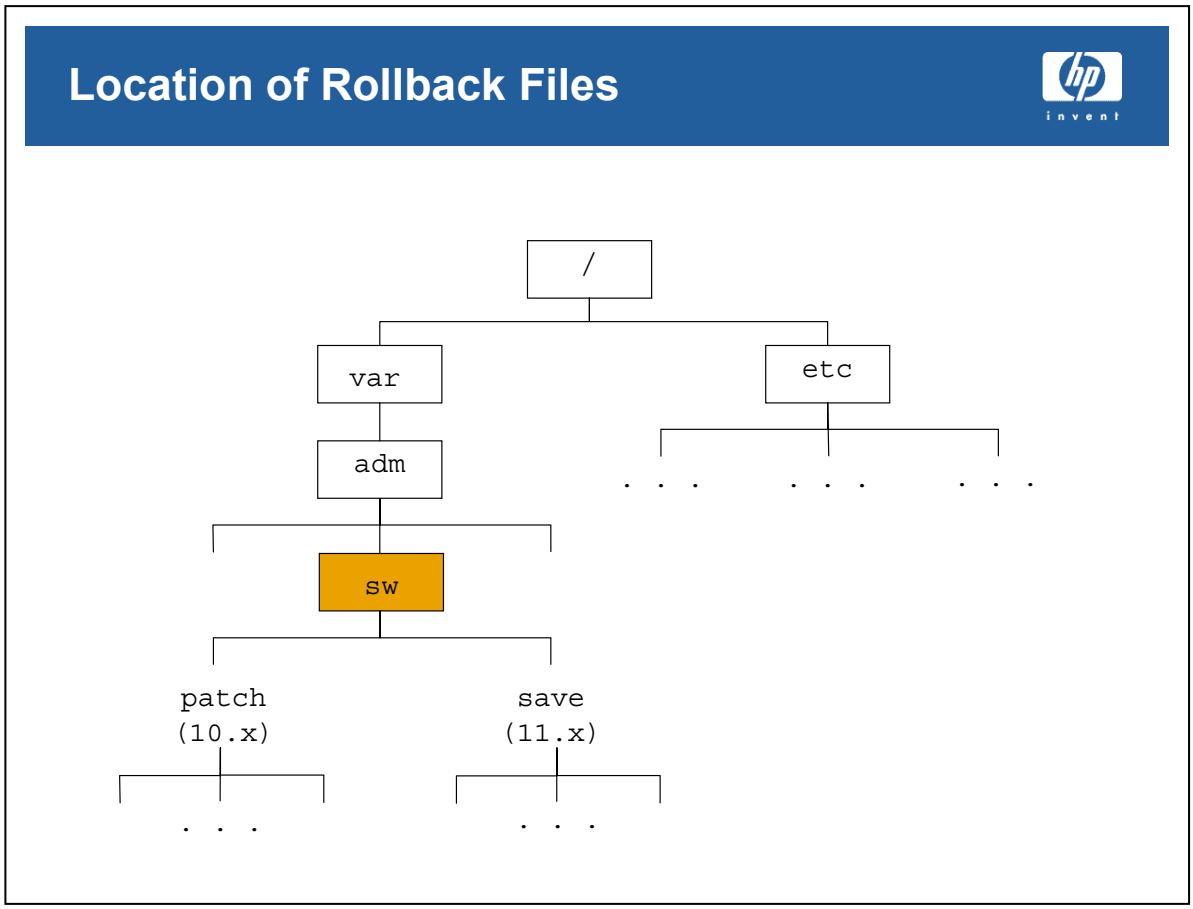
- To list all the older patches a new patch supersedes, enter:

  `# swlist -l patch -a supersedes <new_patch_name>`

More information about the **`supersedes`** and **`superseded_by`** attributes can be found in `sd(5)`.


© 2005 Hewlett-Packard Development Company, L.P.

## 12–6. SLIDE: Location of Rollback Files




### Student Notes

The name of the directory in which the “roll-back” files (the back-out files) are stored changes from HP-UX 10.20 to HP-UX 11.00.

As of HP-UX 11.00, the directory that contains "roll-back" files is **`/var/adm/sw/save`**.




© 2005 Hewlett-Packard Development Company, L.P.

## 12–7. SLIDE: Committing Patches

### Committing Patches

- Patches can be committed with the `swmodify` command.

```
swmodify -x patch_commit=true PHCO_13363.\*
```

- It is possible to commit both superseded and applied patches.
- You must be `root` to commit patches.
- Under normal circumstances, patches can be removed, because the "back out" files have been saved by `swinstall` in the `/var/adm/sw/save` directory.
- If you "commit" a patch, you have removed the "back out" files and now, cannot remove the patch that you just committed.

## Student Notes

When a patch is installed on HP-UX, the original files are saved before the patch overwrites them. Subsequently, if the patch needs to be removed, the system can be returned to the "pre-patch" state (before the patch was installed).

The location of the saved files is **`/var/adm/sw/save`**.

Many times, because of the quantity of patches and size of the saved files, the amount of disk space needed to store all the saved files becomes very large. If it is known that a certain patch will not be removed, then the saved files associated with that patch can be removed by committing the patch.

Committing a patch allows the saved files to be removed for that patch, but it requires the system administrator to *commit* to always keeping that patch on the system.

To commit a specific patch, and thereby free up disk space previously used by saved files, use the following command:

```
#  swmodify -x patch_commit=true  <patch_name>
```


© 2005 Hewlett-Packard Development Company, L.P.

## 12–8. SLIDE: Managing Patches with `swinstall`

### Managing Patches with `swinstall`

- Products and patches can be installed simultaneously if they are in the same depot:

```
swinstall -s <SourceDepot> -x autoreboot=true \
  -x autoselect_patches=true  <SoftwareSelections>
```

- Patches can be committed at install time using swinstall:

```
swinstall -s <SourceDepot> -x patch_commit=true  \
  <Patch_Selections>
```



## Student Notes

Two new features of the **`swinstall`** command introduced with HP-UX 11.00 are:

- Automatic selection and installation of related-patches when a product is installed. Patches must reside in the same depot as the product in order for the automatic selection feature to succeed.
- The administrator can perform patch committals during the installation of the patch, as opposed to using the **`swmodify`** command after the patch has been installed.

By default, on HP-UX 11.00, every time an SD product is installed, the depot containing that product is searched for any related patches corresponding to the product. If a related patch is found, the patch is automatically applied. If two patches for the product exist, only the most current patch is applied.

To disable automatic selection of patches, enter:

```
# swinstall –s <depot> -x autoselect_patches=false  \
<SoftwareSelection>
```


© 2005 Hewlett-Packard Development Company, L.P.

By default, patches are *not* committed during the installation of the patch. This means that saved files are created in case the patch needs to be rolled-back. If saved files are not necessary, commit the patch during the installation as follows:

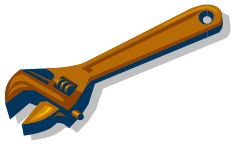
```
# swinstall –s <depot> -x patch_commit=true <PatchSelection>
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 12–9. SLIDE: Additional Patch Management Tools

### Additional Patch Management Tools

- The HP-UX 11i v1 patch `PHCO_27780` and HP-UX 11.0 patch `PHCO_27779` provide three additional patch management tools:
  - `cleanup`
  - `show_patches`
  - `check_patches`





## Student Notes

Auxiliary patch management tools are available. For HP-UX 11i v1, install **`PHCO_27780`**. For HP-UX 11.0, install **`PHCO_27779`**. These patches install 3 additional tools that perform a variety of different functions that are useful when managing a large number of patches.

The additional tools in **`PHCO_27780`** include:

| | |
|---|---|
| **`cleanup`** | The **`cleanup`** tool removes HP-UX 10.x patches from an HP-UX 11.x system, and removes superseded patches from existing depots. |
| **`show_patches`** | The **`show_patches`** tool displays all active or superseded patches on the system. |
| **`check_patches`** | The **`check_patches`** tool checks for common problems and issues related to patch management. This tool incorporates several other tools in one package. |


© 2005 Hewlett-Packard Development Company, L.P.

## 12–10. SLIDE: The `cleanup` Tool

### The `cleanup` Tool

The `cleanup` tool performs two functions:

- It cleans up any remaining 10.x patches after upgrading a system to HP-UX 11.00. It should be executed after every upgrade from HP-UX 10.x to HP-UX 11.00.

```
# cleanup -i
```

- It also cleans up any superseded patches in the same depot as the patch that supersedes it. It should be executed every time patches are added to a depot.

```
# cleanup -d <depot>
```

## Student Notes

The **`cleanup`** tool can perform two functions.

- First, it can be used to remove HP-UX 10.x patches from the installed product database after updating to HP-UX 11.00. The syntax for performing this function is:

  ```
  # cleanup -i
  ```

- Second, it can be used to remove patches from a software depot, if the patches have been superseded by patches available in the same depot. The syntax for performing this function is:

  ```
  # cleanup -d <depot_name>
  ```


© 2005 Hewlett-Packard Development Company, L.P.

## 12–11. SLIDE: The `show_patches` Tool

### The `show_patches` Tool

The `show_patches` tool displays all the active or superseded patches on the system.

- To list all the active patches on the system:

```
# show_patches -a

Active                  Patch
Patch                   Description
----------              ----------------------------------------
PHCO_12555              ioinit patch
PHCO_12577              uucp(1) - fixes multiple hop test failure
PHCO_13205              dd(1) patch for block/unblock conversion
. . .
```

- To list all the superseded patches on the system:

```
# show_patches -s

 Superseded              Patch
 Patch                   Description
 ----------              ----------------------------------------
 PHCO_13363              POSIX:sh patch
 PHCO_13753              HP-UX Patch Tools and White Paper
 PHCO_14084              csh(1) patch
 . . .
```

## Student Notes

Either active or superseded patches can be committed. It is not always clear with the **`swlist`** command if a committed patch is active or superseded.

The **`show_patches`** tool displays all the *active* patches (committed and non-committed) or all the *superseded* patches (committed and non-committed). The output from the **`show_patches`** command is in an easy-to-read, formatted display.

The above slide shows two examples of the **`show_patches`** command: one example displaying all active patches (**`-a`** option), and one example displaying all superseded patches (**`-s`** option).




© 2005 Hewlett-Packard Development Company, L.P.

## 12–12. SLIDE: The `check_patches` Tool

### The `check_patches` Tool

- The `check_patches` utility checks for common problems and issues related to patches on HP-UX 11.x. The utility checks for:
  - Patches missing the SD-UX patch attributes (`-i` option)
  - Patch object modules missing from archive libraries (`-o` option)
  - Patch file sets not in the configured state (`-s` option)
  - Patch file sets that fail `swverify` (`-v` option)
- By default, `check_patches` performs all four checks.

## Student Notes

The **`check_patches`** tool verifies the integrity of the SD components related to patch management. The tool checks for patches missing the SD patch attributes, patch routines missing from archive libraries, patch filesets not in the configured state, and patch filesets that fail **`swverify`**.

### Example

The command below checks for the existence of the **`is_patch`** attribute. If the attribute does not exist, or if the value of the attribute is FALSE, then SD does *not* treat the software as a patch, which can lead to unexpected results.

```
# check_patches -i
Obtaining information on installed patches
Checking for invalid patches
RESULT: Problems found, review /tmp/check_patches.report for
details.
```

CAUTION: There are known issues with earlier versions of the **`check_patches`** tool. Be sure you are using version 1.30 or later.


**© 2005 Hewlett-Packard Development Company, L.P.**

To check:

```
# what /usr/contrib/bin/check_patches
```

Following is a sample output from the **check_patches** tool

```
# check_patches
Obtaining information on installed patches
Checking installed version of /usr/sbin/swconfig
Checking for invalid patches
Checking object module checksums for active patch fileset   395
of   480Unknown type  d
Checking object module checksums for active patch fileset   398
of   480Unknown type  d
Checking object module checksums for active patch fileset   402
of   480Unknown type  d
Checking object module checksums for active patch fileset   478
of   480Unknown type  d
Checking object module checksums for active patch fileset   480
of   480
Checking patch filesets for active patch   238 of   238
Checking state for patch fileset   482 of   482
Checking patch_state for patch fileset   482 of   482
Running swverify on all patch filesets, this may take several
minutes
RESULT: Problems found, review /tmp/check_patches.report for
details.

# more /tmp/check_patches.report
                       Report from check_patches utility
                                 Revision: 1.3
                               01/16/04 13:11:16
                                System: r206c41

==================== swconfig(1M) version ========================

* Determining if the version of the swconfig(1m) command present on
* the system can fully support enforced dependencies on patches.

ERROR:   The version of /usr/sbin/swconfig on the system does
         not provide full support for automatically enforcing
         software dependencies on HP-UX patches.
       * It is recommended that patch PHCO_23578 (or replacement)
         or SW-DIST product version B.11.11.0109 (or higher) be
         installed on the system to provide this functionality.
       * SW-DIST version B.11.11.0109 is first included on the
         HP-UX 11i OE or Application media dated September 2001.
====================== Invalid Patches ==============================

* Determining if any HP-UX 11.11 patches are missing the SD-UX patch
* attributes required for SD-UX to handle the patches properly.  If the
* patch attributes are missing SD-UX does not treat the software as
```




© 2005 Hewlett-Packard Development Company, L.P.

```
* patches, which can lead to unexpected results.

No invalid patches are present.

=================== Object Module Checksums =========================

* Determining if the checksum values of the object modules delivered in
* the currently active patches match the checksum values of the modules
* in the appropriate library.

All object modules checksum values match.

===================== Missing Patch Filesets ========================

* Determining if the active patches have all their applicable filesets
* installed.

All applicable patch filesets for the active patches are present.

======================== Patch Fileset States ========================

* Determining if all patch filesets are in the configured state.

NOTE:    One or more patch filesets are in the installed state, but
         they do NOT include a configure script.  Since these patch
         filesets do not include a configure script, there is no
         difference between the installed or configured state.

         The patch filesets in this condition are:

            PHKL_23246.CORE2-KRN
            PHKL_25871.CORE2-KRN

         If you prefer to have these patch filesets in the configured
         state as opposed to the installed state, they can be
         configured with swconfig(1M), for example:

              swconfig PHKL_25871.CORE2-KRN

         or to configure all the filesets in a patch:

              swconfig PHKL_25871


========================== Patch_States ============================

* Determining if all patch_state attributes are correct.

All patch filesets are in the correct patch_state.

==================== Swverify on Patch Filesets =====================

The swverify(1M) command found no problems with the patch filesets.

=====================================================================

# swconfig PHKL_23246
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
=======  01/16/04 13:16:00 PST  BEGIN swconfig SESSION
         (non-interactive) (jobid=r206c41-0016)

       * Session started for user "root@r206c41".

       * Beginning Selection
       * Target connection succeeded for "r206c41:/".
       * Software selections:
             PHKL_23246.CORE2-KRN,l=/,r=1.0,a=HP-
UX_B.11.11_32/64,v=HP,fr=1.0,fa
=HP-UX_B.11.11_64
       * Selection succeeded.
       * Beginning Analysis
       * Session selections have been saved in the file
         "/root/.sw/sessions/swconfig.last".
       * "r206c41:/":  1 filesets were determined to be skipped in the
         analysis phase.
       * Analysis succeeded.

NOTE:    More information may be found in the agent logfile using the
         command "swjob -a log r206c41-0016 @ r206c41:/".

=======  01/16/04 13:16:04 PST  END swconfig SESSION (non-interactive)
         (jobid=r206c41-0016)
```

**# swconfig PHKL_25871**

```
=======  01/16/04 13:16:13 PST  BEGIN swconfig SESSION
         (non-interactive) (jobid=r206c41-0017)

       * Session started for user "root@r206c41".
       * Beginning Selection
       * Target connection succeeded for "r206c41:/".
       * Software selections:
             PHKL_25871.CORE2-KRN,l=/,r=1.0,a=HP-
UX_B.11.11_32/64,v=HP,fr=1.0,fa
=HP-UX_B.11.11_64
       * Selection succeeded.
       * Beginning Analysis
       * Session selections have been saved in the file
         "/root/.sw/sessions/swconfig.last".
       * "r206c41:/":  1 filesets were determined to be skipped in the
         analysis phase.
       * Analysis succeeded.

NOTE:    More information may be found in the agent logfile using the
         command "swjob -a log r206c41-0017 @ r206c41:/".

=======  01/16/04 13:16:16 PST  END swconfig SESSION (non-interactive)
         (jobid=r206c41-0017)
```

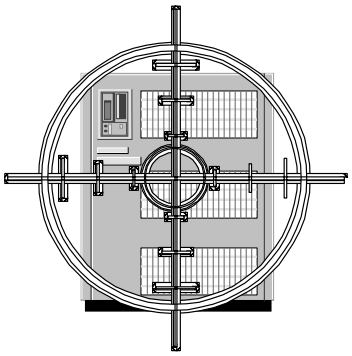



© 2005 Hewlett-Packard Development Company, L.P.

## 12–13. SLIDE: The `security_patch_check` Tool

### The `security_patch_check` Tool

`security_patch_check`



- Downloads the latest data regarding security patches
  - Information from Security bulletins released by HP
- Compares with patches currently installed
- Reports which security patches are missing
- Alerts if any recalled patches are present on system
- Works thru firewalls via a web-proxy

## Student Notes

The **`security_patch_check`** tool regularly downloads security patch information and availability from the HP Response Center and compares it with your installed patches.

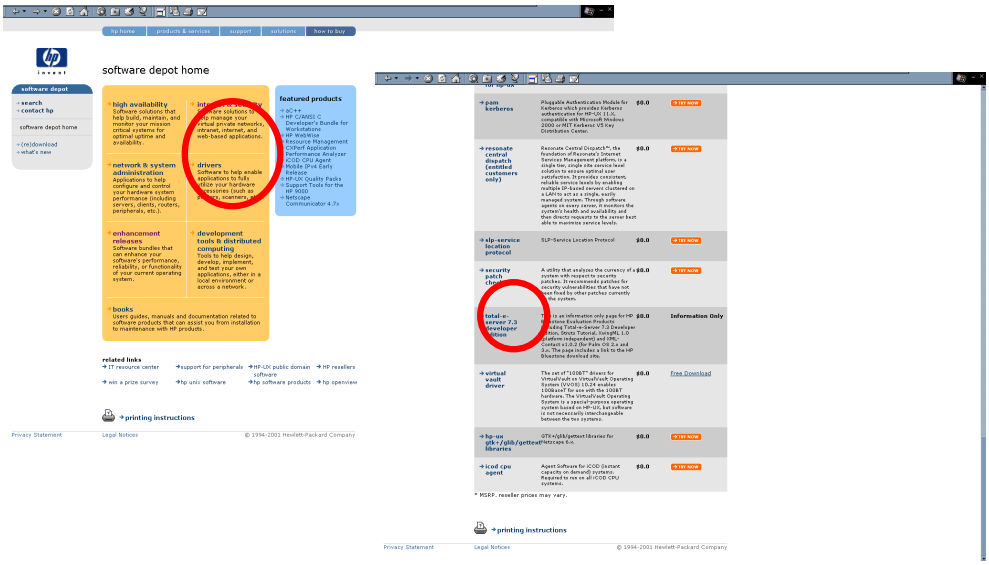
Through a series of routines, the tool reports which security patches are missing, and it alerts if any recalled security patches are presents on your system.

The Perl routine used allows **`security_patch_check`** to get it's information through firewalls.


© 2005 Hewlett-Packard Development Company, L.P.

## 12–14. SLIDE: security_patch_check

security_patch_check




## Student Notes

**security_patch_check** is a Perl script that runs on HP-UX 11.X systems. **security_patch_check** performs an analysis of the filesets and patches installed on an HP-UX machine, and generates a listing (report) of recommended security patches. In order to determine which patches are missing from a system, **security_patch_check** must have access to a listing, or catalog, of security-related patches.

Since new security patches can be released at any time, **security_patch_check** depends on a patch catalog stored on an HP server. This catalog is updated nightly. To help automate the process of checking for security patches missing from a system, **security_patch_check** is able to download the most recently generated catalog from an HP FTP site. It does this by using the LWP Perl module. The LWP module can operate through a firewall. Refer to **security_patch_check(1M)** for more information.

Once **security_patch_check** has access to a security patch catalog, it will create a list of the patches, which are both applicable and not yet installed. Note that although the security patch catalog contains the most recent and highest rated patches, **security_patch_check** will recommend a patch only if it addresses a security problem not already addressed by an installed patch.




© 2005 Hewlett-Packard Development Company, L.P.

Installing the patches that **`security_patch_check`** recommends addresses ONLY those vulnerabilities that are resolved by patches. The security bulletins and advisories from HP sometimes contain other actions (manual steps) to resolve vulnerabilities. Thus, each advisory from the archive of previously released security advisories, which applies to the platform being analyzed, must be examined to determine if any manual steps are required. This archive can be obtained from
`http://itrc.hp.com/cki/bin/doc.pl/screen=ckiSecurityBulletin`


**© 2005 Hewlett-Packard Development Company, L.P.**

## 12–15. SLIDE: Setting Up `security_patch_check`

### Setting Up `security_patch_check`

- Download the latest Perl SD-UX depot:
  - `http://devresource.hp.com/OpenSource/Tools/perl/perl.html`
- Download `security_patch_check` SD-UX depot
  - `http://software.hp.com`
  - Navigate via Internet & Security + security patch check
  - Trial software requires Customers details to download, but is free to download
- Install `security_patch_check`
  - `# swinstall -s $PWD/B6834AA_11_11.depot \*`
- Install Perl
  - `# swinstall -s $PWD/Perl-5.005_03-11.11.tar \*`
- Set up FTP proxy access (if using a proxy server to access the Internet)
  - `# export ftp_proxy=<myproxy>:<proxy>`
  - `# export ftp_proxy=web-proxy.uksr.hp.com:8088`
- Download `security patch catalog` to current working directory
  - Anonymous ftp
    `ftp://ftp.itrc.hp.com/export/patches/security_catalog`
    OR
  - `/opt/sec_mgmt/spc/bin/security_patch_check -r`

## Student Notes

Before using **`security_patch_check,`** it is necessary to ensure the **`Perl 5.005`** is a minimum version of Perl installed. Previous versions of Perl had, in themselves, security vulnerabilities thus, potentially compromising the integrity of **`security_patch_check`** itself.

It may be appropriate to familiarize yourself with the background and working of **`security_patch_check`**. There is online documentation as well as man pages.

README file: **`/opt/sec_mgmt/spc/README`**

Man pages: ensure **`/opt/sec_mgmt/spc/share/man`** is in your MANPATH.

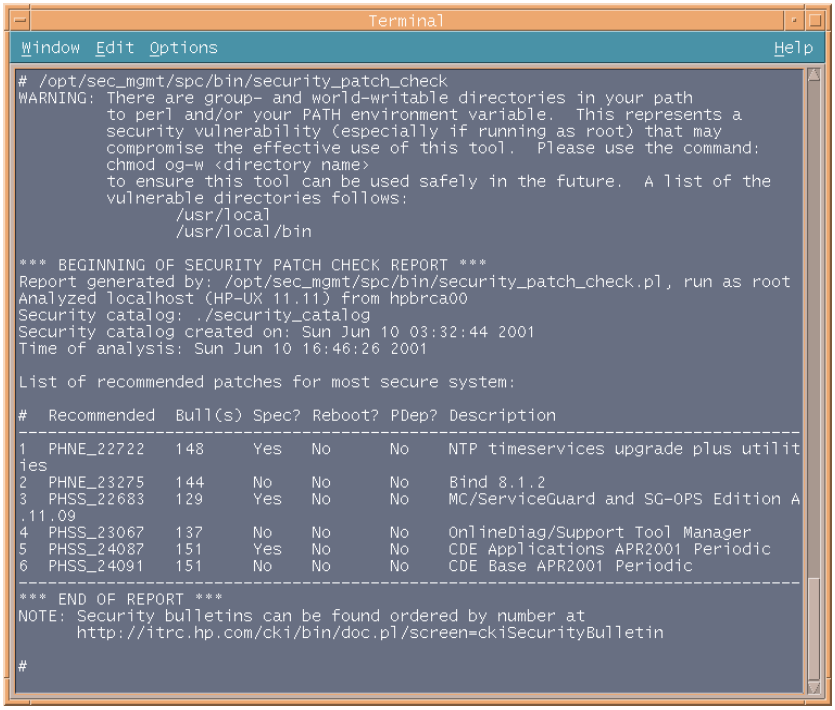
Should you decide to download the security patch catalog directly from the ITRC then ensure you download it before EVERY invocation of **`security_patch_check`**. The security patch catalog is updated nightly, hence your current copy may become out of date quickly. Having **`security_patch_check`** download the catalog directly avoids this issue.




© 2005 Hewlett-Packard Development Company, L.P.

## 12–16. SLIDE: Using `security_patch_check`

### Using `security_patch_check`




## Student Notes

Once the security patch catalog is in place, you can commence running the tool. The report generated reflects HP's recommended patches based on your current loaded software and patches compared against your current security patch catalog. This list of patches should be downloaded from the ITRC and installed as soon as possible. As a depot administrator, you may also consider adding these patches to your patch depot, making them available to other machines on the network.

Following the recommendations of **`security_patch_check`** will result in a system that is up-to-date with HP's security patches. However, the system will not be fully compliant with HP's security bulletins and advisories, and there are additional steps that must be taken to secure the system.

Many security advisories require manual actions on a system. Since some advisories or bulletins contain no patches and others contain both patches and manual actions, these advisories must be read and appropriate action taken. To access an archive of HP-UX security advisories, you must have an account on ITRC. Go to:
`http://itrc.hp.com/cki/bin/doc.pl/screen=ckiSecurityBulletin`


**© 2005 Hewlett-Packard Development Company, L.P.**

## 12–17. SLIDE: Working with SD-UX Logfiles

### Working with the SD-UX Logfiles

Most Software Distributor commands log error messages, summary information about the session, and operation details to a command-specific logfile located (by default) in `/var/adm/sw/<command>.log`.

Examples:

For `swinstall` issues, consult `/var/adm/sw/swinstall.log`

For swagent daemon activity, consult `/var/adm/sw/swagentd.log`

For `swmodify`, consult `/var/adm/sw/swmodify.log`

For `swverify`, consult `/var/adm/sw/swverify.log`

For `swcopy`, consult `/var/adm/sw/swcopy.log`

For `swremove`, consult `/var/adm/sw/swremove.log`



## Student Notes

Most SD-UX commands will log error and status messages to specific command-related logfiles. Checking the directory listing under **`/var/adm/sw`** yields the following logfiles:

```
/var/adm/sw/swagent.log
/var/adm/sw/swagentd.log
/var/adm/sw/swconfig.log
/var/adm/sw/swinstalllog
/var/adm/sw/swmodify.log
/var/adm/sw/swswreg.log
/var/adm/sw/swremove.log
/var/adm/sw/swagent.log
```

The SD-UX commands log messages to **`/var/adm/sw/sw<task>.log`**.

A **`swagent`** process performs the actual individual commands (on the designated target) that generate the logfiles above. However, looking through the **`swagent.log`** file can also be helpful in troubleshooting issues.
Further, if networking issues are suspected during the installation of products/patches, inspection of the **`swagentd.log`** file might be warranted


© 2005 Hewlett-Packard Development Company, L.P.

## 12–18. SLIDE: The Installed Products Database (IPD)

### The Installed Products Database

The Installed Products Database (IPD):

- Describes the products and patches that are installed on a system
- Should not be manipulated at the file level
- Should only be modified via SD-UX commands

**What if the IPD becomes corrupt?**

## Student Notes

The Installed Products Database (IPD) tracks all products and patches installed on your system. Further, the IPD tracks states of patches and products.

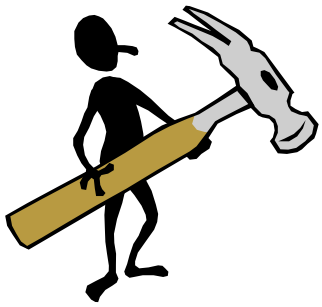
While the IPD is a flat file (`/var/adm/sw/products/INDEX`), it should not be edited directly. Instead, use the SD-UX commands to manipulate software and patches.


© 2005 Hewlett-Packard Development Company, L.P.

## 12–19. SLIDE: Repairing a Corrupt IPD

### Repairing a Corrupt IPD

- A corrupt IPD will often prevent the administrator from executing Software Distributor commands
- It may be necessary to rebuild the IPD.





## Student Notes

Occasionally, the Installed Products Database (IPD) becomes corrupt. This has the nasty effect of preventing certain SD-UX commands from running correctly (or at all).

Following is a sample dialog showing the output you might see if the IPD were corrupt:

```
# swlist
# Initializing...
# Contacting target "rbox"...
ERROR:   The expected depot or root does not exist at "/".
ERROR:   There is currently no installed software on host "rbox"
at
         location "/".
#
```

This error is usually caused by the INDEX file getting deleted or removed. The "master" INDEX file contains the product definitions for all the installed software on the system. It is located in **/var/adm/sw/products**.


© 2005 Hewlett-Packard Development Company, L.P.

### To Repair a Corrupt IPD:

First, make a backup copy of the **INDEX** file (if it exists). If problems are encountered, this file can be moved back into place, and the product definitions will be restored to their former state.

Then, to rebuild the **INDEX** file, use the following procedure which will install a "fake" product on the system. During the installation of this product, the **INDEX** file will be regenerated.

```
# cd /var/adm/sw/products
# mv INDEX INDEX.orig
# cd /tmp
# vi void.psf
   product
   tag void
   fileset
   tag void
   :wq!
# swpackage -s /tmp/void.psf
# swinstall void
# swremove void
# rm void.psf
# swremove -d void
```


© 2005 Hewlett-Packard Development Company, L.P.

## 12–20. SLIDE: Module Objectives

### Module Objectives

You should now be able to:

- List four new patch management attributes introduced with HP-UX 11.00.
- List three additional patch management tools that can be added to an HP-UX 11.x system through a patch.
- Describe the procedure for committing a patch.
- Describe options when installing patches.
- Describe the use and purpose of `security_check_patch` tool.
- Describe the purpose of various SD-UX logfiles.
- Understand the purpose of the Installed Products Database (IPD).

## Student Notes

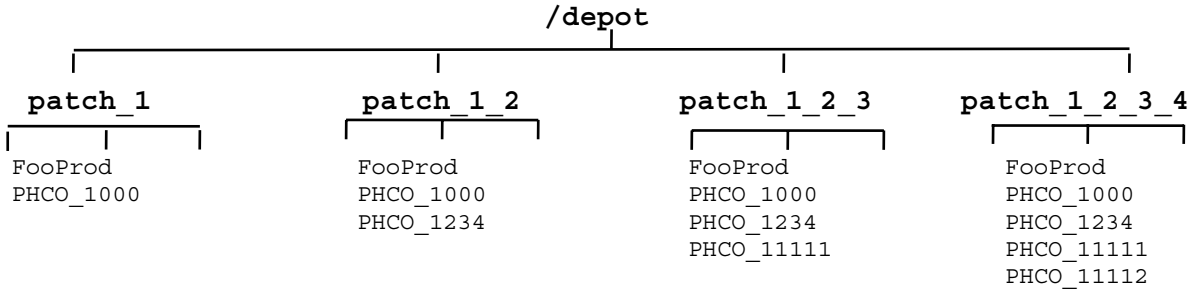



© 2005 Hewlett-Packard Development Company, L.P.

# 12–21. LAB: Troubleshooting Patch Installation

## Directions

This lab exposes the student to patch installation and patch commit scenarios. The lab makes use of four depots installed under the **/depot** directory. The depots are organized as follows:




### Automatic Patch Selection

Demonstrate the **autoselect_patch** feature that automatically selects all patches for an existing product. The patch must be in the same depot as the product.

1. Install the **FooProd** product from the **patch_1** directory.

   ```
   # swinstall -s /depot/patch_1 FooProd
   ```

   Note the output from the "Analysis Phase." The patch **PHCO_1000** was automatically selected for the **FooProd** product.

2. List all products "not contained in a bundle."

   ```
   # swlist
   ```

   The product and the patch should be listed.


© 2005 Hewlett-Packard Development Company, L.P.

3. List the files that were patched by the **PHCO_1000** patch:

```
# swlist -l file PHCO_1000
```

There should be a file called **/usr/bin/foo**.

4. Display the contents of the patched file, **/usr/bin/foo**:

```
# more /usr/bin/foo
```

Note the last line indicating the product has been patched by PHCO_1000.

5. The **save** directory is the location of the original file, before being patched. List the **save** directory:

```
# ll /var/adm/sw/save
```

6. Note the name of the **PHCO_1000** patch. Under this directory are copies of the original files before the patch was installed.

7. Display the contents of the original file.

```
# more /var/adm/sw/save/PHCO_1000/FOO-MIN/usr/bin/foo
```




© 2005 Hewlett-Packard Development Company, L.P.

8. Verify that this file has been restored upon removal of the patch.

```
# swremove PHCO_1000
# more /usr/bin/foo
```

## Patch Selection of the Most Current Patch

Demonstrate that if two patches exist in a depot for the same product, the "most current" patch is selected.

1. Reinstall the **FooProd** product from a directory containing two patches: **patch_1234** supersedes **patch_1000**.

```
# swinstall -s /depot/patch_1_2 -x reinstall=true FooProd
```

2. Note that the most current patch is applied.

   Verify that the most current patch was applied:

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

## Reinstall the Product with the Most Recent Patch

Demonstrate that the more current patch (in this case **PHCO_11111**) will supersede the existing patch (**PHCO_1234**), if the product is reinstalled and a more current patch exists.

1. Reinstall the **FooProd** product with a more current patch. Do *not* delete the current patch already installed.

```
# swinstall -s /depot/patch_1_2_3 -x reinstall=true FooProd
```


**© 2005 Hewlett-Packard Development Company, L.P.**

2. Verify that the old patch and product were removed by the new installation.

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

### Apply a New Patch without Removing the First Patch

1. Install another patch (**PHCO_11112**) without removing any existing patches. This new patch has a dependency patch. Verify that the previously patched files are saved, and the dependency patch (**PHSS_5555**) is loaded with the new patch.

```
# swinstall -s /depot/patch_1_2_3_4 PHCO_11112
```

2. Verify that the new patch was installed without removing the existing patch.

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

### Attempt to Remove a Patch needed for Rollback

1. The **FooProd** product contains two applied patches. **PHCO_11111** was applied first, and **PHCO_11112** was applied second. Try to remove patch **PHCO_11111**.

```
# swremove PHCO_11111
```

This should fail, because removing a patch needed for rollback is an “unsupported” operation.




© 2005 Hewlett-Packard Development Company, L.P.

## Verify Patch Rollback

1. Remove the **PHCO_11112** patch. Verify that the files roll back to the patch **PHCO_11111** state.

```
# swremove PHCO_11112
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

2. Remove the **PHCO_11111** patch. Verify that the files roll back to the original state.

```
# swremove PHCO_11111
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

3. Remove the **FooProd** product. Verify that all related patches are also removed.

```
# swremove FooProd
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

## Test Applying Patches to Existing Products

1. Install the **FooProd** product. Specifically indicate *not* to install patches at this time.

```
# swinstall -x autoselect_patches=false -s /depot/patch_1 FooProd
```

2. Verify only the product was installed:

```
# swlist
```


**© 2005 Hewlett-Packard Development Company, L.P.**

3. Install patches to all existing, applicable products:

```
# swinstall -x patch_match_target=true -s /depot/patch_1_2_3_4
```

4. Verify that the most current patch has been installed:

```
# swlist
```

## Test Loading an Old Patch

1. Try to load a patch that has been "superseded" by an already loaded patch:

```
# swinstall -s /depot/patch_1_2 PHCO_1234
```

2. This should fail. Review log file for specific details:

```
# tail -50 /var/adm/sw/swagent.log | more
```

3. Remove the **`FooProd`** product. Verify that all related patches are removed.

```
# swremove FooProd
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```




© 2005 Hewlett-Packard Development Company, L.P.

## 12–22. LAB: Troubleshooting a corrupt IPD

### Directions

This lab exposes the student to troubleshooting a corrupt IPD.

1. Let's create a scenario of a missing IPD and then we will rebuild it. Please rename your **`INDEX`** file in preparation for the rest of the lab.

2. Now, list the products that are currently installed on your system. Were you successful?

3. Using the steps from this module, rebuild your IPD.

4. Now let's ensure the above procedure properly rebuilt the IPD.


**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 13 — Troubleshooting Network Problems: Part 1

### Objectives

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

- Relate the OSI 7-Layer Model to network addressing, display, and test command output.
- Identify and resolve LAN card configuration problems.
- Identify and resolve subnetting problems.
- Identify and resolve routing problems.
- Display and modify tunable network parameters with **`ndd`**.




© 2005 Hewlett-Packard Development Company, L.P.

## 13–1. SLIDE: Module Objectives

### Module Objectives

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

- Relate the OSI 7-Layer Model to network addressing, display, and test command output
- Identify and resolve LAN card configuration problems
- Identify and resolve subnetting problems
- Identify and resolve routing problems
- Display and modify tunable network parameters with `ndd`

## Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 13–2. SLIDE: Addressing the Correct Layer

### Addressing the Correct Layer

| Layer | Configuration | Display | Test |
|---|---|---|---|
| Application Hostname<br>Presentation Hostname | /usr/bin/hostname<br>/sbin/set_parms<br>/etc/rc.config.d/netconf<br>/sbin/init.d/hostname<br>/sbin/rc1.d/S320hostname<br>/etc/hosts<br>/etc/resolv.conf<br>/etc/nsswitch.conf | /usr/bin/hostname<br>/usr/bin/nslookup<br>/usr/contrib/bin/nsquery<br>/usr/sbin/arp<br>/usr/bin/netstat -i | /usr/bin/hostname<br>/usr/sbin/ping *foreign-hostname*<br>/usr/bin/nslookup *hostname*<br>*/usr/contrib/bin/nsquery hosts hostname*<br>/usr/contrib/bin/nsquery *hosts hostname* |
| Session Socket | IPAddress.Port | /usr/bin/netstat -[apif][n]<br>/usr/bin/find / -type s | ----- |
| Transport Port | /etc/services<br>/etc/protocols<br>/usr/sbin/rpcbind | /usr/bin/netstat -[apif][n] | ----- |
| Network IP Address | /usr/sbin/ifconfig<br>/etc/rc.config.d/netconf<br>/sbin/init.d/net /sbin/init.d/net.init<br>/sbin/rc2.d/S008net.init /sbin/rc2.d/S340net<br>/sbin/rc1.d/K992net.init /sbin/rc1.d/K660net | /usr/sbin/ifconfig *lanic*<br>/usr/bin/nslookup<br>/usr/contrib/bin/nsquery<br>/usr/sbin/arp<br>/usr/bin/netstat -i | /usr/sbin/ping *foreign-IPAddress*<br>/usr/bin/nslookup *hostname*<br>/usr/contrib/bin/nsquery *hosts IPAddress* |
| Data Link MAC | Unique MAC address | /usr/sbin/lanscan<br>/usr/sbin/lanadmin /usr/sbin/landiag | /usr/sbin/linkloop 0x*HexMACAddress* |
| Physical | Physically connect hardware | ----- | ----- |

## Student Notes

Whether you choose to troubleshoot top-down or bottom-up, knowing the right addressing level with the right command is key. The top-down troubleshooter starts at OSI Model Layer 7 (Application) and runs test commands down the chart until a non-error condition is encountered. The bottom-up troubleshooter starts at OSI Model Layer 1 (Physical) and runs test commands up the chart until an error condition is encountered. Either way, this chart will help identify network address-related problems.

In practice, most network troubleshooting starts in the middle, at layer three. **`ping`** is easy and fast, and in most cases lets you know whether a connection is possible. However, some networks are configured to filter out ping packets and allow “application” packets only.

Once the adjacent layers are identified (for example, layer 2 tests are successful, but all tests above layer 3 fail), troubleshooting can focus on the failing layer configuration. This chart correlates the layers of the OSI 7-layer model with the addressing levels, initialization commands and files, display, and test commands.

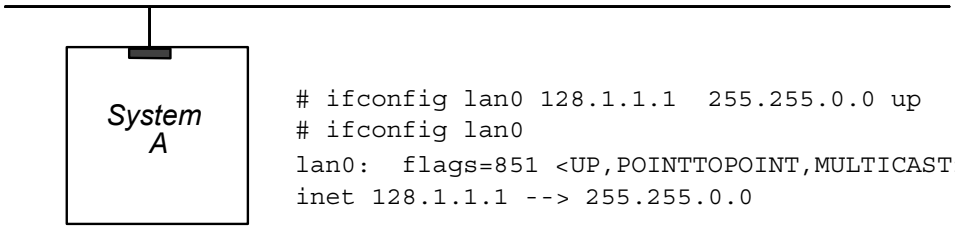



© 2005 Hewlett-Packard Development Company, L.P.

## 13–3. SLIDE: Troubleshooting LAN Card Configuration Problems

### Troubleshooting LAN Card Configuration Problems

- Using the `ifconfig` command incorrectly results in an incorrect LAN configuration.




```
# ifconfig lan0 128.1.1.1  255.255.0.0 up
# ifconfig lan0
lan0:  flags=851 <UP,POINTTOPOINT,MULTICAST>
inet 128.1.1.1 --> 255.255.0.0
```

Do you see the problem?

What is the solution?

## Student Notes

One of the most common commands used for LAN card configuration is **`ifconfig`**. It is used both to display network interface card settings and to configure a network interface card. If **`ifconfig`** is used incorrectly, your LAN card will have erroneous configuration information associated with it.

The above example uses **`ifconfig`** to configure an IP address of **`128.1.1.1`** and a subnet mask of **`255.255.0.0`**. Unfortunately, it does not work. In the above command, the subnet mask is being specified without the keyword “netmask” preceding it. Thus, the LAN card is being configured as a point-to-point interface, which is incorrect. The correct way to configure the LAN card is:

```
# ifconfig lan0 128.1.1.1 netmask 255.255.0.0 up
```

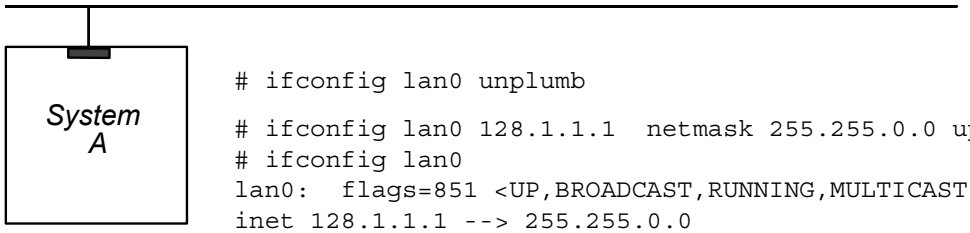
If you issue this command at this point, it still will not fix this interface. Instead, you must "tear down" the lan card configuration and build it back up again.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 13–4. SLIDE: Solving the LAN Card Configuration Problem

### Solving the LAN Card Configuration Problem

- Tear down the LAN configuration, then reconfigure the LAN card properly.




```
# ifconfig lan0 unplumb

# ifconfig lan0 128.1.1.1  netmask 255.255.0.0 up
# ifconfig lan0
lan0:  flags=851 <UP,BROADCAST,RUNNING,MULTICAST>
inet 128.1.1.1 --> 255.255.0.0
```

## Student Notes

If you ran into a situation where your LAN card was improperly configured, as in the case of the example illustrated on the previous slide, one way to solve the problem is to tear down the current LAN card configuration and reconfigure it correctly.

How do you do this? The commands below illustrate how:

```
# ifconfig lan0 unplumb
# ifconfig lan0 128.1.1.1 netmask 255.255.0.0 up
# ifconfig lan0
```

The sequence of commands noted above is one solution for fixing an incorrectly configured LAN card. The first command uses the **`unplumb`** option. This option tears down the LAN card's configuration. Once this command is run, the LAN card exists as if it had never been configured. The second two command lines simply configure the LAN card and verify the configuration.

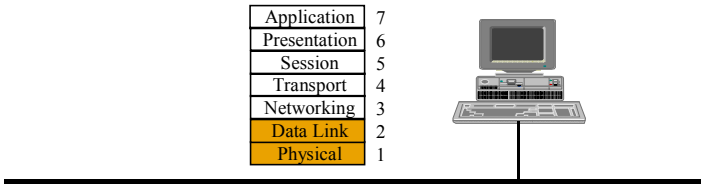
In most cases, unplumbing the interface is unnecessary; simply using **`ifconfig`** with the correct settings will make the card work. Remember to make the corrections in the **`/etc/rc.config.d/netconf`** file also.




© 2005 Hewlett-Packard Development Company, L.P.

## 13–5. SLIDE: Troubleshooting: Using the `lanadmin` Command

### Troubleshooting: Using the `lanadmin` Command




- The `lanadmin` command is a LAN diagnostic tool, available specifically for HP 9000 business servers and workstations.
- Capabilities of the command:
  - Resets the LAN interface card
  - Changes the maximum packet size for the LAN card
  - Changes the speed setting of the LAN card
  - Displays driver statistics for the LAN card
  - Resets the driver statistics to zero for the LAN card

## Student Notes

**`lanadmin`** allows you to do the following:

- Display and change the station address.
- Display and change the maximum packet size (max transmission unit, or MTU) for the LAN card.
- Display and change the maximum speed, and duplex settings for the LAN card.
- Gather LAN interface statistics.
- Reset the interface card.
- Execute the interface self-test to check for hardware problems.

Following are the **`lanadmin`** command options:

**`-e`** Echoes the input commands on the output device. This command is useful if you want to redirect your output to a file.

**`-t`** Suppresses the display of the command menu before each command prompt. This command is the same as the test selection mode, **`terse`** command.


© 2005 Hewlett-Packard Development Company, L.P.

**-a** Display current station address corresponding to PPA number. The **-A** argument is used to change the station address.

**-m** Display current MTU size corresponding to PPA number. The **-M** argument is used to change the MTU size.

**-s** Display current speed setting corresponding to PPA number. The **-S** argument is used to change the speed setting.

**-h** Display online help related to the syntax of the command.

Conventional execution of this command (without parameters) yields the following:

```
# /usr/sbin/lanadmin

       LOCAL AREA NETWORK ONLINE ADMINISTRATION, Version 1.0
                      Fri, May 27,1999 16:38:54

               Copyright 1999 Hewlett Packard Company.
                     All rights are reserved.

Test Selection mode.

   lan     = LAN Interface Administration
   menu    = Display this menu
   quit    = Terminate the Administration
   terse   = Do not display command menu
   verbose = Display command menu

Enter command: lan
```

When you invoke **lanadmin**, you are in the test selection mode. From here, there is only one useful choice. Enter the diagnostic by entering **lan** or just the first letter, **l**.

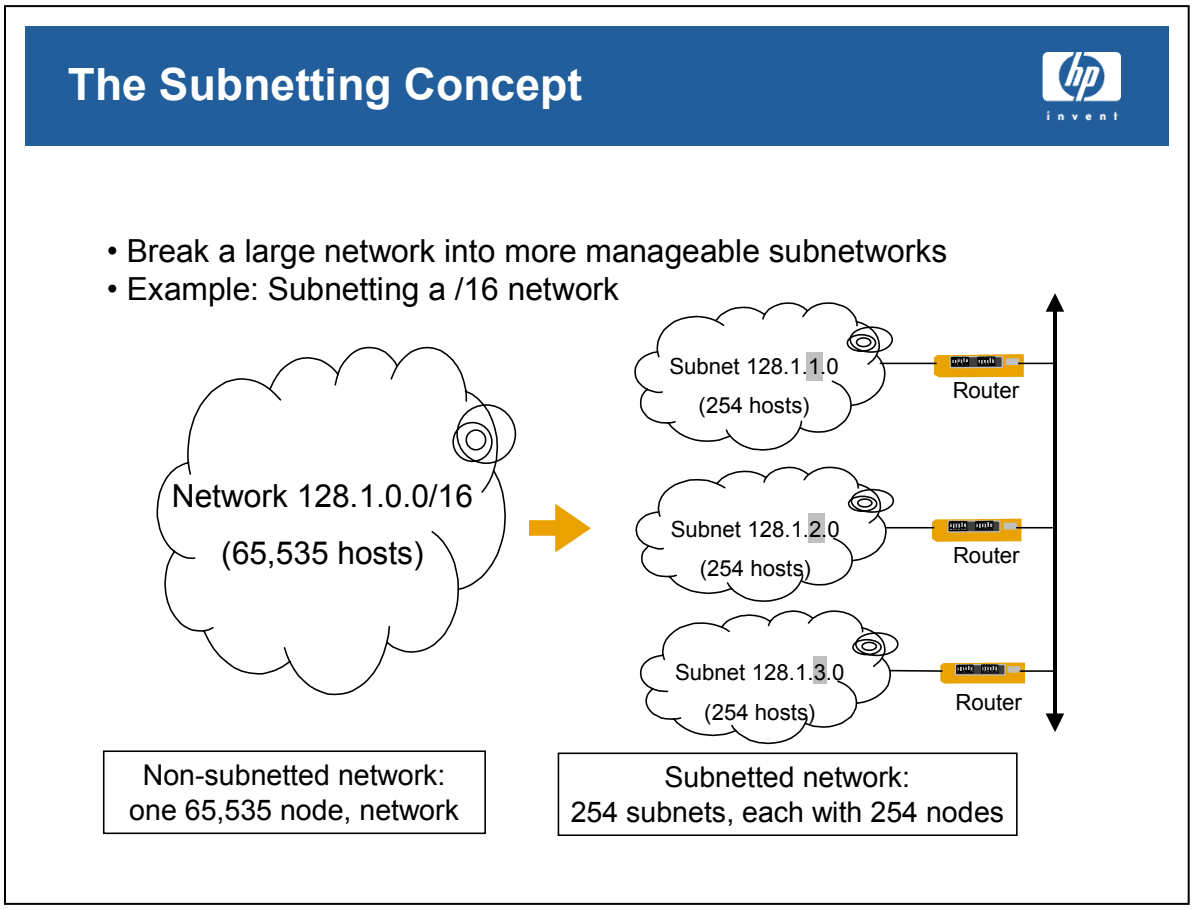
The LAN interface diagnostic allows you to test your LAN hardware (layers 1 and 2 of the OSI model).

*NOTE:* **lanadmin** requires the device file **/dev/dlpi** and the kernel driver **dlpi**.




© 2005 Hewlett-Packard Development Company, L.P.

## 13–6. SLIDE: The Subnetting Concept




## Student Notes

Subnetting makes it possible to divide a large network IP address space into several smaller, more manageable "subnets."

The example on the slide shows a subnetted /16 network. Without subnetting, the 128.1.0.0/16 network would have 65 thousand hosts on the same physical network, which could easily lead to excessive collisions.

This network, however, has been subdivided into 254 subnets. Each of these subnets could potentially have up to 254 hosts.

```
Subnet Addresses
----------------
128.1.1.0
128.1.2.0
...
128.1.253.0
128.1.254.0
```


© 2005 Hewlett-Packard Development Company, L.P.

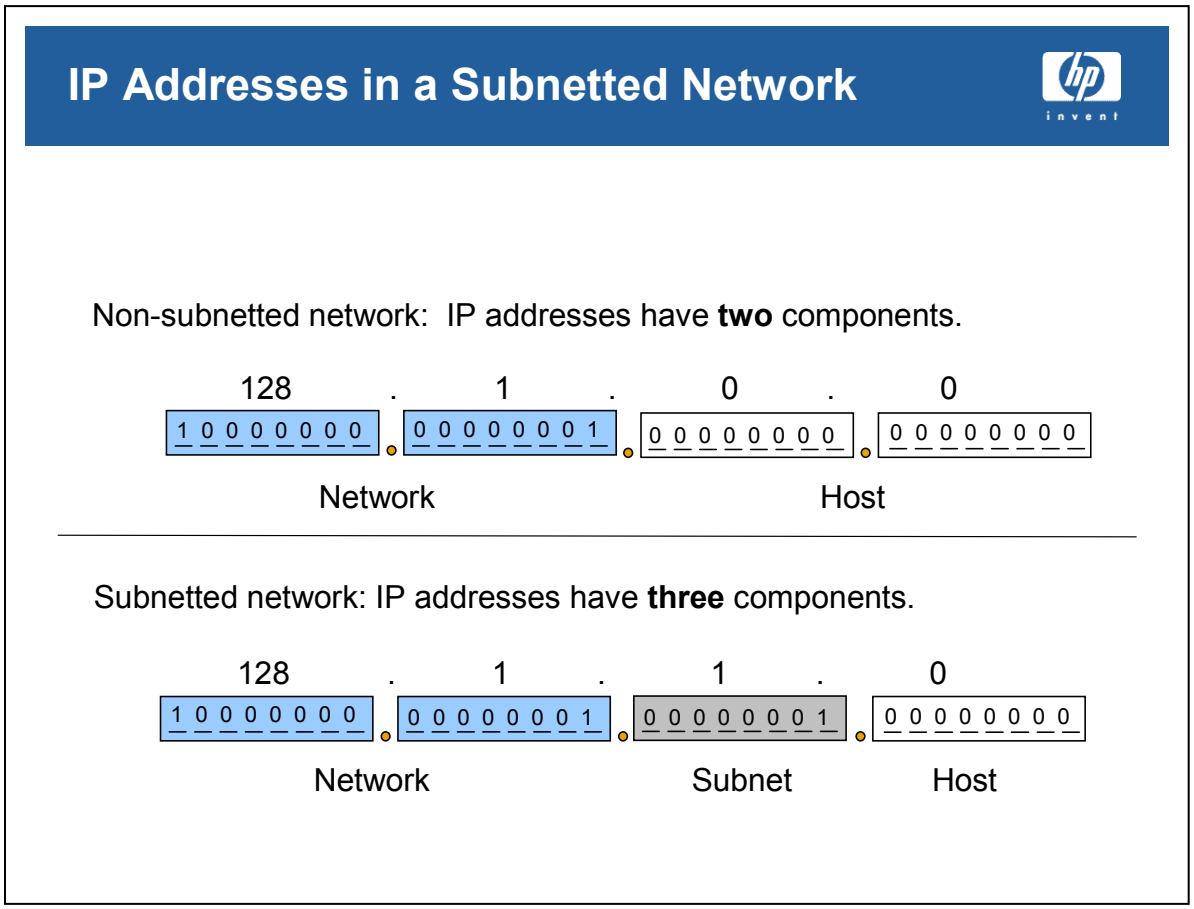
Subnets are separated from one another by routers, which overcome both the collision and topological issues discussed on the previous slide.

Subnetting also makes it easy for the network administrator to delegate authority for portions of the IP network address space to other entities within the organization. Simply assign each department a separate subnet. Each network administrator then becomes responsible for a subnet within the larger corporate network.




© 2005 Hewlett-Packard Development Company, L.P.

## 13–7. SLIDE: IP Addresses in a Subnetted Network




## Student Notes

In a non-subnetted network, each IP address has just two components. A portion of the IP's bits identifies the network to which a host is attached, and the remaining bits uniquely define individual hosts on the network.

Subnetted IP addresses have a third component as well: a portion of the IP address's host bits is used to define the subnet to which the host belongs.

Returning to the 128.1.0.0/16 network example: Normally, a host on a /16 network has 16 host bits. When implementing subnetting, 8 of those bits are used to define the host's subnet, leaving 8 remaining bits to define the individual host address.

The number of subnet bits may vary. Increasing the number of subnet bits allows more subnets, but fewer hosts on each subnet. Decreasing the number of subnet bits decreases the number of addressable subnets, but allows more hosts on each subnet.


© 2005 Hewlett-Packard Development Company, L.P.

## 13–8. SLIDE: Netmasks in a Subnetted Network

### Netmasks in a Subnetted Network

- The netmask masks network and subnet bits with 1s.
- Netmask for a non-subnetted /16 network:

| 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 | 0 0 0 0 0 0 0 0 | 0 0 0 0 0 0 0 0 | = 255.255.0.0 |
|---|---|---|---|---|
| Network | | Host | | Netmask |

- Netmask for /24 subnetworks on a /16 network:

| 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 | 1 1 1 1 1 1 1 1 | 0 0 0 0 0 0 0 0 | = 255.255.255.0 |
|---|---|---|---|---|
| Network | | Subnet | Host | Netmask |

## Student Notes

The text on the previous page noted that the number of subnet bits can vary. So how do routers and other network devices determine where the network/subnet portion of an IP address ends, and where the host portion of an IP address begins on a subnetted network?

In printed form, the boundary between the network/subnet portion of the IP and the host portion of an IP is typically indicated via the "/" suffix on the end of the IP. The number following the "/" indicates the total number of network/subnet bits. All remaining bits are assumed to be host bits. Consider the example on the bottom of the slide. The IP address in the example has 16 network bits and 8 subnet bits. Since 16+8=24, IP addresses on these subnets would be represented as x.x.x.x/24 addresses.

UNIX identifies the network/ subnet host boundary in an IP address via the IP netmask. On a non-subnetted network, the 1s in the netmask identify network bits. On a subnetted network, the 1s in the netmask mask both network and subnet bits.

The example on the slide shows a netmask that consists of 24 "1" bits, followed by 8 "0" bits. Thus, the network/subnet portion of the IP addresses on this network appears to span the first three octets, while the final octet represents the host portion of each IP address.

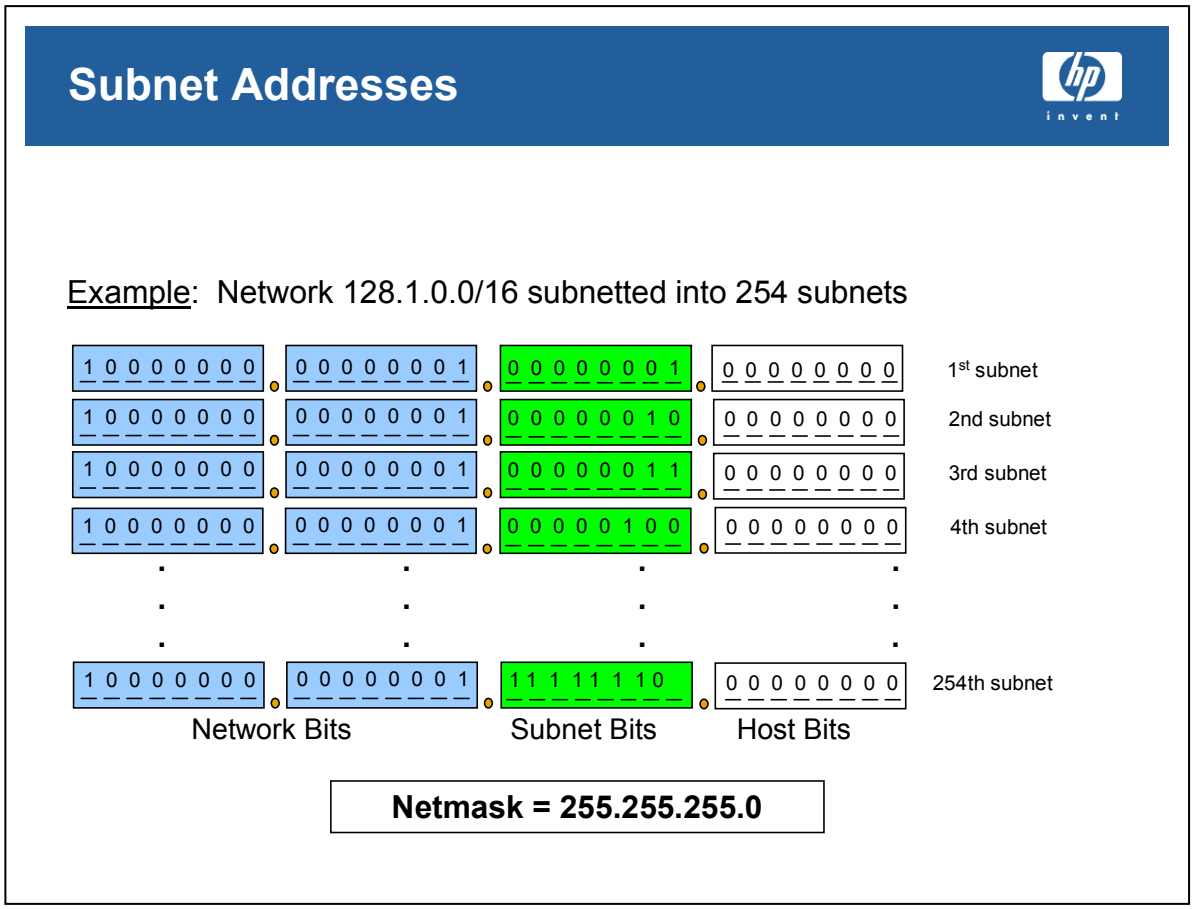



© 2005 Hewlett-Packard Development Company, L.P.

Since the number of subnet bits varies from network to network, the netmask varies from network to network as well. In a subnetted network, you must define the netmask for each LAN interface card.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 13–9. SLIDE: Subnet Addresses




## Student Notes

A single network may contain multiple subnets. The network bits for all hosts on all of the subnets within a network will be the same. However, each subnet is assigned a unique subnet address. The subnet address is defined in the subnet bits specified by the netmask.

Continuing the example started in the previous slides, this slide shows the subnet addresses for the 128.1.0.0/16 network. The 255.255.255.0 netmask tells us that the third octet defines the subnet portion of the IP addresses on this network.

With eight subnet bits, it is possible to represent 256 addresses:

```
00000000 =   0          Not allowed by some devices.
00000001 =   1
00000010 =   2
00000011 =   3
...
11111101 = 253
11111110 = 254
11111111 = 255          Not allowed by some devices.
```




© 2005 Hewlett-Packard Development Company, L.P.

Although it is possible to represent 256 subnet addresses with 8 subnet bits, some devices and do not allow all-0 or all-1 subnets. Eliminating these addresses leaves the following subnet addresses:

```
128.1.1.0/24
128.1.2.0/24
...
128.1.253.0/24
128.1.254.0/24
```

## All-0 and All-1 Subnet Bits in HP-UX

Before HP-UX 11i, HP-UX did not support IP addresses that had all 0s or all 1s in the subnet portion of an IP address. Starting at HP-UX 11i, all-0 and all-1 subnet addresses are supported, but only if the **ip_check_subnet_addr** tunable network parameter has been set to "0". Network tunable parameters, including **ip_check_subnet_addr**, can be both viewed and set using the **ndd** command:

```
# ndd -get /dev/ip ip_check_subnet_addr     Check the current value
# ndd -set /dev/ip ip_check_subnet_addr 0   Enable all-0/all-1 subnets
# ndd -set /dev/ip ip_check_subnet_addr 1   Disable all-0/all-1 subnets
```

By default, this parameter is set to 0, and all-0 and all-1 subnet addresses are allowed.

Changes made via **ndd** are lost at reboot time, unless they are recorded in the **/etc/rc.config.d/nddconf** file:

```
# vi /etc/rc.config.d/nddconf
  TRANSPORT_NAME[1]=ip
  NDD_NAME[1]=ip_check_subnet_addr
  NDD_VALUE[1]=0
```

This is just one of many parameters that may be tuned via the **ndd** command. For a full list of tunable **ndd** parameters, type **ndd -h**.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 13–10. SLIDE: Host IP Addresses on a Subnet

### Host IP Addresses on a Subnet

- The host address with all 0s represents the address for the entire subnet.
- The host address with all 1s represents the broadcast address for the subnet.
- All other addresses within the subnet may be used for hosts.
- Examples: IP addresses for subnet 128.1.1.0/24:

```
Subnet #1 :   10000000.00000001.00000001.00000000 = 128.1.1.0/24
Host #1   :   10000000.00000001.00000001.00000001 = 128.1.1.1/24
Host #2   :   10000000.00000001.00000001.00000010 = 128.1.1.2/24
Host #3   :   10000000.00000001.00000001.00000011 = 128.1.1.3/24
   :                                  :                          :

Host #253 :   10000000.00000001.00000001.11111101 = 128.1.1.253/24
Host #254 :   10000000.00000001.00000001.11111110 = 128.1.1.254/24
Broadcast :   10000000.00000001.00000001.11111111 = 128.1.1.255
```

**Netmask = 255.255.255.0**

## Student Notes

Each subnet may contain multiple hosts. Within a subnet, all network and subnet bits must be identical for every host. However, each host must have a unique sequence of host bits to distinguish it from all the other hosts on the subnet.

Consider the 128.1.1.0/24 subnet from the previous page. Each host on this subnet will have an IP address that begins with 128.1.1. This leaves eight host bits.

With eight bits, it is possible to represent 256 values:

```
00000000 =   0
00000001 =   1
00000010 =   2
00000011 =   3
...
11111101 = 253
11111110 = 254
11111111 = 255
```




© 2005 Hewlett-Packard Development Company, L.P.

The address formed by setting all the host bits to 0 is used to define routes to the subnet in the network routing tables. This address should not be assigned to a specific node.

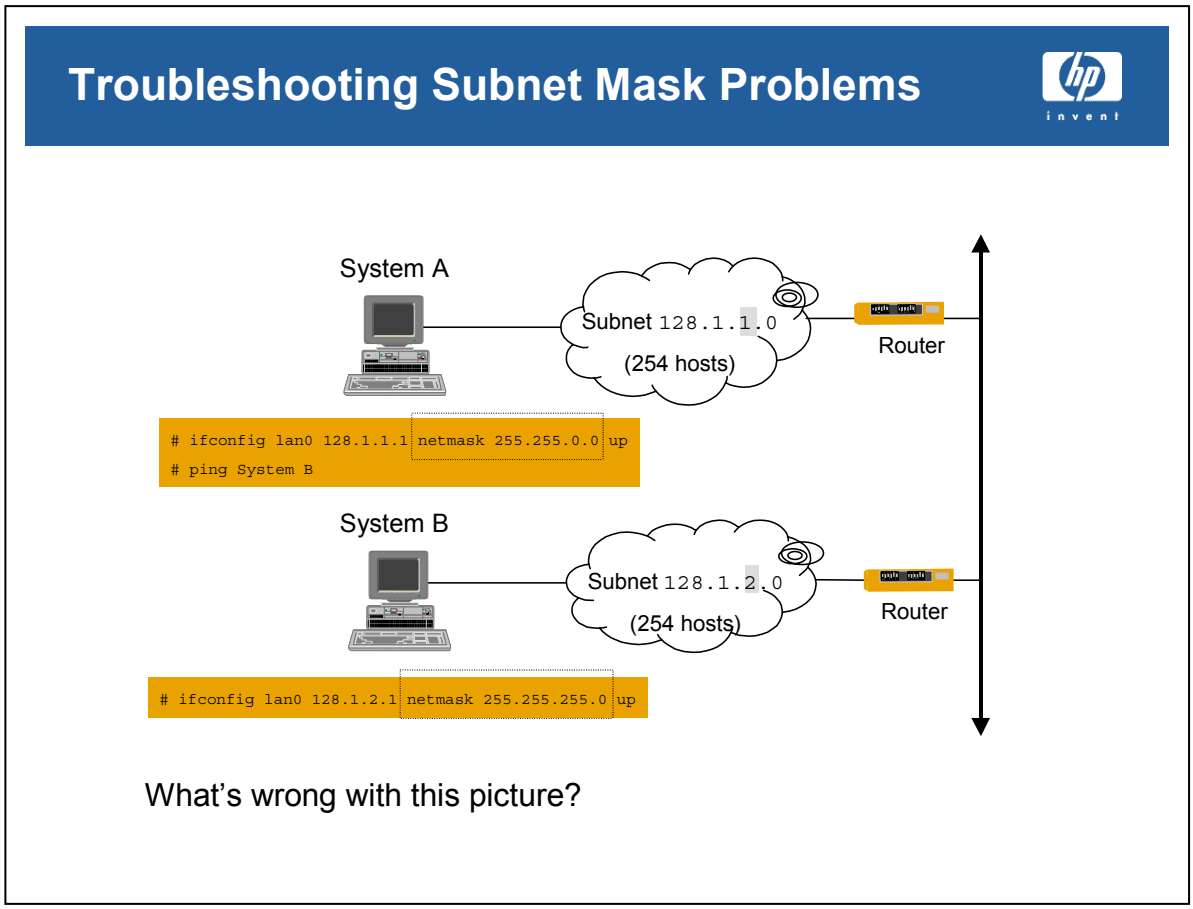
The address formed by setting all the host bits to 1 is a reserved address as well. It is the subnet broadcast address.

All remaining addresses may be assigned to hosts in the subnet. Valid addresses for hosts on the 128.1.1.0/24 subnet, then, include:

```
128.1.1.1/24
128.1.1.2/24
128.1.1.3/24
...
128.1.1.253/24
128.1.1.254/24
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## 13–11. SLIDE: Troubleshooting Subnet Mask Problems




## Student Notes

A subnet mask is used to extract the network bits from a local system's IP address. These network bits are compared to all destination IP addresses. If the network bits are identical, the remote system can be found on the same network. If the network bits are not identical, the remote system is found on a different network.

In the above slide, System A has configured its LAN card with the following information:

IP address = **`128.1.1.1`**
netmask = **`255.255.0.0`**

In the above slide, System B has configured its LAN card with the following information:

IP address = **`128.1.2.1`**
netmask = **`255.255.255.0`**

If System A issues the command, **`ping 128.1.2.1`**, System A will never gain access to System B. Why? When System A attempts to communicate with System B, System A will compare its network bits to the network bits of the destination address. When this




© 2005 Hewlett-Packard Development Company, L.P.

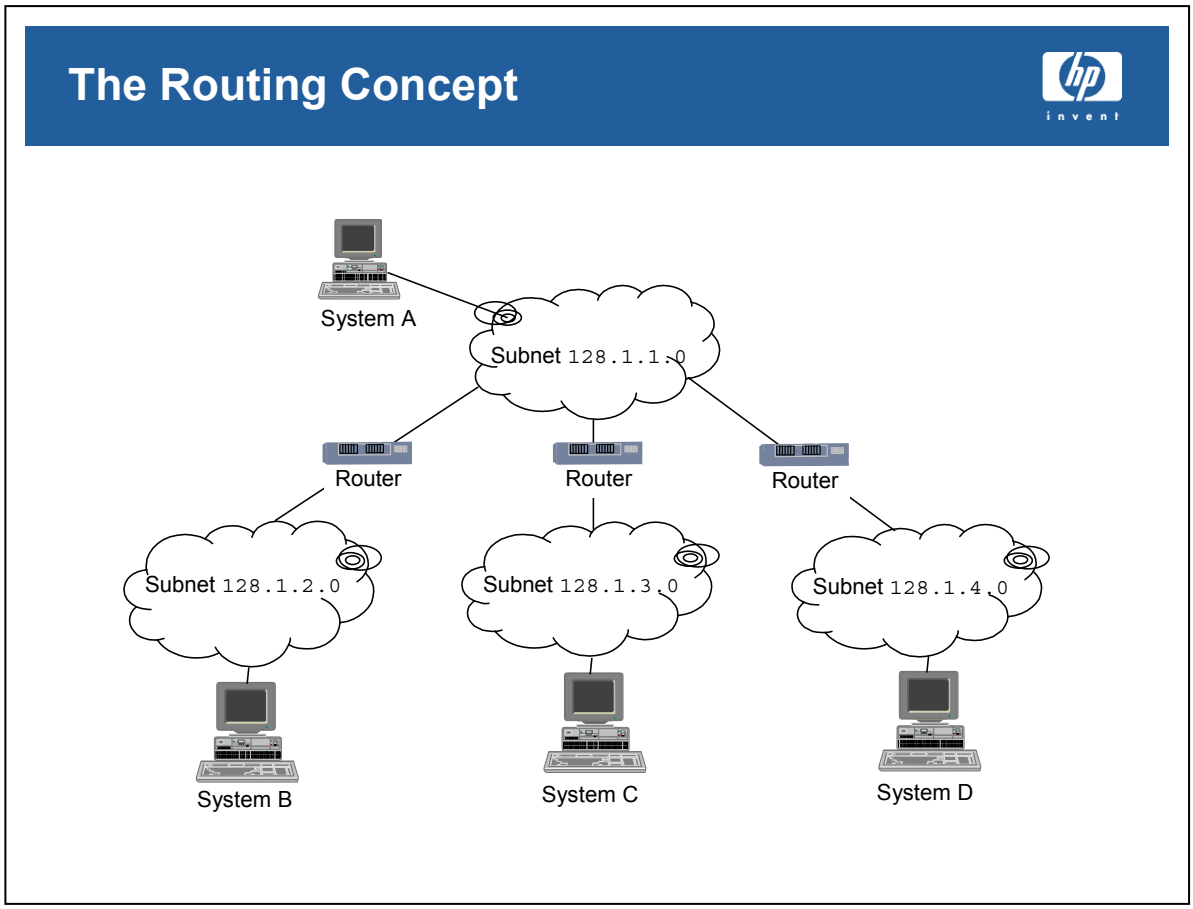
comparison is done, the network bits match. Therefore, System A believes System B is on the same network and attempts to communicate on that basis.

How do you fix a problem like this?

Modify System A's subnet mask to reflect the fact that the third octet is part of the network bits. Once this occurs and System A issues the command, **`ping 128.1.2.1`**, System A will recognize that System B is on a different network. This will force System A to go through a router to access System B.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 13–12. SLIDE: The Routing Concept




## Student Notes

The Internet is composed of many physical network segments interconnected by routers and gateways. The routers and gateways route packets between networks.

Routing refers to the process of choosing a path over which to send a packet.

The term **router** refers to any computer making such choices. Documentation often refers to routers as gateways. The term router is from OSI terminology.

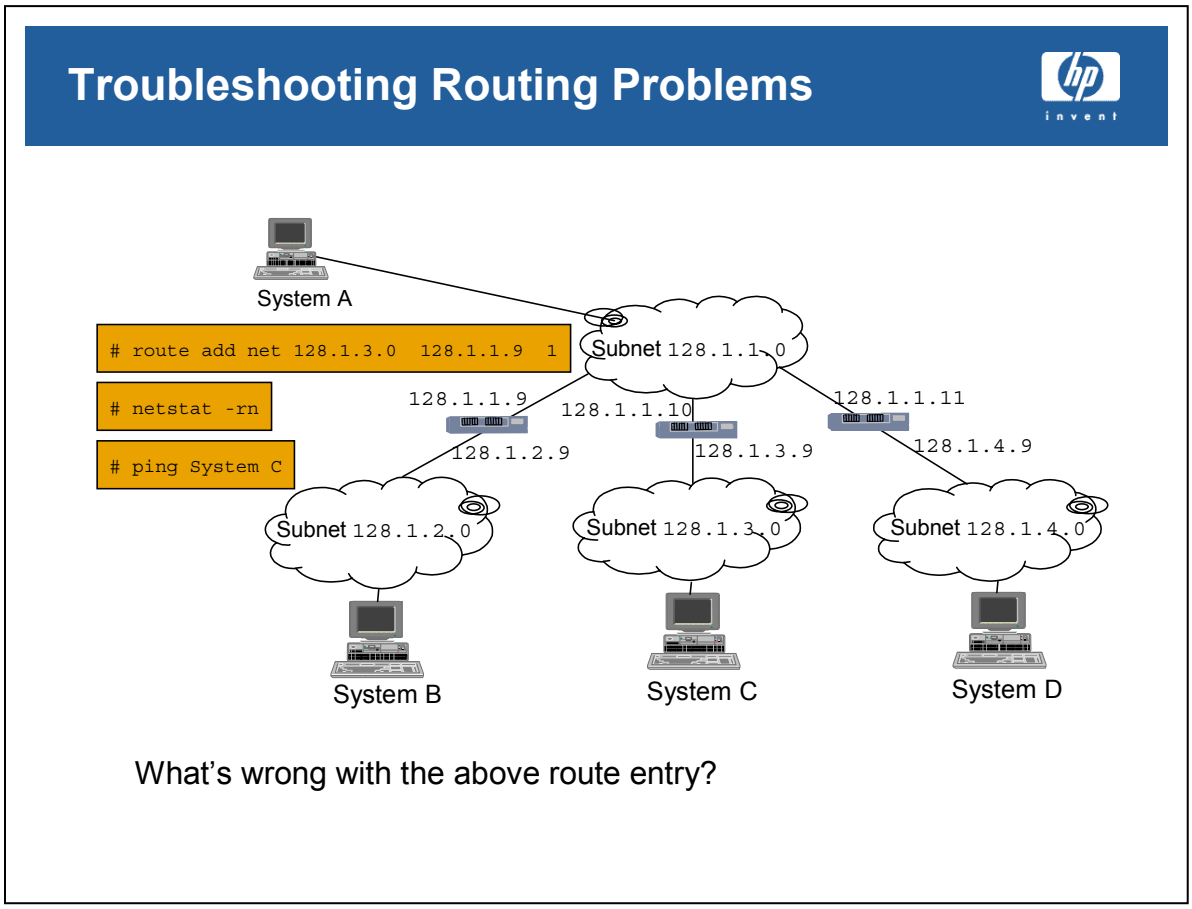
To visualize how routing works, imagine a large internet with many networks interconnected by routers, but with only two hosts at the far ends. When one host wants to send to the other, it sends a packet to its nearest router. When the packet reaches that router, the router selects the next router along the path toward the destination. Packets pass from router to router until they reach a router that can deliver them directly.

Table-driven IP routing is considered address-only routing. This means that the packet traveling across the internet takes along only the destination address. Along the route, the packet is “told where to turn” by routers.




© 2005 Hewlett-Packard Development Company, L.P.

## 13–13. SLIDE: Troubleshooting Routing Problems




## Student Notes

Anytime a system attempts to communicate with a system, including itself, it must be directed by a route entry in the route table. Types of routes found in route tables consist of host, network, and default routes. The definitions of these routes are as follows:

| | |
|---|---|
| Host route: | Route to be used if access to a specific host is desired. |
| Network route: | Route to be used if access to a specific network is desired. |
| Default route: | Route to be used if no other route has been defined for either a specific host or specific network. |

In the slide above, a network route entry is being defined for System A. This entry states that when System A attempts to communicate with another system on network **`128.1.3`**, System A should forward packets through the router configured with IP address **`128.1.1.9`**.

Is this correct? No! How do you fix a problem like this?

Remove the incorrect network route. Thereafter, define the correct network route as it should be.


**© 2005 Hewlett-Packard Development Company, L.P.**

The solution:

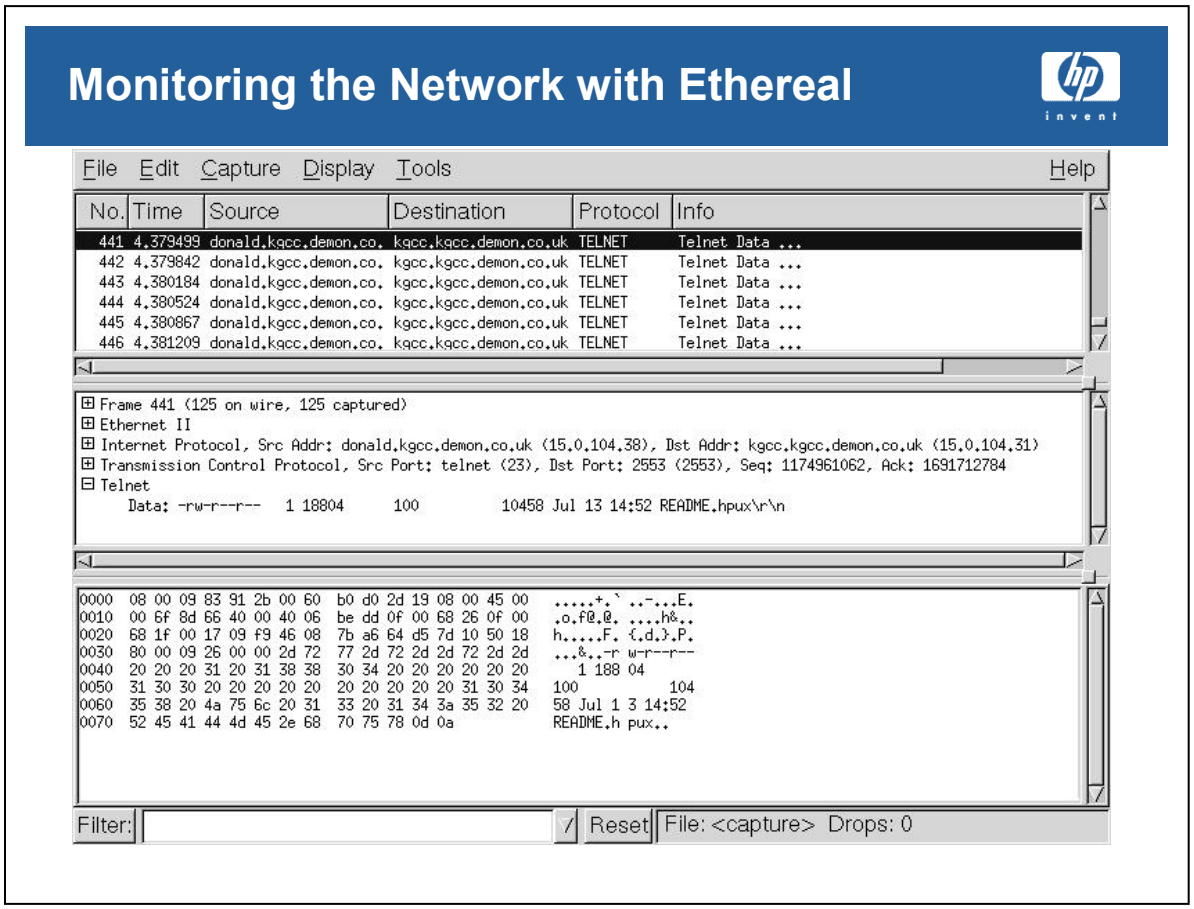
```
# route delete net 128.1.3   128.1.1.9
# route add net 128.1.3   128.1.1.10  1
# netstat -rn
```

The above sequence of commands removes the network route, then establishes the correct network route in the route table.




© 2005 Hewlett-Packard Development Company, L.P.

## 13–14. SLIDE: Monitoring the Network with Ethereal




## Student Notes

As well as the HP supplied and supported network-tracing tools, there are OpenSource tools, such as Ethereal. While HP does not support such tools, they can still provide useful functionality.

Ethereal's home web site is

**http://www.ethereal.com**

Ethereal can also be downloaded from http://software.hp.com . You can either order the Internet Express CD (which contains many useful utilities), or opt to download the Ethereal software by itself.

Example: Using Ethereal to trace a telnet session:

1. Select Edit ➔ Capture filters…

| | |
|---|---|
| Filter name | `telnet` |
| Filter strings | `port 23` |


© 2005 Hewlett-Packard Development Company, L.P.

Click **New** and then **Close**.

2. Select **Capture** ➔ **Start**.

   Select your filter and any options you might like, such as **promiscuous mode** (capturing packets not addressed to your system), or list in real time.

   Then select **OK**

3. A packet counter window will then pop up. This also has the **stop capturing** button on it.

One advantage that **Ethereal** has over some other trace capturing utilities is that it has built-in formatters for many common network applications such as **telnet**, as seen in this example. As well as being capable of formatting the raw Ethernet packet, or the **tcp** part of the packet, it also understands the format of the **telnet** portion of the packet.




© 2005 Hewlett-Packard Development Company, L.P.

## 13–15. SLIDE: Tuning the Network with `ndd` (HP-UX 11.x)

### Tuning the Network with `ndd` (HP-UX 11.x)

`/usr/bin/ndd`

Display or modify tunable network parameters: Times are in milliseconds.

- `ndd -c` # Read `/etc/rc.config.d/nddconf` to set network device tunable parameters.
- `ndd -h [supported|unsupported]` # Display help on all, or only supported or unsupported tunable network parameters.
- `ndd -get /dev/tcp ?` # Display `tcp` tunable parameters
- `ndd -set /dev/ip ip_forwarding 0` # Disable `ip_forwarding` by setting it to zero.
- `ndd -get /dev/arp arp_cache_report` # Display `arp cache`
- `ndd -set /dev/arp arp_cleanup_interval 604800000` # Purge `arp` entries not referenced in 7 days.

## Student Notes

**`/usr/bin/ndd`** is another tool to help display and tune network parameters. **`ndd`** displays and allows modification of network interface device files.

When coming up to run-level 2, **`/sbin/init.d/net.init`** references **`/etc/rc.config.d/nddconf`** to initialize network tunable parameters.

| | |
|---|---|
| **`ndd -h [supported|unsupported]`** | Help on all or only supported or unsupported tunable network parameters. |
| **`ndd -get`** ***`parameter net_dev`*** | Display *parameter* for *net_dev*. |
| **`ndd -set`** ***`parameter net_dev val`*** | Set *parameter* for *net_dev* to *val*. |
| **`ndd -c`** | Set network parameters to values for network devices as specified in **`/etc/rc.config.d/nddconf`**. |


© 2005 Hewlett-Packard Development Company, L.P.

## 13–16. SLIDE: Module Objectives

### Module Objectives

You should now be able to do the following:

- Relate the OSI 7-Layer Model to network addressing, display, and test command output.
- Identify and resolve LAN card configuration problems.
- Identify and resolve subnetting problems.
- Identify and resolve routing problems.
- Display and modify tunable network parameters with `ndd`.

## Student Notes




© 2005 Hewlett-Packard Development Company, L.P.

## 13–17. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What option is available with the `ifconfig` command to tear down the present LAN card configuration?

2. What is the purpose of a **subnet mask**? If it is not set properly, what problems are likely to ensue?

3. What is the purpose of **routing**? If entries within a route table are incorrect, what will happen?


© 2005 Hewlett-Packard Development Company, L.P.

# 13–18. LAB: Using `ndd` to Set Network Parameters

## Purpose

This lab is designed to give the student experience with displaying and modifying tunable network parameters with **`ndd`**.

## Directions

### Displaying and Modifying Tunable Network Parameters

1. Display all tunable network parameters with **`ndd`**.

2. Display the current value of **`ip_forwarding`** with **`ndd`**.

3. Set **`ip_forwarding`** to 0 with **`ndd`**.

4. Return the value of **`ip_forwarding`** to its original value with **`ndd`**.

5. Verify that **`ip_forwarding`** has returned to its original value with **`ndd`**.




© 2005 Hewlett-Packard Development Company, L.P.

# 13–19. LAB: Configuring the LAN Card

## Purpose

The goal of this portion of the lab exercise is to configure a new IP address and host name for all the machines in the classroom.

## Preliminary Steps

1. Just in case something goes wrong during this lab, make a backup copy of all of your network configuration files. There is a shell script in your **`labs`** directory designed specifically for this purpose. The shell script will save a tar archive backup of your network configuration files in the file you specify. Add the **`-l`** option to verify your backup.

   ```
   # /home/labs/netfiles.sh -s ORIGINAL
   # /home/labs/netfiles.sh -l
   # /home/labs/netfiles.sh -l ORIGINAL
   ```

2. Portions of this lab may disable your lan0 interface card. If you are using remote lab equipment, login via the MP console interface for the duration of the lab.

3. Changing your host name and IP on a running system can wreak havoc on CDE and other applications. Kill CDE before going any further:

   ```
   # /sbin/init.d/dtlogin.rc stop
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

# New LAN Card Configuration

## Directions

Your instructor will assign you a host name from the table below.

The first two octets in the IP addresses will vary from classroom to classroom, but should be consistent across all hosts within your classroom. Ask your instructor what the first two octets should be set to. The last two octets must be set in accordance with the table below.

| Host Name | IP Address |
|---|---|
| corp | ____.____.0.1 |
| sanfran | ____.____.1.1 |
| oakland | ____.____.1.2 |
| la | ____.____.1.3 |
| chicago | ____.____.2.1 |
| peoria | ____.____.2.2 |
| rockford | ____.____.2.3 |
| atlanta | ____.____.3.1 |
| athens | ____.____.3.2 |
| macon | ____.____.3.3 |
| nyc | ____.____.4.1 |
| albany | ____.____.4.2 |
| buffalo | ____.____.4.3 |
| paris | ____.____.5.1 |
| lyon | ____.____.5.2 |
| grenoble | ____.____.5.3 |
| london | ____.____.6.1 |
| leeds | ____.____.6.2 |
| ipswich | ____.____.6.3 |
| bonn | ____.____.7.1 |
| berlin | ____.____.7.2 |
| hamburg | ____.____.7.3 |
| tokyo | ____.____.8.1 |
| kyoto | ____.____.8.2 |
| osaka | ____.____.8.3 |

1. There should be a script in the **`/home/labs`** directory called **`netsetup.sh`**. This script will ask you for your instructor-assigned hostname, and the first two IP octets that your instructor should also provide. After you enter the requested parameter, use the displayed information to configure your new hostname, IP address, and **`/etc/hosts`** file. Be sure to reboot to ensure the new settings take effect. The new hostname and IP address will be used in this module and the next. Run the script as follows:

   ```
   # /home/labs/netsetup.sh
   ```




© 2005 Hewlett-Packard Development Company, L.P.

2. After the reboot, you should be able to **`ping`** the instructor's system. Enter the following command:

```
#  ping corp
```

3. Were you successful? If so, continue. If not, redo question 1.

4. Let's make another "checkpoint" backup copy of this new network configuration as NEW. In the subsequent labs, we may need to restore back to this NEW configuration if something goes wrong.

```
# /home/labs/netfiles.sh -s NEW
```

5. Execute the following command line:

```
# /home/labs/mod13.lab1.sh
```

Using the knowledge gained in the current module, fix your system so that you can **`ping`** the instructor's system.


**© 2005 Hewlett-Packard Development Company, L.P.**

# 13–20. LAB: Host Communication Problems

## Purpose

This lab is designed to give you experience troubleshooting host communication problems.

## Directions

Read and answer the following questions.

1. Execute the **`mod13.lab2.sh`** script.

   ```
   # cd /home/labs
   # ./mod13.lab2.sh
   ```

   After running this script, fix your system so that you can **`ping`** the instructor's system (corp).

2. Execute the **`mod13.lab3.sh`** script.

   ```
   # cd /home/labs
   # ./mod13.lab3.sh
   ```

   After running this script, fix your system so you can **`ping`** the second LAN card connected to the instructor's system. Ask your instructor for the IP address of the second LAN card.

3. Execute the **`mod13.lab4.sh`** script. **(Optional)**

   ```
   # cd /home/labs
   # ./mod13.lab4.sh
   ```

   On the reboot, you should notice some failures in the rc scripts. In fact, it will probably hang at "Start NFS Client Subsystems". Break out of it and determine the problem.




© 2005 Hewlett-Packard Development Company, L.P.

**© 2005 Hewlett-Packard Development Company, L.P.**

# Module 14 — Troubleshooting Network Problems: Part II

### Objectives

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

- Discuss NFS server setup issues.
- Troubleshoot a stale NFS file handle.
- Explain DNS server configuration files and setup issues.
- Understand the usage of **`nslookup`** and **`nsquery`**




© 2005 Hewlett-Packard Development Company, L.P.

## 14–1. SLIDE: Module Objectives

### Module Objectives

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

- Discuss NFS server setup issues.
- Troubleshoot a stale NFS file handle.
- Explain DNS server configuration files and setup issues.
- Explain the differences between `nslookup` and `nsquery`

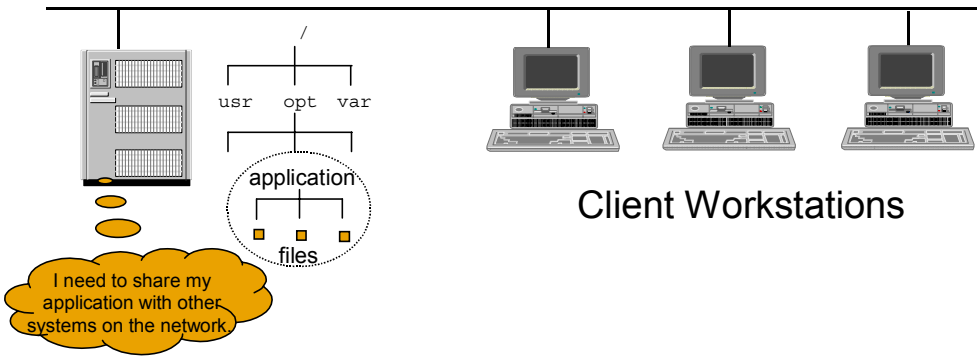
### Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 14–2. SLIDE: NFS Functionality

### NFS Functionality

- NFS is a network-based tool for sharing files and directories across the network.
- NFS allows transparent access to files from any node on the LAN.




## Student Notes

The NFS protocol provides transparent remote access to shared file systems across networks. The NFS protocol is designed to be independent of machine, operating system, network architecture, and transport protocols. This independence is achieved with remote procedure call (RPC) primitives built upon an eXternal data representation (XDR). Implementations of the NFS protocol exist for a variety of machines, from personal computers to supercomputers.

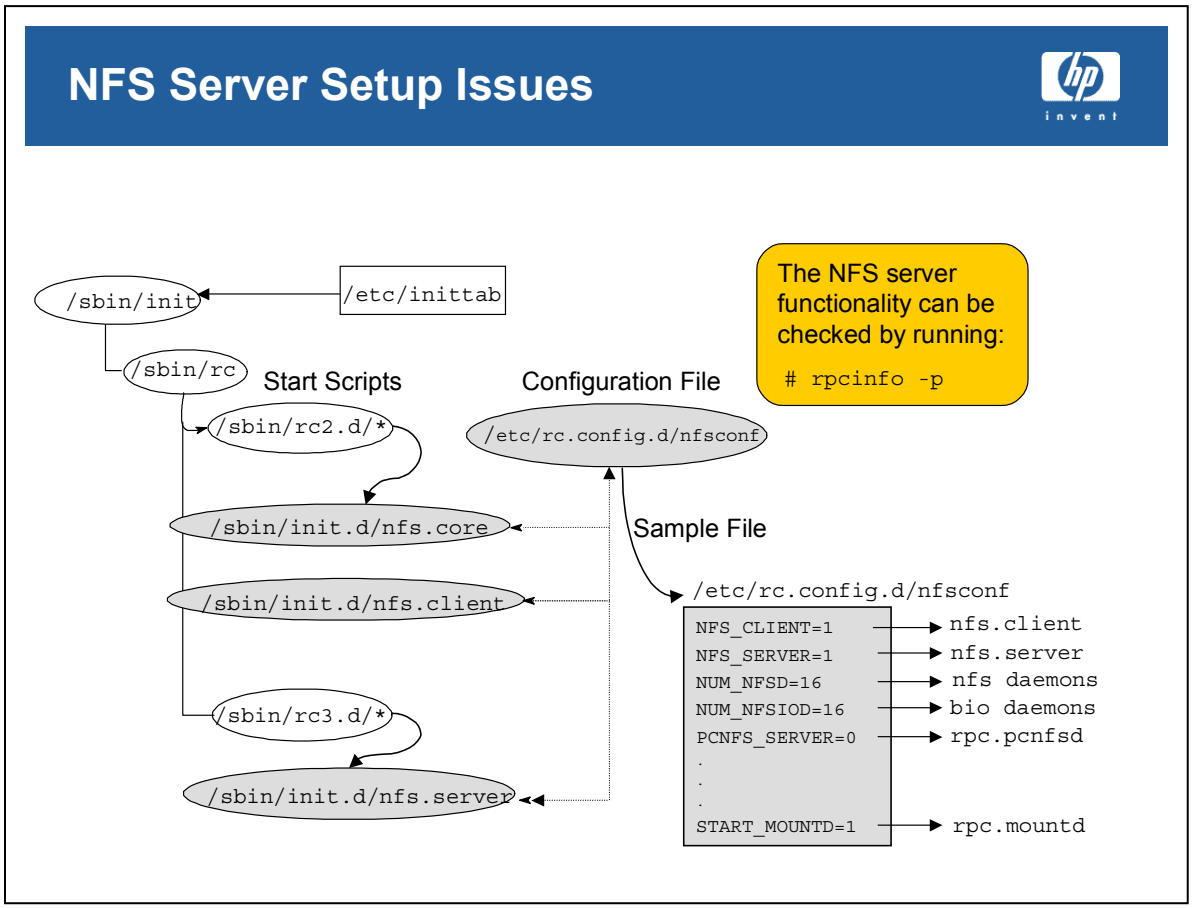
The supporting MOUNT protocol performs the operating system-specific functions that allow clients to attach remote directory trees to a point within the local file system. The mount process also allows the server to grant remote access privileges to a restricted set of clients via export control.

The Lock Manager provides support for file locking when used in the NFS environment. The Network Lock Manager (NLM) protocol isolates the inherently stateful aspects of file locking into a separate protocol.




© 2005 Hewlett-Packard Development Company, L.P.

## 14–3. SLIDE: NFS Server Setup Issues




## Student Notes

When the system starts up to run level 2 or higher, the startup scripts (link scripts) in **`/sbin/rc2.d`** will be executed to start NFS core and NFS client functionality. When the system starts up to run level 3 or higher, the startup scripts (link scripts) in **`/sbin/rc3.d`** will also be executed to start up NFS server functionality.

The startup scripts are linked to the run scripts that reside in **`/sbin/init.d`**. The run scripts fetch the configurable parameters from the **`/etc/rc.config.d/nfsconf`** configuration script, but the daemons are invoked only if the control variables are set to 1.

*NOTE:* By default, HP-UX attempts to start the Automounter at boot time from the NFS-client start script. Since this requires the networking configuration to be working, networking failures earlier in the boot sequence can cause the boot process to *hang* at this point.

You can break out of the boot in these cases by issuing a **`<cntl> \`** (the default quit sequence). This will terminate the **`/sbin/rc`** script, but still allow you to login on the console and fix the problem.


**© 2005 Hewlett-Packard Development Company, L.P.**

*NOTE:* Because NFS client functionality starts at run level 2 and the NFS server functionality doesn't start until run level 3, there can be issues where 2 systems need to cross mount each others' file systems. Options include: 1) Use the background option for the mounts, and 2) use Automounter/AutoFS.

### `/usr/bin/rpcinfo`

The **`/usr/bin/rpcinfo`** command verifies whether certain NFS programs are running. It makes an RPC call to an RPC server and reports what it finds.

The options of the syntax are as follows:

**`-p`** Probes the portmapper on the host and prints a list of all registered RPC programs. If a host is not specified, it defaults to the value returned by hostname.

**`-u`** Makes an RPC call to procedure `0` of the program on the specified host using UDP, and reports whether a response was received.

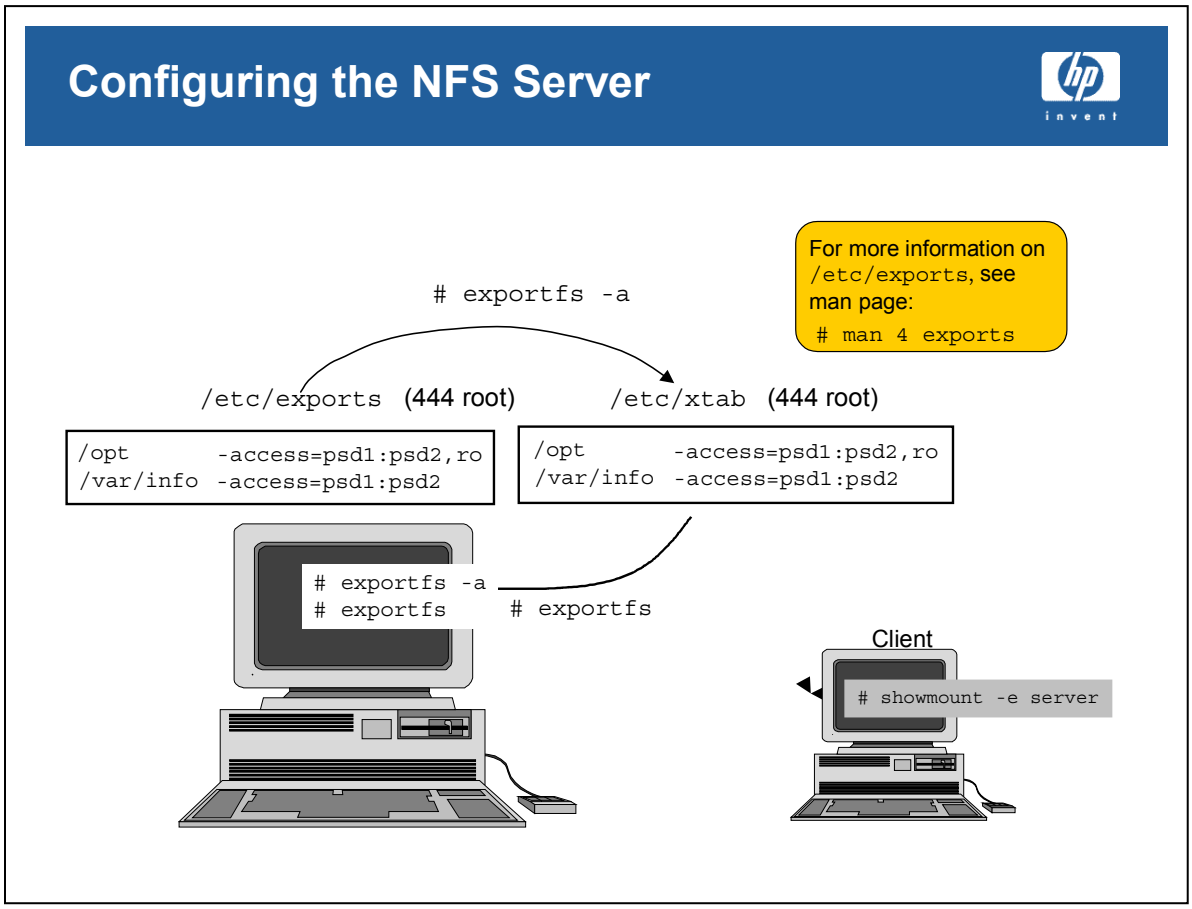
**`-t`** Makes an RPC call to procedure `0` of the program on the specified host using TCP, and reports whether a response was received.

*NOTE:* If you cannot mount an NFS file system, a possible reason is that the **`mount`** daemon is not running on the server. You can check this very easily with **`rpcinfo`**.




© 2005 Hewlett-Packard Development Company, L.P.

## 14–4. SLIDE: Configuring the NFS Server




## Student Notes

An NFS server machine exports a file system, a directory, or a file via the **/etc/exports** file to allow NFS client systems mount access. When you boot the NFS server, the necessary daemons are automatically invoked and the **exportfs -a** command is executed. The **exportfs -a** command looks in **/etc/exports** and transfers the export entries to the **/etc/xtab** file. The **/etc/exports** ASCII file is the configuration file, which can be edited with vi or the editor of your choice. **/etc/xtab** is be accessed by the mount daemon (**mountd**) each time a mount request is received. Simply put, **/etc/exports** is the configuration file, and **/etc/xtab** contains the enabled export entries.

The concept of exporting is the most important security feature of an NFS server system. You can export to selected client systems, and you can specify certain access options (for example, read-only).

You can manually export and unexport directories after the server is up, or change access permissions of the exported directory. The superuser can run the **exportfs** command at any time to alter the list of characteristics of exported directories. You must invoke **exportfs** every time **/etc/exports** is modified.


© 2005 Hewlett-Packard Development Company, L.P.

The table below shows the most common export option combinations. The first column shows common export options. The remaining three columns indicate which clients would be able to access each file system, and how, given the access option listed on the left (**rw**="read and write access allowed", **ro**="read-only access allowed").

**Table 1**

| Export Options Used | hosta | hostb | Others |
|---|---|---|---|
| /home -access=hosta | rw | — | — |
| /home -access=hosta:hostb | rw | rw | — |
| /home | rw | rw | rw |
| /home -rw=hosta:hostb | rw | rw | ro |
| /home -rw=hosta | rw | ro | ro |
| /home -ro | ro | ro | ro |
| /home -access=hosta:hostb,ro | ro | ro | — |
| /home -access=hosta,ro | ro | — | —— |

By default, **root** on the client systems is treated as user **nobody** when processing files on NFS servers. In order to grant NFS clients **root** access, the **root** option to the **export** command must be used. If a file system is exported to a client with the **root** option, then the user **root** on that client will have **root** permission on that file system. The table below shows several examples using the **root** export option:

**Table 2**

| Export Options Used | hosta | hostb | Others |
|---|---|---|---|
| /home -root=hosta,access=hosta | root+rw | — | — |
| /home -root=hosta,access=hosta:hostb | root+rw | rw | — |
| /home -root=hosta | root+rw | rw | rw |
| /home -root=hosta,rw=hosta:hostb | root+rw | rw | ro |
| /home -root=hosta,rw=hosta | root+rw | ro | ro |
| /home -root=hosta,rw=hosta,access=hosta:hostb | root+rw | ro | — |
| /home -root=hosta,access=hosta | root+rw | — | — |

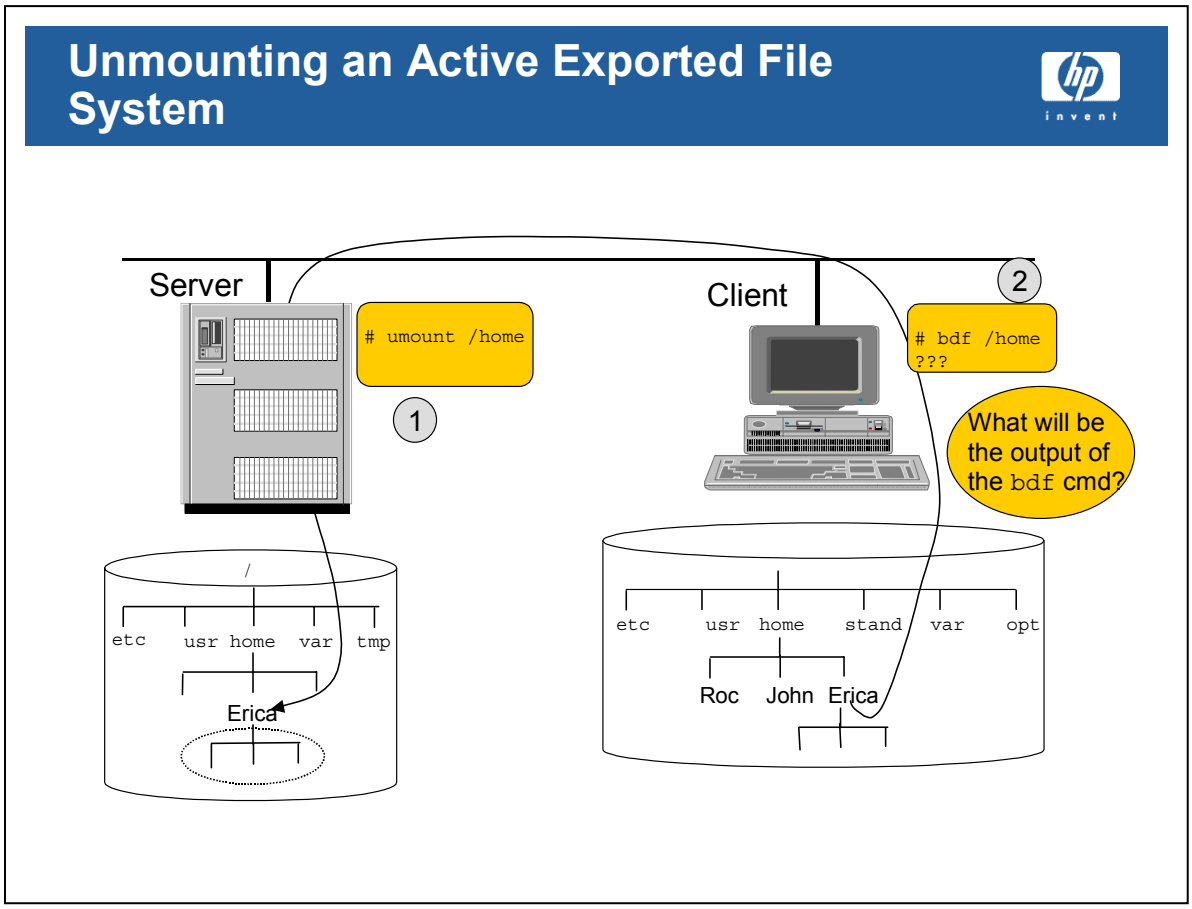
A common problem experienced with configuring an NFS server is using incorrect syntax within the **/etc/exports** file. Setting up specific access privileges for specific hosts can be tricky. If you experience problems, consult the man pages on the **/etc/exports** file:

```
# man 4 exports
```




© 2005 Hewlett-Packard Development Company, L.P.

## 14–5. SLIDE: Unmounting an Active Exported File System




## Student Notes

If an NFS mounted directory is unexported with **`exportfs -u`**, any access to the directory by the client causes an error.

On the server:

```
server # exportfs -u /home/Erica
```

On the client:

```
client # ls
. unreadable
```

The mounted area is immediately inaccessible to the client system, but **`/etc/mnttab,`** which contains information about all mounted file systems on the client, still contains an entry for this file system. This entry can be removed with the **`umount`** command. Simply issue the following command on the client system:

```
client # umount /home/Erica
```


**© 2005 Hewlett-Packard Development Company, L.P.**

If any files or directories on mounted file system were in use, the **umount** command will fail with a device busy error.

```
client # umount /home/Erica
nfs umount: nfs_unmount: /home/Erica: is busy
```

In order to **umount** the file system, you need to find all the processes using it. The **fuser** command can be used to list these processes. When using **fuser** to list the users of a file system you can either give **fuser** the device file the file system is built on, or you can use the **-c** option with the **fuser** command and the mountpoint directory. With NFS, you don't have a device file, so you must use the **-c** option. If you **fuser** without the **-c** option on the mountpoint it will only show the processes using the directory itself, not anything under that directory.

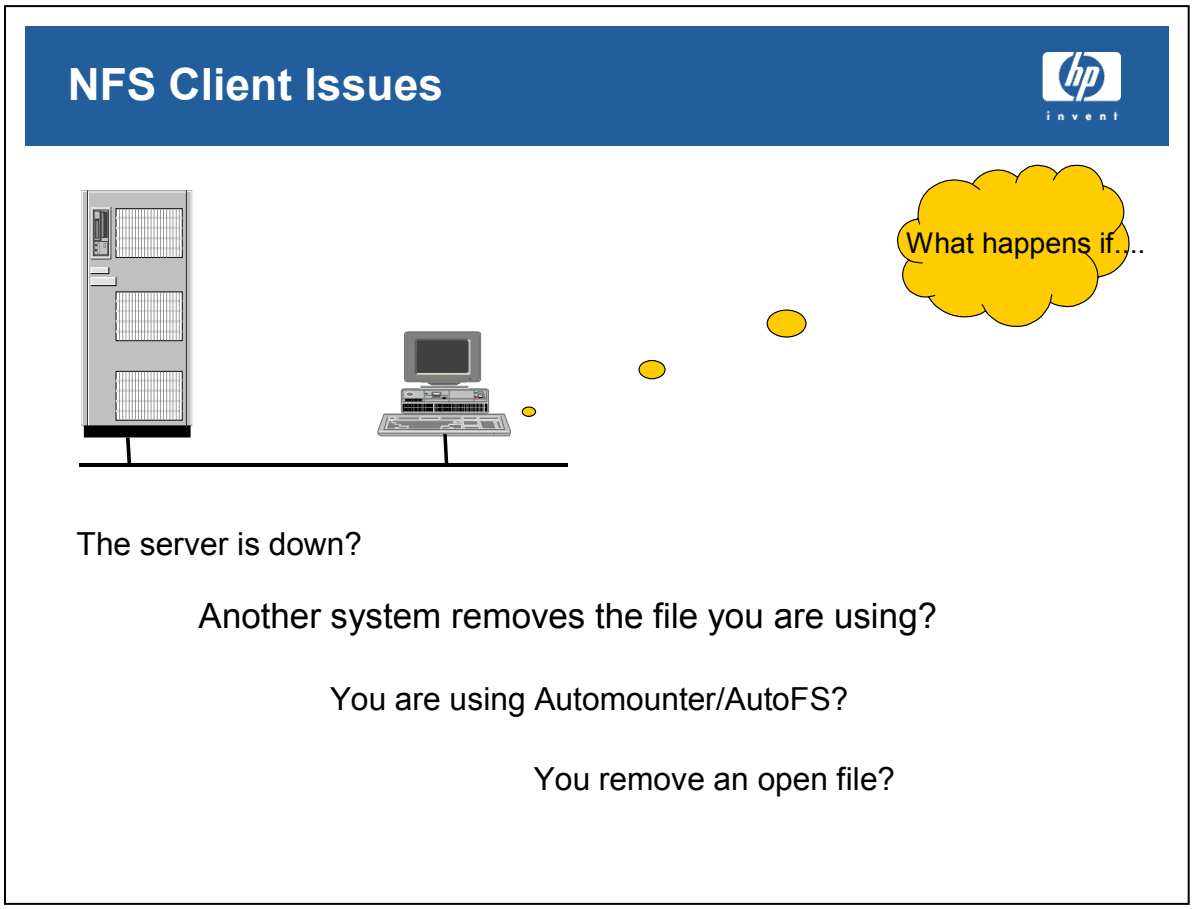
```
client # fuser -cu /home/Erica
/home/Erica:     1201c(root)    1347c(root)
```

*NOTE:* The **fuser -k** option performs a **kill -9** on all the processes listed. This does not give the processes a chance to exit cleanly, which can cause data corruption problems in many applications.




© 2005 Hewlett-Packard Development Company, L.P.

## 14–6. SLIDE: NFS Client Issues




## Student Notes

NFS is stateless. That is, NFS servers and clients do not know each others' current states. If a server becomes unavailable, the client is not notified. The next time the client attempts to contact the server, it may hang, (depending on mount options used), until the server is rebooted or becomes available again. Some NFS mount options that affect this behavior are:

1. Soft or hard mounts (default=hard)

   With soft mounts, the client request will time out and return an error.
   With hard mounts, the client will continue the retry request until the server responds

2. **`intr`** or **`nointr`** (default=**`intr`**)

   This option controls the behavior when working with hard mounts and the server is not responding.

   With the **`intr`** option, the client system will periodically check to see whether the application has been killed; if so, it wakes up the hung thread, allowing the process to die. Note that this is not immediate, and can take over a minute to finish.

With the **`nointr`** option, this checking does not occur and the client thread will stay in the high priority sleep state until either the NFS server does respond or the system is rebooted.


**© 2005 Hewlett-Packard Development Company, L.P.**

Since the thread is in a high priority sleep state, it cannot wake up to handle signals, so even a **`kill -9`** will not resolve this problem. The only way to kill the process is to perform a reboot.

This was the original behavior of NFS from its earliest days. This option probably only exists to offer backward compatibility, as it is unlikely that anyone would want to use this option in practice.

Normally, if you delete a file that is currently in use, the only thing actually removed is the filename. The internal references to the file still exist and therefore, so does the file. With NFS, the server has no state information about the client and therefore has no internal references to keep. Thus, there are two situations that can happen when removing files within NFS mounted directories or file systems.

1. The client deletes a file that it is using.

```
server / # exportfs -i /var/tmp


client / # mkdir /tmp/nfsTest
client / # mount owl:/var/tmp /tmp/nfsTest
```

In this example, we will NFS the directory **`/var/tmp`** from the server onto the client.

```
client / # cd  /tmp/
client /nfsTest #
client /nfsTest # while true
> do
> date
> sleep 1
> done > output.file &
[1]     13514
```

Create a file and keep it open.

```
client /nfsTest # ls -ltr | tail -4
-rw-rw-rw-   1 root       sys           30 Aug 10 22:44
WAAa1NBAa10180
-rw-rw-rw-   1 root       sys            0 Aug 10 22:45 BAAa10180
-rw-rw-rw-   1 root       sys            0 Aug 10 22:46 sh10568.1
-rw-rw-rw-   1 4294967294 nogroup     1711 Aug 11 01:26
output.file
```

Now remove the file.

```
client /nfsTest # rm output.file
client /nfsTest # ls -ltr | tail -4
-rw-rw-rw-   1 root       sys           30 Aug 10 22:44
WAAa1NBAa10180
-rw-rw-rw-   1 root       sys            0 Aug 10 22:45 BAAa10180
-rw-rw-rw-   1 root       sys            0 Aug 10 22:46 sh10568.1
-rw-rw-rw-   1 4294967294 nogroup     2610 Aug 11 01:27 .nfsEEB7
```




© 2005 Hewlett-Packard Development Company, L.P.

The file was not removed, but rather renamed.

2. Another system, the server or another client deletes the file you are using.

```
client /nfsTest # ./testprog1 testfile &
```

The test program opens and regularly accesses **`testfile`**. Now delete **`testfile`** from the server.

```
server # rm /var/tmp/testfile
```

On the client, we now see the following error:

```
client /nfsTest # read error: Stale NFS file handle

[1] +  Done(2)                    ./testprog1 testfile &
```

As we can see, the program receives a stale NFS file handle error.

If the deleted file is the program file, the next time the client needs to access the program file, it should die with a "text modification error."

Another area where problems can occur with the client side of NFS is with the Automounter. From HP-UX 11.0 and later, there are two versions of the Automounter shipped: The older NFS PV2 Automounter, and the newer NFS PV3 AutoFS. By default, Automounter is configured.

The older Automounter has several problems that the newer AutoFS attempts to resolve:

1. It is single threaded. It can only handle one request at a time. If a second process attempts to access a new file system via the Automounter while it is still handling a earlier request, the second process must wait to the first to finish.

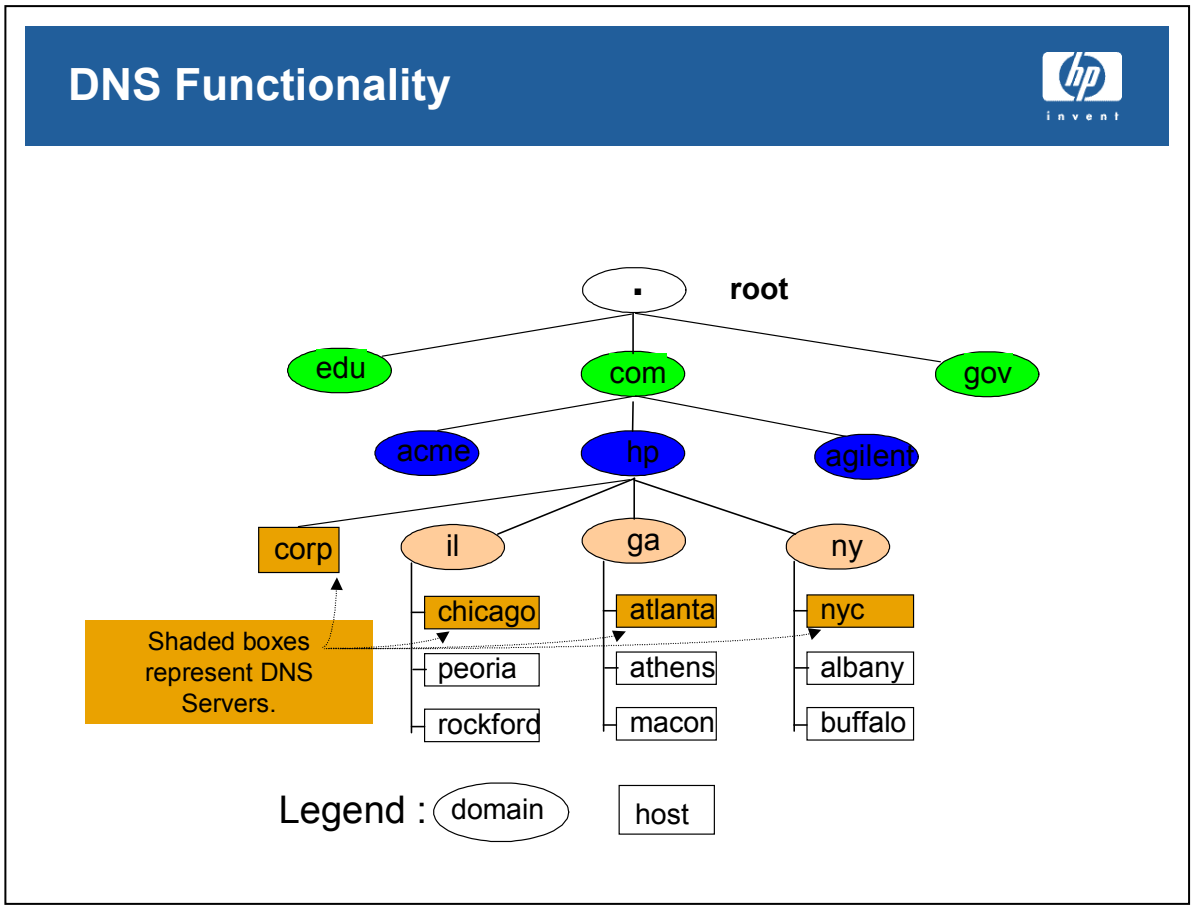
2. If the Automounter is killed or dies, the system needs to be rebooted, as any processes accessing mount directories served by automount will hang.

3. It is not always possible to shutdown the Automounter cleanly.

4. Any changes to the **`auto_master`** map require that Automounter be restarted.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 14–7. SLIDE: DNS Functionality




## Student Notes

DNS name space is a hierarchical method of naming network nodes based on a tree structure. The leveling of the tree is roughly based on organizational network boundaries.

Each subtree of the namespace is called a domain, and each domain has a label. The label that composes a domain name is printed from left to right, separated by dots: **`chicago.il.hp.com`**.

The node **`chicago`** belongs to the domain **`il`**; the domain **`il`** belongs to the domain **`hp`**; the domain **`hp`** belongs to the domain **`com`**. The domain "**`com`**" is below the "**`root domain`**".

The last dot at the end of a domain name indicates an absolute domain name, as opposed to a relative domain, like **`chicago.il`**.

The resolver routine translates relative names to absolute names.




© 2005 Hewlett-Packard Development Company, L.P.

## Resolving Host Names in the Local Domain

When querying host names in the local domain, the DNS client sends the request to the local DNS name server to resolve the host name into an IP address. If the DNS name server(s) do not respond, the user program hangs a short while before timing out and trying to contact an alternate server if specified in the **`resolv.conf`** file.

## Resolving Host Names in Other Domains

When querying host names in other domains, the DNS client sends the request to the local DNS name server to resolve the host name into an IP address. Since the requested host name is not found within the local zone data, the DNS server performs a recursive search of other domains for the hostname.

The sequence of events that occurs when performing the recursive search is described below:

The root server is queried. It provides the best answer it can: the address of the name server closest to the destination.

The local DNS server then queries the next name server, which responds with a referral to the next server. Finally, we get the server that has the responsibility (i.e. zone data) of the requested host. The provided answer is said to be an authoritative answer. This answer is definitive, as this server has the complete set of information for the requested host. The local DNS name server caches the addresses of all the name servers, plus the final answer.

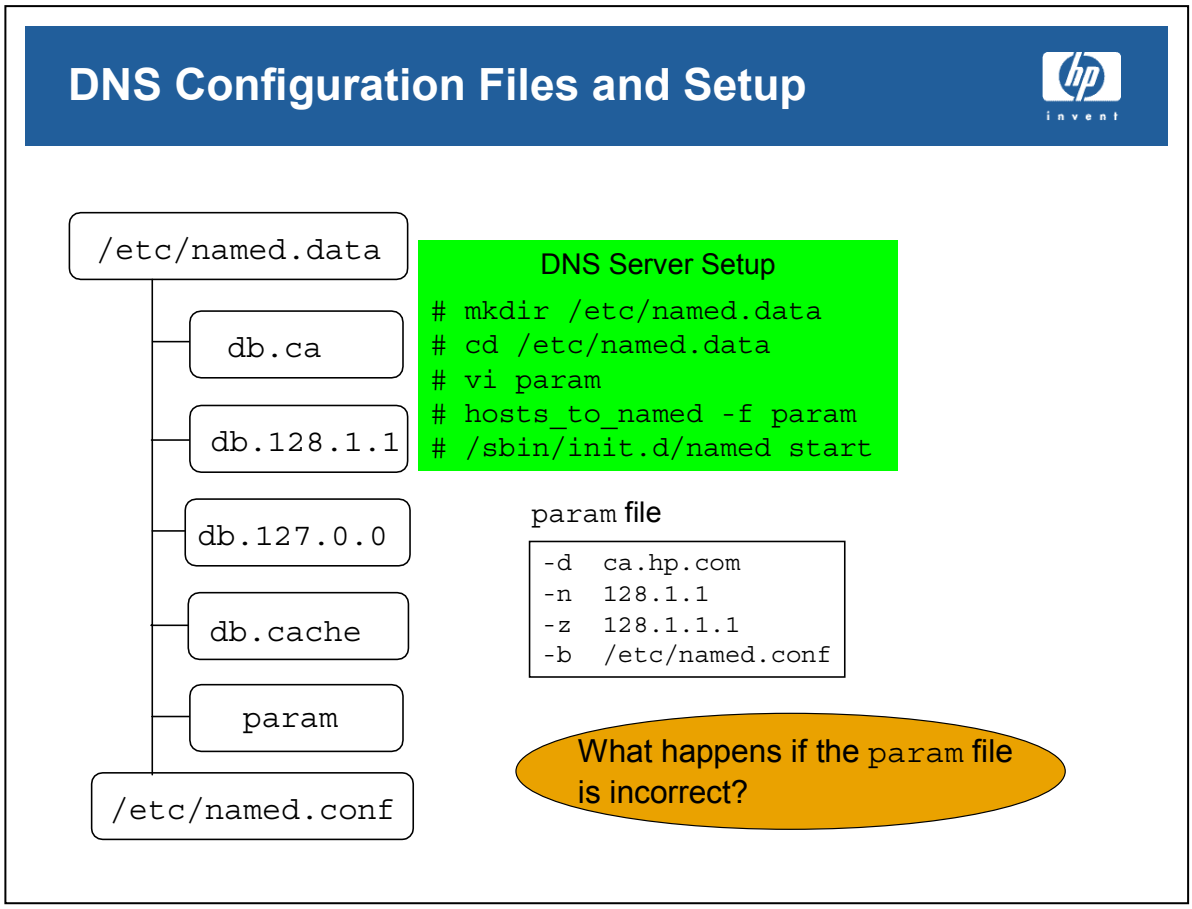
If another query is done to the local DNS server for the same host name, the local DNS server responds immediately with the cached data in the form of a non-authoritative answer.

Note that a server does not initially know its parent server or the servers in other branches of the name space. However, it will cache their addresses after a name resolution.

The root server is the primary server of the root domain. It does not have a specific configuration. Its task is not to own hosts, but rather to delegate to the domains below it. There are usually several root servers for backup purposes.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 14–8. SLIDE: DNS Configuration Files and Setup




## Student Notes

The step-by-step procedure for configuring a primary name server is shown below. The notes assume the node **sanfran** is being configured as a primary name server for the domain **ca.hp.com**.

1. Fully qualify hostnames in **/etc/hosts**.

   The **hosts_to_named** utility provided with HP-UX can create the DNS data files on your primary server using the information already in your **/etc/hosts** file. However, in order for this to work, all of the entries in your **hosts** file need to be converted to fully qualified hostnames. The old hostnames can be used as aliases.

   If you wish, you can delete lines in the **/etc/hosts** file that refer to domains for which your name server is not responsible. (Note, however, that the **localhost** entry must remain.) The example below shows the changes that are required on system **sanfran**:

```
vi /etc/hosts
127.0.0.1       localhost
128.1.1.1       sanfran.ca.hp.com  sanfran
128.1.1.2       oakland.ca.hp.com  oakland
128.1.1.3       la.ca.hp.com       la
```




© 2005 Hewlett-Packard Development Company, L.P.

2. Create a directory for the DNS database files.

   The **hosts_to_named** program will create several DNS data files. These data files are typically stored in a directory called **/etc/named.data**. Create the **/etc/named.data** directory manually with **mkdir**.

   ```
   # mkdir /etc/named.data
   # cd /etc/named.data
   ```

3. Create a **param** file for **hosts_to_named** in the directory created in step 2.

   The **hosts_to_named** utility is a powerful tool for building DNS database files. **hosts_to_named** can reference a **param** file to determine which domains your name server will serve.

   – Include a **-d** entry for each domain for which this name server will be responsible. Since some name servers serve multiple domains, you may have multiple **-d's**.

   – Include a **-n** entry for each (sub)net included in this domain. Since many domains include hosts on several subnets, the **param** file may have multiple **-n's**.

   – The **-b** option determines where your DNS boot configuration file will be stored. **/etc/named.conf** is the standard location.

   – Any secondary name servers that you create need to download a configuration file containing the IP address of the primary name server and other information about the domain. The **-z** option in the **param** file creates this configuration file, which is downloaded by the secondary name servers.

   – Other options can be specified in this file as well — see the **hosts_to_named** man page for details.

   The **param** file for the **sanfran** name server looks like this:

   ```
   vi /etc/named.data/param
      -d ca.hp.com.          ← Use your domain name(s) here
      -n 128.1.1             ← Use your subnet address(es) here
      -z 128.1.1.1           ← Use your primary server's IP here
      -b /etc/named.conf
   ```

4. Create the DNS data and boot files with **hosts_to_named.**

   The **hosts_to_named** utility automatically creates all the DNS data files needed to resolve hostnames and IP addresses in your domain using your **/etc/hosts** file and the options defined in your **param** file.

   ```
   # hosts_to_named -f param
   Translating /etc/hosts to lower case ...
   Collecting network data ...
           128.1
   Creating list of multi-homed hosts ...
   Creating "A" data (name to address mapping) for net 128.1 ...
   Creating "PTR" data (address to name mapping) for net 128.1 ...
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
Creating "MX" (mail exchanger) data ...
Building default named.boot file ...  ← Note: hosts_to_named still says
                                                          named.boot
Building default db.root file ...
Building default boot.sec.save for secondary servers ...
Building default boot.sec for secondary servers ...
Building default boot.cacheonly for caching only servers ...
done
```

5. Download a **db.cache** file with pointer to the root server.

   The **hosts_to_named** utility creates all of the necessary DNS database files except one. You must download the **db.cache** file with the addresses of the root-level name servers. You can **ftp** a file containing the current root server list from the internic web site. Since the list of root servers changes from time to time, you will need to download this file on a regular basis.

   For the exercises in class, we will download this file from the instructor's station at IP address **128.1.0.1**, rather than the internic.

```
# cd /etc/named.data
# ftp 128.1.0.1
    ftp > get /etc/named.data/db.cache
    ftp > quit
```

6. Modify **/etc/rc.config.d/namesvrs**. Start **named** manually. A reboot is not needed.

   In order to ensure that the name server daemon, **named**, starts during the boot process, set the NAMED variable in the **/etc/rc.config.d/namesvrs** configuration file to "1."

```
# vi /etc/rc.config.d/namesvrs
```

```
...
NAMED=1
NAMED_ARGS=""   (optional)
...
```

```
# /sbin/init.d/named start
```

**named** can start with arguments, but usually the **NAMED_ARGS** variable is not set.

Configuring client functionality is covered later in the module.




© 2005 Hewlett-Packard Development Company, L.P.

## Options in the `param` File

The options in the **`param`** file are as follows:

**`d`** Specifies the default domain. The **`db.ca`** file is created. Other domains can be served by adding lines with this option.

**`n`** Specifies data to be created for the given network number.

**`r`** Indicates that this primary server is also a root server. The **`db.root`** file is created.

**`z`** *and* **`Z`** Creates **`boot.sec.save`** and **`boot.sec`**, which are transferred to a secondary server's **`boot`** file.

**`b`** Specifies the name and location of the **`boot`** file.

See the **`man`** page for other **`hosts_to_named(1M)`** options.

## The Created Data Files

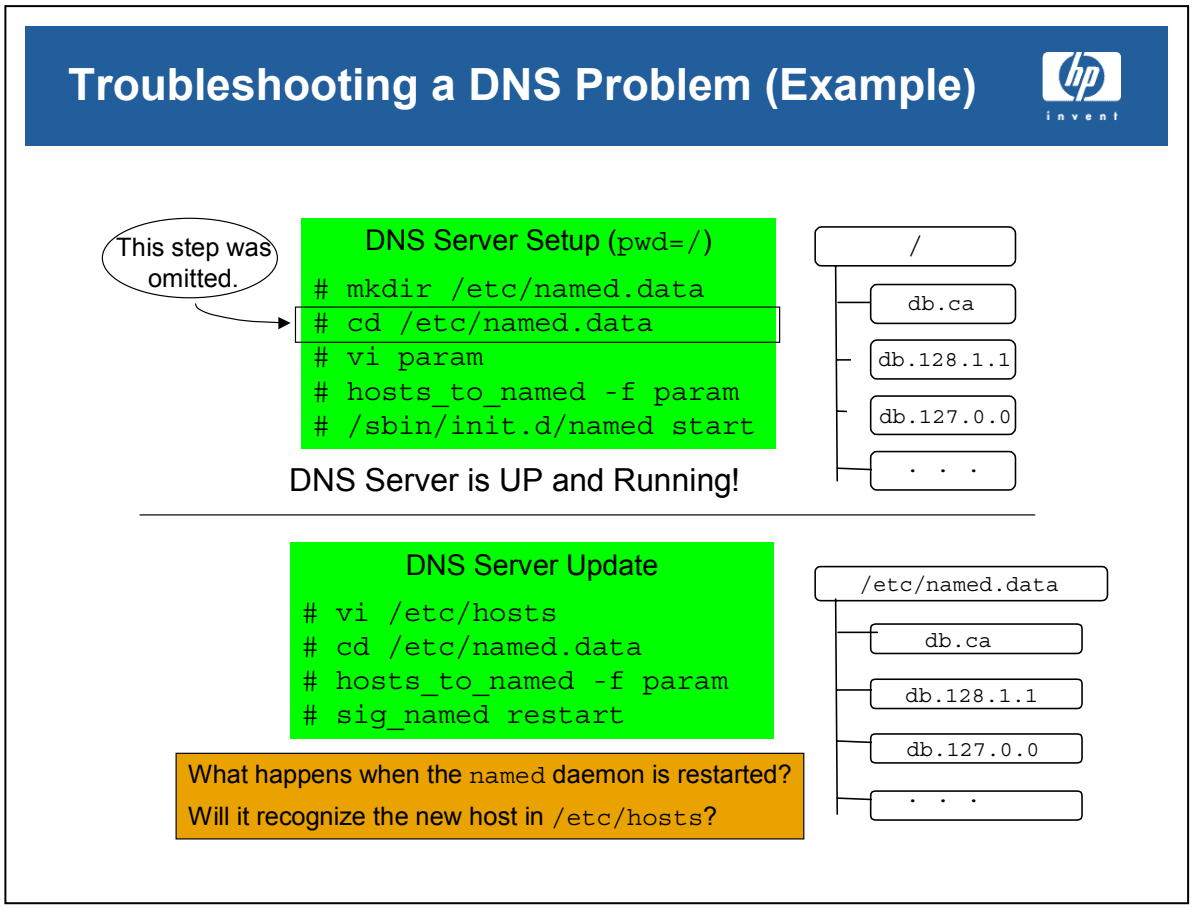
The following data files are created by the **`hosts_to_named`** command:

| | |
|---|---|
| `/etc/named.conf` | Configuration file for **`named`** |
| `/etc/named.data/db.ca` | Database file for domain **`ca`**; maps host names to IP addresses |
| `/etc/named.data/db.127.0.0` | Database file for the local loopback network; maps IP addresses to host names |
| `/etc/named.data/db.128.1` | Database file for network 128.1; maps IP addresses to host names |
| `/etc/named.data/db.root` | For root servers only |
| `/etc/named.data/boot.sec` | Transfer to secondary servers |
| `/etc/named.data/boot.sec.save` | Transfer to secondary servers |
| `/etc/named.data/boot.cacheonly` | Transfer to cache only servers |

*NOTE:* If the **`param`** file is incorrect, the DNS data files created by **`host_to_named`** will not be accurate. The **`param`** file controls DNS data files created and their contents.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 14–9. SLIDE: Troubleshooting a DNS Problem (Example)




## Student Notes

A very common DNS server problem is noted above. Initially, when the DNS server was being set up, the working directory was "/". When the **hosts_to_named** command was issued, all the DNS data files were created in "/". Consequently, the DNS boot file, **/etc/named.conf,** references root as the directory that holds the DNS data files.

Later, when a new host is added to the DNS domain, the **/etc/hosts** file is updated by adding the new host. The working directory is changed to **/etc/named.data**. The **hosts_to_named** command is issued to create the new DNS data files. Finally, the **named** daemon is restarted in an attempt to read the new DNS data files.

When the named daemon is restarted, it reads the DNS boot file, **/etc/named.conf**. The DNS boot file has a pointer to the directory where the DNS data files are located. Unfortunately, the pointer points to the original directory in which the DNS data files were created (in this case, "/"), which holds the old, unmodified DNS data files. Therefore, when a query is made referencing the new host, the **named** daemon is not able to resolve the new host name.




© 2005 Hewlett-Packard Development Company, L.P.

## 14–10. SLIDE: Testing Name Servers with `nslookup`

### Testing Name Servers with `nslookup`

## Student Notes

You can ensure that your DNS name servers are functioning properly using the **`nslookup`** command. If your host has already been configured with DNS client functionality, simply type:

```
# nslookup corp.hp.com   (simple host name lookup)
# nslookup 128.1.0.1     (simple IP address lookup)
```

**`nslookup`** uses your default name server as configured in **`/etc/resolv.conf`** and responds back with the IP or host name you requested.

Alternately, if you haven't yet configured DNS client functionality, or if you wish to override the default name server listed in **`/etc/resolv.conf`**, you may wish to run **`nslookup`** interactively:

```
# nslookup
> server 128.1.1.1  (try some lookups using the master server)
> corp.hp.com
> 128.1.0.1
> server 128.1.1.2  (now test the slave server, too)
```


© 2005 Hewlett-Packard Development Company, L.P.

```
> corp.hp.com
> 128.1.0.1
> exit
```

There are many other commands available within **nslookup** for troubleshooting your DNS name servers. At the "**>**" prompt, you can enter a "**?**" for a list of available tools within **nslookup**.

### Question

You may notice that **nslookup** sometimes returns a "Non-authoritative answer."
In fact, if you look up the same host name twice, you may notice that only the second response from **nslookup** is marked as "Non-authoritative." Can you guess why?




© 2005 Hewlett-Packard Development Company, L.P.

## 14–11. SLIDE: Testing Resolvers with `nsquery`

### Testing Resolvers with `nsquery`

```
# nsquery hosts sacramento
```

```
Using "dns [NOTFOUND=continue] files" for the hosts policy

Searching dns for sacramento.ca.hp.com
sacramento was NOTFOUND
Switch configuration: Allows fallback

Searching /etc/hosts for sacramento.ca.hp.com
Hostname: sacramento.ca.hp.com
Aliases:
Address:  128.1.1.4
Switch configuration: Terminates search
```

## Student Notes

At HP-UX 11.x, you should use the **`nsquery`** command to test your resolver configuration:

```
# nsquery hosts sacramento.ca.hp.com
# nsquery hosts 128.1.1.4
```

The **`nsquery`** command first checks your **`/etc/nsswitch.conf`** file to determine which switch policy you have chosen to use. If you have chosen **`/etc/hosts`**, then **`nsquery`** simply searches the **`/etc/hosts`** file for the host name or IP address you have specified.

If you have chosen to use DNS as a lookup source, **`nsquery`** checks **`/etc/resolv.conf`** to find the address of your default name server, and forwards the resolution request accordingly. If the first name server times out, **`nsquery`** will try the second name server listed in **`/etc/resolv.conf`**. If none of the name servers in **`/etc/resolv.conf`** respond, **`nsquery`** displays a message indicating that the DNS lookup failed, then follows the "fallback" policy defined in your switch file to choose another lookup service.


**© 2005 Hewlett-Packard Development Company, L.P.**

**`nsquery`** reports the result of each lookup service consulted, so you can determine if your switch policy behaves as expected.

---

***CAUTION***: At HP-UX 10.20, use the **`nslookup`** command to test your resolvers. At HP-UX 11.0, **`nslookup`** was unable to interpret the **`/etc/nsswitch.conf`** file properly. The **`nsquery`** command is now the preferred command for testing the fallback resolution method defined in the **`/etc/nsswitch.conf`** file.

---




© 2005 Hewlett-Packard Development Company, L.P.

## 14–12. SLIDE: Module Objective

### Module Objectives

You should now be able to do the following:

- Discuss NFS server setup issues.
- Troubleshoot a stale NFS file handle.
- Explain DNS server configuration files and setup issues.
- Explain the differences between `nslookup` and `nsquery`

### Student Notes


© 2005 Hewlett-Packard Development Company, L.P.

## 14–13. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What happens to a mounted client-side NFS file system if it gets unexported on the NFS server system? What do you do to rid the problem from the NFS client system?

2. What NFS server-side configuration file can cause problems when attempting to export NFS file systems? What is one of the most common problems experienced with this file? How can you fix that?

3. What is the name of the first DNS configuration file that is read when the **`named`** daemon is started? What is the purpose of this file? What could happen if the information contained within is incorrect?




© 2005 Hewlett-Packard Development Company, L.P.

# 14–14. LAB: NFS Problems

## Directions

In this lab you will work with a partner to experiment with NFS problems. One of your systems will function as an NFS server, and the other will function as an NFS client. You should work *together* throughout the lab to ensure that you feel comfortable with both the client and server functionalities of NFS. At this point, decide between yourselves who will be the server and who will be the client.

Host name of server: _____

Host name of client: _____

## Preliminary Steps

1. **(Client)**

   Install the lab files needed on your client:

   ```
   # cd /home/labs

   # ./mod14.lab1.sh
   ```

   You should now have two new user accounts defined in your **`/etc/passwd`** file: "mickie" and "minnie." The passwords for the new accounts are "mickie" and "minnie" respectively. Note that neither user has a home directory on your machine. You will mount their home directories from your partner's NFS server.

2. **(Server)**

   Install the lab files needed on your server:

   ```
   # cd /home/labs

   # ./mod14.lab1.sh
   ```

   This script creates several new files and directories, and two new user accounts, "mickie" and "minnie," in your **`/etc/passwd`** file. The passwords for the new accounts are "mickie" and "minnie" respectively. The script also creates home directories for mickie and minnie.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 1: Exporting and Mounting NFS File Systems

1. **(Server)**

   Your clients need to access several files on your server machine. Export the following with the export options set as noted. Make the file systems available to clients immediately, but also ensure that they will be available after the next system boot by adding them to **`/etc/exports`**.

   | | |
   |---|---|
   | `/home` | **`rw`** for your partner's machine, no access for other hosts |
   | `/opt/phone` | **`rw`** for your partner's machine, read only for all others |
   | `/opt/fun` | read only for everyone on the LAN |

2. **(Server)**

   What command can you use to see what file systems you have made available? Can you tell which export options you used?

3. **(Client)**

   Create mount points for the file systems your neighbor exported, and mount them:

   ```
   /home/mickie
   /home/minnie
   /opt/fun
   /opt/phone
   ```

4. **(Client)**

   What file needs to be modified to ensure that these NFS file systems are automatically mounted after every system boot? Make it so. (For now, use the "defaults" mount option.)

   Syntax errors in the **`/etc/fstab`** file may cause the next system boot to fail. Do a **`mount -a`** to ensure that you didn't make any mistakes in **`fstab`** file.

   Finally, use **`mount -v`** to ensure that all the NFS file systems actually mounted properly.




© 2005 Hewlett-Packard Development Company, L.P.

## Part 2: What If the NFS Client Loses LAN Connectivity?

1. During the remainder of the lab, you will be asked to shutdown your LAN card several times. Exit out of your X-windows/CDE login session, reconnect via the MP, and shutdown CDE.

   ```
   # /sbin/init.d/dtlogin.rc stop
   ```

2. **(Server and Client)**

   What happens if the NFS client loses LAN connectivity to the server? Execute the following commands and note the output. First note the client's behavior when the server is up. (It should be normal.)

   ```
   client# cd /opt/fun
   client# ls
   ```

   Now take the **server's** LAN card down and note what happens to the client:

   ```
   server# ifconfig lan0 down
   client# ls                    #This will hang indefinitely
   ```

   Move on to next step.

   What happens when the client regains connectivity to the NFS server?

   ```
   server# ifconfig lan0 up
   client# ls
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 3: What If the NFS Server Unexports a Client-Mounted File System?

**(Server and Client)**

1. What happens if the NFS server unexports a file system while an NFS client has it mounted? Do the following and note the output from the commands. First note the client's behavior when the NFS server's file system is exported. (It should be normal.)

```
client# cd /opt/fun
client# ls
client# bdf
```

Now unexport the NFS server's file system and note what happens to the NFS client:

```
server# exportfs -uv /opt/fun
client# bdf                    #You should see a stale NFS file handle!
```

Move on to next step.

How do you get rid of this "*stale file handle*" from the NFS client?

```
client# cd /
client# umount /opt/fun
client# bdf                    #The bad entry should be gone!
```

What happens when the NFS server re-exports and the client re-mounts the file system?

```
server# exportfs /opt/fun
server# exportfs
client# mount -F nfs "server":/opt/fun /opt/fun
client# bdf
```

Now **`umount`** all NFS file systems.

```
client# umount -aF nfs
```




© 2005 Hewlett-Packard Development Company, L.P.

## Part 4: Troubleshooting a Nonresponsive NFS Server

**(Client and Server)**

1. You have seen the effect that a downed NFS server has on NFS clients. What can the administrator on the client side do to determine what might be wrong on the server side? Start by doing some experiments while both your server and client are functioning properly. Make sure the LAN cards are in an "UP" state on both systems:

   ```
   server# ifconfig lan0 up
   client# ifconfig lan0 up
   ```

   Now test connectivity from the client to your NFS server:

   ```
   client# ping "server"
   client# rpcinfo -p "server"
   ```

**(Client and Server)**

2. Now shut down NFS on your NFS server.

   ```
   server# /sbin/init.d/nfs.server stop
   ```

   From the client, try mounting **/opt/fun**.

   `client# mount /opt/fun` # *After you see the error, hit* **Ctrl** + **c**.

**(Client)**

3. From the client, try your connectivity test commands again:
   ```
   client# ping "server"
   client# rpcinfo -p "server"
   ```

   Can you still **ping** the server?

   Which RPC programs are no longer available on the server?

   Will clients be able to mount NFS file systems now? Why?


**© 2005 Hewlett-Packard Development Company, L.P.**

Will clients be able to access already mounted NFS file systems? Why?

.

## Part 5: Cleaning Up

Before moving on to the next part of the lab, restore your network configuration parameters to what you started with (prior to the NFS lab) by typing:

```
# /home/labs/netfiles.sh -r NEW
```




© 2005 Hewlett-Packard Development Company, L.P.

# 14–15. LAB: DNS Data File Problem

## Purpose

This lab is designed to give you experience troubleshooting DNS data file problems.

1. Execute the **`mod14.lab2.sh`** script.

   ```
   # cd /home/labs
   # ./mod14.lab2.sh
   ```

   After running this script, your system will be configured as a DNS server and client. The DNS server will be able to resolve all host names to IP addresses in the **`/etc/hosts`** file except for one entry. That entry is as follows:

   ```
   x.y.9.1   philly.pa.hp.com
   ```

   (where x and y match the first two octets of your current IP address)

   Using the knowledge gained in the current module, fix the DNS server so that it recognizes and resolves the host name, **`philly.pa.hp.com`**, to the appropriate IP address.

### Cleaning Up

When you are finished with all labs, restore your ORIGINAL network configuration as follows:

```
# /home/labs/netfiles.sh -r ORIGINAL
```


© 2005 Hewlett-Packard Development Company, L.P.

# Solutions


© 2005 Hewlett-Packard Development Company, L.P.

© 2005 Hewlett-Packard Development Company, L.P.

# 1–14. QUIZ: Review Questions

## Directions

Answer the following questions. These questions serve as a review for this module.

1. List four HP-UX troubleshooting resources.

   **Answer:**

   Manuals, websites, and newgroups
   Printouts from system commands and configuration files
   System log files
   System logbook

2. Which "troubleshooting resource" do you value more than others? Why?

   **Answer:**

   The "troubleshooting resource" can be any resource listed in question #1.


© 2005 Hewlett-Packard Development Company, L.P.

# 1–15. LAB: System Log Files

## Purpose

This lab is designed to give you experience with logfiles.

## Directions

Read and answer the following questions, which serve as a review of the module on "Your Troubleshooting Resources."

1. Normally, HP-UX's **inetd** does not log connections to **/var/adm/syslog/syslog.log**. Executing **inetd -l** enables the logging of network connections through **syslogd**. If this functionality is needed all the time, then it can be enabled at boot time by setting the option

   **export INETD_ARGS="-l"** *(the letter "ell")*

   in the file **/etc/rc.config.d/netdaemons**.

   First, examine the **/var/adm/syslog/syslog.log** file. Note the kinds of information it contains:

   ```
   # vi /var/adm/syslog/syslog.log
   ```

   | | |
   |---|---|
   | *NOTE:* | We really do not want to accidentally write to the system log file via vi. vi makes a copy of the file, edits it, and then writes back to the original. Any messages logged while in vi would be lost if you saved the file. |

   a. How much physical memory does your system contain?

   b. Is the **ftp** service available?

   c. Is the **rlogin** service available?

   d. Is the **telnet** service available?

   e. Is the **remsh** service available?

   f. Are there any **telnet** requests from other machines?

2. Look at the **inetd** process using the standard **ps** command:

   ```
   # ps -ef | grep inetd
   ```

   Turn on more extensive logging.

   Properly kill the **inetd** process:

   ```
   # inetd -k
   ```

3. Restart **inetd**, but this time with logging enabled, by using the letter "*ell*" command-line option (not the number "*one*" ):


© 2005 Hewlett-Packard Development Company, L.P.

```
# inetd -l
```

| | |
|---|---|
| *NOTE:* | You can enable logging for **inetd** simply by executing **inetd -l**, but at the next shutdown, logging will be lost. Later in this lab, we will enable logging permanently. |

Repeat the same **ps** command you used earlier in this same lab question:

```
# ps -ef | grep inetd
```

4. After obtaining permission, **telnet** to a system near you. Login in as root; do a simple **ll** command; and then log out. After the person near you has **telnet**ed to your machine and logged out, re-examine **syslog.log**:

```
# cd /var/adm/syslog

# vi syslog.log
```

Do you find the **telnet** entries for the person sitting near you?

Turn on the logging capability so that at the next reboot, **inetd** will restart automatically with logging. First, change to the subdirectory where most of the startup configuration information is kept:

```
# cd /etc/rc.config.d
```

In this subdirectory, **vi** the **netdaemons** file. Notice that the comments near the top of the file instruct us to add the **-l** ("ell") option to turn on logging at bootup. Using **vi**, add this option:

```
# vi netdaemons
```

5. Properly shutdown and reboot your machine:

```
# shutdown -ry 0
```

When the machine has completely rebooted, look again at the **inetd** process using the standard **ps** command:

```
# ps -ef | grep inetd
```

6. **telnet** a second time to a person's machine sitting near you. Login as root, do a simple **ll** command, then log out. After the person near you has **telnet**ed to your machine and logged out, re-examine **syslog.log** file:

```
/var/adm/syslog/syslog.log
```

Do you find the **telnet** entries for the person sitting near you?

7. Set up your machine as an anonymous **ftp** site. The easiest way to do this is with SAM. First, access SAM, and then follow the instructions to get to SAM's correct submenu:

```
# sam

From the top level of SAM, double-click on "Networking and Communications"
```


© 2005 Hewlett-Packard Development Company, L.P.

```
From the second level of SAM, double-click on "Network Services"
From the third level of SAM, SINGLE-click on "Anonymous FTP", and then
                             SINGLE-click on "Actions", and then
                             SINGLE-click on "Enable".
```

8. Notice that a **/home/ftp** subdirectory now exists. Copy just *one* of the following files into the **pub** subdirectory below **/home/ftp**:

   **/etc/passwd** **/etc/hosts**
   **/etc/group** **/etc/inetd.conf**
   **/var/adm/syslog/syslog.log** **/etc/rc.config.d/netdaemons**

   Be sure that the file you copy has permissions so that it can be read by anyone.

   ```
   # chmod 644 /home/ftp/pub/< filename >
   ```

   Now enable logging within the **ftp** daemon itself.

   ```
   vi /etc/inetd.conf
   ```

   Find the line for the **ftp** daemon and change the option **-l** to **-L**.

   ```
   ftp      stream tcp nowait root /usr/lbin/ftpd     ftpd -L
   ```

   Save the file, and restart **inetd**.

   ```
   inetd –c
   ```

9. Ask a person sitting near you to use anonymous **ftp** to grab a copy of the file you placed in the **pub** directory. That person should do the following:

   a. **ftp** to your machine.

   b. At the login prompt, login as **anonymous** or **ftp**.

   c. At the **passwd** prompt, enter your complete email address.

   From the **ftp>** prompt, **cd** into the **pub** directory:

   ```
   ftp> cd pub
   ```

   In the **pub** directory, execute the **ftp "dir"** command to see the files currently in the **pub** directory:

   ```
   ftp> dir
   ```

   While still in the **pub** directory, **get** the file the other person previously placed there:

   ```
   ftp> get < file_name >
   ```

   And then exit **ftp**:

   **ftp> bye**

10. Finally, **telnet** a second time to a system near you. Login as root, do a simple **ll** command, and then log out. After the person near you has **telnet**ed to your machine and logged out, re-examine **syslog.log** file:

    **# vi /var/adm/syslog/syslog.log**


**© 2005 Hewlett-Packard Development Company, L.P.**

Do you find the **`telnet`** entries for the person sitting near you? Did you notice that, in addition to showing the anonymous **`ftp`** login, **`syslog.log`** also shows you what FTP operations the person did while there!


© 2005 Hewlett-Packard Development Company, L.P.

# 1–16. LAB: MANPATH Issue

## Purpose

This lab will investigate the MANPATH variable.

1. Change to the **`/home/labs`** directory.

2. Run the script: **`./mod1.lab1.sh`**

3. Then log out, and log back in, and try the following command:

   ```
   # man make_net_recovery
   ```

4. What happened? What do you need to do to fix this?

   **Answer:**

   The manual pages for HP-UX are stored in many directories. The environment variable **MANPATH** tells the man command which directories to search. This variable is set at logon time from within the file **/etc/MANPATH**.

   When Ignite-UX is installed, its manual directories, **/opt/ignite/share/man/*,** are added to the **MANPATH** file. However, not all software does this. In this case, these directories have been removed from the **/etc/MANPATH** file.

   To correct this, we can simply put the above directories back into **/etc/MANPATH**.


© 2005 Hewlett-Packard Development Company, L.P.

## 2–18. LAB: Creating a `make_net_recovery` archive

### Directions

Create a **`make_net_recovery`** archive of your system. Use the server designated by your instructor.

1. Verify that the Ignite-UX software is loaded. If not, install it. Ask your instructor for a depot location.

   ```
   # swlist -l product |grep Ignite-UX
   ```

2. Run **`make_net_recovery`** and choose a full vg00 archive.

   ```
   # make_net_recovery -s Ignite_Server -x inc_entire=vg00 \
           -x exclude=/tmp
   ```

3. When the above command completes, logon to the Ignite server and inspect the **`/var/opt/ignite/clients/<your client>/recovery/<date,time>`** directory. You should see several files that correspond to your archive.

   When the **`make_net_recovery`** completes check the file **`/var/opt/ignite/clients/<your client>/recovery/<date,time>flist`** to see how many files will be archived?

   ```
   # wc -l flist
   ```

   Will **`/home`** be included in the archive?

   ```
   # grep '^[0-9][0-9]* tmp/' flist
   ```

4. This archive will function as a backup in case something goes wrong in a future module.


© 2005 Hewlett-Packard Development Company, L.P.

## 2-19. LAB: Creating a `make_tape_recovery` archive (Optional)

### Directions

If you have a tape drive available and if you have time, perform a **`make_tape_recovery.`**

1. Insert a tape in the tape drive. Run **`make_tape_recovery`** in preview mode, specifying that you wish to create a minimum core recovery tape. (Ignore any vxvm errors)

   ```
   # make_tape_recovery -p
   ```

2. When the above command completes change directories to **`/var/opt/ignite/recovery`**. Do a long listing. Change to the **`latest`** directory. Do another long listing.

   ```
   # cd /var/opt/ignite/recovery
   # ll
   # cd latest
   # ll
   ```

   When the **`make_tape_recovery`** completes, check the file **`/var/opt/ignite/recovery/latest/flist`** to see how many files will be archived?

   ```
   # wc -l flist
   ```

   Will **`/home`** be included in the archive?

   ```
   # grep '^[0-9][0-9]* home/' flist
   ```

3. Run **`make_tape_recovery`** in preview mode, specifying that you wish to create an archive containing all the files and directories in the root volume group excluding **`/tmp`**. (Ignore any vxvm errors)

   ```
   # make_tape_recovery -p -x inc_entire=vg00 -x exclude=/tmp
   ```

4. When the above command completes change directories to **`/var/opt/ignite/recovery`**. Do a long listing. Change to the latest directory. Do another long listing. When **`make_tape_recovery`** runs in preview mode it renames the directory "**`latest`**" therefore you will need to **`cd`** out of the renamed "**`latest`**" directory and then **`cd`** back to latest to ensure that you are looking at the current information.

   ```
   # cd /var/opt/ignite/recovery
   # ll
   # cd latest
   # ll
   ```

   How many files will be archived?

   ```
   # wc -l flist
   ```


© 2005 Hewlett-Packard Development Company, L.P.

Will **/home** be included in the archive?

```
# grep '^[0-9][0-9]* home/' flist
```

Will **/tmp** be included in the archive?

```
# grep '^[0-9][0-9]* tmp/' flist
```

5. Create a recovery tape that includes all the files in all the volume groups containing the essential files, calling for an interactive restore. This step will take quite a while.

```
# make_tape_recovery -A -I
```

6. After the tape has been created, check the log file in **/var/opt/ignite/recovery/latest/recovery.log**.

## Optional

Check with your instructor for a convenient time to try to restore from your tape.


© 2005 Hewlett-Packard Development Company, L.P.

# 3–26. LAB: Recovering LVM Disks and Structures

## Directions

Carefully follow the instructions below. If you are using an Integrity system for this lab, check with your instructor for the appropriate disks to use for your new volume group.

## Part 1: Preliminary Steps

1. Configure a two-disk vg01 volume group. If you already have a vg01 volume group, you can use it in this lab, but be sure that the volume group has only two disks, and no logical volumes.

   **Answer:**

   ```
   # pvcreate -f /dev/rdsk/cxtxdx
   # pvcreate -f /dev/rdsk/cytydy
   # mkdir /dev/vg01
   # mknod /dev/vg01/group c 64 0x010000
   # vgcreate vg01 /dev/dsk/cxtxdx \
                   /dev/dsk/cytydy
   ```

2. Create a 16M non-mirrored logical volume called "nonmirrored" on the first disk in the volume group. Create a VxFS file system in the logical volume, and mount the file system on **/nonmirrored**, and copy a few files to the file system from **/usr/bin**.

   **Answer:**

   ```
   # lvcreate -n nonmirrored vg01
   # lvextend -L 16 /dev/vg01/nonmirrored /dev/dsk/cxtxdx
   # newfs /dev/vg01/rnonmirrored
   # mkdir /nonmirrored
   # mount /dev/vg01/nonmirrored /nonmirrored
   # cp /usr/bin/a* /nonmirrored
   ```

3. Create a 16M mirrored logical volume called "mirrored". Create a VxFS file system in the logical volume, mount the file system on /mirrored, and copy a few files to the file system from **/usr/bin**.

   **Answer:**

   ```
   # lvcreate -n mirrored vg01
   # lvextend -L 16 /dev/vg01/mirrored /dev/dsk/cxtxdx
   # lvextend -m 1 /dev/vg01/mirrored /dev/dsk/cytydy
   # newfs /dev/vg01/rmirrored
   # mkdir /mirrored
   # mount /dev/vg01/mirrored /mirrored
   # cp /usr/bin/a* /mirrored
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 2: Recreating Missing Device Files

1. Remove all the device files in the **/dev/vg01** directory.

   ```
   # rm /dev/vg01/*
   ```

2. How are the following commands impacted?

   ```
   # ls /mirrored
   # vgdisplay -v vg01
   # lvextend -L 32 /dev/vg01/mirrored
   ```

   **Answer:**

   The **ls** succeeds since the file system was already mounted. The LVM commands fail since the device files they reference no longer exist.

3. Recreate the volume group device file.

   **Answer:**

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

4. You should be able to **vgdisplay -v vg01** now, but something is missing from the output. What is it?

   **Answer:**

   ```
   # vgdisplay -v vg01
   ```

   The logical volumes aren't included in the output any more.

5. Recreate the logical volume device files, too.

   **Answer:**

   ```
   # mknod /dev/vg01/nonmirrored  b 64 0x010001
   # mknod /dev/vg01/rnonmirrored c 64 0x010001
   # mknod /dev/vg01/mirrored     b 64 0x010002
   # mknod /dev/vg01/rmirrored    c 64 0x010002
   ```

6. Verify the creation of the logical volume device files by executing step 4 again.

   **Answer:**

   ```
   # vgdisplay -v vg01
   ```

   The logical volumes are present in the output again.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 3: Rebuilding a Missing /etc/lvmtab

What happens when the **/etc/lvmtab** file is lost or corrupted? In this part of the lab you will find out!

1. Move **/etc/lvmtab** to **/etc/lvmtab.bkp**

   **Answer:**

   ```
   # mv /etc/lvmtab /etc/lvmtab.bkp
   ```

2. What impact does this have on the following commands?

   ```
   # ls /mirrored
   # vgdisplay -v vg01
   # lvextend -L 32 /dev/vg01/mirrored
   ```

   **Answer:**

   The **ls** command succeeds. Most other LVM commands fail.

3. What command can you use to recreate the **/etc/lvmtab** file? Make it so! How does this command determine which disks belong to which volume groups?

   **Answer:**

   ```
   # vgscan
   ```

   The command searches the PVRA/VGRA headers to determine which volume group each disk belongs to.

4. Verify your recovery succeeded. How can you view the contents of the **/etc/lvmtab** file?

   **Answer:**

   ```
   # strings /etc/lvmtab
   ```

## Part 4: What Happens when a Disk Fails?

This portion of the lab demonstrates the effect of a downed disk on volume group activation. We will simulate a disk failure by overwriting the LVM headers on the first disk in your volume group.

1. Write 64M of zeroes to the first disk in your volume group. This should be more than adequate to clobber the LVM headers.

   ```
   # dd if=/dev/zero of=/dev/rdsk/cxtxdx bs=8192 count=8192
   ```

2. Does **vgdisplay** indicate that there is a problem? Explain!


© 2005 Hewlett-Packard Development Company, L.P.

**Answer:**

No. **vgdisplay** uses cached information, so it isn't aware of the problem at this point.

3. Unmount your **/mirrored** and **/nonmirrored** file systems, deactivate the volume group, then try to reactivate. Does this succeed? Explain!

   **Answer:**

   ```
   # umount /mirrored
   # umount /nonmirrored
   # vgchange -a n vg01
   # vgchange -a y vg01
   ```

   Re-activation fails because activation quorum isn't present.

4. What can you do to activate the volume group without quorum? Make it so!

   **Answer:**

   ```
   # vgchange -a y -q n vg01
   ```

   This will still generate error messages, but at least the volume group can be activated.

5. Now does **vgdisplay** indicate that there is a problem?

   **Answer:**

   ```
   # vgdisplay -v vg01
   ```

   Yes! **vgdisplay** displays the volume group, but includes error messages regarding the missing physical volume.

6. Can you access your logical volumes? Attempt to mount **/dev/vg01/nonmirrored** and **/dev/vg01/mirrored**. What happens? Why?

   **Answer:**

   ```
   # mount /dev/vg01/mirrored /mirrored
   # mount /dev/vg01/nonmirrored /nonmirrored
   ```

   The mirrored logical volume should mount successfully, since the logical volume's data is still accessible via the remaining mirror.

   The non-mirrored logical volume mount request should fail, since the sole disk in the logical volume is no longer accessible.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 5: Replacing a Failed Disk

Now let's see if you can restore the failed LVM disk.

1. Use **vgcfgrestore** to restore the LVM headers to your failed disk.

   **Answer:**

   ```
   # vgcfgrestore -n vg01 /dev/rdsk/cxtxdx
   ```

2. Does LVM recognize the fact that the headers were restored? Execute **vgdisplay -v** to find out.

   **Answer:**

   ```
   # vgdisplay -v vg01
   ```

   **vgdisplay** still displays errors. LVM won't realize that the headers have been restored until you reactivate the volume group.

3. Do whatever is necessary to force LVM to recognize that the disk has been recovered – without taking the volume group offline again.

   **Answer:**

   ```
   # vgchange -a y vg01
   ```

   When executed on an already-active volume group, **vgchange** causes LVM to re-scan the headers. This should eliminate the **vgdisplay** error messages.

## Part 6: Restoring Logical Volumes

You restored the LVM headers, but your logical volumes still require some attention.

1. View the **lvdisplay -v** output for the **/dev/vg01/nonmirrored** logical volume. What is the LV Status?

   **Answer:**

   ```
   # lvdisplay -v /dev/vg01/nonmirrored
   ```

   The LV Status appears to be "Available", and the extents appear to be "Current". But is the data still in tact?

2. What is the status of the file system in the logical volume? Try running **fsck** to find out!

   **Answer:**

   ```
   # fsck -F vxfs /dev/vg01/nonmirrored
   ```


© 2005 Hewlett-Packard Development Company, L.P.

You should a variety of nasty error messages. The file system metadata should have been corrupted by the **dd** command.

3. Recreate the file system metadata structures in **/dev/vg01/rnonmirrored**, then remount the file system. How can you restore the data that was in this file system?

   **Answer:**

   ```
   # newfs -F vxfs /dev/vg01/rnonmirrored
   # mount /dev/vg01/nonmirrored /nonmirrored
   ```

   Unfortunately, since this logical volume wasn't mirrored, the data must be restored from tape.

4. Now view the **lvdisplay -v** output for the **/dev/vg01/mirrored** logical volume. What is the LV Status?

   **Answer:**

   ```
   # lvdisplay -v /dev/vg01/mirrored
   ```

   The LV Status appears to be "Stale", because the mirror on the replaced disk no longer contains an up-to-date image of the logical volume's data.

5. Recall that you already successfully mounted the file system in this logical volume, even while one of the mirrors was unavailable. Did the user data survive the disk failure, too?

   **Answer:**

   ```
   # ls /mirrored
   ```

   The user data should be fine, too.

6. Although you probably discovered that the data in **/dev/vg01/mirrored** survived the disk failure, the mirror on the failed drive is no longer participating in the logical volume updates. Long term, you will probably want to resynchronize the logical volume. Make it so!

   **Answer:**

   There are two ways to do this. If you can afford some downtime, simply deactivate and reactivate the volume group.

   ```
   # vgchange -a n vg01
   # vgchange -a y vg01
   ```

   You can avoid downtime, though, by simply running **vgsync**. This command resynchronizes the logical volume while the system remains up and running.

   ```
   # vgsync vg01
   ```


© 2005 Hewlett-Packard Development Company, L.P.

7. Wait a minute or two, then re-run the **lvdisplay** command to verify that the command succeeded.

   **Answer:**

   ```
   # lvdisplay -v /dev/vg01/mirrored
   ```

   The logical volume should now be fully re-synced.

## Part 7: Cleanup

Before you continue on to the next chapter, remove the logical volumes that you created during this lab. However, you can keep the **vg01** volume group for use in the next lab.

```
# umount /mirrored
# umount /nonmirrored
# lvremove -f /dev/vg01/mirrored
# lvremove -f /dev/vg01/nonmirrored
```


© 2005 Hewlett-Packard Development Company, L.P.

## 4–30. QUIZ: Review Questions

1. Write the command that lists the contents of the LIF area after the system is fully up (at run level 2 or 3).

   **Answer:**

   ```
   # lifls -l /dev/rdsk/c0t6d0
   ```

2. Write the command that displays the contents of the **AUTO** file after the system is fully up (at run level 2 or 3).

   **Answer:**

   ```
   # lifcp /dev/rdsk/c0t6d0:AUTO -
   ```

3. From the ISL prompt, write the command that list the contents of the **AUTO** file.

   **Answer:**

   ```
   ISL> lsautofl
   ```

4. Write a single command that will restore a damaged LIF area when the system is fully up (at run level 2 or 3).

   **Answer:**

   ```
   # mkboot /dev/rdsk/c0t6d0
   ```

5. List five binary startup and/or core command files.

   **Answer:**

   ```
   /sbin/init
   /sbin/ioinit
   /sbin/insf
   /sbin/mount
   /sbin/vgcfgrestore
   /sbin/reboot
   /sbin/lvlnboot
   /sbin/vgchange
   /sbin/newfs
   ```

6. Which binary startup file ranks as one of the most important?

   **Answer:**

   ```
   /sbin/init
   ```


© 2005 Hewlett-Packard Development Company, L.P.

# 4–31. LAB: Booting: Corrupt Stable Storage and Bad LIF Area

**NOTE: The following labs should only be performed on a PA-RISC system**

## Part 1: Corrupt Stable Storage

This lab is designed to give you experience troubleshooting a corrupt stable storage.

1. Login to the system console. If you are accessing the system remotely, access the GSP console interface.

2. Execute the **`mod4.lab1.sh`** script.

   ```
   # cd /home/labs
   # ./mod4.lab1.sh
   ```

   This will simulate the condition such that, when rebooting your machine, the boot and startup processes will fail, because stable storage is corrupt. Use the knowledge gained in the current module to fix the problem.

   **Answer:**

```
  *** FINAL System shutdown message from r206c41 # ***

System going down IMMEDIATELY
Jul 25 21:46:14 rpcbind: terminate: rpcbind terminating on signal. Restart
with "rpcbind -w"
Jul 25 21:46:14 automount[854]: exiting

sync'ing disks (0 buffers to flush):
0 buffers not flushed
0 buffers still dirty

Closing open logical volumes...
Done


Firmware Version  1.1

Duplex Console IO Dependent Code (IODC) revision 1

-------------------------------------------------------------------------
   (c) Copyright 1995-1998, Hewlett-Packard Company, All rights reserved
-------------------------------------------------------------------------

Processor   Speed            State           Coprocessor State  I/D Cache
---------  --------   ---------------------  -----------------  ----------
   0      367 MHz    Active                 Functional         512 KB/1 MB

  Central Bus Speed (in MHz) :        118

  Available memory (bytes)    : 268435456
  Good memory required (bytes):   21610496
```


© 2005 Hewlett-Packard Development Company, L.P.

```
  Primary boot path:    2/0/1.66
  Alternate boot path:  2/0/1.55
  Console path:         SERIAL_1.9600.8.none
  Keyboard path:        PS2


- Main Menu -------------------------------------------------------------
        Command                          Description
        -------                          -----------
        BOot [PRI|ALT|<path>]            Boot from specified path
        PAth [PRI|ALT|CON|KEY] [<path>] Display or modify a path
        SEArch [DIsplay|IPL] [<path>]    Search for boot devices

        COnfiguration [<command>]        Access Configuration menu/commands
        INformation [<command>]          Access Information menu/commands
        SERvice [<command>]              Access Service menu/commands

        DIsplay                          Redisplay the current menu
        HElp [<menu>|<command>]          Display help for menu or command
        RESET                            Restart the system
-------
```

*NOTE:* The system did not attempt to autoboot; so autoboot is probably disabled. This can be checked and fixed from the configuration menu, from ISL, or from the **`setboot`** command after the system is booted. It is generally easier to work from the normal UNIX shell environment; so we will fix this after the machine is up.

```
Main Menu: Enter command > bo
Interact with IPL (Y, N, Q)?> n

Booting...
Not a Valid Boot Path.
```

*NOTE:* If you look above at the path information, it does not match our system. Using the **`search`** command, we can check for valid boot paths.

```
------- Main Menu ---------------------------------------------------------

        Command                          Description
        -------                          -----------
        BOot [PRI|ALT|<path>]            Boot from specified path
        PAth [PRI|ALT|CON|KEY] [<path>] Display or modify a path
        SEArch [DIsplay|IPL] [<path>]    Search for boot devices

        COnfiguration [<command>]        Access Configuration menu/commands
        INformation [<command>]          Access Information menu/commands
        SERvice [<command>]              Access Service menu/commands

        DIsplay                          Redisplay the current menu
        HElp [<menu>|<command>]          Display help for menu or command
```


© 2005 Hewlett-Packard Development Company, L.P.

```
        RESET                           Restart the system
-------
Main Menu: Enter command > sea

Searching for potential boot device(s)...
This may take several minutes.

To discontinue search, press any key (termination may not be immediate).


   Path Number        Device Path              Device Type
   -----------        -----------------        -----------
   P0                 SESCSI.0.0               HP      C1537A
   P1                 LAN.15.0.104.31.3.254    15.0.104.31
   P2                 FWSCSI.6.0               IBM     DDRS-34560WS
   P3                 FWSCSI.5.0               IBM     DDRS-34560WS
```

Let's look at the pri and alt boot paths.

```
Main Menu: Enter command > path

  Primary boot path:    2/0/1.66

  Alternate boot path:  2/0/1.55

  Console path:         SERIAL_1.9600.8.none

  Keyboard path:        PS2
```

The paths don't look right. Let's fix them.

```
Main Menu: Enter command > path pri FWSCSI.6

  Primary boot path:    FWSCSI.6.0

Main Menu: Enter command > path alt sescsi.0

  Alternate boot path:  SESCSI.0.0
```

We can now attempt to boot the system.

```
Main Menu: Enter command > bo
Interact with IPL (Y, N, Q)?> n

Booting...
Boot IO Dependent Code (IODC) revision 1


HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL booting  hpux
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
Boot
: disk(8/0/19/0.6.0.0.0.0.0;0)/stand/vmunix
7503872 + 1757184 + 1261424 start 0x1e1f68
.
.
.
GenericSysName [HP Release B.11.11] (see /etc/issue)
Console Login: root
Password:
Please wait...checking for disk quotas
.
.
.
Value of TERM has been set to "hpterm".
WARNING:  YOU ARE SUPERUSER !!
```

Now that we're logged in, we can check the paths and reset the autoboot mode.

```
r206c41 # setboot
Primary bootpath : 8/0/19/0.6.0
Alternate bootpath : 8/16/5.0.0

Autoboot is OFF (disabled)
Autosearch is OFF (disabled)

r206c41 #
r206c41 # setboot -b on
r206c41 # setboot
Primary bootpath : 8/0/19/0.6.0
Alternate bootpath : 8/16/5.0.0

Autoboot is ON (enabled)
Autosearch is OFF (disabled)
```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 2: Bad LIF Area

This lab is designed to give you experience troubleshooting a bad LIF area.

1. Login to the system console. If you are accessing the system remotely, access the GSP console interface.

2. Execute the **mod4.lab2.sh** script:

   ```
   # cd /home/labs
   # ./mod4.lab2.sh
   ```

   This script will simulate the condition such that, when rebooting your machine, the boot and startup processes will fail, because of a bad LIF area. Using the knowledge gained in the current module, fix this problem.

   **Answer:**

```
Firmware Version  42.19

Duplex Console IO Dependent Code (IODC) revision 1

-----------------------------------------------------------------------
(c) Copyright 1995-2001, Hewlett-Packard Company, All rights reserved
-----------------------------------------------------------------------

Processor   Speed            State           CoProcessor State  Cache Size
Number                                       State              Inst
Data
-------  --------   --------------------  -----------------  ------------
 0      650  MHz   Active                 Functional         750 KB 1.5 MB

  Central Bus Speed (in MHz)  :        120
  Available Memory            :     655360  KB
  Good Memory Required        :      25820  KB

   Primary boot path:    0/0/1/1.15
   Alternate boot path:  0/0/2/0.2
   Console path:         0/0/4/1.643
   Keyboard path:        0/0/4/0.0


Processor is booting from first available device.

To discontinue, press any key within 10 seconds.

10 seconds expired.
Proceeding...

Trying Primary Boot Path
------------------------
Booting...
```


© 2005 Hewlett-Packard Development Company, L.P.

```
Boot IO Dependent Code (IODC) revision 1


HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL booting  hpux

   Not an isl command or utility.


     Auto-execute file is inconsistent.  Autoboot aborted.
```

Looks like HPUX is missing or corrupt

```
ISL>reset
```

Boot from the Recovery CD

```
Main Menu: Enter command or menu > bo alt
Interact with IPL (Y, N, or Cancel)?> n

...
...

-----------------------------------------------------------------------
-----------------------------------------------------------------------


                 HP-UX CORE MEDIA SYSTEM RECOVERY
                            MAIN MENU


        s.  Search for a file
        b.  Reboot
        l.  Load a file
        r.  Recover an unbootable HP-UX system
        x.  Exit to shell


This menu is for listing and loading the tools contained on the core
media.
Once a tool is loaded, it may be run from the shell. Some tools require
other files to be present in order to successfully execute.

Select one of the above: r
Loading commands for recovery ....
```


© 2005 Hewlett-Packard Development Company, L.P.

```
-----------------------------------------------------------------------



        DEVICE FILE VERIFICATION MENU

    This menu is used to specify the path of the root file system.
    When the information is correct, select 'a'.

        INFORMATION to verify:
          Device file used for '/'(ROOT) is c1t15d0
          The hardware path to disk is 0/0/1/1.15.0

        Select one of the following:
          a.  The above information is correct.
          b.  WRONG!! The device file used for '/'(ROOT) is incorrect.

          m.  Return to the 'HP-UX Recovery MENU.'
          x.  Exit to the shell.

     Selection: a
The disk is LVM
Setting boot and root device file for c1t15d0...


-------------------------------------------------------------------


                      HP-UX Recovery MENU

        Select one of the following:
        a.  Mount the root disk and exit to a shell only.
        b.  Recover the bootlif/os partitions.
        c.  Replace the kernel on the root file system.
        d.  Both Options: b and c
        v.  Read information about LVM Recovery (LVM.RECOVER)

        m.  Return to 'HP-UX Recovery Media Main Menu'.
        x.  Exit to the shell.

    Use this menu to select the level of recovery desired.

    Selection: a
Entering file system checking ...



-------------------------------------------------------------------

                FILE SYSTEM CHECK
                     MENU
       The file system check '/sbin/fs/vxfs/fsck -y
/dev/rdsk/c1t15d0s1lvm'
       will now be run.
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
Select one of the following:
 a.  Run fsck -y.
 b.  Prompt for the fsck run string on c1t15d0s1lvm.
 m.  Return to the 'HP-UX Recovery MENU.'

 Selection: a
```

```
** /dev/rdsk/c1t15d0s1lvm
** Last Mounted on /stand
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
49 files, 0 icont, 28652 used, 114921 free (121 frags, 14350 blocks)
Going to mount /dev/dsk/c1t15d0s1lvm to /ROOT/stand as read only
Umount /ROOT/stand
Doing fsck on /ROOT file system
/sbin/fs/vxfs/fsck -y /dev/rdsk/c1t15d0s2lvm
file system is clean - log replay is not required
 Mount /ROOT file system
/sbin/fs/vxfs/mount /dev/dsk/c1t15d0s2lvm /ROOT
Mount /ROOT/stand again
/sbin/fs/hfs/mount  -F hfs /dev/dsk/c1t15d0s1lvm /ROOT/stand

==================================================================
The root disk has been mounted under /ROOT, if you want to chroot
to this directory type the commands:
     loadfile chroot
     chroot /ROOT /sbin/sh
```

Since it appears all that is wrong is a missing or corrupt HPUX in the LIF area, we might be able to get away with a simple
"mkboot /dev/rdsk/c1t15d0"

Let's try it.

```
# loadfile chroot
# chroot /ROOT /sbin/sh
# vgchange -a y vg00
Activated volume group
Volume group "vg00" has been successfully changed.
# mount -a
mount: /dev/vg00/lvol1 on /stand : Device busy
# mkboot /dev/rdsk/c1t15d0
```

Since **mkboot** rebuilds the whole LIF, let's make sure the **AUTO** file has the right info.

```
# mkboot -a hpux /dev/rdsk/c1t15d0
```

Since we rebuilt the entire LIF, we now have a fresh LABEL file, and the pointer to the boot lvol will not be present.


© 2005 Hewlett-Packard Development Company, L.P.

```
# lvlnboot -v
Boot Definitions for Volume Group /dev/vg00:
Physical Volumes belonging in Root Volume Group:
        /dev/dsk/c1t15d0 (0/0/1/1.15.0) -- Boot Disk
No Boot Logical Volume configured
Root: lvol3     on:     /dev/dsk/c1t15d0
Swap: lvol2     on:     /dev/dsk/c1t15d0
Dump: lvol2     on:     /dev/dsk/c1t15d0, 0

# lvlnboot -b /dev/vg00/lvol1
Volume Group configuration for /dev/vg00 has been saved in
/etc/lvmconf/vg00.conf
```

Verifying...

```
# lvlnboot -v
Boot Definitions for Volume Group /dev/vg00:
Physical Volumes belonging in Root Volume Group:
        /dev/dsk/c1t15d0 (0/0/1/1.15.0) -- Boot Disk
Boot: lvol1     on:     /dev/dsk/c1t15d0
Root: lvol3     on:     /dev/dsk/c1t15d0
Swap: lvol2     on:     /dev/dsk/c1t15d0
Dump: lvol2     on:     /dev/dsk/c1t15d0, 0

#
# reboot
```


**© 2005 Hewlett-Packard Development Company, L.P.**

# 4–32. LAB: Recovering an Unbootable System — Missing Kernel

## Directions

This lab gives you experience troubleshooting a system when all kernel files have been accidentally removed or corrupted.

1. Login to the system console. If you are accessing the system remotely, access the GSP console interface.
2. Execute the following sequence of commands:

```
# cd /stand
# rm vmunix*

# cd /
# shutdown -ry 0
```

Using the knowledge gained in the current module, recover the system.

**Answer:**

**Boot from Recovery CD**

```
Main Menu: Enter command or menu > bo alt
Interact with IPL (Y, N, or Cancel)?> n
...


                 HP-UX CORE MEDIA SYSTEM RECOVERY
                            MAIN MENU


        s.  Search for a file
        b.  Reboot
        l.  Load a file
        r.  Recover an unbootable HP-UX system
        x.  Exit to shell


This menu is for listing and loading the tools contained on the core
media.
Once a tool is loaded, it may be run from the shell. Some tools require
other files to be present in order to successfully execute.

Select one of the above: r
Loading commands for recovery ....

----------------------------------------------------------------


        DEVICE FILE VERIFICATION MENU

    This menu is used to specify the path of the root file system.
```


© 2005 Hewlett-Packard Development Company, L.P.

```
    When the information is correct, select 'a'.

        INFORMATION to verify:
          Device file used for '/'(ROOT) is c1t15d0
          The hardware path to disk is 0/0/1/1.15.0

        Select one of the following:
          a.  The above information is correct.
          b.  WRONG!! The device file used for '/'(ROOT) is incorrect.

          m.  Return to the 'HP-UX Recovery MENU.'
          x.  Exit to the shell.

     Selection: a
The disk is LVM
Setting boot and root device file for c1t15d0...


-----------------------------------------------------------------


                      HP-UX Recovery MENU

        Select one of the following:
        a.  Mount the root disk and exit to a shell only.
        b.  Recover the bootlif/os partitions.
        c.  Replace the kernel on the root file system.
        d.  Both Options: b and c
        v.  Read information about LVM Recovery (LVM.RECOVER)

        m.  Return to 'HP-UX Recovery Media Main Menu'.
        x.  Exit to the shell.

    Use this menu to select the level of recovery desired.

    Selection: a
Entering file system checking ...


-----------------------------------------------------------------



                FILE SYSTEM CHECK
                     MENU
       The file system check '/sbin/fs/vxfs/fsck -y
/dev/rdsk/c1t15d0s1lvm'
       will now be run.

       Select one of the following:
        a.  Run fsck -y.
        b.  Prompt for the fsck run string on c1t15d0s1lvm.
        m.  Return to the 'HP-UX Recovery MENU.'

        Selection: a
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
** /dev/rdsk/c1t15d0s1lvm
** Last Mounted on /ROOT/stand
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Check Pathnames
** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
48 files, 0 icont, 1700 used, 141873 free (121 frags, 17719 blocks)
Going to mount /dev/dsk/c1t15d0s1lvm to /ROOT/stand as read only
Umount /ROOT/stand
Doing fsck on /ROOT file system
/sbin/fs/vxfs/fsck -y /dev/rdsk/c1t15d0s2lvm
file system is clean - log replay is not required
 Mount /ROOT file system
/sbin/fs/vxfs/mount /dev/dsk/c1t15d0s2lvm /ROOT
Mount /ROOT/stand again
/sbin/fs/hfs/mount  -F hfs /dev/dsk/c1t15d0s1lvm /ROOT/stand

==================================================================
The root disk has been mounted under /ROOT, if you want to chroot
to this directory type the commands:
     loadfile chroot
     chroot /ROOT /sbin/sh

# loadfile chroot
# chroot /ROOT /sbin/sh
# vgchange -a y vg00
Activated volume group
Volume group "vg00" has been successfully changed.
# fsck -P
/dev/vg00/rlvol1: 48 files, 0 icont, 1700 used, 141873 free (121 frags,
17719 blocks)
log replay in progress
replay complete - marking super-block as CLEAN
file system is clean - log replay is not required
file system is clean - log replay is not required
file system is clean - log replay is not required
file system is clean - log replay is not required
file system is clean - log replay is not required
file system is clean - log replay is not required
# mount -a
mount: /dev/vg00/lvol1 on /stand : Device busy

# cd /stand
# ls
bootconf    dlkm        kernrel     krs_lkg     lost+found  system
build       ioconfig    krs         krs_tmp     rootconf    system.d
# mk_kernel
Generating module: krm...
Compiling /stand/build/conf.c...
Loading the kernel...
Generating kernel symbol table...
# pwd
/stand
# kmupdate
```


© 2005 Hewlett-Packard Development Company, L.P.

```
  Kernel update request is scheduled.

  Default kernel /stand/vmunix will be updated by
  newly built kernel /stand/build/vmunix_test
  at next system shutdown or startup time.

# /sbin/init.d/kmbuild stop
       * The newly built kernel has been moved to "/stand/vmunix".
       * The newly built kernel function set directory has been moved
         to "/stand/dlkm".
# reboot
Shutdown at 15:56 (in 0 minutes)
System shutdown time has arrived
-----------------------------------------------------------------

Firmware Version  42.19

Duplex Console IO Dependent Code (IODC) revision 1

-------------------------------------------------------------------------
 (c) Copyright 1995-2001, Hewlett-Packard Company, All rights reserved
-------------------------------------------------------------------------

Processor   Speed            State           CoProcessor State  Cache Size
Number                                       State              Inst  Data
-------  --------   ---------------------  -----------------  ------------
 0      650  MHz   Active                 Functional         750 KB 1.5 MB

  Central Bus Speed (in MHz)  :        120
  Available Memory            :     655360  KB
  Good Memory Required        :      25820  KB

   Primary boot path:    0/0/1/1.15
   Alternate boot path:  0/0/2/0.2
   Console path:         0/0/4/1.643
   Keyboard path:        0/0/4/0.0


Processor is booting from first available device.

To discontinue, press any key within 10 seconds.

10 seconds expired.
Proceeding...

Trying Primary Boot Path
------------------------
Booting...
Boot IO Dependent Code (IODC) revision 1


HARD Booted.

ISL Revision A.00.43  Apr 12, 2000

ISL booting  hpux
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
Boot
: disk(0/0/1/1.15.0.0.0.0.0;0)/stand/vmunix
10268672 + 1777664 + 1542264 start 0x1fc1e8




alloc_pdc_pages: Relocating PDC from 0xf0f0000000 to 0x27b00000.
gate64: sysvec_vaddr = 0xc0002000 for 2 pages
NOTICE: nfs3_link(): File system was registered at index 3.
...
...
 Starting the Apache subsystem ..................................... N/A
 Start CDE login server ............................................ OK

The system is ready.

HP-UX r206c41 B.11.11 9000/800/A400-6X
Console Login: root
Password:
Please wait...checking for disk quotas
(c)Copyright 1983-2000 Hewlett-Packard Co.,  All Rights Reserved.
...
WARNING:  YOU ARE SUPERUSER !!


root@r206c41   [/root]

# ll /stand/vmunix
-rwxr-xr-x   1 root       sys        27573440 Jan 15 12:56 /stand/vmunix
```


© 2005 Hewlett-Packard Development Company, L.P.

# 5–41. LAB SOLUTION: Booting HP-UX 11i v2 via the EFI

## Directions

Carefully follow the instructions below.

## Part 1: Interacting with the EFI Boot Manager and Shell

This part of the lab provides an opportunity to explore some of the commands and utilities that are available at the EFI shell prompt.

1. Access the MP console interface, and login as user root.

2. Reboot your system, interrupt the autoboot process, and select the EFI shell option from the boot manager menu.

   **Answer:**

   ```
   # shutdown -ry 0
   EFI Boot Manager ver 1.10 [14.60] Firmware ver 1.61 [4241]
   Please select a boot option

       HP-UX Primary Boot
       HP-UX HA Alternate Boot
       EFI Shell [Built-in]
       Boot option maintenance menu
       Security/Password Menu
   ```

3. Use the EFI info –b all command to view your system configuration.

   a. What is your system model string?

   b. How many CPUs do you have?

   c. How much memory do you have?

   d. How many LAN interfaces do you have? What are their MAC addresses?

   **Answer:**

   ```
   Shell> info -b all
   ```

   The hardware configuration will vary from system to system.


© 2005 Hewlett-Packard Development Company, L.P.

4. Use the map –r –fs command to view a list of potential boot devices. Record the fs*n*: names below, and see if you can determine which hardware path each fs*n*: name represents. You may need to refer back to the EFI hardware addressing slide earlier in the book to interpret the hardware paths.

   **Answer:**

   ```
   Shell> map -b -r -fs
   ```

   The actual fs*n*: labels and hardware paths will vary, but you should have at least four EFI file system partitions. The hardware paths that include the string Part1 usually represent OS partitions, and paths that include the string Part3 usually represent HPSP partitions.

5. Type fs*n*: to select the system partition on your primary boot disk, where fs*n*: is the drive name that you identified in the previous step.

   **Answer:**

   Drive names will vary, but after selecting the desired drive name, your EFI shell prompt should display the drive that you're currently accessing.

6. Use the EFI file system navigation commands that you learned in the chapter to explore the file system and find the hpux.efi OS loader and the auto file. What are the full pathnames for these files?

   **Answer:**

   ```
   fsn:\> cd \efi\hpux
   fsn:\> ls
   ```

   \efi\hpux\hpux.efi is the full pathname for the HP-UX OS loader.
   \efi\hpux\auto is the full pathname for the auto file.

7. Which EFI shell command can be used to view the contents of the auto file? Try it!

   **Answer:**

   ```
   fsn:\> type \efi\hpux\auto
   ```

8. Use the appropriate bcfg arguments to dump the current boot menu to the screen. (Note: the **bcfg** dump command output may be greater than 80 characters wide. You may need to adjust your terminal emulator settings to 132 columns if you want to see the entire line)

   **Answer:**

   ```
   fsn:\> bcfg boot dump
   ```


© 2005 Hewlett-Packard Development Company, L.P.

9. Add a new entry to the boot menu just below your primary boot disk. Since we don't have another actual bootable device available on the system, simply add a second a second boot menu option for your primary boot disk. Use "My primary boot disk", or perhaps a more creative label, for your new boot menu option.

   **Answer:**

   ```
   fsn:\> bcfg boot add 2 fsn:\efi\hpux\hpux.efi "My primary boot
   disk"
   ```

10. Dump the EFI boot menu to the screen again to verify your work.

    **Answer:**

    ```
    fsn:\> bcfg boot dump
    ```

11. Since we don't really want two entries in the boot menu for the same disk. Remove the boot menu option that you just added.

    **Answer:**

    ```
    fsn:\> bcfg boot rm 2
    ```

12. Dump the EFI boot menu to the screen again to verify your work.

    **Answer:**

    ```
    fsn:\> bcfg boot dump
    ```

13. Boot the system from the EFI shell. When given an opportunity to interrupt the HP-UX kernel loader autoboot, press Escape.

    **Answer:**

    ```
    fsn:\> \efi\hpux\hpux.efi
    Escape
    ```


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 2: Interacting with the HPUX Kernel Loader

In this part of the lab, you will have a chance to explore the HPUX kernel loader.

1. If you didn't interrupt the autoboot process in the previous part of the lab, reboot again, and press Escape when the HPUX autoboot message appears.

   ```
   HP-UX Boot Loader for IA64  Revision 1.71
   Press Any Key to interrupt Autoboot
   \EFI\HPUX\AUTO ==> boot vmunix
   Seconds left till autoboot – 10  Escape
   Type 'help' for help
   ```

2. Type the help command to view a list of available commands.

   **Answer:**

   ```
   HPUX> help
   ```

3. Execute the command to list the kernels available in /stand.

   **Answer:**

   ```
   HPUX> ll
   ```

4. Which command can you use to view the contents of the auto file? What is the purpose of the auto file?

   **Answer:**

   ```
   HPUX> showauto
   ```

   The auto file determines the default boot string.

5. Boot vmunix to single-user mode.

   **Answer:**

   ```
   HPUX> boot vmunix -is
   ```

6. Once you reach single-user mode, use the HP-UX `init 3` command to transition to run-level 3.

   **Answer:**

   ```
   # init 3
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 3: Mirroring the Boot Disk: Configuring Partitions

The next few parts of the lab walk you through the process required to mirror an HPUX 11i v2 boot disk.

1. Use lvlnboot to determine the location of your current boot disk, then use ioscan and diskinfo to identify a disk large enough to serve as a mirror.

   **Answer:**

   ```
   # lvlnboot -v
   # ioscan -funC disk
   # diskinfo /dev/rdsk/cxtxdx
   ```

2. Use the idisk command to create a 500MB system partition, an OS partition, and a 400MB service partition on your new boot disk.

   **Answer:**

   ```
   # vi /tmp/idf
     3
     EFI 500MB
     HPUX 100%
     HPSP 400MB
   # idisk -wf /tmp/idf /dev/rdsk/cxtxdx
   ```

3. Create device files for the new partitions.

   **Answer:**

   ```
   # insf -eC disk
   ```

4. Use the idisk command to verify the partition table on the new boot disk.

   **Answer:**

   Near the end of the idisk output, you should see three partitions.

   ```
   # idisk /dev/rdsk/cxtxdx
   ```

5. Verify that the partition device files were created properly. There should be block and raw *cxtxdx*s1, *cxtxdx*s2, and *cxtxdx*s3 device files for the new boot mirror.

   **Answer:**

   ```
   # ioscan -funC disk
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 4: Mirroring the Boot Disk: Configuring the System Partition

After you create the partition table, you can initialize and populate the EFI system partition.

1. Populate the /efi/hpux directory in the new EFI partition.

   **Answer:**

   ```
   # mkboot -e -l /dev/rdsk/cxtxdx
   ```

2. Change the auto file on both boot disks so the host can boot without quorum. Change the boot string in the file to boot vmunix –lq. (Hint: Be sure to specify the s1 partition!)

   **Answer:**

   ```
   # vi /tmp/auto
     boot vmunix -lq
   # efi_cp -d /dev/rdsk/cxtxdxs1 /tmp/auto /efi/hpux/auto
   # efi_cp -d /dev/rdsk/cytydys1 /tmp/auto /efi/hpux/auto
   ```

3. Verify the contents of the system partition. (Hint: Be sure to specify the s1 partition!)

   **Answer:**

   ```
   # efi_ls -d /dev/rdsk/cxtxdxs1
   ```

4. Verify the contents of the auto file in the system partition.

   **Answer:**

   ```
   # efi_cp -d /dev/rdsk/cxtxdxs1 -u /efi/hpux/auto /tmp/auto
   # cat /tmp/auto
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 5: Mirroring an 11i v2 Boot Disk: Configuring the OS Partition

Configure the OS partition, essentially an encapsulated LVM boot disk.

1. Initialize and add the new OS partition to vg00 (Hint: Be sure to specify s2, the OS partition!).

   **Answer:**

   ```
   # pvcreate -fB /dev/rdsk/cxtxdxs2
   # vgextend vg00 /dev/dsk/cxtxdxs2
   ```

2. Mirror *all* of the logical volumes in vg00.

   **Answer:**

   ```
   # for lv in /dev/vg00/lvol*
     do
        lvextend -m 1 $lv
     done
   ```

3. Add a line to /stand/bootconf so SDUX knows which disks are boot disks (Hint: Be sure to specify **s2**, the OS partition!).

   ```
   # vi /stand/bootconf
       l /dev/dsk/c0t0d0s2
       l /dev/dsk/c0t1d0s2
   ```

4. Verify that the disk was added to vg00, and that the logical volumes are synchronized. The LV Status field for each logical volume should report syncd.

   **Answer:**

   ```
   # vgdisplay -v vg00
   ```

5. Verify that the BDRA was updated properly. You should see references to both disks in the boot, root, and swap portions of the output.

   **Answer:**

   ```
   # lvlnboot -v
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 6: Configuring an HP-UX 11i v2 Boot Disk: Configuring the EFI Boot Manager Menu

Add the new mirrored disk to the EFI boot manager menu. This ensures that the EFI will automatically attempt to boot the mirrored boot disk if the primary boot disk fails.

1. Ensure that your initial boot disk is configured as your primary boot source, and add your new boot disk to the EFI boot manager as the second (i.e. high availability alternate) boot source.

   **Answer:**

   Replace the hardware paths below with the hardware paths for your first and second boot disks. Your hardware paths may be different.

   ```
   # setboot -p 0/1/1/0.0.0
   # setboot -h 0/1/1/0.1.0
   ```

2. Verify that your new boot path was recorded properly in NVRAM.

   **Answer:**

   ```
   # setboot
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 7: Mirroring the Boot Disk: Testing the New Mirror

Boot the system from the mirror disk.

1. Use the shutdown command to initiate the boot sequence. When given an opportunity to interrupt the EFI autoboot sequence by pressing the down arrow key.

   **Answer:**

   ```
   # shutdown -ry 0
   ```

   When the EFI Boot Manager menu appears, press the down arrow key.

2. Select your mirrored disk from the boot manager selection menu and proceed with the boot process.

   ```
   EFI Boot Manager ver 1.10 [14.60] Firmware ver 1.61 [4241]
   Please select a boot option
      HP-UX Primary Boot
      HP-UX Alternate Boot
      EFI Shell [Built-in]
   ```

3. When the boot process finishes, check the syslog.log file to verify which disk/kernel you booted from.

   ```
   # grep “Boot device’s HP-UX HW path” /var/adm/syslog/syslog.log
     vmunix: Boot device's HP-UX HW path is: 0.0.0.0.1.0
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 8: Replacing a Failed Mirror: Rebuilding the Partition Table

The next three sections of the lab exercise walk through the process of replacing a failed HP-UX 11i v2 boot disk mirror.

1. First, overwrite the first 1MB of your system and OS partitions on the primary boot disk to render it unbootable. Reboot the system to see what impact this has on your machine. Watch the console messages carefully.

   ```
   # dd if=/dev/zero of=/dev/rdsk/cxtxdxs1 bs=1024 count=1024
   # dd if=/dev/zero of=/dev/rdsk/cxtxdxs2 bs=1024 count=1024
   # shutdown -ry 0
   ```

   **Answer:**

   The EFI boot manager should attempt to boot the primary mirror. When the primary mirror fails to boot, the EFI should automatically attempt to boot the alternate mirror instead.

2. Once your system reboots from the remaining mirror, begin the recovery process. Type ioscan and insf to verify that the disk's device files are still in tact (they should be).

   **Answer:**

   ```
   # ioscan -fC disk
   # insf -eC disk
   ```

3. Recreate the system, OS, and service partitions.

   **Answer:**

   ```
   # vi /tmp/idf
     3
     EFI 500MB
     HPUX 100%
     HPSP 400MB
   # idisk -wf /tmp/idf /dev/rdsk/cxtxdx
   ```

4. Recreate and verify device files for the partitions if necessary.

   **Answer:**

   ```
   # insf -eC disk
   ```

5. Verify that the partition table was successfully recreated.

   **Answer:**

   ```
   # idisk /dev/rdsk/cxtxdx
   ```


© 2005 Hewlett-Packard Development Company, L.P.

6. Verify that the partition device files were created properly.

```
# ioscan -funC disk
```

## Part 9: Replacing a Failed Mirror: Rebuilding the System Partition

After you recreate the partition table, repopulate the system partition using the same procedure you used earlier.

1. Populate the /efi/hpux/ directory in the new EFI partition.

   **Answer:**

   ```
   # mkboot -e -l /dev/rdsk/cxtxdx
   ```

2. Change the auto file on the replacement disk to hpux –lq so the host can boot without quorum.

   **Answer:**

   ```
   # vi /tmp/auto
     boot vmunix -lq
   # efi_cp -d /dev/rdsk/cxtxdxs1 /tmp/auto /efi/hpux/auto
   ```

3. Verify the contents of the system partition.

   **Answer:**

   ```
   # efi_ls -d /dev/rdsk/cxtxdxs1
   ```

4. Verify the contents of the auto file in the system partition.

   **Answer:**

   ```
   # efi_cp -d /dev/rdsk/cxtxdxs1 -u /efi/hpux/auto /tmp/auto
   # cat /tmp/auto
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 10: Replacing a Failed Mirror: Rebuilding the OS Partition

Rebuild the OS partition. Use vgcfgrestore to restore the LVM headers, then resynchronize the mirrored logical volumes from the remaining disk.

1. Restore LVM headers to the disk.

   **Answer:**

   ```
   # vgcfgrestore -n vg00 /dev/rdsk/cxtxdxs2
   ```

2. Reactivate the volume group to recognize the recovered disk.

   **Answer:**

   ```
   # vgchange -a y vg00
   ```

3. Resynchronize mirrors.

   **Answer:**

   ```
   # vgsync vg00
   ```

## Part 11: Replacing a Failed Mirror: Rebuilding the EFI Boot Menu

Update the EFI boot menu so it references the GUID of the disk's new GUID rather than its old GUID.

1. Use the **setboot** command to re-define the primary boot path.

   **Answer:**

   ```
   # setboot -p 0/1/1/0.0.0
   # setboot -h 0/1/1/0.1.0
   ```

2. Run **setboot** without any options to verify your work.

   **Answer:**

   ```
   # setboot
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 12: Cleaning Up

Before proceeding, remove any EFI boot manager menu items that you may have added during the lab.

1. Reboot your system, interrupt the EFI boot manager autoboot process, and access the EFI shell.

```
EFI Boot Manager ver 1.10 [14.60]  Firmware ver 1.61 [4241]
Please select a boot option

    HP-UX Primary Boot
    HP-UX HA Alternate Boot
    EFI Shell [Built-in]
    Boot option maintenance menu
    Security/Password Menu
```

2. Dump the contents of the EFI boot manager menu to the screen.

   **Answer:**

```
Shell> bcfg boot dump
```

3. Use the bcfg command to remove any EFI boot menu items that you added over the course of the lab. Verify your work by dumping the remaining menu items to the screen.

   **Answer:**

```
Shell> bcfg boot rm 2
Shell> bcfg boot dump
```


© 2005 Hewlett-Packard Development Company, L.P.

## 6–10. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. List five ASCII startup or start link files.

   **Answer:**

   ```
   /etc/inittab
   /etc/fstab
   /etc/passwd
   /etc/group
   /etc/hosts
   /sbin/rc3.d/S990dtlogin.rc
   ```

2. Which ASCII startup file ranks as one of the most important?

   **Answer:**

   The **`/etc/inittab`** file supplies the script to the boot **`init`** daemon in its role as a general process dispatcher.

3. What is the purpose and function of **`/etc/rc.config.d`**?

   **Answer:**

   **`/etc/rc.config.d`** is a directory that holds configuration files. These configuration files are used by the startup scripts located in **`/sbin/init.d`**.

4. What two ways are used to get into single-user mode?

   **Answer:**

   Single-user mode can be accessed using the **`shutdown`** command and **`hpux -is`** from the **`ISL>`** prompt.


© 2005 Hewlett-Packard Development Company, L.P.

# 6–11. LAB: Recovering an ASCII Startup File, Start Link, and Startup-Script Configuration File

## Lab Directions

Since the exercises in this lab will cause your system to reboot, be sure to login to your system via the MP. This way, you will see the entire shutdown and startup process.

## Part 1

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links and/or startup-script configuration files.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.

2. Execute the **mod6.lab1.sh** script

```
# cd /home/labs
# ./mod6.lab1.sh
```

When your system reboots, the boot and startup processes will partially succeed. This occurs because one of the ASCII startup files is corrupt or missing. Use the knowledge gained in this module to fix this problem.

**Answer:**

```
...

The system is ready.

HP-UX system1 B.11.11 9000/777/C110
Console Login:
```

You may find that attempts to login to the console result in the following message:
`No utmp entry. You must exec "login" from the lowest level "sh".`
If so, reboot the system and bring back up in single user mode.

Otherwise, login now

```
# who -r
  -> The system is in run-level 1.


# export TERM=dtterm
```

In this exercise, **/etc/inittab** shows the incorrect default run level and must be modified.
Before we can edit anything, we need to be sure the **TERM** variable is set; otherwise, **vi** won't work.

```
# echo $TERM
```


© 2005 Hewlett-Packard Development Company, L.P.

```
# export TERM=dtterm
```

It might also be necessary to mount all file systems.

```
# mount -a

# vi /etc/inittab

  -> Change default run-level to "3"
  -> init:3:initdefault:

# reboot
```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 2

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links and/or startup-script configuration files.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.

2. Execute the **mod6.lab2.sh** script

```
# cd /home/labs
# ./mod6.lab2.sh
```

When your system reboots, the boot and startup processes will partially succeed. This occurs because one of the ASCII startup files is corrupt or missing. Use the knowledge gained in this module to fix this problem.

**Answer:**

The system begins to boot, but fails with the message:
"ioinit: Rebooting the system because /etc/ioconfig and /stand/ioconfig files are inconsistent

The system reboots, but there are many failures. Starting early in the boot sequence, just after the lines that state the installed memory, there are errors accessing **/stand**.

In the section where **bcheckrc** runs, there are no lines about any of the "normal" file systems. There is only the following:

```
/sbin/bcheckrc:
Checking for LVM volume groups and Activating (if any exist)
Volume group "/dev/vg00" has been successfully changed.
Activated volume group
Volume group "/dev/vg01" has been successfully changed.
vgsync: Error, "vgsync" is not a valid LVM command.
vgsync: Error, "vgsync" is not a valid LVM command.
vxfs fsck: sanity check: root file system OK (mounted read/write)
Checking hfs file systems
HFS file systems are OK, not running fsck
Checking vxfs file systems
```

After the copyright messages, there are a couple of errors.

```
/sbin/auto_parms: DHCP access is disabled (see /etc/auto_parms.log)

/sbin/rc2.d/S620xfs[10]: env:  not found
/sbin/rc2.d/S620xfs[10]: grep:  not found
```

In the first start-up script screen, the status codes (OK, N/A, FAIL) display at many times the usual display rate. In addition, there are numerous errors:


© 2005 Hewlett-Packard Development Company, L.P.

```
HP-UX Start-up in progress                                    Status
__________________________                                    ______

Configure system crash dumps ............................ [ OK ]
Mount file systems ...................................... [FAIL]  *
Update kernel and loadable modules ...................... [N/A ]
Initialize loadable modules ............................. [N/A ]
Setting hostname ........................................ [FAIL]  *
Start Kernel Logging facility ........................... [N/A ]
Set privilege group ..................................... [N/A ]
Display date ............................................ [N/A ]
Copy processor logs to /var/tombstones .................. [N/A ]
Save system crash dump if needed ........................ [N/A ]
Enable auxiliary swap space ............................. [FAIL]  *
Start syncer daemon ..................................... [N/A ]
Configure Loopback interfaces (lo0) ..................... [FAIL]  *
Configuring all unconfigured software filesets ........... [FAIL]  *
Recover editor crash files .............................. [ OK ]

* - An error has occurred !
* - Refer to the file /etc/rc.log for more information.
```

The most significant of these errors is the **"Mount file systems"** line.

After a few of these screens flash by, you then get to one where the boot will appear to hang on the line about the NFS client. It does not actually hang at this stage. You just have to be prepared to wait several minutes.

Attempting to break out of the **/sbin/rc** startup process with **Ctrl** + **** may not be effective.

```
HP-UX Start-up in progress                                    Status
__________________________                                    ______

Start NIS+ client subsystem ..................................... [ OK ]
Start NIS server subsystem ...................................... [ OK ]
Start ldap client daemon ........................................ [N/A ]
Start NIS/LDAP server subsystem ................................. [N/A ]
Start NIS client subsystem ...................................... [ OK ]
> Start NFS client subsystem .................................... [busy]
INIT: Command is respawning too rapidly.bsystem ................. [    ]
Will try again in 5 minutes.s for MAC/LLA access ................ [    ]
Check for possible errors.ng daemon ............................. [    ]
id:cons "/usr/sbin/getty console console           # system console". ]
Start dynamic routing daemon .................................... [    ]
Start router discover protocol daemon ........................... [    ]
Configuring PPP Interface ....................................... [    ]
Start RARP protocol daemon ...................................... [    ]
Start remote system status daemon ............................... [    ]

  * - An error has occurred !
  * - Refer to the file /etc/rc.log for more information.
```

**init** complains that **getty** is respawning too rapidly, and you do not get a chance to log on.


© 2005 Hewlett-Packard Development Company, L.P.

Here, you have no option but to restart the system, either by turning it off and back on, or if the system has an access port or GSP, resetting it.

Since the kernel seem to boot normally and many other things seem to work, it is worth an attempt to boot into single user mode, rather than trying to use the recovery shell, which is really a last resort tool. You should try other approaches first.

```
Main Menu: Enter command or menu > boot pri
Interact with IPL (Y or N)?> y

Booting...
Boot IO Dependent Code (IODC) revision 0

HARD Booted.

ISL Revision A.00.38  OCT 26, 1994

ISL> hpux -is

...
... Entering single-user mode
...
```

The **rc** screens said there was a problem and referred us to **/etc/rc.log**

```
# cat /etc/rc.log
Old /etc/rc.log moved to /etc/rc.log.old

*************************************************
HP-UX Start-up in progress
Sat Jul 28 21:45:43 BST 2001
*************************************************

Configure system crash dumps
Output from "/sbin/rc1.d/S080crashconf start":
----------------------------
EXIT CODE: 0

Mount file systems
Output from "/sbin/rc1.d/S100localmount start":
----------------------------
mountall: file system table may be corrupt
EXIT CODE: 1
"/sbin/rc1.d/S100localmount start" FAILED

Update kernel and loadable modules
Output from "/sbin/rc1.d/S110kmbuild start":
----------------------------
"/sbin/rc1.d/S110kmbuild start" SKIPPED

Initialize loadable modules
Output from "/sbin/rc1.d/S112kminit start":
----------------------------
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
EXIT CODE: 2 - loadable module feature is not initialized.
"/sbin/rc1.d/S112kminit start" SKIPPED

Setting hostname
Output from "/sbin/rc1.d/S320hostname start":
----------------------------
/sbin/rc1.d/S320hostname[41]: hostname:  not found
EXIT CODE: 127
/sbin/rc1.d/S320hostname[51]: setuname:  not found
EXIT CODE: 127
"/sbin/rc1.d/S320hostname start" FAILED
```

The mounts failed and, as you would expect, we are getting **"not found"** errors.

```
# cat /etc/fstab

## System /etc/fstab file. Static information about the file systems
## See fstab(4) and sam(1M) for further details on configuring devices
#/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
```

Manually mount **/usr** and **/var** file systems.

```
# mount -F vxfs /dev/vg00/lvol7 /usr
# mount -F vxfs /dev/vg00/lvol8 /var

# vi /etc/fstab
I don't know what kind of terminal you are on - all I have is
'unknown'.
[Using open mode]
"/etc/fstab" 11 lines, 502 characters
## System /etc/fstab file.  Static information about the file systems
:q
#
```

In HP-UX 11.11, single user mode does not run the start up scripts, so none of your normal environment has been built. You might have already noticed that the backspace key does not work.

Since the profiles have not run, your TERM variable has not been set. So, **vi** does not know what terminal type you are using.

```
# export TERM=dtterm
```

**Note:** If you are running this lab in the RAIL environment (or are telnet'd into your system), it is probably a good idea to set your terminal emulation mode to a vt102. With Reflection terminal emulation software, this can be accomplished by accessing the "Setup" selection in the menu bar, then "terminal", then select the "vt102" radio button and close.


© 2005 Hewlett-Packard Development Company, L.P.

```
# vi /etc/fstab

  -> Uncomment all commented entries

  -> #/dev/vg00/lvol3    /       vxfs  log        0 1
     #/dev/vg00/lvol1    /stand  hfs   defaults   0 1
     #/dev/vg00/lvol8    /var    vxfs  delaylog   0 2
     #/dev/vg00/lvol7    /usr    vxfs  delaylog   0 2
     #/dev/vg00/lvol6    /tmp    vxfs  delaylog   0 2
     #/dev/vg00/lvol5    /opt    vxfs  delaylog   0 2
     #/dev/vg00/lvol4    /home   vxfs  delaylog   0 2

  -> /dev/vg00/lvol3     /       vxfs  log        0 1
     /dev/vg00/lvol1     /stand  hfs   defaults   0 1
     /dev/vg00/lvol8     /var    vxfs  delaylog   0 2
     /dev/vg00/lvol7     /usr    vxfs  delaylog   0 2
     /dev/vg00/lvol6     /tmp    vxfs  delaylog   0 2
     /dev/vg00/lvol5     /opt    vxfs  delaylog   0 2
     /dev/vg00/lvol4     /home   vxfs  delaylog   0 2
```

This lab underscores the importance of keeping a hardcopy of the /etc/fstab file in your logbook. If the /etc/fstab file had been deleted instead of artificially corrupted, there would be a lot more work to do that simply remembering that /usr is lvol7 and /var is lvol8.

```
# init 3
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 3

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links or startup-script configuration files.

1. Login to the system console. If you are accessing the system remotely, access the MP console interface.

2. Execute the **mod6.lab3.sh** script

```
# cd /home/labs
# ./mod6.lab3.sh
```

When your system reboots, the boot and startup processes will partially succeed. This occurs because one of the startup-script configuration files is bogus, corrupt, or missing. Using the knowledge gained in this module, fix this problem.

**Answer:**

Again, the system begins to boot, but fails with the message:
"ioinit: Rebooting the system because /etc/ioconfig and /stand/ioconfig files are inconsistent

After the reboot, you see the following:

```
                          RESTRICTED RIGHTS LEGEND
Use, duplication, or disclosure by the U.S. Government is subject to
restrictions as set forth in sub-paragraph (c)(1)(ii) of the Rights in
Technical Data and Computer Software clause in DFARS 252.227-7013.

                          Hewlett-Packard Company
                          3000 Hanover Street
                          Palo Alto, CA 94304 U.S.A.

Rights for non-DOD U.S. Government Departments and Agencies are as set
forth in FAR 52.227-19(c)(1,2).
/sbin/rc[67]: syntax error at line 1 : `'' unmatched
```

In addition, you might also see a message like the following:

```
INIT: Command is respawning too rapidly.
Will try again in 5 minutes.
Check for possible errors.
id:cons "/usr/sbin/getty console console            # system console"
```

There are two problems here:
First, the alledged syntax error for **rc** on line 67. This line reads: **. /etc/rc.config**, which is correct.

The rest of the error suggests that there is a problem on line 1, where there is an imbalanced set of quotes. Line 1 of **/etc/rc.config** is a comment. This script


© 2005 Hewlett-Packard Development Company, L.P.

processes all the scripts in **/etc/rc.config.d**, so the problem can most likely be found there.

If you did see the following:

```
INIT: Command is respawning too rapidly.
Will try again in 5 minutes.
Check for possible errors.
id:cons "/usr/sbin/getty console console           # system console"
```

There is a more immediate problem: **init** cannot start a **getty**; so you cannot log in. If this happened, you'll need to restart the system in Single User Mode to troubleshoot.

Unfortunately, **rc** crashed before creating a new **/etc/rc.log**. As an aside, **/etc/rc.log** probably will not help, since it does not log the **rc.config.d** scripts.

```
# cd /etc/rc.config.d
# ls
Dmiconfig        crashconf        i4lmd            pd
LANG             cron             jb               pdcinfo
Rpcd             dce              kl               ptydaemon
.
.
.
clean_uucp       hpfddi4conf      nfsconf          xfs
comsec           hpgelanconf      opcagt           ypldapd
#
```

All of the above scripts are supposed to set variables; so we could try running them all to see which one gave an error.

```
#
# for i in *
> do
> sh ./$i
> done
./jb: Syntax error at line 1 : `'' is not matched.

So the error is in the jb script.

# cat jb
xyz='
# rm jb
# init 3
```

Another approach might have been to list the files in the **/etc/rc.config.d** directory in order of time modified ( **# ll -rt** ). The most recently modified files will be at the end of the list.


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 4 (Optional)

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links or startup-script configuration files.

1. If you are attending this course in the RAIL environment, please skip this lab.

   If you are accessing your system directly (not via RAIL or telnet), login at the CDE login screento the system console.

2. Execute the **`mod6.lab4.sh`** script

```
# cd /home/labs
# ./mod6.lab4.sh
```

   When your system reboots, the boot and startup processes will partially succeed. This occurs because one of the start links is corrupt or missing. Using the knowledge gained in this module, fix this problem.

   **Answer:**

```
...
```

   As system comes up, CDE will not start. You will only see a "Console Login:" prompt.

```
HP-UX system1 B.11.11 9000/800/A400-6X
Console Login: root
Password:

...

Value of TERM has been set to "70092".
WARNING:  YOU ARE SUPERUSER !!

# cd /sbin/rc3.d

# ll
total 0
lrwxr-xr-x   1 root       sys      23 Apr  3 12:36 S100nfs.server ->
/sbin/init.d/nfs.server

lrwxr-xr-x   1 root       sys      19 Apr  3 13:16 S200tps.rc ->
/sbin/init.d/tps.rc

lrwxr-xr-x   1 root       sys      25 Apr  3 13:23 S900apache_httpd ->
/sbin/init.d/apache_httpd
```

   The above listing is absent a symbolic link that points to the CDE start script in **`/sbin/init.d`**. This link must be re-created.

```
# ln -s /sbin/init.d/dtlogin.rc /sbin/rc3.d/S990dtlogin.rc
```


© 2005 Hewlett-Packard Development Company, L.P.

```
# ll
total 0
lrwxr-xr-x   1 root       sys    23 Apr  3 12:36 S100nfs.server ->
/sbin/init.d/nfs.server

lrwxr-xr-x   1 root       sys    19 Apr  3 13:16 S200tps.rc ->
/sbin/init.d/tps.rc

lrwxr-xr-x   1 root       sys    25 Apr  3 13:23 S900apache_httpd ->
/sbin/init.d/apache_httpd

lrwxr-xr-x   1 root       sys    23 Apr 10 15:29 S990dtlogin.rc ->
/sbin/init.d/dtlogin.rc
```

Now start CDE manually.

```
# /sbin/rc3.d/S990dtlogin.rc start
```


**© 2005 Hewlett-Packard Development Company, L.P.**

## Part 5 (Optional)

This lab is designed to give you experience troubleshooting corrupt or missing ASCII startup files, start links or startup-script configuration files.

1. In this exercise, work with a lab partner who can telnet to your system after you have executed the script below.

2. Execute the **mod6.lab5.sh** script

```
# cd /home/labs
# ./mod6.lab5.sh
```

Now have your lab partner attempt to login to your system. Then will not be successful because an important ASCII file is not present. Using the knowledge gained in this module, fix this problem.

**Answer:**

```
r206c44 # rlogin r206c41
rcmd: Lost connection
r206c44 # telnet r206c41
Trying...
Connected to r206c41.
Escape character is '^]'.
 Local flow control off
 Connection closed by foreign host.
r206c44 #  ftp r206c41
Connected to r206c41.
421 Service not available, remote server has closed connection
ftp>

r206c41 #  ps -e | grep inetd
   926 ?         0:00 inetd

r206c41 #  netstat -an -f inet | grep 23
tcp        0      0  *.23                   *.*          LISTEN
udp        0      0  *.1023                 *.*

r206c41 #  tail /var/adm/syslog/syslog.log
.
.
.
Jul 31 22:09:16 r206c41 inetd[3479]: getpwnam: root: No such user
Jul 31 22:09:21 r206c41 inetd[3480]: getpwnam: root: No such user
Jul 31 22:09:28 r206c41 inetd[3481]: getpwnam: root: No such user
r206c41 #
```

Hmmmm, no such user as root?


© 2005 Hewlett-Packard Development Company, L.P.

```
r206c41 #  ll /etc/passwd
/etc/passwd not found
```

You could copy a vanilla /etc/passwd file from **/usr/newconfig**.... or just use the copy we made.

```
r206c41 #  mv /etc/passwd.real /etc/passwd
```


© 2005 Hewlett-Packard Development Company, L.P.

## 7–10. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What is the minimum run-level that allows users to log in?

   **Answer:**

   The minimum run-level is 2.

2. What starts **`getty`** processes during system startup?

   **Answer:**

   The **`init`** process starts **`getty`** processes as configured in the **`/etc/inittab`** file.

3. What starts **`telnetd`** servers upon telnet client requests?

   **Answer:**

   The **`inetd`** process starts **`telnetd`** servers on demand.

4. What run-level does CDE start in?

   **Answer:**

   CDE starts in run-level 3.

5. List three files or directories on which CDE is dependent for proper operation.

   **Answer:**

   ```
   /etc/hosts
   /sbin/rc3.d/S990dtlogin.rc
   /tmp
   ```


© 2005 Hewlett-Packard Development Company, L.P.

# 7–11. LAB: Problems Logging On

## Part 1: Login Problems

This lab is designed to give you experience troubleshooting user login problems.

1. Use the Common Desktop Environment (CDE) for this lab. If using the RAIL environment, choose the ReflectionX option to access your system.

2. Execute the **mod7.lab1.sh** script:

   ```
   # cd /home/labs
   # ./mod7.lab1.sh
   ```

   After running this script, **user1** will have problems logging in. Using the knowledge gained in the current module, fix this problem such that **user1** can log into system. **user1**'s password is **class1**.

   **Answer:**

   ```
   ...
   Console Login: user1
   Password:
   Please wait...checking for disk quotas
   No shell

   HP-UX system1 B.11.11 9000/777/C110
   Console Login:
   ```

   The above message indicates that user1 does not have a valid shell defined in the **/etc/passwd** file. Issuing the **pwck** command will give us more insight to why **user1**'s shell is invalid.

   ```
   Console Login: root
   Password:
   Please wait...checking for disk quotas

   ...

   # pwck

   user1:fnnmD.DGyptLU:301:301:student:/home/user1:/usr/bin/sh
           Optional shell file not found
   ```

   In taking a closer look at **user1**'s shell declaration in the **/etc/passwd** file, there is a space appended. Instead of the shell being **/usr/bin/sh**, it is **/usr/bin/sh** . In vi, there is a *trick* to seeing problems like this. Enter the command **:set list**. This shows TABS as **^I** and end of lines as **$**. You van disable this mode with **:set nolist**.

   To fix this problem, simply remove the space appended to the end of the shell name.

   ```
   # vi /etc/passwd
   ```
   Remove the space appended to user1's shell program and re-attempt to login as user1.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 2: `getty` Problems

This lab is designed to give you experience troubleshooting **`getty`** process problems.

1. Use the Common Desktop Environment (CDE) for this lab. If using the RAIL environment, choose the ReflectionX option to access your system.

2. Execute the **`mod7.lab2.sh`** script:

```
# cd /home/labs
# ./mod7.lab2.sh
```

After running this script, there will be a non-existent **`getty`** process to the console. Using the knowledge gained in the current module, fix this problem.

**Answer:.**

```
Main Menu: Enter command or menu > boot pri
Interact with IPL (Y or N)?> y

Booting...
Boot IO Dependent Code (IODC) revision 0

HARD Booted.

ISL Revision A.00.38  OCT 26, 1994

ISL> hpux -is

...
... Entering single-user mode
...

# cat /etc/inittab

 -> init:1:initdefault:
    ...
    ...
    cons:6:respawn:/usr/sbin/getty console console         # system console
    ...
    ...
```

The default run-level is incorrect. The run-level(s) that are specified for **`getty`** entries are also incorrect.

```
# mount -a
# vi /etc/inittab

  -> init:3:initdefault:
     ...
     ...
     cons:123456:respawn:/usr/sbin/getty console console    # system console
     ...
     ...

# reboot
```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 3: Telnet Access Problems

This lab is designed to give you experience troubleshooting **telnet** client access problems.

1. Use the Common Desktop Environment (CDE) for this lab. If using the RAIL environment, choose the ReflectionX option to access your system.

2. Execute the **mod7.lab3.sh** script:

```
# cd /home/labs
# ./mod7.lab3.sh
```

After running this script, **telnet** client requests will be denied. Using the knowledge gained in the current module, fix this problem such that **telnet** client requests are processed.

**Answer:**

```
...

# telnet localhost
Trying...
Connected to localhost.
Escape character is '^]'.
Local flow control off
telnetd: Telnet device drivers missing: No such file or directory
Connection closed by foreign host.

# cd /dev
# ll telnet*
total 0
```

The telnet master device file, **/dev/telnetm**, does not exist. This file must be created.

```
#  mknod /dev/telnetm c 72 0x000012

# ll /dev/telnetm
crw-r--r--   1 root     sys       72 0x000012 Apr 10 16:23 /dev/telnetm
```

The telnet master device file has been recreated.

```
# telnet localhost
Trying...
Connected to localhost.
Escape character is '^]'.
Local flow control on
Telnet TERMINAL-SPEED option ON

HP-UX system1 B.11.11 C 9000/777 (ta)

login:
```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 4: CDE Session Problems

This lab is designed to give you experience troubleshooting problems involving the establishment of CDE sessions.

1. Execute the **mod7.lab4.sh** script:

```
# cd /home/labs
# ./mod7.lab4.sh
```

When the system reboots, no one will be able to establish a CDE session. Using the knowledge gained in the current module, fix this problem.

**Answer:**

After system reboots, CDE will not start and the following message will be displayed:

```
************************************************************************
*
* The X-server cannot be started on display system1:0...
*
************************************************************************


************************************************************************
*
* Suspending Desktop Login...
*
* Press [Enter] for a login prompt.
*
* Log in. Desktop Login will resume shortly after you log out.
*
************************************************************************
```

Log in and view the permissions on the **/tmp** directory.

```
HP-UX system1 B.11.11 9000/777/C110
Console Login: root
Password:
Please wait...checking for disk quotas

...

# ll -d /tmp
d---------   4 bin        bin           1024 Apr 11 09:56 /tmp
```

The permissions set on **/tmp** are **000**. If the permissions are not set to 777, CDE will not start. Simply change the permissions and log out. Afterwards, CDE will start.

```
# chmod 777 /tmp
# exit
```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 5: Command-Line Login Option Not Available

This lab is designed to give you experience troubleshooting a grayed out Command-Line Login option from the **dtgreet** menus.

1. Execute the following commands:

```
# init 2

login: root
password: hp

# /sbin/init.d/dtlogin.rc start
```

After issuing the above commands, the Command-Line Login option will be grayed out from the **dtgreet** menus. Using the knowledge gained in the current module, fix this problem.

**Answer:**

```
-> Follow the below sequence of steps:

  CDE (dtgreet screen)
   Option
    Session
     Failsafe Session
       "Login as root"
       # ps -ef | grep console
       # kill -1 "PID w/ shell running on console"  (kill minus one...)
       # exit
```

Command-line login option should no longer be grayed out.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 6: No Pseudo-Terminals Available

This lab is designed to give you experience troubleshooting non-existent pseudo-terminals.

1. Execute the **mod7.lab6.sh** script:

```
# cd /home/labs
# ./mod7.lab6.sh
```

After running this script, you will not be able to open another pseudo-terminal window. Using the knowledge gained in the current module, fix this problem.

**Answer:**

After attempting to open a **dtterm** window, the following message will appear:

```
Terminal - Warning
  unable to get pty!
```

Either all **dtterms** are in use, or the device files do not exist.

```
# ll /dev/pts/[0-9] /dev/pts/[0-9][0-9]
/dev/pts/[0-9] not found
/dev/pts/[0-9][0-9] not found
```

There are no **dtterm** device files. They must be recreated.

```
# kmtune -q npty -q nstrpty -q nstrtel

Parameter         Value
==============================================
npty              256
nstrpty           256
nstrtel           60


# insf -d pts -s 256 -e   #< where 256 is the value of nstrpty >
insf: Installing special files for pseudo driver pts

# insf -d pty -n 256 -e   #< where 256 is the value of npty >
insf: Installing special files for pseudo driver ptym
insf: Installing special files for pseudo driver ptys

# insf -d tels -s 60 -e   #< where 60 the value of nstrtel >

# ll /dev/pts/[0-9] /dev/pts/[0-9][0-9]
crw-rw-rw-   1 root    sys     157 0x000000 Apr 11 10:29 /dev/pts/0
crw-rw-rw-   1 root    sys     157 0x000001 Apr 11 10:29 /dev/pts/1
crw-rw-rw-   1 root    sys     157 0x00000a Apr 11 10:29 /dev/pts/10
crw-rw-rw-   1 root    sys     157 0x00000b Apr 11 10:29 /dev/pts/11
crw-rw-rw-   1 root    sys     157 0x00000c Apr 11 10:29 /dev/pts/12
crw-rw-rw-   1 root    sys     157 0x00000d Apr 11 10:29 /dev/pts/13
crw-rw-rw-   1 root    sys     157 0x00000e Apr 11 10:29 /dev/pts/14

...
```


© 2005 Hewlett-Packard Development Company, L.P.

. . .

Exit "Failsafe Session." Establish normal CDE session, and open a **`dtterm`** window.


© 2005 Hewlett-Packard Development Company, L.P.

## 8–12. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What is a file system?

   **Answer:**

   A file system is a structure on a device such as a whole disk or logical volume that is used to manage files and directories.

2. What are the major components of a HFS and VxFS file system?

   **Answer:**

   Major components of a HFS file system are as follows:

   - superblock
   - cylinder group
     1. alternate superblock
     2. inode table
     3. bitmaps
     4. data blocks

   Major components of a VxFS file system are as follows:

   - superblock
   - intent log
   - allocation unit
     5. inode table
     6. bitmaps
     7. data blocks

3. What major feature does a VxFS file system have that's not present with a HFS file system?

   **Answer:**

   A VxFS file system has and uses an intent log. This intent log is used to track pending file system metadata updates. Before a transaction is begun, the intent log is updated. If a system crash occurs, **`fsck`** needs to only look in the intent log and not search through the entire file system as is the case with HFS.

4. What is the purpose of the **`fsck`** command?

   **Answer:**

   **`fsck`** audits and repairs inconsistant conditions found within file systems. It is the primary file system repair utility used.


© 2005 Hewlett-Packard Development Company, L.P.

5. What file systems can **`fsck`** run on?

   **Answer:**

   **`fsck`** can run on HFS and VxFS file systems. To view file system specific versions of **`fsck`**, view the man page on the following commands:

   ```
   fsck_hfs
   fsck_vxfs
   ```

6. What are the **`fsck`** phases for a HFS and VxFS file system?

   **Answer:**

   Phases for a HFS file system are as follows:

   - Check blocks and sizes
   - Check path names
   - Check connectivity
   - Check reference counts
   - Check cylinder groups
   - Salvage cylinger groups

   Phases for a VxFS file system are as follows:

   - Check structural files
   - Check inode sanity and blocks
   - Check directory linkage
   - Check reference counts
   - Check resource maps

7. Is it possible to run a full **`fsck`** on a VxFS file system? If so, how?

   **Answer:**

   Yes, it is possible to run a full **`fsck`** on a VxFS file system, demonstrated as follows:

   ```
   # fsck -F vxfs -o full /dev/vg00/rlvol#
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

# 8–13. LAB: Recovering a Corrupt VxFS File System

## Directions

For many of the lab questions, there are no answers. You will be walked through the lab step-by-step. For other lab questions, the questions and answers are printed together.

1. Create a 20MB Logical Volume named **`test`** , build a JFS file system in it, create a mount point called **`/mod8`** and mount the file system to your new mount point.

   ```
   # lvcreate -L 20 –n test /dev/vg01
   # newfs –F vxfs /dev/vg01/rtest
   # mkdir /mod8
   # mount /dev/vg01/test /mod8
   ```

2. Copy some scratch files into this new file system with the following command:

   ```
   # cd /sbin
   # cp vg* /mod8
   ```

3. Now, **`umount`** the file system.

   ```
   # umount /mod8
   ```

4. Extend this logical volume to the new size of 36 MB.

   ```
   # lvextend -L 36 /dev/vg01/test
   ```

   You should see something like the following:

   ```
   # lvextend -L 36 /dev/vg01/test
   Logical volume "/dev/vg01/lvol2" has been successfully extended.
   Volume Group configuration for /dev/vg01 has been saved in \
   /etc/lvmconf/vg01.conf
   ```

   Special Note: Be sure to read the entire question before you try to execute it.

   Next, we use the **`extendfs`** command. **Immediately** after you hit **Return** to begin execution of the **`extendfs`** command, hit **Ctrl**+**c** interrupt to abort the execution.

   ```
   # extendfs -F vxfs /dev/vg01/rtest
   < ctrl> C
   ```

5. What message(s) do you see on your screen?

   **Answer:**

   You should see something like the following:

   ```
   # extendfs -F vxfs /dev/vg01/rlvol2
   vxfs extendfs: Invocation of the fsck program terminated abnormally.
   The file system is marked bad.  Run full fsck manually.
   (e.g.  fsck -F vxfs /dev/vg01/rlvol2)
   ```

6. Now, run a normal **`fsck`** on the VxFS file system.

   ```
   # fsck -F vxfs /dev/vg01/rtest
   ```


© 2005 Hewlett-Packard Development Company, L.P.

What does the screen output say?

**Answer:**

You should see something like the following:

```
# fsck -F vxfs /dev/vg01/rtest
file system is clean - log replay is not required
```

7. Yet the previous error message (from question 6 above) says that there is a problem with the structural integrity of the file system within **/dev/vg01/test**.

   So, we will run a full **fsck**.

```
# fsck -F vxfs -o full /dev/vg01/rtest
```

   What does the screen output now look like?

   **Answer:**

```
# fsck -F vxfs -o full /dev/vg01/rtest
pass0 - checking structural files
pass1 - checking inode sanity and blocks
pass2 - checking directory linkage
pass3 - checking reference counts
pass4 - checking resource maps
OK to clear log? (ynq)y
set state to CLEAN? (ynq)y
```

8. Now, we again mount the file system.

```
# mount /dev/vg01/test /mod8
```

   Are all the **vg** files still present?

   **Answer:**

   You should see something like the following:

```
# ll /mod8
total 13024
drwxr-xr-x   2 root     root         96 Mar 27 10:30 lost+found
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgcfgbackup
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgcfgrestore
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgchange
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgcreate
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgdisplay
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgexport
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgextend
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgimport
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgreduce
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgremove
-r-sr-xr-x  11 root     sys      606208 Mar 27 10:30 vgscan
```

   Yes, all the files are still present.


**© 2005 Hewlett-Packard Development Company, L.P.**

9. Remove the **/mod8** file system and the mount point.

**Answer:**

```
# cd /
# umount /mod8
# lvremove /dev/vg01/test
# rmdir /mod8
```


© 2005 Hewlett-Packard Development Company, L.P.

## 8–14. LAB: Using `fsdb`

### Directions

For many of the lab questions, there are really no answers. You will be walked through the lab step-by-step. For other lab questions, the questions and answers are printed together.

In this lab, we explore the **fsdb** tool. We will be changing data pointers within inodes, but we'll also be working with an expendable file system.

1. We first set up an HFS file system for this lab.

   a. Create a logical volume within the new volume group:

   ```
   # lvcreate -L 100 –n disk5 /dev/vg01
   ```

   b. Next, build a file system within the logical volume:

   ```
   # newfs -F hfs /dev/vg01/rdisk5
   ```

   c. Create a mount point for the new file system:

   ```
   # mkdir /disk5
   ```

   d. Last, mount the file system:

   ```
   # mount -F hfs /dev/vg01/disk5 /disk5
   ```

2. Copy some files into **/disk5** and then use **fsdb** to cause corruption. Use the following commands:

   a. First, copy some files into the **disk5** file system:

   ```
   # cd /disk5
   # cp /usr/sbin/ioscan .
   # cp /usr/sbin/lanscan .
   # cp /usr/sbin/diskinfo .
   ```

   b. Now, check the inode numbers for the files — you will need them when you run **fsdb.**

   ```
   # ls -i
   ```

   You should see something like the following:

   ```
   6  diskinfo      4  ioscan     5  lanscan   3 lost+found
   ```

   c. Next, **umount /disk5**.

   ```
   # cd /
   # umount /disk5
   ```


© 2005 Hewlett-Packard Development Company, L.P.

or

```
# cd /
# umount /dev/vg01/disk5
```

d. Run the **fsdb** tool on this file system.

```
# fsdb -F hfs /dev/vg01/rdisk5
```

You should see something like the following:

```
File system size = 1024000(frags) isize/cyl group=48(Kbyte blocks)
primary block size=8192(bytes)
fragment size=1024
no. of cyl groups = 42
```

e. Notice that **fsdb** has no prompt. (You are supposed to know what you are doing.☺ ) So, while in **fsdb**, look inside inode 5:

**5i** This command will show inode information for **lanscan**.

You should see something like the following:

```
i#:5  md:  fu--r-xr-xr-x  ln:  1  uid:  0  gid:  3  sz: 40960  ci:  0
a0:    128   a1:   136   a2:    240   a3:    248   a4:  256   a5:   0
a6:      0   a7:     0   a8:      0   a9:      0   a10:   0
a11:     0  a12:     0  a13:      0  a14:      0
at:  Thu  Aug 30  13:16:31 2001
mt:  Thu  Aug 30  13:16:31 2001
ct:  Thu  Aug 30  13:16:31 2001
```

f. Now, let us look inside of inode 6.

**6i** This will show inode information for **diskinfo.**

```
i#:6  md:  f---r-xr-xr-x  ln:  1  uid:  0  gid:  3  sz: 24576  ci: 0
a0:    264   a1:   272   a2:    280   a3:    0   a4:    0   a5:   0
a6:      0   a7:     0   a8:      0   a9:    0  a10:    0  a11:   0
a12:     0   a13:    0  a14:      0
at:  Thu  Aug 30  13:16:31 2001
mt:  Thu  Aug 30  13:16:31 2001
ct:  Thu  Aug 30  13:16:31 2001
```

g. Next, we deliberately cross a few pointers by pointing inode 6 pointers to inode 5 data blocks.

```
a0=128   ( where 128 is the value of a0 from inode #5 )
a1=136   ( where 136 is the value of a1 from inode #5 )
a2=240   ( where 240 is the value of a2 from inode #5 )
a3=248   ( where 248 is the value of a3 from inode #5 )
a4=256   ( where 256 is the value of a4 from inode #5 )
```


© 2005 Hewlett-Packard Development Company, L.P.

h. Now, look at inode 6 again to see the changes.

**6i** This will show inode information for **diskinfo.**

i. Last, we exit **fsdb**.

```
< ctrl-d > to exit fsdb
```

3. Now, with the **/disk5** file system still unmounted, run **fsck** on **/disk5** file system. Answer *n* (no) to all questions.

```
# fsck -F hfs /dev/vg01/rdisk5
```

You will see something like the following:

```
** /dev/vg01/rdisk5
** Last Mounted on /disk5
** Phase 1 - Check Blocks and Sizes
128 DUP I=6
129 DUP I=6
130 DUP I=6
131 DUP I=6
132 DUP I=6
133 DUP I=6
...
...
...
247 DUP I=5
248 DUP I=5
249 DUP I=5
250 DUP I=5
251 DUP I=5
** Phase 2 - Check Pathnames
DUP/BAD  I=5
OWNER=root MODE=104555
SIZE=40960 MTIME=Aug 26 13:16 2001
FILE=/lanscan

REMOVE? n

DUP/BAD  I=6
OWNER=root MODE=100555
SIZE=24576 MTIME=Aug 26 13:16 2001
FILE=/diskinfo

REMOVE? n

** Phase 3 - Check Connectivity
** Phase 4 - Check Reference Counts
** Phase 5 - Check Cyl groups
28 BLK(S) MISSING
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
BAD CYLINDER GROUPS
FIX? n

5 files, 0 icont, 177 used, 99464 free (8 frags, 12432 blocks)
```

4. What type of problems do you encounter?

   **Answer:**

   Duplicate Blocks

5. Which files were reported to have problems?

   **Answer:**

   **`lanscan`** and **`diskinfo`**

6. Mount the **`/disk5`** file system and execute the files found to have duplicate blocks. Which command does not work properly?

   ```
   # mount /dev/vg01/disk5 /disk5
   ```

   The **`/disk5/lanscan`** file and the **`/disk5/diskinfo`** file should now be executed.

   ```
   # cd /disk5
   # ./lanscan
   # ./diskinfo
   ```

   **Answer:**

   ```
   # cd /disk5

   # ll
   total 408
   -r-xr-xr-x   1 root       sys          28672 Mar 26 13:16 diskinfo
   -r-xr-xr-x   1 root       sys         126976 Mar 26 13:16 ioscan
   -r-sr-xr-x   1 root       sys          32768 Mar 26 13:16 lanscan
   drwxr-xr-x   2 root       root          8192 Mar 26 13:15 lost+found

   # ./lanscan
   Hardware Station         Crd Hardware Net-Interface   NM  MAC     ...
   Path     Address         In# State    NameUnit State  ID  Type    ...
   8/16/6   0x0060B0179863  0   UP       lan0     UP     4   ETHER   ...
   ```

   Note: To save space in this book, the above screen output was truncated and is incomplete. But, as you can see, the **`lanscan`** command does indeed execute.

   ```
   # ./diskinfo
   sh: ./diskinfo: Execute permission denied.
   ```

   Conclusion: The **`diskinfo`** file does not work properly. The **`diskinfo`** command fails to execute and generates a **`permission denied`** message, even though the permission bits are set appropriately. This is due to the duplicate blocks with **`lanscan`**. This is an example of the problems that can result from a corrupted file system, problems that can be detected and corrected by **`fsck`**.


© 2005 Hewlett-Packard Development Company, L.P.

7. Umount the **disk5** file system and run **fsck** again. Remove only the **diskinfo** file, for example, answer *y* (yes) to any question that refers to **diskinfo** (**FILE=diskinfo**), or the inode for **diskinfo** (probably inode 6).

   Also, answer **y** (yes) to any question concerning the superblock or a cylinder group.

   ```
   # cd /
   # umount /disk5
   # fsck -F hfs /dev/vg01/rdisk5
   ```

8. Was the **diskinfo** file placed in the **lost+found** directory?

   ```
   # mount /dev/vg01/disk5 /disk5
   # ls /disk5/lost+found
   ```

   **Answer:**

   No files found in the **lost+found** directory.

In summary, while we didn't use **fsdb** to correct problems (which material will be covered in a more advanced course), we did get a chance to see how it could be used to change pointers in an inode.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 8–15. LAB: Using fsadm

## Directions

1. Run a directory and extent fragmentation report on the **/var** file system. At this point, don't do anything to repair the fragmentation. Record the following values:

   a. Directories to reduce:
   b. Average number of extents per file:

   **Answer:**

   ```
   # fsadm -F vxfs -DE /var
   ```

2. Defragment the **/var** file system.

   **Answer:**

   ```
   # fsadm -F vxfs -de /var
   ```

3. How successful was the defragmentation utility? Run a second fragmentation report and compare the results to the values you recorded above in question 1. Do you see any improvement?

   **Answer:**

   ```
   # fsadm -F vxfs -DE /var
   ```

   Since these systems were installed at the beginning of the week, you will likely see very little difference between the reports. At a bare minimum, however, you should see the "Dirs to reduce" parameter decrease.


© 2005 Hewlett-Packard Development Company, L.P.

## 9–12. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What is a process?

   **Answer:**

   A process is an instance of a running program.

2. Explain the purpose of the **`fork()`** and **`exec()`** system calls.

   **Answer:**

   The **`fork()`** system call creates new processes.
   The **`exec()`** system call executes a file.

3. List valid process & thread states.

   **Answer:**

   Valid process states are as follows:

   - Idle (SIDL)
   - Inuse (SINUSE)
   - Zombie (SZOMB)

   Valid thread states are as follows:

   - Idle (TSIDL)
   - Running (TSRUN)
   - Sleeping (TSSLEEP)
   - Waiting to execute (TSRUN)
   - Stopped (TSSTOP)
   - Suspended (TSSUSP) Unused
   - Zombie (TSZOMB)

4. What action does the **`kill`** command perform?

   **Answer:**

   The **`kill`** command sends signals to processes.

5. What is a zombie process?

   **Answer:**

   Zombies are defunct processes. They are dead processes that have not yet been cleared from the process table.


© 2005 Hewlett-Packard Development Company, L.P.

6. What steps should you take to clear a zombie process?

   **Answer:**

   Steps to clear zombie processes are as follows:

   a. `kill -18` "Parent process of zombie"
   b. `kill -15` "Other processes of parent"
   c. `kill -15` "Parent process of zombie"
   d. As a last resort, reboot the system.

7. What "right out the box" commands can help identify CPU-intensive processes?

   **Answer:**

   `top(1)` and `ps(1)` are "right out of the box" commands that can be used to identify CPU-intensive processes.


© 2005 Hewlett-Packard Development Company, L.P.

# 9–13. LAB: Process Hierarchy

## Purpose

This lab is designed to give you hands on experience in determining the process hierarchy for any given user.

## Directions

Read and answer the following questions.

1. Define the appropriate environmental variable such that the **-H** (Hierarchy) option becomes available to the **ps** command and use the **ps** command to generate a report of every process on the system in a process hierarchical format.

   **Answer:**

   ```
   # UNIX95=true ps -efH
   ```

2. Use the **ps** command to generate a report of every process belonging to root in a process hierarchical format.

   **Answer:**

   ```
   # ps –fHu root
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## 9–14. LAB: Zombie Processes

### Directions

This lab provides you with an opportunity to create and review zombie processes. In the standard **ps -ef** command, these processes are seen as defunct processes.

1. Enter these commands to view the source code for this lab:

```
# cd /home/labs
# more mod9.lab3.c.
```

2. Since this file, **mod9.lab3.c** is raw source code, we need to compile it into a binary executable that we will name **mod9**. To do so, enter these commands:

```
# cc mod9.lab3.c -o mod9
```

3. Once you have the **#** prompt back, verify that you have a file named **mod9** in your current directory.

4. Now we are ready to execute the program that will create 10 zombie processes. When executing the binary, make it run in the background.

```
# ./mod9 &
```

   Your # prompt should return almost immediately.

5. To see the 10 zombie (defunct) processes, use the standard **ps-ef** command, and **grep** for **defunct**:

```
# ps -ef | grep defunct
```

   **Answer:**

   For the simple **ps -ef | grep defunct** command, you should see something like the following:

```
# ps -ef | grep defunct

root  3482  3481  1 10:43:58 pts/1     0:00 <defunct>
root  3487  3481  1 10:43:58 pts/1     0:00 <defunct>
root  3488  3481  0 10:43:58 pts/1     0:00 <defunct>
root  3489  3481  0 10:43:58 pts/1     0:00 <defunct>

root  3484  3481  0 10:43:58 pts/1     0:00 <defunct>
root  3491  3481  0 10:43:58 pts/1     0:00 <defunct>
root  3483  3481  0 10:43:58 pts/1     0:00 <defunct>
root  3486  3481  0 10:43:58 pts/1     0:00 <defunct>
root  3485  3481  0 10:43:58 pts/1     0:00 <defunct>
root  3490  3481  0 10:43:58 pts/1     0:00 <defunct>
root  3509  3504  1 10:47:42 ttyp2     0:00 grep defunct
```


© 2005 Hewlett-Packard Development Company, L.P.

6. For a more complete listing, pipe the **`ps -ef`** to more:

   `ps -ef | more`

   **Answer:**

   For the full **`ps -ef | more`** command, your output will resemble the following:

```
UID    PID  PPID  C   STIME TTY       TIME COMMAND
root     0     0  0  Apr 10  ?        0:02 swapper
root     1     0  0  Apr 10  ?        0:00 init
root     2     0  0  Apr 10  ?        0:00 vhand
root     3     0  0  Apr 10  ?        1:01 statdaemon
root     4     0  0  Apr 10  ?        0:04 unhashdaemon
root     7     0  0  Apr 10  ?        0:00 ttisr
root    13     0  0  Apr 10  ?        0:00 lvmkd
root    14     0  0  Apr 10  ?        0:00 lvmkd
root    15     0  0  Apr 10  ?        0:00 lvmkd
root    16     0  0  Apr 10  ?        0:00 lvmkd
root    17     0  0  Apr 10  ?        0:00 nvsisr
root    18     0  0  Apr 10  ?        0:00 supsched
root    19     0  0  Apr 10  ?        0:00 strmem
root    20     0  0  Apr 10  ?        0:00 strweld
root  1026     1  0  Apr 10  console  0:00 /usr/sbin/getty console console
root   455     1  0  Apr 10  ?        0:00 /usr/sbin/syslogd -D
root  1089     1  0  Apr 10  ?        0:04 /usr/dt/bin/ttsession -s
root    25     0  0  Apr 10  ?        0:09 vx_sched_thread
root    26     0  0  Apr 10  ?        0:05 vx_iflush_thread
root    27     0  0  Apr 10  ?        0:03 vx_ifree_thread
root    28     0  0  Apr 10  ?        0:03 vx_inactive_cache_thread
root    29     0  0  Apr 10  ?        0:00 vx_delxwri_thread
root    30     0  0  Apr 10  ?        0:01 vx_logflush_thread
root    31     0  0  Apr 10  ?        0:00 vx_attrsync_thread
root    32     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    33     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    34     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    35     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    36     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    37     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    38     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    39     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    40     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    41     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    42     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    43     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    44     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    45     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    46     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    47     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    48     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    49     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    50     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    51     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    52     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    53     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    54     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    55     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    56     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    57     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    58     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    59     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    60     0  0  Apr 10  ?        0:00 vx_inactive_thread
root    61     0  0  Apr 10  ?        0:00 vx_inactive_thread
root   692     1  0  Apr 10  ?        0:02 /usr/sbin/snmpdm
root  3482  3481  1 10:43:58 pts/1    0:00 <defunct>
root  1105  1102  0  Apr 10  pts/0    0:00 /sbin/sh
root   583     1  0  Apr 10  ?        0:00 /usr/sbin/biod 4
root   673     1  0  Apr 10  ?        0:00 sendmail: accepting onnections
root   551     1  0  Apr 10  ?        0:00 /usr/sbin/portmap
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
daemon  1041  1040  0  Apr 10  ?  26:41 /usr/bin/X11/X :0  -auth /var/dt/Ar19w2-b01040
root   329     1  0  Apr 10  ?        0:19 /usr/sbin/syncer
root   584     1  0  Apr 10  ?        0:00 /usr/sbin/biod 4
root   464     1  0  Apr 10  ?        0:00 /usr/sbin/ptydaemon
root   582     1  0  Apr 10  ?        0:00 /usr/sbin/biod 4
root  3481  1106  0 10:43:58 pts/1    0:00 ./mod.8
root   800     1  0  Apr 10  ?        0:25 /opt/dce/sbin/rpcd
root   483     1  0  Apr 10  ?        0:00 /usr/lbin/nktl_daemon  0 0 0 0 1 -2
root  3487  3481  1 10:43:58 pts/1    0:00 <defunct>
daemon  1046  1041  0  Apr 10  ?       0:03 pexd /tmp/to_pexd_1041.1 /dev/pty/ttyp1 6
root   585     1  0  Apr 10  ?        0:00 /usr/sbin/biod 4
root  1115  1114  0  Apr 10  ?        0:01 /usr/dt/bin/dtterm
root  1023     1  0  Apr 10  ?        0:00 /usr/dt/bin/dtrc /usr/dt/bin/dtrc
root   591     1  0  Apr 10  ?        0:00 /usr/sbin/rpc.statd
root   597     1  0  Apr 10  ?        0:00 /usr/sbin/rpc.lockd
root   603     1  0  Apr 10  ?        0:00 /usr/sbin/automount -f /etc/auto_master
root   623     1  0  Apr 10  ?        0:00 /usr/sbin/inetd
root  3488  3481  0 10:43:58 pts/1    0:00 <defunct>
root  1117  1115  0  Apr 10  pts/2    0:00 /sbin/sh
root   851     1  0  Apr 10  ?        0:00 /usr/sbin/vtdaemon
daemon  1045     1  0  Apr 10  ?       0:00 grmd
root  1106  1101  0  Apr 10  pts/1    0:02 /sbin/sh
root  3489  3481  0 10:43:58 pts/1    0:00 <defunct>
lp     889     1  0  Apr 10  ?        0:00 /usr/sbin/lpsched
root   910     1  0  Apr 10  ?        0:00 /usr/sbin/cron
root  1042  1040  0  Apr 10  ?        0:00 /usr/dt/bin/dtlogin
root  3484  3481  0 10:43:58 pts/1    0:00 <defunct>
root  1040  1023  0  Apr 10  ?        0:00 /usr/dt/bin/dtlogin
root  1060  1042  0  Apr 10  ?        0:07 /usr/dt/bin/dtsession
root  1100  1060  0  Apr 10  ?        0:31 dtwm
root   931     1  0  Apr 10  ?        0:18 /opt/audio/bin/Aserver
root   932   931  0  Apr 10  ?        0:00 /opt/audio/bin/Aserver
root  1090   623  0  Apr 10  ?        0:00 /usr/dt/bin/rpc.ttdbserver
root  1102  1060  0  Apr 10  ?        0:01 /usr/dt/bin/dtterm -session dta01550
root  1101  1060  0  Apr 10  ?        1:33 /usr/dt/bin/dtterm -session dta01551
root   971     1  0  Apr 10  ?        0:00 /usr/sbin/rpc.mountd
root  3491  3481  0 10:43:58 pts/1    0:00 <defunct>
root   978   977  0  Apr 10  ?        0:00 /usr/sbin/nfsd 4
root   977     1  0  Apr 10  ?        0:00 /usr/sbin/nfsd 4
root  3483  3481  0 10:43:58 pts/1    0:00 <defunct>
root   982   977  0  Apr 10  ?        0:00 /usr/sbin/nfsd 4
root  3486  3481  0 10:43:58 pts/1    0:00 <defunct>
root   984   977  0  Apr 10  ?        0:00 /usr/sbin/nfsd 4
root  3485  3481  0 10:43:58 pts/1    0:00 <defunct>
root  3490  3481  0 10:43:58 pts/1    0:00 <defunct>
root  3512  1117  5 10:48:04 pts/2    0:00 ps -ef
root  3499  1106  0 10:45:02 pts/1    0:00 vi mod.8.lab.3
```

7. Use the "gentle kill" command on one of the zombie processes: **`kill 3488`** (or whatever the PID number is your machine). Did the process die?

   **Answer:**

   No

8. Next, use an "absolute kill" command on one of the zombie processes: **`kill -9 3488`** (or whatever the PID number is your machine). Did the process die?

   **Answer:**

   No

9. Now kill the parent process; just a normal **`kill`** will be OK.

   **Answer:**

   `# kill "PPID"`


© 2005 Hewlett-Packard Development Company, L.P.

10. Check whether there are zombies left on the system.

    **Answer:**

    `ps -ef | grep defunct` or `ps -ef | more`

11. Did the zombie processes die?

    **Answer:**

    Yes


© 2005 Hewlett-Packard Development Company, L.P.

## 9–15. LAB: CPU-Intensive Processes

### Purpose

This lab is designed to give you experience troubleshooting CPU-intensive processes.

### Directions

Read and answer the following questions.

8. Execute the following commands.

```
#  cd /tmp
#  vi cpuhog

   #!/sbin/sh
   while true
   do
      :
   done

#  chmod +x cpuhog
#  ./cpuhog
```

9. Open up another pseudo-terminal window.

10. Identify and suspend this CPU-intensive process.

    Issue the **top** command.

```
# top
# kill -25 PID
```

11. Issue the appropriate commands ensuring that the process was suspended.

```
# ps -lp PID
```

    The state column (**S**) should be a "T", which means stopped/suspended.

12. **renice** the CPU-intensive process using an offset of 12.

```
# renice -n 12 PID
# kill -26 PID (resumes process)
```

13. When finished, kill the **cpuhog** PID.


© 2005 Hewlett-Packard Development Company, L.P.

# 10-14. LAB: Troubleshooting the HP-UX 11i v2 Kernel

## Directions

Carefully follow the instructions below.

## Preliminary Steps

1. Portions of this lab require console access. Login as root via your Integrity server's MP console interface.

2. Portions of this lab require access to a web browser, too. If you are accessing your Integrity server via HP's remote lab infrastructure, launch Internet Explorer from your portal server. If you are accessing your Integrity server directly via a PC desktop, simply launch Internet Explorer from your Start menu.

3. If your instructor tells you to disable your browser's proxy settings, do so now. Ask your instructor for details.

## Part 1: Configuring and using `kcweb`

Our goal in this portion of the lab is to configure, launch, and navigate the **`kcweb`** interface.

1. Point your web browser to the Apache/`hpwebadm` daemon at **`https://`*`server`*`:1188/casey/top.cgi`**. Don't forget the "**`s`**" on "**`https`**". Also, it may be necessary to fully qualify your server's hostname. What happens?

   **Answer:**

   This should fail since the daemon isn't running yet.

2. In your console window, configure SSL certificates for your server and manually start the Apache/`hpwebadm` daemon. Which **`kcweb`** option can you use to verify that the daemon is running?

   **Answer:**

   ```
   # kcweb -s startssl
   # kcweb -s status
   ```

3. Click the **`Reload`** or **`Refresh`** button in your web browser to attempt to contact the Apache/`hpwebadm` server again. What happens?

   **Answer:**

   It should successfully connect this time!

4. You may see a dialog box noting that the server is using a certificate issued by "a company you haven't chosen to trust". Click "yes" when asked if you wish to proceed.


© 2005 Hewlett-Packard Development Company, L.P.

5. When prompted for your username and password, enter the root username and password.

6. Explore the **kcweb** interface to answer the following questions:
   a. What is the current value of the **nproc** kernel parameter?
   b. If you changed the value of the **nproc** parameter would a reboot be required?
   c. Is the **fddi4** module currently linked into the kernel?
   d. If you changed the state of the **fddi4** module would a reboot be required
   e. Are any alarms defined on your system?

   **Answer:**

   The value of **nproc** and the state of the **fddi4** module may vary from system to system. **nproc** is dynamically tunable, but **fddi4** is a static kernel module. By default, three alarms are preconfigured. When any monitored kernel resource reaches 90%, **kcmond** sends a message to the console, root's email, and the **syslog.log** file.

7. Exit out of the **kcweb** interface.

8. Shutdown the Apache/hpwebadm server daemon.

   **Answer:**

```
# kcweb -s stop
```

9. To streamline future attempts to access **kcweb**, configure the **kcweb autowa** service in **/etc/inetd.conf** and **/etc/services**. After making the change, use the **netstat** command to verify that **inetd** is listening on port 1110.

   **Answer:**

```
# waconf -a on
# netstat -an | grep 1110
        tcp 0 0 *.1110 *.* LISTEN
```

Reconnect to the server via the autostart functionality. Point your browser to **http://server:1110/casey/top.cgi**. (Note: Be sure to use **http** rather than **https** and port 1110 rather than 1188).

## Part 2: Logging and Troubleshooting

1. Reboot your system. When you get to the HPUX OS loader stage, terminate the autoboot sequence.

   **Answer:**

```
# cd /
# shutdown -ry 0
```


© 2005 Hewlett-Packard Development Company, L.P.

```
HP-UX Boot Loader for IA64  Revision 1.71
Press Any Key to interrupt Autoboot
 \EFI\HPUX\AUTO ==> boot vmunix
 Seconds left till autoboot - 10  Escape
 Type 'help' for help
HPUX>
```

2. Boot to tunable maintenance mode. What distinguishes tunable maintenance mode from a normal boot?

   **Answer:**

   ```
   HPUX> boot vmunix –tm
   ```

   When booting to tunable maintenance mode, all tunable parameters are set to the failsafe values defined by the module developers. These failsafe parameters are designed to ensure a successful boot to single-user mode so the administrator can troubleshoot kernel problems.

3. Once you reach tunable maintenance mode, mount all of your file systems. Then use the **`more`** command to review recent kernel changes that were recorded in the kernel change log.

   **Answer:**

   ```
   # mount –a
   # more /var/adm/kc.log
   ```

   At this point, if you needed to, you could reverse the changes found in `/var/adm/kc.log` to bring you back to a working kernel.

4. Reboot your system again and allow it to come up automatically with **`vmunix.`**

   **Answer:**

   ```
   # reboot
   HP-UX Boot Loader for IA64  Revision 1.71
   Press Any Key to interrupt Autoboot
    \EFI\HPUX\AUTO ==> boot vmunix
   ```


**© 2005 Hewlett-Packard Development Company, L.P.**

# 11-30. QUIZ: Review Questions

## Directions

Read and answer the following questions.

1. What is a system panic?

   **Answer:**

   A panic is when the system halts itself because it encountered a situation in which it did not know how to respond.

2. What is a crash dump?

   **Answer:**

   A crash dump is a core image stored to the dump area upon a system panic.

3. Which command reports configured-memory classes and dump declarations on a system?

   **Answer:**

   `crashconf -v`

4. What startup scripts are used for system dump configuration during boot? List the startup configuration files.

   **Answer:**

   Startup scripts:

   ```
   /sbin/init.d/crashconf
   /sbin/init.d/savecrash
   ```

   Startup configuration files:

   ```
   /etc/rc.config.d/crashconf
   /etc/rc.config.d/savecrash
   ```


© 2005 Hewlett-Packard Development Company, L.P.

# 11–31. LAB: Kernel Crash Dump Configuration

## Purpose

This lab is designed to give you hands on experience configuring memory classes and dump devices.

## Directions

Read and answer the following questions.

1. Use the **crashconf** command to view the current system crash dump configurations.

   ```
   # crashconf -v
   ```

2. Use the **crashconf** command to include the memory classes "***KCODE***" and "***USERPG***" within the system crash dump configuration.

   ```
   # crashconf –i kcode –i userpg
   ```

3. Use the **crashconf** command to exclude the "***KDDATA***" memory class from the system crash dump configuration.

   ```
   # crashconf –e kddata
   ```

4. Create a 128-MB logical volume to use as a dump device. Afterwards, dynamically declare that logical volume as additional dump space using the **crashconf** command.

   ```
   # lvcreate -L 128 -n dumplvol -C y -r n /dev/vg00
   # crashconf /dev/vg00/dumplvol
   ```

5. Configure your system so that the above system crash dump configurations occur automatically during boot. Note: On Integrity systems, the following **CRASH_INCLUDED_PAGES** and **CRASH_EXCLUDED_PAGES** entries do not apply. Instead, the administrator would issue **crashconf –e *class*** or **crashconf –i *class*** commands. To make the changes permanent, issue the **–t** option in the above commands.

   ```
   # vi /etc/rc.config.d/crashconf
     CRASH_INCLUDED_PAGES="kcode userpg"
     CRASH_EXCLUDED_PAGES="kddata"
     CRASHCONF_READ_FSTAB=1
   ```

   **NOTE**: For purposes of this lab, we are excluding kernel dynamic data (**kddata**), which provides an example of how to exclude a memory class. However, in real life, excluding "kddata" would render a crash dump virtually useless.

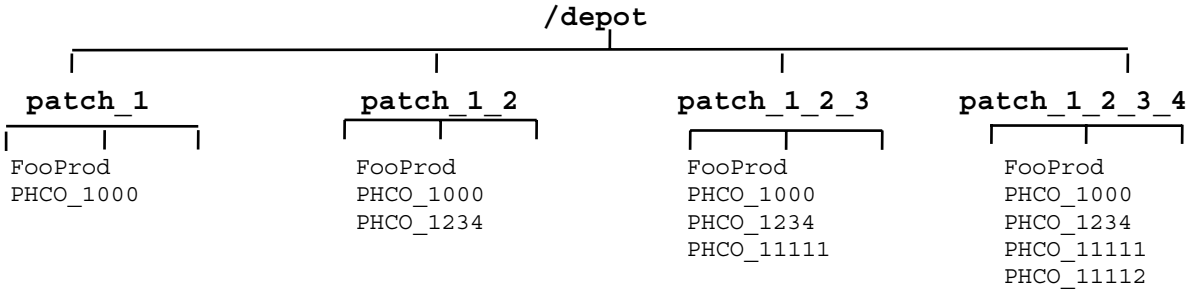
   ```
   # vi /etc/fstab
     /dev/vg00/dumplvol /       dump  defaults   0 0
   ```


© 2005 Hewlett-Packard Development Company, L.P.

# 12–21. LAB: Troubleshooting Patch Installation

## Directions

This lab exposes the student to patch installation and patch commit scenarios. The lab makes use of four depots installed under the **/depot** directory. The depots are organized as follows:




### Automatic Patch Selection

Demonstrate the **autoselect_patch** feature that automatically selects all patches for an existing product. The patch must be in the same depot as the product.

1. Install the **FooProd** product from the **patch_1** directory.

   ```
   # swinstall -s /depot/patch_1 FooProd
   ```

   Note the output from the "Analysis Phase." The patch **PHCO_1000** was automatically selected for the **FooProd** product.

2. List all products "not contained in a bundle."

   ```
   # swlist
   ```

   The product and the patch should be listed.

3. List the files that were patched by the **PHCO_1000** patch:

   ```
   # swlist -l file PHCO_1000
   ```

   There should be a file called **/usr/bin/foo**.

4. Display the contents of the patched file, **/usr/bin/foo**:

   ```
   # more /usr/bin/foo
   ```

   Note the last line indicating the product has been patched by `PHCO_1000`.


© 2005 Hewlett-Packard Development Company, L.P.

5. The **save** directory is the location of the original file, before being patched. List the **save** directory:

```
# ll /var/adm/sw/save
```

6. Note the name of the **PHCO_1000** patch. Under this directory are copies of the original files before the patch was installed.

7. Display the contents of the original file.

```
# more /var/adm/sw/save/PHCO_1000/FOO-MIN/usr/bin/foo
```

8. Verify that this file has been restored upon removal of the patch.

```
# swremove PHCO_1000
# more /usr/bin/foo
```

### Patch Selection of the Most Current Patch

Demonstrate that if two patches exist in a depot for the same product, the "most current" patch is selected.

1. Reinstall the **FooProd** product from a directory containing two patches: **patch_1234** supersedes **patch_1000**.

```
# swinstall -s /depot/patch_1_2 -x reinstall=true FooProd
```

2. Note that the most current patch is applied.

   Verify that the most current patch was applied:

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

### Reinstall the Product with the Most Recent Patch

Demonstrate that the more current patch (in this case **PHCO_11111**) will supersede the existing patch (**PHCO_1234**), if the product is reinstalled and a more current patch exists.

1. Reinstall the **FooProd** product with a more current patch. Do *not* delete the current patch already installed.

```
# swinstall -s /depot/patch_1_2_3 -x reinstall=true FooProd
```

2. Verify that the old patch and product were removed by the new installation.

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

### Apply a New Patch without Removing the First Patch

1. Install another patch (**PHCO_11112**) without removing any existing patches. This new patch has a dependency patch. Verify that the previously patched files are saved, and the dependency patch (**PHSS_5555**) is loaded with the new patch.


© 2005 Hewlett-Packard Development Company, L.P.

```
# swinstall -s /depot/patch_1_2_3_4 PHCO_11112
```

2. Verify that the new patch was installed without removing the existing patch.

```
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```

### Attempt to Remove a Patch needed for Rollback

1. The **FooProd** product contains two applied patches. **PHCO_11111** was applied first, and **PHCO_11112** was applied second. Try to remove patch **PHCO_11111**.

   ```
   # swremove PHCO_11111
   ```

   This should fail, because removing a patch needed for rollback is an "unsupported" operation.

### Verify Patch Rollback

1. Remove the **PHCO_11112** patch. Verify that the files roll back to the patch **PHCO_11111** state.

   ```
   # swremove PHCO_11112
   # swlist
   # more /usr/bin/foo
   # ll /var/adm/sw/save
   ```

2. Remove the **PHCO_11111** patch. Verify that the files roll back to the original state.

   ```
   # swremove PHCO_11111
   # swlist
   # more /usr/bin/foo
   # ll /var/adm/sw/save
   ```

3. Remove the **FooProd** product. Verify that all related patches are also removed.

   ```
   # swremove FooProd
   # swlist
   # more /usr/bin/foo
   # ll /var/adm/sw/save
   ```

### Test Applying Patches to Existing Products

1. Install the **FooProd** product. Specifically indicate *not* to install patches at this time.

   ```
   # swinstall -x autoselect_patches=false -s /depot/patch_1 FooProd
   ```

2. Verify only the product was installed:

   ```
   # swlist
   ```

3. Install patches to all existing, applicable products:


© 2005 Hewlett-Packard Development Company, L.P.

```
# swinstall -x patch_match_target=true -s /depot/patch_1_2_3_4
```

4. Verify that the most current patch has been installed:

```
# swlist
```

### Test Loading an Old Patch

1. Try to load a patch that has been "superseded" by an already loaded patch:

```
# swinstall -s /depot/patch_1_2 PHCO_1234
```

2. This should fail. Review log file for specific details:

```
# tail -50 /var/adm/sw/swagent.log | more
```

3. Remove the **FooProd** product. Verify that all related patches are removed.

```
# swremove FooProd
# swlist
# more /usr/bin/foo
# ll /var/adm/sw/save
```


© 2005 Hewlett-Packard Development Company, L.P.

## 12–22. LAB: Troubleshooting a corrupt IPD

### Directions

This lab exposes the student to troubleshooting a corrupt IPD.

1. Let's create a scenario of a missing IPD and then we will rebuild it. Please rename your **INDEX** file in preparation for the rest of the lab.

   **Answer:**

   ```
   # cd /var/adm/sw/products
   # mv INDEX INDEX.bak
   ```

2. Now, list the products that are currently installed on your system. Were you successful?

   **Answer:**

   ```
   # swlist -l product
   ```

   Since the **INDEX** file (the IPD) is not present, this operation will not succeed.

3. Using the steps from this module, rebuild your IPD.

   **Answer:**

   ```
   # cd /tmp
   # vi void.psf
           product
           tag void
           fileset
           tag void
           :wq!
   # swpackage -s /tmp/void.psf
   # swinstall void
   # swremove void
   # rm void.psf
   # swremove -d void
   ```

4. Now let's ensure the above procedure properly rebuilt the IPD.

   **Answer:**

   ```
   # swlist -l product
   ```


© 2005 Hewlett-Packard Development Company, L.P.

## 13–17. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What option is available with the **`ifconfig`** command to tear down the present LAN card configuration?

   **Answer:**

   `# ifconfig lanX unplumb`

2. What is the purpose of a *subnet mask*? If it is not set properly, what problems are likely to ensue?

   **Answer:**

   The subnet mask is used by the local system to determine if the destination system is on the same network. If destination system is on the same network, communication is direct. If destination system is on a different network, communication occurs through a router.

3. What is the purpose of *routing*? If entries within a route table are incorrect, what will happen?

   **Answer:**

   Routing determines the path to use when sending packets. If entries within the route table are incorrect, packets will likely not reach their destination.


© 2005 Hewlett-Packard Development Company, L.P.

# 13–18. LAB: Using `ndd` to set network parameters

## Purpose

This lab is designed to give the student experience with displaying and modifying tunable network parameters with **`ndd`**.

## Directions

### Displaying and Modifying Tunable Network Parameters

1. Display all tunable network parameters with **`ndd`**.

   **Answer**

   ```
   # ndd -h
   ```

2. Display the current value of **`ip_forwarding`** with **`ndd`**.

   **Answer**

   ```
   # ndd -get /dev/ip ip_forwarding
   ```

3. Set **`ip_forwarding`** to 0 with **`ndd`**.

   **Answer**

   ```
   # ndd -set /dev/ip ip_forwarding 0
   ```

4. Return the value of **`ip_forwarding`** to its original value with **`ndd`**.

   **Answer**

   ```
   # ndd -set /dev/ip ip_forwarding 2
   ```

5. Verify that **`ip_forwarding`** has returned to its original value with **`ndd`**.

   **Answer**

   ```
   # ndd -get /dev/ip ip_forwarding
   ```


© 2005 Hewlett-Packard Development Company, L.P.

# 13–19. LAB: Configuring the LAN Card

## Purpose

The goal of this portion of the lab exercise is to configure a new IP address and host name for all the machines in the classroom.

## Preliminary Steps

1. Just in case something goes wrong during this lab, make a backup copy of all of your network configuration files. There is a shell script in your **labs** directory designed specifically for this purpose. The shell script will save a tar archive backup of your network configuration files in the file you specify. Add the **-l** option to verify your backup.

   ```
   # /home/labs/netfiles.sh -s ORIGINAL
   # /home/labs/netfiles.sh -l
   # /home/labs/netfiles.sh -l ORIGINAL
   ```

2. Portions of this lab may disable your lan0 interface card. If you are using remote lab equipment, login via the GSP/MP console interface for the duration of the lab.

3. Changing your host name and IP on a running system can wreak havoc on CDE and other applications. Kill CDE before going any further:

   ```
   # /sbin/init.d/dtlogin.rc stop
   ```


© 2005 Hewlett-Packard Development Company, L.P.

# New LAN Card Configuration

## Directions

Your instructor will assign you a host name from the table below.

The first two octets in the IP addresses will vary from classroom to classroom, but should be consistent across all hosts within your classroom. Ask your instructor what the first two octets should be set to. The last two octets must be set in accordance with the table below.

| Host Name | IP Address |
|---|---|
| corp | ____.____.0.1 |
| sanfran | ____.____.1.1 |
| oakland | ____.____.1.2 |
| la | ____.____.1.3 |
| chicago | ____.____.2.1 |
| peoria | ____.____.2.2 |
| Rockford | ____.____.2.3 |
| Atlanta | ____.____.3.1 |
| athens | ____.____.3.2 |
| macon | ____.____.3.3 |
| nyc | ____.____.4.1 |
| albany | ____.____.4.2 |
| buffalo | ____.____.4.3 |
| paris | ____.____.5.1 |
| lyon | ____.____.5.2 |
| grenoble | ____.____.5.3 |
| london | ____.____.6.1 |
| leeds | ____.____.6.2 |
| ipswich | ____.____.6.3 |
| bonn | ____.____.7.1 |
| berlin | ____.____.7.2 |
| hamburg | ____.____.7.3 |
| tokyo | ____.____.8.1 |
| kyoto | ____.____.8.2 |
| osaka | ____.____.8.3 |

1. There should be a script in the **`/home/labs`** directory called **`netsetup.sh`**. This script will ask you for your instructor-assigned hostname, and the first two IP octets that your instructor should also provide. After you enter the requested parameter, use the displayed information to configure your new hostname, IP address, and **`/etc/hosts`** file. Be sure to reboot to ensure the new settings take effect. The new hostname and IP address will be used in this module and the next. Run the script as follows:
   ```
   # /home/labs/netsetup.sh
   ```


© 2005 Hewlett-Packard Development Company, L.P.

2. After the reboot, you should be able to **ping** the instructor's system. Enter the following command:

   ```
   #  ping corp
   ```

3. Were you successful? If so, continue. If not, redo question 1.

4. Let's make another "checkpoint" backup copy of this new network configuration as NEW. In the subsequent labs, we may need to restore back to this NEW configuration if something goes wrong.

   ```
   # /home/labs/netfiles.sh -s NEW
   ```

5. Execute the following command line:

   ```
   # /home/labs/mod13.lab1.sh
   ```

   Using the knowledge gained in the current module, fix your system so that you can **ping** the instructor's system.

   ```
   # ifconfig lan0
   lan0: flags=851<UP,POINTOPOINT,RUNNING,MULTICAST>
           inet 128.1.1.3 --> 255.255.0.0 netmask ffff0000
   ```

   LAN card is configured as a point-to-point interface. the configuration must be torn down and rebuilt.

```
# ifconfig lan0 unplumb

# ifconfig lan0
ifconfig: no such interface


# ifconfig lan0 128.1.1.3 netmask 255.255.0.0 up

# ifconfig lan0
lan0: flags=843<UP,BROADCAST,RUNNING,MULTICAST>
        inet 128.1.1.3 netmask ffff0000 broadcast 128.1.255.255
```

Validating access to "corp".

```
# ping corp
PING 128.1.0.1: 64 byte packets
64 bytes from 128.1.0.1: icmp_seq=0. time=14. ms
64 bytes from 128.1.0.1: icmp_seq=1. time=4. ms
64 bytes from 128.1.0.1: icmp_seq=2. time=1. ms
64 bytes from 128.1.0.1: icmp_seq=3. time=2. ms
64 bytes from 128.1.0.1: icmp_seq=4. time=0. ms

----corp PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/4/14
```


**© 2005 Hewlett-Packard Development Company, L.P.**

# 13–20. LAB: Host Communication Problems

## Purpose

This lab is designed to give you experience troubleshooting host communication problems.

## Directions

Read and answer the following questions.

1. Execute the **mod13.lab2.sh** script.

```
# cd /home/labs
# ./mod13.lab2.sh
```

After running this script, fix your system so that you can **ping** the instructor's system (corp).

2. Execute the **mod13.lab3.sh** script.

```
# cd /home/labs
# ./mod13.lab3.sh
```

After running this script, fix your system so you can **ping** the second LAN card connected to the instructor's system. Ask your instructor for the IP address of the second LAN card.
**Answer:**

Issuing **ping** command.

```
# ping 192.1.1.1
PING 192.1.1.1: 64 byte packets

----192.1.1.1 PING Statistics----
11 packets transmitted, 0 packets received, 100% packet loss
```

"host 192.1.1.1" is not responding to the ping command. Evaluating the current LAN card configuration.

```
# ifconfig lan0
lan0: flags=843<UP,BROADCAST,RUNNING,MULTICAST>
        inet 128.1.1.3 netmask ffff0000 broadcast 128.1.255.255
```

Lan card configuration is good. Evaluate route table.

```
# netstat -rn
Routing tables
Dest/Netmask      Gateway         Flags  Refs       Use  Interface  Pmtu
127.0.0.1         127.0.0.1       UH       0        739  lo0        4136
128.1.1.3         128.1.1.3       UH       0          0  lan0       4136
192.1.1.1         128.1.1.9       UGH      0          0  lan0       1500
128.1.0.0         128.1.1.3       U        2          0  lan0       1500
127.0.0.0         127.0.0.1       U        0          0  lo0        4136
```


© 2005 Hewlett-Packard Development Company, L.P.

The route table shows a route entry for “host 192.1.1.1.” The path taken to access “host 192.1.1.1” is through “gateway 128.1.1.9.” The gateway does not exist. This route entry must be deleted.

```
# route delete host 192.1.1.1 128.1.1.9
delete host 192.1.1.1: gateway 128.1.1.9
```

A new route entry must be defined that will allow access to “host 192.1.1.1”.

```
# route add net 192.1.1 128.1.0.1 1
add net 192.1.1: gateway 128.1.0.1
```

Evaluating route table.

```
# netstat -rn
Routing tables
Dest/Netmask    Gateway       Flags  Refs       Use  Interface  Pmtu
127.0.0.1       127.0.0.1     UH       0        739  lo0        4136
128.1.1.3       128.1.1.3     UH       0          0  lan0       4136
128.1.0.0       128.1.1.3     U        2          0  lan0       1500
192.1.1.0       128.1.0.1     UG       0          0  lan0       1500
127.0.0.0       127.0.0.1     U        0          0  lo0        4136
```

Checking access by issuing the **ping** command.

```
# ping 192.1.1.1
PING 192.1.1.1: 64 byte packets
64 bytes from 192.1.1.1: icmp_seq=0. time=1045. ms
64 bytes from 192.1.1.1: icmp_seq=1. time=37. ms
64 bytes from 192.1.1.1: icmp_seq=2. time=26. ms
64 bytes from 192.1.1.1: icmp_seq=3. time=70. ms
64 bytes from 192.1.1.1: icmp_seq=4. time=66. ms

----192.1.1.1 PING Statistics----
5 packets transmitted, 5 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 26/248/1045
```

3. Execute the **mod13.lab4.sh** script. **(Optional)**

```
# cd /home/labs
# ./mod13.lab4.sh
```

On the reboot, you should notice some failures in the rc scripts. In fact, it will probably hang at "Start NFS Client Subsystems". Break out of it and determine the problem.

**Answer:**

When the system reboots you should notice failures for the "Configure LAN interfaces" line.


**© 2005 Hewlett-Packard Development Company, L.P.**

```
  Create PCI FDDI device files. ............................... [ OK ]
  Configure HP 100BASE-T interfaces ........................... [ OK ]
  Configure LAN interfaces .................................... [FAIL]
*
  Start name server daemon .................................... [N/A ]
  Start NFS core subsystem .................................... [ OK ]
```

The boot will then hang on the "Start NFS client subsystem" line. You can break out with "<cntl> \", but it won't be immediate, so be patient.

```
  Start NIS client subsystem .................................. [ OK ]
> Start NFS client subsystem ................................... [busy]
  GenericSysName [HP Release B.11.11] (see /etc/issue)........... [    ]
  Console Login: pseudo devices for MAC/LLA access ............. [    ]
  Start multicast routing daemon .............................. [    ]
```

Since the problem occurred during the **rc** startup script, check **/etc/rc.log.**

```
 # vi /etc/rc.log
.
.
.
Configure LAN interfaces
Output from "/sbin/rc2.d/S340net start":
----------------------------
WARNING: Missing INTERFACE_NAME for corresponding IP_ADDRESS,
SUBNET_MASK,
BROADCAST_ADDRESS or INTERFACE_STATE in the file
/etc/rc.config.d/netconf.  Excess variables will be ignored.
ERROR:   lan0:1 interface: primary interface must first be
configured
ERROR:  Failed to add route entry because its interface is not
        yet initialized.  May need to add this route entry with
        a route commad after the interface is up :
        add net default: gateway 15.0.105.130: Network is
unreachable
"/sbin/rc2.d/S340net start" FAILED
```


© 2005 Hewlett-Packard Development Company, L.P.

Let's check **/etc/rc.config.d/netconf**.

```
 # vi /etc/rc.config.d/netconf
.
.
.

INTERFACE_NAME[0]="lan0"
IP_ADDRESS[0]="15.0.104.67"
SUBNET_MASK[0]="0xfffff800"
BROADCAST_ADDRESS[0]="15.0.111.255"
INTERFACE_STATE[0]=""
DHCP_ENABLE[0]=0
```

So far, so good.

Then, down the bottom of the file

```
INTERFACE_NAME[0]="lan0:1"
IP_ADDRESS[1]="192.168.168.67"
SUBNET_MASK[1]="0xffffff00"
BROADCAST_ADDRESS[1]="192.168.168.255"
INTERFACE_STATE[1]=""
DHCP_ENABLE[1]=0
```

Note the array index for **INTERFACE_NAME[0]**, should be 1 !

This is a very common mistake. Unfortunately, this happens frequently with "cut and paste" editing.

Fix the line and reboot.


**© 2005 Hewlett-Packard Development Company, L.P.**

## 14–13. QUIZ: Review Questions

### Directions

Read and answer the following questions.

1. What happens to a mounted client-side NFS file system if it gets unexported on the NFS server system? What do you do to rid the problem from the NFS client system?

   **Answer:**

   The client-side NFS file system becomes stale and inaccessible if it is un-exported by the server. The client must un-mount this file system and attempt to remount it once the server re-exports the file system.

2. What NFS server-side configuration file can cause problems when attempting to export NFS file systems? What is one of the most common problems experienced with this file? How can you fix that?

   **Answer:**

   The configuration file that can cause problems when attempting to export nfs file systems is **`/etc/exports`**. Common problems experienced with this file are syntactical. Viewing the manual page on **`/etc/exports`** is a sure-fire way of addressing this problem.

3. What happens to a functional NIS client if the NIS server goes down? List two scenarios that can play out.

   **Answer:**

   The NIS client will issue a broadcast request attempting to bind to an alternate NIS server that is part of the same NIS domain. If an alternate NIS server is available, it re-binds to that system. If an alternate NIS server were not available, an error message would be echoed to the NIS client system. That message would indicate something was wrong with the NIS server. In some cases, the NIS client will hang.

   If the NIS client is being booted and the NIS server is down, the following will occur: The NIS client will hang on the “NIS client subsystem“ message displayed in the startup checklist. Another possibility is the NIS client will echo a “FAIL” error message under the status column displayed in the startup checklist.

4. What is the name of the first DNS configuration file that is read when the **`named`** daemon is started? What is the purpose of this file? What could happen if the information contained within is incorrect?

   **Answer:**

   **`/etc/named.conf`** is read by the **`named`** daemon. **`/etc/named.conf`** is the boot file containing information about where the name server gets its initial data. If the information in this file is incorrect, the DNS daemon, **`named,`** will most likely fail to work properly.


© 2005 Hewlett-Packard Development Company, L.P.

# 14–14. LAB: NFS Problems

## Directions

In this lab, you will work with a partner to experiment with NFS problems. One of your systems will function as an NFS server, and the other will function as an NFS client. You should work *together* throughout the lab to ensure that you feel comfortable with both the client and server functionalities of NFS. At this point, decide between yourselves who will be the server and who will be the client.

Host name of server: _____

Host name of client: _____

## Preliminary Steps

1. **(Client)**

   Install the lab files needed on your client:

   ```
   # cd /home/labs

   # ./mod14.lab1.sh
   ```

   You should now have two new user accounts defined in your **`/etc/passwd`** file: "mickie" and "minnie." The passwords for the new accounts are "mickie" and "minnie" respectively. Note that neither user has a home directory on your machine. You will mount their home directories from your partner's NFS server.

2. **(Server)**

   Install the lab files needed on your server:

   ```
   # cd /home/labs

   # ./mod14.lab1.sh
   ```

   This script creates several new files and directories, and two new user accounts, "mickie" and "minnie," in your **/etc/passwd** file. The passwords for the new accounts are "mickie" and "minnie" respectively. The script also creates home directories for mickie and minnie.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 1: Exporting and Mounting NFS File Systems

1. **(Server)**

   Your clients need to access several files on your server machine. Export the following with the export options set as noted. Make the file systems available to clients immediately, but also ensure that they will be available after the next system boot by adding them to **/etc/exports**.

   | | |
   |---|---|
   | /home | **rw** for your partner's machine, no access for other hosts |
   | /opt/phone | **rw** for your partner's machine, read only for all others |
   | /opt/fun | read only for everyone on the LAN |

   **Answer:**

   ```
   server # vi /etc/exports
   /home         -access="client"
   /opt/phone    -rw="client"
   /opt/fun      -ro
   ```

2. **(Server)**

   What command can you use to see what file systems you have made available? Can you tell which export options you used?

   **Answer:**

   ```
   server # exportfs
   ```

   The `exportfs` command shows what is exported, and which export options were used.

   ```
   server# showmount -e {otherservername}
   ```

   The **showmount** command shows who can mount the file systems, but does NOT indicate what export options were used.

3. **(Client)**

   Create mount points for the file systems your neighbor exported, and mount them:

   ```
   /home/mickie
   /home/minnie
   /opt/fun
   /opt/phone
   ```

   **Answer:**

   ```
   client# mkdir /home/mickie /home/minnie /opt/fun /opt/phone

   client# mount server:/home/mickie /home/mickie

   client# mount server:/home/minnie /home/minnie

   client# mount server:/opt/fun /opt/fun

   client# mount server:/opt/phone /opt/phone
   ```


© 2005 Hewlett-Packard Development Company, L.P.

4. **(Client)**

   What file needs to be modified to ensure that these NFS file systems are automatically mounted after every system boot? Make it so. (For now, use the "defaults" mount option.)

   Syntax errors in the **/etc/fstab** file may cause the next system boot to fail. Do a **mount -a** to ensure that you didn't make any mistakes in **fstab** file.

   Finally, use **mount -v** to ensure that all the NFS file systems actually mounted properly.

   **Answer:**

```
client# vi /etc/fstab
server:/home/mickie /home/mickie nfs defaults 0 0
server:/home/minnie /home/minnie nfs defaults 0 0
server:/opt/fun     /opt/fun     nfs defaults 0 0
server:/opt/phone   /opt/phone   nfs defaults 0 0
```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 2: What If the NFS Client Loses LAN Connectivity?

1. During the remainder of the lab, you will be asked to shutdown your LAN card several times. Exit out of your X-windows/CDE login session, reconnect via the MP, and shutdown CDE.

   ```
   # /sbin/init.d/dtlogin.rc stop
   ```

2. **(Server and Client)**

   What happens if the NFS client loses LAN connectivity to the server? Execute the following commands and note the output. First note the client's behavior when the server is up. (It should be normal.)

   ```
   client# cd /opt/fun
   client# ls
   ```

   Now take the **server's** LAN card down and note what happens to the client:
   If you started CDE after the last lab, stop CDE now:

   ```
   server# /sbin/init.d/dtlogin.rc stop
   ```

   and log back into the server.

   ```
   server# ifconfig lan0 down
   client# ls                    #This will hang indefinitely
   ```

   Move on to next step.

   What happens when the client regains connectivity to the NFS server?

   ```
   server# ifconfig lan0 up
   client# ls
   ```

   **Answer:**

   This should work without any problems.

   The **`ls`** hangs indefinitely. Momentarily, you should get an **`NFS server not responding`** error.

   The **`ls`** command executes.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 3: What If the NFS Server Unexports a Client-Mounted File System?

**(Server and Client)**

1. What happens if the NFS server unexports a file system while an NFS client has it mounted? Do the following and note the output from the commands. First note the client's behavior when the NFS server's file system is exported. (It should be normal.)

```
client# cd /opt/fun
client# ls
client# bdf
```

Unexport the NFS server's file system and note what happens to the NFS client:

```
server# exportfs -uv /opt/fun
client# bdf                    #You should see a stale NFS file handle!
```

Move on to next step.

How do you get rid of this "*stale file handle*" from the NFS client?

```
client# cd /
client# umount /opt/fun
client# bdf                    #The bad entry should be gone!
```

What happens when the NFS server re-exports and the client re-mounts the file system?

```
server# exportfs /opt/fun
server# exportfs
client# mount -F nfs "server":/opt/fun /opt/fun
client# bdf
```

Now **umount** all NFS file systems.

```
client# umount -aF nfs
```


© 2005 Hewlett-Packard Development Company, L.P.

## Part 4: Troubleshooting a Nonresponsive NFS Server

**(Client and Server)**

1. You have seen the effect that a downed NFS server has on NFS clients. What can the administrator on the client side do to determine what might be wrong on the server side? Start by doing some experiments while both your server and client are functioning properly. Bring the LAN cards on both machines to an "UP" state:

   ```
   server# ifconfig lan0 up
   client# ifconfig lan0 up
   ```

   Now test connectivity from the client to your NFS server:

   ```
   client# ping "server"
   client# rpcinfo -p "server"
   ```

**(Client and Server)**

2. Now shut down NFS on your NFS server.

   ```
   server# /sbin/init.d/nfs.server stop
   ```

   From the client, try mounting **/opt/fun**.

   `client# mount /opt/fun` # After you see the error, hit **Ctrl** + **c**.

**(Client)**

3. From the client, try your connectivity test commands again:
   ```
   client# ping "server"
   client# rpcinfo -p "server"
   ```

   Can you still **ping** the server?

   Which RPC programs are no longer available on the server?

   Will clients be able to mount NFS file systems now? Why?

   Will clients be able to access already mounted NFS file systems? Why?

   **Answer:**

   You can still **ping** the server, but the **nfs rpcs** are no longer registered with the server's **rpcbind**. The client can't mount the file system since the **mountd** daemon isn't running on the server anymore. Nor can the client access already mounted NFS file systems since the **nfs** daemons are not running.


© 2005 Hewlett-Packard Development Company, L.P.

## Part 5: Cleaning Up

Before moving on to the next part of the lab, restore your network configuration parameters to what you started with (prior to the NFS lab) by typing:

```
# /home/labs/netfiles.sh -r NEW
```


© 2005 Hewlett-Packard Development Company, L.P.

# 14–15. LAB: DNS Data File Problem

## Purpose

This lab is designed to give you experience troubleshooting DNS data file problems.

1. Execute the **mod14.lab2.sh** script.

```
# cd /home/labs
# ./mod14.lab2.sh
```

After running this script, your system will be configured as a DNS server and client. The DNS server will be able to resolve all host names to IP addresses in the **/etc/hosts** file except for one entry. That entry is as follows:

```
x.y.9.1   philly.pa.hp.com
```

(where x and y match the first two octets of your current IP address)

Using the knowledge gained in the current module, fix the DNS server so that it recognizes and resolves the host name, **philly.pa.hp.com**, to the appropriate IP address.

**Answer:**

```
root@sanfran # nslookup philly
Name Server:  sanfran.hp.com
Address:  128.1.1.1

Trying DNS
Trying NIS
looking up FILES
*** No address information is available for "philly"
```

The hostname **philly** is not known to DNS. Check another hostname (that we would expect to find), just to make sure DNS is working.

```
root@sanfran # nslookup la
Name Server:  sanfran.hp.com
Address:  128.1.1.1

Trying DNS
Name:    la.hp.com
Address:  128.1.1.3
```

Check the **hosts** file for **philly**.

```
root@sanfran # grep philly /etc/hosts
128.1.9.1       philly.pa.hp.com
```


© 2005 Hewlett-Packard Development Company, L.P.

So, **/etc/hosts** knows about the hostname **philly**. Check where DNS's data files are. The **named.conf** file lists the directory for storing the RR datafiles.

```
root@sanfran # grep directory /etc/named.conf
directory "/";  // running directory for named
```

So, DNS's files are being stored in "/" . This is almost certainly a mistake. Check these files.

```
root@sanfran # ls /db.*
/db.127.0.0   /db.128.1     /db.128.1.1 /db.hp

root@sanfran # grep philly /db.hp
```

So the name **philly** is not listed in these files. Where are DNS's files supposed to be?

```
root@sanfran # find /etc /var -name db.hp
/etc/named.data/db.hp

root@sanfran # grep philly /etc/named.data/db.hp
philly.pa       IN      A       128.1.9.1
philly.pa       IN      MX      10      philly.pa.hp.com.
```

So it looks like the DNS files are supposed to be in **/etc/named.data**, and that the copies in here are up to date.

Before we can easily move to the use the versions of the files in the **/etc/named.data** directory we need to check out issues with the db file version/serial numbers.

```
root@sanfran # grep Serial /db.*
/db.127.0.0:                                    1       ; Serial
/db.128.1:                                      1       ; Serial
/db.15.0.104:                                   1       ; Serial
/db.hp:                                          1       ; Serial
```

So, all of these files have a serial number of "1". The files under **/etc/named.data** will have to have their serial numbers updated to be greater than this. This entails editing all the **db.*** files.

```
root@sanfran # cd /etc/named.data
root@sanfran # ls
boot.cacheonly  conf.sec        db.127.0.0      db.15.0.104     param
conf.cacheonly  conf.sec.save   db.128.1        db.hp
root@sanfran #
root@sanfran # for i in db.*
do
ed $i <<EOF
2s/1/2/
w
q
EOF
```


**© 2005 Hewlett-Packard Development Company, L.P.**

```
done
245
245
1114
1114
250
250
3696
3696
root@sanfran[named.data] grep Serial db.*
db.127.0.0:                                     2       ; Serial
db.128.1:                                       2       ; Serial
db.15.0.104:                                    2       ; Serial
db.hp:                                          2       ; Serial
```

You might prefer to just use vi.

Now we need to update **/etc/named.conf** to reference the correct directory.

```
root@sanfran # vi /etc/named.conf
options {
        .
        .
        .
        directory "/etc/named.data";    // running directory for named
        .
        .
        .
};
```

After updating the version numbers and fixing **named.conf**, we just need to restart the name server.

```
root@sanfran # sig_named restart
Name server restarted
```

Then test to see whether DNS now recognizes the name philly.

```
root@sanfran # nslookup philly
Name Server:  sanfran.hp.com
Address:  128.1.1.1

Trying DNS
Name:    philly.pa.hp.com
Address:  128.1.9.1
```


© 2005 Hewlett-Packard Development Company, L.P.

## Cleaning Up

When you are finished with all labs, restore your ORIGINAL network configuration as follows:

```
# /home/labs/netfiles.sh -r ORIGINAL
```


© 2005 Hewlett-Packard Development Company, L.P.