AIX 6 System Administration II: Problem Determination
(Course code AU16)

Student Exercises

ERC 14.0

IBM certified course material
Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM® is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

AIX®
DS4000™
General Parallel File System™
Notes®
POWER5™
POWER Gt3™
RS/6000®
Tivoli®

AIX 5L™
eServer™
GPFS™
POWER™
POWER6™
pSeries®
SP™
TotalStorage®

DB2®
FlashCopy®
Micro-Partitioning™
POWER4™
POWER Gt1™
Redbooks®
System p™
xSeries®

Alerts® is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX® is a registered trademark of The Open Group in the United States and other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

December 2007 edition

The information contained in this document has not been submitted to any formal IBM test and is distributed on an “as is” basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will result elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

© Copyright International Business Machines Corporation 1997, 2007. All rights reserved.
This document may not be reproduced in whole or in part without the prior written permission of IBM.
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
Contents

Trademarks ................................................................. v

Exercise Description .................................................. vii

Exercise 1. Problem Determination Introduction .................. 1-1

Exercise 2. The Object Data Manager (ODM) ....................... 2-1

Exercise 3. System Initialization Part 1 ............................. 3-1

Exercise 4. System Initialization Part 2 ............................. 4-1

Exercise 5. LVM Tasks and Problems ............................... 5-1

Exercise 6. Mirroring rootvg ......................................... 6-1

Exercise 7. Exporting and Importing Volume Groups ............. 7-1

Exercise 8. Saving and Restoring a User Volume Group .......... 8-1

Exercise 9. Error Log and syslogd ................................. 9-1

Exercise 10. Diagnostics ............................................. 10-1

Exercise 11. System Dump ........................................... 11-1

Exercise 12. Basic Performance Commands ....................... 12-1

Exercise 13. Performance Diagnostic Tool ....................... 13-1

Exercise 14. Authentication and ACLs ............................ 14-1

Appendix A. Auditing .................................................. A-1
Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM® is a registered trademark of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

- AIX®
- DS4000™
- General Parallel File System™
- Notes®
- POWER5™
- POWER Gt3™
- RS/6000®
- Tivoli®
- AIX 5L™
- eServer™
- GPFS™
- POWER™
- POWER6™
- SP™
- TotalStorage®
- DB2®
- FlashCopy®
- Micro-Partitioning™
- POWER4™
- POWER Gt1™
- pSeries®
- System p™
- xSeries®

Alerts® is a registered trademark of Alphablox Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX® is a registered trademark of The Open Group in the United States and other countries.

Linux® is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.
Exercise Description

Each exercise in this course is divided into sections as described below. Select the section that best fits your method of performing exercises. You may use a combination of these sections as appropriate.

Exercise Instructions

This section tells you what to accomplish. There are no definitive details on how to perform the tasks. You are given the opportunity to work through the exercise given what you learned in the unit presentation, utilizing the Student Notebook, your past experience, and maybe a little intuition.

Exercise Instructions with Hints

This section is also an exact duplicate of the Exercise Instructions and contains solutions and additional tips for the students. If very inexperienced students take part in this course, they should work with this section.

Students can use this part to compare their work with the solutions.

When showing the SMIT method to accomplish a task, each line in bold represents a submenu or selector screen. You will need to press the Enter key after selecting each item as listed. When you reach the dialog screen, the field descriptions will be in regular text and the items you need to fill in will be in bold. Only the items that need to be changed will be shown, not the entire screen. Once you have reached the dialog screen portion of SMIT, press Enter ONLY after all indicated entries have been made.

The SMIT steps will be shown for the ASCII version of SMIT. Under most circumstances these steps match the steps taken if using the graphics version of SMIT. The exceptions relate to the use of the function keys. When instructed to press the F3 key back to a particular menu, when in graphics SMIT, you will instead click the Cancel box at the bottom of the screen. When instructed to press the F9 key to shell out, in graphics mode, simply open another window.

Note: The Web-based System Manager interfaces are currently not shown.

Optional Exercise Parts

Some labs provide additional practice on a particular topic. Specific details and hints are provided to help step you through the Optional Exercises, if needed. Not all exercises include Optional Exercises.
According to the group, the instructor can decide to do them or not. If there is time, the optional part should be executed by the students.
Text highlighting

The following text highlighting conventions are used throughout this book:

**Bold** Identifies file names, file paths, directories, user names, principals, menu paths, and menu selections. Also identifies graphical objects such as buttons, labels, and icons that the user selects.

*Italics* Identifies links to Web sites, publication titles, is used where the word or phrase is meant to stand out from the surrounding text, and identifies parameters whose actual names or values are to be supplied by the user.

**Monospace** Identifies attributes, variables, file listings, SMIT menus, code examples, and command output that you would see displayed on a terminal, and messages from the system.

**Monospace bold** Identifies commands, subroutines, daemons, and text the user would type.
Exercise 1. Problem Determination Introduction

What This Exercise Is About

This exercise will acquaint you with the system that you will be using throughout this course. You will recall some basic administration commands. This exercise also provides you with an opportunity to work with the Service Update Management Assistant (SUMA).

What You Should Be Able to Do

At the end of the lab, you should be able to:

- List volume groups, physical, and logical volumes on your system
- Identify real memory and paging space on your system
- Identify the hardware platform and processor type of your system
- Use the Service Update Management Assistant (SUMA)

Introduction

In this exercise, you will obtain and record information about your system using some basic administration commands with which you are probably already familiar. You will also use the Service Update Management Assistant (SUMA).

You will require root authority to complete this exercise.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Recording system information

__ 1. Using commands rather than SMIT, collect and record the following information regarding your system:
   __ a. The volume groups on your system:

   __ b. The physical volumes for your system:

   __ c. The logical volumes in rootvg on your system:

   __ d. All paging space areas for your system:

   __ e. Real memory on your system:

   __ f. Hardware platform (architecture) of your system:

   __ g. Processor type of your system:

__ 2. Identify the logical volumes that reside on your hdisk0.
   Write down the command you used:

   From the fact that the number of LPs is equal to the number of PPs, what can you conclude?
Using the Service Update Management Assistant (SUMA)

__3. Display the man page for the suma command.
   Locate the examples illustrating use of this command. How many such examples are there?

__4. As illustrated in the suma man page examples (example 7), enter a suma command that will create and schedule a repeating (-a Repeats=y) task that will check for the latest level of the bos.rte.install fileset on the 15th of each month at 3:30 AM.

__5. Enter a suma command that will list information regarding the SUMA task you just created.

End of exercise
Exercise 2. The Object Data Manager (ODM)

What This Exercise Is About

This exercise will review some of the most important ODM files and how they are used in device configuration. You will use the ODM command line interface.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Describe some of the most important ODM files
- Use the ODM command line interface
- Explain how ODM classes are used by device configuration commands

Introduction

This exercise has three parts:

1. Review of device configuration ODM classes (PdDv, PdAt, CuDv, CuAt, CuDep, CuDvDr).
2. Role of ODM during device configuration.
3. Optional Part: Creating self-defined ODM classes.

All instructions in this exercise require root authority.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Review of device configuration ODM classes

__ 1. Execute the `lsdev` command and identify all devices that are supported on your system. Tell the `lsdev` command to provide column headers in the output.
   
   What is the command you used?

   __________________________________________________________

   Which ODM class is used by the `lsdev` command to generate this output?

   __________________________________________________________

__ 2. Execute the `lsdev` command and identify all disk devices that are currently attached to your system. Tell the `lsdev` command to provide column headers in the output.
   
   What is the command you used?

   __________________________________________________________

   Which ODM class is used by the `lsdev` command to generate this output?

   __________________________________________________________

__ 3. Request the same listing as above, except customize the reported fields needed to complete the following list for disk `hdisk0`:

   Name: ___________________________________

   Status: _________________________________

   Location: ______________________________

   Physical location: ______________________

   Description: ___________________________
5. From this output please answer the following questions.

What is the meaning of the descriptor \texttt{chgstatus}?

The \texttt{lsdev} command provides a description field, which is not part of ODM class \texttt{CuDv}. Where does the description come from?

6. Execute the \texttt{lsattr} command and identify the \textit{physical volume identifier} for your \texttt{hdisk0}.

What is the command you used?

Write down the physical volume ID of the disk:

\texttt{pvid}: ______________________________________________________________________

7. Use the ODM command line interface, and list the ODM object that stores the \textit{physical volume identifier}.

What is the command you used?

8. The \texttt{/dev} directory contains the special files to access the devices. Write down the major and minor number of the special file for \texttt{hdisk0}.

Major number: ________________  
Minor Number: ________________

Which ODM class is used to identify the major number and minor number for the device driver?
9. List all your logical volumes that are part of the **rootvg**.
What is the command you used?

---------------------
Query the ODM class **CuDep** and identify all logical volumes that belong to **rootvg**.
What is the command you used?

---------------------

**Role of ODM during device configuration**

10. During the following steps we will simulate the configuration of an SCSI disk without using **cfgmgr**.

**Important:** This is just a simulation. You do not attach a real disk in this exercise.
The ability to simulate a non-existent disk depends on having a physical SCSI adapter which is in an available state.

For lab systems which only have virtual SCSI, this exercise section ends after this step (you may wish to go to the start of the following optional part).

The ODM contains predefined objects to support many types of different disks.
Use the **lsdev** command to list all predefined devices of class **disk**.
Write down the command you used.

---------------------
Locate the predefine device object that support the disks your system has configured. Use the PdDvLn value you previously recorded in step 4 to make the match up.

For example, if you have physical disks you might have a disk type of **osdisk** and a subclass of **scsi** (Other SCSI Disk Drive). Or, if you have virtual disks, you would have a disk type of **vdisk** and a subclass of **vscsi**.

Use **odmget** to identify the object in **PdDv** that describes your disk type and subclass.
Write down the command you used.

---------------------
From the output complete the following:

- type: __________
- class: __________
- subclass: __________
- prefix: __________
Device Driver: _________
Configuration Method: _________

11. We will use the device scsi0 (or an alternative device such as scsi1 or scsi2 if specified by your instructor) as the parent device for the disk we are configuring. This is where the new disk will be attached to the system.

12. Before configuring the device, a free SCSI ID must be identified. List all ODM objects in CuDv where scsi0 (or the alternative device specified by your instructor) is stored as the parent device.

Write down the command you used.

From the output, write down the SCSI IDs which are in use:
SCSI IDs in use: __________________________
Choose a free SCSI ID and write it down in the table. You need to specify this ID later.
Free SCSI ID: __________________________

13. Get the disk into the defined state using the mkdev -d command. You need to pass the following information to mkdev:
- Device class
- Device subclass
- Device type
- Parent device
- SCSI address

Write down the command you used to define the disk.

What device name has been assigned to the disk?
Device name: __________________________

14. Using this newly assigned name, list the object that stores information about your disk in the customized database.

Write down the command you used:

15. Try to configure the disk using mkdev -l. What happens?

__________________________________________________________
__________________________________________________________

16. Finally, remove the disk from the system using rmdev.

Write down the command you used:
Examine your CuDv object class. Did you find the removed disk in this object class?

---

**Optional Part: Creating self-defined ODM classes**

--- 17. Before creating an ODM class you need to specify the descriptors that are contained in the class. Create the directory `/tmp/odm` to hold the specification file and `cd` to that directory.

Using an editor, create a file `parts.cre` (in your new working directory) with the following class structure:

```c
class parts {
  long        part_number;
  char        part_description[128];
  char        warehouse[4];
  long        contained_in;
}
```

--- 18. Create the ODM class using this class structure and check the structure of this class. Write down the commands you used:

---

Identify in your working directory, which files have been created during this step.

---

What do you think is the purpose of these files?

---

Where does the ODM class `parts` reside?

---

19. Create some objects in ODM class `parts`, using the following data:

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Warehouse</th>
<th>Contained In</th>
</tr>
</thead>
<tbody>
<tr>
<td>10001</td>
<td>Wheel</td>
<td>a12</td>
<td>50001</td>
</tr>
<tr>
<td>10003</td>
<td>Frame</td>
<td>a19</td>
<td>50001</td>
</tr>
<tr>
<td>10005</td>
<td>Saddle</td>
<td>a01</td>
<td>50001</td>
</tr>
<tr>
<td>Part Number</td>
<td>Description</td>
<td>Warehouse</td>
<td>Contained In</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------</td>
<td>-----------</td>
<td>--------------</td>
</tr>
<tr>
<td>10006</td>
<td>Front wheel brake</td>
<td>a03</td>
<td>50001</td>
</tr>
<tr>
<td>10007</td>
<td>Rear wheel brake</td>
<td>a03</td>
<td>50001</td>
</tr>
<tr>
<td>50001</td>
<td>City Bike Easy Rider</td>
<td>x99</td>
<td></td>
</tr>
</tbody>
</table>

20. List all objects that are contained in part 50001 (the City Bike Easy Rider). Write down the command you used:

__________________________________________________________

21. Change the warehouse location for part Wheel to b10.

22. Finally, remove ODM class parts from the system. Write down the command you used.

__________________________________________________________

End of exercise
Exercise 3. System Initialization Part 1

What This Exercise Is About

This exercise will review the hardware boot process of an AIX system.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Boot a machine in maintenance mode
- Repair a corrupted boot logical volume
- Alter bootlists

Introduction

This exercise has three parts:

1. Identify the bootlists of your system
2. Identify LVM information of your system
3. Repair a corrupted boot logical volume

All instructions in this exercise require root authority.

Requirements

- The program /home/workshop/ex3pro1
- Bootable media that matches the version and release of your system or a NIM server setup that can be used to execute a remote boot).
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Working with bootlists and identifying information on your system

1. What is the boot sequence of your system for a normal boot?
   • Boot device:___________________________
   • What is the command you used, to determine the bootlist? ________________

2. Does your model support a customized service bootlist? ___________
   • What command did you use? ______________________________
   • If your model supports a service bootlist for maintenance mode, write down the boot sequence for this boot mode:
     1) Boot device:_______________________________
     2) Boot device:_______________________________
     3) Boot device:_______________________________
     4) Boot device:_______________________________

3. Identify which disks are contained within the rootvg:
   ____________________________________________
   • What command did you use? _________________________________
   • Which disk is the bootable disk? (That means the disk that contains the boot logical volume hd5): ______________________
     What command did you use? ________________________________
• What is the logical volume type of hd5? _________________________
  What command did you use? ________________________________
Part 2 - Identify LVM information from your system

4. Alter the boot sequence of your system using the `bootlist` command. Set the normal bootlist so it contains only the bootable hard disk.

5. The Logical Volume Manager uses names and IDs when storing information. Complete the following table that maps names to IDs:

<table>
<thead>
<tr>
<th>rootvg VGID</th>
<th>First disk PVID</th>
<th>Second disk PVID</th>
</tr>
</thead>
</table>

- What command did you use to determine the `rootvg` VGID? ________________

- What command did you use to determine the physical volume IDs? __________

- Using `odmget`, identify the attribute `pvid` of one of your disks from ODM class `CuAt`.
  - What command did you use?_________________________________________

- What difference do you see with the ID value?
Part 3 - Booting to maintenance mode

6. Before creating any boot problems, verify that you can boot into maintenance mode. This will be crucial to fixing the problem.

For local machines, where your system console is a physical display, the procedure is fairly straightforward:

   i. Place the provided installation media in the CD/DVD drive.
   ii. Shut down your machine by using the shutdown -F command from the root level prompt.
   iii. Power on your machine.
   iv. Soon after the LED reads E1F1, the keyboard is discovered and a distinctive beeping will sound. Between that time and before the system begins to search for the boot image, press either the F6 or numeric 6 key (depending on type of machine) to invoke the service mode bootlist.
   v. The console should eventually display the Installation and Maintenance menu.
   vi. Power off your machine and power it back on (which allows it to boot normally to multi-user mode).

For remote machines the procedure, at a high level, is as follows:

   i. Shut down your AIX operating system by running the shutdown -F command from the AIX root level command prompt.
   ii. Access the HMC and locate the icon for your LPAR.
   iii. Activate the LPAR into maintenance mode, using the service bootlist.
   iv. Shut down the LPAR from the current maintenance mode.
   v. Start the LPAR back up into multi-user mode.

Except for the shutdown of a running AIX operating system, details of this will depend on the level of HMC with which you are working. The details for the different HMC versions are on the following pages:
If using **WebSM** with a version of **HMC prior to version 7**: 

i. **Access the HMC and locate your LPAR:**
   1. Start the Web-based System Manager client (icon on your lab workstation desktop). Note that the lab workstation may be a portal machine at the remote server location.
   2. Enter the IP address of your assigned HMC (provided by instructor) and click **OK**.
   3. Enter the provided userid and password to login.
   4. In the navigation area, click on the address of the HMC.
   5. In the content area on the right, double-click the **Server and Partition** icon.
   6. Double-click the **Server Management** icon.
   7. Click the + next to the name of the assigned managed system (provided by instructor).
   8. Click the + next to **Partitions**.
   9. Look for the name of your assigned partition (provided by instructor).

ii. **Activate your LPAR into maintenance mode (with customized service bootlist):**
   1. When the partition state is **Not Activated**, proceed to activate the partition.
   2. Select the partition and right click to get the context menu.
   3. On the menu, click **Close Terminal Connection** and confirm when prompted.
   4. Select the partition and right-click to get the context menu.
   5. On the menu, click **Activate**.
   6. When the **Activate Logical Partition** panel appears, click the box next to **open terminal window**, and click **OK** (the intent is to have you manually signal which bootlist to use later in step 9).
   [Alternatively, you may choose to click the “Advanced” button and specify “Diagnostic with Stored Bootlist” as the boot mode. In that case you would not need to follow step 9.]
   7. A terminal console window should appear.
   8. The virtual console will display the discovered devices (soon after the HMC shows an operator panel value of E1F1).
   9. When the keyboard is discovered, a distinctive beeping will sound and the “keyboard” key word will appear. Between that time and before the system begins to search for the boot image, press the numeric 6 key to invoke the service mode bootlist. This happens very quickly for LPARs, and soon after
staring the partition you may want to simply press the 6 key periodically right after activation.

10) The console should eventually display the **Installation and Maintenance** menu.

iii. Shut down the partition from maintenance mode (if accessing the root level command prompt, simply run `shutdown -F`):
   1) In the HMC **Server and Partition** panel, right-click your partition icon.
   2) On the resulting menu, click **Close Terminal Connection** and confirm when prompted.
   3) On the **Server and Partition** panel, right-click your partition icon, and select **Shutdown Partition**.
   4) On the **Shutdown Partitions** panel, select **immediate** and click **OK**.

iv. Start your partition in multiuser mode (normal bootlist):
   1) When the partition state is **Not Activated** proceed to activate the partition.
   2) Select the partition and right-click to get the context menu.
   3) On the menu, click **Activate**.
   4) When the **Activate Logical Partition** panel appears, click the box next to **open terminal window**, and click **OK**.
   5) A terminal console window should appear.
   6) The **Server and Partition** operator panel will display the LED values during boot up.
   7) When the **init** process starts, the virtual console will display the remaining boot progress.
   8) You should eventually receive a login prompt.

v. Opening a **virtual console**
   1) Access the HMC and locate your LPAR (as described in earlier).
   2) Select the partition and right-click to get the context menu.
   3) Select **Close Terminal Connection** and confirm when prompted.
   4) Select the partition and right-click to get the context menu.
   5) Select **Open Terminal Window**.
If using a Web browser with HMC version 7 or later:

i. Access the HMC and locate your LPAR:
   1) Start a browser on your lab workstation (note that the workstation may be a portal machine at a remote location).
   2) Enter a URL of: https://<IP address of your HMC>.
      This will take you to an HMC status window which has 3 status indicators and a link with the text:
      “Log on and Launch the Hardware Management Console web application”

   3) Click the log-on link to launch the HMC logon panel.
   4) Enter the userid of hscroot and the password (either abc123 or abc1234) and click the logon button. This should launch the HMC Web interface.
   5) In the left navigation area click Systems Management. The Systems Management item should expand to show “Servers” and “Custom Groups.”
   6) Click the Servers item. The “Servers” item should expand to show the managed systems.
   7) Click the managed system which is assigned to your team. In the Content Area on the right, you should see a list of logical partitions defined for your assigned system.
   8) Select your assigned logical partition by clicking the box under “Select” for your LPAR. After a short delay you should see a small menu icon appear to the right of your LPAR name, and the Tasks Area on the bottom half of the panel should update to reflect operations which are appropriate for the selected target.
   9) If you left-click the new menu icon (to right of the LPAR name), then you should see a menu which is similar to what you see in the Tasks Area.

ii. Activate your LPAR into maintenance mode (with service bootlist):
   1) When the partition state is Not Activated, proceed to activate the partition.
   2) Select the partition (if not already selected).
   3) When the small menu icon appears, click it to show the menu and click the Operations task.
   4) When the subtasks appear, click the Activate subtask.
   5) In the pop-up window labeled Activate Logical Partition: <your lpar name>, click the small box next to Open a terminal window or console session and also click the Advanced button. This should result in a new pop-up window labeled “Activate Logical Partition - Advanced”.

Course materials may not be reproduced in whole or in part without the prior written permission of IBM.
6) In the new pop-up window, click the menu icon to the right of “Boot Mode” and select **Diagnostic with stored bootlist**.

You may alternately boot in Normal mode, but you would then have to press numeric 6 at the appropriate time (right after keyboard discovery) to cause a service boot. Click **OK** to exit this pop-up.

7) On the **Activate Logical Partition: <your lpar name>**, click **OK**. A virtual terminal window should appear and you should see information about the progress of the BOOTP request, followed by a “Welcome to AIX” display, and finally followed by a prompt to “Define the System Console”.

8) Respond to the various boot prompts until you see the “Maintenance” menu.

iii. Shut down the partition from maintenance mode (if accessing the root level command prompt, simply run `shutdown -F`):

1) On the HMC Content Area, make sure your LPAR (and only your LPAR) is currently selected.

2) Click the menu icon, click the **Operations** task and then click the **Shutdown** subtask. This should result in a pop-up window.

3) In the shutdown window, select **Immediate** and then click **OK**.

4) The partition shutdown is complete when the “Status” field for your LPAR changes from “Running” to “Not Active”.

iv. Start your partition in multiuser mode (normal bootlist):

1) When the partition state is **Not Activated**, proceed to activate the partition.

2) Select the partition (if not already selected).

3) When the small menu icon appears, click it to show the menu and click the **Operations** task.

4) When the subtasks appear, click the **Activate** subtask.

5) In the pop-up window labeled “Activate Logical Partition: <your lpar name>”, click the small box next to “Open a terminal window or console session” (unless you already have a virtual console window open) and also click **OK**.

6) You should eventually see a login prompt appear in the virtual console window.

v. Opening a **virtual console**

1) Locate and select your LPAR, as described earlier.

2) Left-click the menu to the right of your LPAR name.
3) Left-click the “Console Window” item.
4) Left-click the “open terminal connection” item.
Part 4 - Repair a corrupted boot logical volume

___ 7. Execute the program /home/workshop/ex3pro1. When the prompt is returned, shut down and reboot the system.

___ 8. What happens on your system during the reboot?

___ 9. Boot to maintenance mode to do the repair:

___ 10. Repair the boot logical volume.

   The procedure for using the maintenance menu to repair the boot logical volume is the same for all environments:
   • Access the rootvg with all mounted file systems.

___ 11. In the maintenance shell, check that hdisk0 is in the normal bootlist and that the rootvg actually it has a boot logical volume on it. Correct if needed.

___ 12. In the maintenance shell, rebuild the boot image on the boot logical volume. Write down the command you used.

   »

___ 13. If the command executes successfully, reboot your system in normal mode.

End of exercise
Exercise 4. System Initialization Part 2

What This Exercise Is About

This exercise will review the software boot process of an AIX system.

What You Should Be Able to Do

At the end of the lab, you should be able to:

• Boot a machine in maintenance mode
• Repair a corrupted log logical volume
• Analyze and fix an unknown boot problem

Introduction

This exercise has two parts:

1. Repair a corrupted log logical volume
2. Analyze and fix a boot failure

All instructions in this exercise require root authority.

Required Material

• Program /home/workshop/ex4pro1
• Bootable media that matches the version and release of your system or a NIM server setup that can be used to execute a remote boot)
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Repair a Corrupted Log Logical Volume

Before starting the lab, read the following paragraph carefully.

If you have any questions, ask the instructor before starting the exercise steps.

Files or directories which are created or updated are stored with their i-nodes and the superblock of the file system in memory first. Most write requests are handled in memory first to improve system performance. Every minute or 16 KB of changes the syncd daemon writes the changes from memory to disk.

All changes in the JFS file systems (superblock, i-nodes, list of free data blocks, and so forth) are recorded in a log logical volume. The rootvg uses as default the log logical volume /dev/hd8. When the changes are written to the disk, the JFS transactions are removed from the log logical volume. This guarantees the integrity of a file system. Until the file system changes are written to disk, the changes are recorded and held in the log logical volume.

In this part of the lab, we corrupt the jfslog to stress a boot failure.

1. Check to see if your rootvg file systems are JFS or JFS2. You will need this information later in this exercise.

2. Execute the program /home/workshop/ex4pro1. This program may take as long as 30 seconds to run. It will shut down your machine.

3. Power on your system.

4. What happens during the reboot? Examine your Student Guide to find an explanation for the boot failure.

_____________________________________________________________

_____________________________________________________________
5. Boot your machine in maintenance mode.

6. From the maintenance menu access the rootvg before mounting the file systems. You need to do this, because mounting the file systems in rootvg will fail due to the corrupted log logical volume.

7. Reformat the journal log logical volume. Be sure to do a file system check for all file systems that use /dev/hd8. If you like, use set -o emacs or set -o vi.

8. Shut down your system and reboot your system in normal mode.
**Part 2 - Analyze and fix a boot failure**

9. What happens during the reboot of the system? Write down the last LED code that is shown. What type of problem is this indicative of?

_________________________________________________________________

_________________________________________________________________

_________________________________________________________________

10. Reboot the system to maintenance mode.

11. Examine your system and find the corrupted file that leads to the boot failure.

   Be sure to set the `TERM` variable to `lft`, if you are working on a graphical display. Otherwise `vi` or SMIT will not work correctly in the maintenance shell.

12. Repair the corrupted file. You will find an example in your student notebook. If you are not able to fix the boot failure, contact your instructor.

**End of exercise**
Exercise 5. LVM Tasks and Problems

What This Exercise Is About

This exercise has two parts. In the first part, you will be asked to complete some common LVM tasks. In the second part, you will analyze and fix LVM-related ODM problems. Two different lab sessions will be devoted to this exercise. So, you should stop after completing the first part of the exercise and not continue with the second part of the exercise.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Complete common LVM tasks
- Analyze an LVM-related ODM problem
- Fix an LVM-related ODM problem associated with the rootvg

Introduction

This exercise has two parts:

1. In the first part, you will complete some common LVM tasks. This may be a review for some members of the class.
   
   You should do this part during the first lab session allotted to this exercise.

2. In the second part, which has two sections, you will be asked to analyze and fix LVM-related ODM problems:
   
   a. Analyze and fix an LVM ODM failure manually.
   
   b. Analyze and fix an LVM ODM failure by using rvgrecover.

   You should do this part during the second lab session allotted to this exercise.

You will need root authority to complete this exercise.

Requirements

- /home/workshop/ex5_corrupt_pvid
- /home/workshop/ex5_corrupt_odm
- /home/workshop/rvgrecovery
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1: Basic LVM Tasks

__ 1. If you have a disk which is not part of the rootvg, extend the rootvg to include that disk. You may get an error if the additional disk appears to already belong to a volume group. This can happen if the disk was not properly removed from a previous volume group (reducevg); in that case, use the force option (-f).

__ 2. Using `smit mklv`, create a *mirrored logical volume* with the name `mirrorlv`. Make it two logical partitions in size.

Use `lslv -m` to identify the physical partitions that have been assigned to your logical partitions. Use the output produced to complete the table below:

<table>
<thead>
<tr>
<th>LP</th>
<th>PP1</th>
<th>PV1</th>
<th>PP2</th>
<th>PV2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Finally, remove the logical volume `mirrorlv`.

__ 3. Use `lspv -p` to determine a region (outer edge, outer middle, center, inner middle, inner edge) on `hdisk0` (or some other disk if so directed by your instructor) that has space available.

Record the region or regions with free partitions in the space provided below:

____________________________________________________________

__ 4. Use `smit mklv` to created an unmirrored logical volume `lvtmp1` on `hdisk0` with a size of one logical partition. Choose an intra-physical policy that will choose physical partitions in a region where free partitions exist.

Use `lspv -p` to check where the partitions of `lvtmp1` reside.

__ 5. Using `smit chlv` or the `chlv` command, change the intra-physical policy to another disk region. Have the partitions been moved to another region?

____________________________________________________________

If not, use the `reorgvg` command. Use the `man` pages to identify how to reorganize a logical volume.

*Note: Do not reorganize the complete rootvg, because this takes too much time!*
Write down the command you used:

Using `lspv -p` check where the partitions of `lvtmp1` reside now.

6. Find two free partitions on a disk. Write down the partition numbers:

Create a logical volume `lvtmp2` that uses an allocation map. The logical volume should have a size of two partitions and should use the two partitions you identified before. Here is an example for an allocation map:

`hdisk0:82-83`

After creating the logical volume, check where the partitions reside.

7. What would be the maximum number of disks allowed if we created a volume group using the following command:

`# mkvg -B -t 4 -y homevg hdisk11 hdisk99`

End of Part 1

*(If you are doing “Part 1” of this exercise, stop here. Do not go on to “Part 2.”)*
Part 2 - Fixing LVM-related ODM Problems

If you are doing “Part 2” of this exercise, start here.

Section 1: Analyze and Fix an LVM-related ODM Problem

__ 7. Execute `lspv` without any options to list all physical volumes in your system.
   Complete the following table.

<table>
<thead>
<tr>
<th>Disk name</th>
<th>PVID</th>
<th>Volume group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

__ 8. Execute `lsvg -p` to list all physical volumes that are part of your `rootvg`.
   Complete the following table:

<table>
<thead>
<tr>
<th>PV_NAME</th>
<th>PV STATE</th>
<th>TOTAL PPs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

__ 9. Execute `odmget -q` to see the disk information stored in ODM. Write down the command you used:

____________________________________________________________________

<table>
<thead>
<tr>
<th>PV_Name</th>
<th>PVID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Also, write down the structure of the stanza (that is, information labels) output by the above command. You will need this information in a later lab step.

____________________________________________________________________

____________________________________________________________________

____________________________________________________________________

____________________________________________________________________

____________________________________________________________________

____________________________________________________________________

____________________________________________________________________
__10. Execute the program /home/workshop/ex5_corrupt_pvid.

__11. Repeat the `lspv` command to list your physical volumes. Complete the table and compare with the table from step 1.

<table>
<thead>
<tr>
<th>Disk name</th>
<th>PVID</th>
<th>Volume group</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

__12. Repeat the `lsvg -p` command you used earlier to list the physical volumes in `rootvg`.

What is the output from the command?

_____________________________________________________________

_____________________________________________________________

_____________________________________________________________

_____________________________________________________________

__13. You learned that LVM stores information about volume groups, physical volumes and logical volumes in the ODM. Consider the output of `lspv` and `lsvg -p`. What data is missing? Where is the problem?

- Volume group objects?
- Physical volume objects?
- Logical volume objects?

Write down what you suspect:

_____________________________________________________________

__14. Depending on your suspicion, identify the ODM entries which are shown in your student notes in Unit 5.

Find out which objects in which ODM class are missing by reviewing the material from your student notes.

_____________________________________________________________

__15. Before you fix the problem, please consult one VGDA for each of the VGs on your system and compare the missing information with the data in the VGDA. Be sure that the information you wrote down in the tables above is correct, otherwise you will not be able to fix the problem.

What command allows you to query a VGDA?
16. Fix the ODM problem by adding the missing objects into the ODM. Please work very carefully in this step!

Recover the missing entries for all disks that show a problem in the `lspv` listing, not just the ones that are part of a particular volume group.

Use your student notes to find out the layout of the corresponding ODM class. Write down the steps you executed to fix the problem.

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

17. Repeat the commands `lspv` and `lsvg -p` to check whether your fix works.

If you still have problems, the stanza file you created contains a typo. Find the typo, delete the objects you just created, and add the fixed file. Did you remember to include the 16 trailing zeros on your `pvid` value?

Section 2: Analyze and Fix an LVM-related ODM Problem Using `rvgrecover`

18. Execute the program `/home/workshop/ex5_corrupt_odm`.

19. Verify the following information:
   a. Check whether your volume groups are ok. Use `lsvg`.
   b. Check whether your physical volumes are ok. Use `lspv`.
   c. Check whether your logical volumes are ok. List all logical volumes that are part of your `rootvg`. Use `lsvg -1 rootvg`.

What happens?
________________________________________________________________________
________________________________________________________________________
________________________________________________________________________
20. Display information for logical volume **hd2**. Use **lslv hd2**.

What happens?

____________________________________________________________________

____________________________________________________________________

____________________________________________________________________

21. Analyze the ODM problem by reviewing your student notebook. Compare the ODM entries for logical volumes from “Unit 5” with the ODM objects from your system.

What causes the ODM problems?

____________________________________________________________________

____________________________________________________________________

____________________________________________________________________

22. Examine the **/home/workshop/rvgrecovery** script and modify it if necessary to match your situation (the specified disk must be one in your **rootvg**).

After making any required changes to the script, fix the ODM problem by executing **/home/workshop/rvgrecovery**. Ignore the messages. This may take up to 1 minute, depending upon the speed of your lab system.

Check that your ODM problems have been fixed. Repeat **lsvg -1 rootvg** and **lslv hd2**. They should work now without problems.

23. Look into **/home/workshop/rvgrecovery**. What two main steps fix your ODM problem?

24. Remove the logical volumes **lvtmp1** and **lvtmp2** that were created in the first part of this exercise.

*End of Part 2*

*End of exercise*
Exercise 6. Mirroring rootvg

What This Exercise Is About

This exercise covers the process required to mirror the rootvg.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Mirror the rootvg
- Describe physical volume states
- Unmirror the rootvg

Introduction

In this exercise, you will mirror and unmirror rootvg. Completion of this exercise requires root authority.

Requirements

- /home/workshop/ex6_diskfailure
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Exercise: Mirror and Unmirror the Complete rootvg

__ 1. Write down on which disks your rootvg resides.

________________________________________________________________
________________________________________________________________

You might have a mixed installation, where the rootvg logical volumes are spread over two disks. Knowing which logical volumes reside on which disk is important because later (if you use mklvcopy), you will need to specify the target disk for the new mirror for each logical volume.

Which command displays the logical volumes that are contained on a disk?

________________________________________________________________

Execute this command for each disk in the rootvg on your system.

__ 2. Now mirror each logical volume as described in your student notebook. If you have a mixed rootvg installation, you must be careful when specifying the target disk name. Do not synchronize the logical volumes in this step.

Write down the commands you executed in this step:

________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________
________________________________________________________________

__ 3. Display information about your rootvg using the command lsvg rootvg.

Use the output of the lsvg rootvg command to record the following information:

- Stale physical volumes: __________________
- Stale physical partitions: ________________

4. Now, synchronize your rootvg. Depending on your system, this step may take 5 to 10 minutes to complete, depending on the speed of your system. This may be a good point to take a break.

Write down the command you executed:

```
```

5. Check by using lsvg rootvg that all partitions have been updated.

6. Enter the following command to see which disks on the system are bootable:

```
# ipl_varyon -i
```

Examine the BOOT DEVICE column in the output produced by this command. Does the value YES appear in this column for both of the physical volumes in rootvg?

7. Update your boot logical volumes and your bootlist (increase the size of the /tmp filesystem, if needed).

Write down the commands you executed:

```
```

8. Use the ipl_varyon command again to see which disks on the system are bootable now.

Examine the BOOT DEVICE column in the output produced by this command. Does the value YES now appear in this column for both of the physical volumes in rootvg?

9. Use the -m flag of the lslv command to confirm that there is a copy of the logical partition for the boot logical volume on the second disk in your rootvg.

10. Now, reboot your system. When the system comes back up, check to see what device the system booted from.

Write down the commands you executed.

```
```

11. The procedure /home/workshop/ex6_diskfailure simulates a disk failure. This procedure requires that your rootvg is mirrored completely.

Clear the error log and then execute the program:

```
/home/workshop/ex6_diskfailure
```

Create some files in the /tmp file system after running the program.

12. Analyze your AIX error log. Detailed information about working with the AIX error log is not provided until later in this course, so use SMIT to display the information in the error log. You may first want to look at the summary report.
13. The detailed report can get very long if you do not filter it, so you might only request hardware errors. In our case, we saw both hardware and software errors and the summary report was not very long; thus request the detailed report without any filtering.

Browse through your error report and identify the error log entries that have been created by the LVM. Note that the only information identifying the disks involved are the major and minor numbers, which are in hex and need converting to decimal before comparing to an `lsdev` listing. Here is some room for you to make notes about the error log entries:

_________________________________________________________
_________________________________________________________
_________________________________________________________
_________________________________________________________
_________________________________________________________
_________________________________________________________
_________________________________________________________
_________________________________________________________

14. Use the command `lspv hdiskx` to display information about each of the physical volumes in `rootvg`. Analyze the PV STATE field to check the state of each disk. Also check the STALE PARTITIONS entry for each of the disks.

Which disk is causing problems? Which PV STATE has been allocated to the failing disk?

_________________________________________________________
_________________________________________________________
_________________________________________________________
_________________________________________________________

15. Review the page in your Student Notebook that describes “Physical Volume States.” Find the physical volume state of the failing disk on the visual.

16. Execute a `varyonvg rootvg` and check whether this fixes the disk problem.

What happens? Check the PV STATE of the failing disk.

_________________________________________________________
_________________________________________________________
17. By reviewing your *Student Notebook*, determine which command will bring the disk back into the active state.

_________________________________________________________

Execute this command and check the PV STATE of the failing disk.

_________________________________________________________

18. Check whether your `rootvg` still contains stale partitions.

_________________________________________________________

19. Check your *Student Notebook* again. What command is the best to fix the stale partitions?

_________________________________________________________

Execute the command and check if the stale partitions are fixed.

20. Unmirror the `rootvg` of your system.

*Important*: Unmirror your `rootvg` in a way so that one disk is completely empty. We need an empty disk in our next exercise.

Decide which of your disks you want to remove all mirrors.

Write down the command you executed to unmirror your `rootvg`:

_________________________________________________________

What recommendation regarding the `chpv` command do you get when you execute the `unmirrorvg` command?

_________________________________________________________

Follow this recommendation and clear the boot record.

_________________________________________________________

21. Check that all logical volumes have been removed from the disk and remove the “empty disk” from `rootvg`.

_________________________________________________________

_________________________________________________________

22. Finally update your boot logical volume and your bootlist.

Then, reboot your system. When the system comes back up, check to see what device the system booted from.

Write down the commands you executed:

_________________________________________________________

_________________________________________________________
End of exercise
Exercise 7. Exporting and Importing Volume Groups

What This Exercise Is About

This exercise describes the steps to export and import volume groups.

What You Should Be Able to Do

At the end of the lab, you should be able to:

• Export a volume group
• Import a volume group

Introduction

As you have learned in this course, export and import can be used to move data from one system to another. Sometimes it is the only way to correct ODM failures with non-rootvg volume groups.

This exercise has two parts:

• Export and import a volume group
• Analyze import messages (Optional)

This exercise requires one disk to be completely empty. This disk will be used to create a new volume group. This volume group will be exported and imported.

All instructions in this exercise require root authority.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Export and import a volume group

__ 1. Create a new volume group named datavg on a disk that is empty. Check that this disk does not belong to another volume group. Set the physical partition size to 16 MB.

Write down the command you executed to create the new volume group:

______________________________________________________________________________

__ 2. Check if the new volume group has been varied on automatically. Write down the command you used.

______________________________________________________________________________

__ 3. Use the fastpath smit mklv to create a logical volume in datavg with the following characteristics:
   • Logical volume name: lv_raw
   • Number of logical partitions: 1

__ 4. Use the fastpath smit jfs to create two standard journaled file systems in datavg with the following characteristics:
   • Size of file systems: 16 MB (65536 512-byte blocks)
   • Mount points:
     - File system 1: /home/jupiter
     - File system 2: /home/mars

Note: Do NOT build these file systems on the lv_raw logical volume created in the previous step.
5. Verify the new logical volumes are in `datavg` with the `lsvg` command. Fill in the following table with the logical volume information in `datavg`:

<table>
<thead>
<tr>
<th>LV NAME</th>
<th>TYPE</th>
<th>MOUNT POINT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. Mount the file systems and create some files in both file systems.

7. Export the `datavg` volume group from your system. Write down all the steps you executed to export the volume group.

8. Analyze your system to see if it contains any reference to the exported volume group. For example, check whether the file systems you have created exist. (Check `/etc/filesystems`.)

9. Import the volume group into your system. Specify volume group name `datavg`, otherwise the system will generate a new volume group name. Write down the command you executed:

10. Check whether the imported volume group, `datavg`, is varied on.

Check to see if the file system information is back.
11. Mount the `/home/jupiter` and `/home/mars` file systems. Check that no files have been lost.
**Part 2: Analyze import messages (Optional)**

In *Part 1: Export and import a volume group*, the export and import worked without problems, as the logical volumes and file systems did not exist during the import of the volume group.

This part will show what will happen when a volume group that is being imported has the same logical volume names of those that already exist on the system.

___ 12. Export the *datavg* volume group again. Repeat the steps from the last export.

___ 13. Use the fastpath `smit mklv` to create a logical volume in *rootvg* with the following characteristics:
   - Logical volume name: *lv_raw*
   - Number of logical partitions: 1

___ 14. Use the fastpath `smit jfs` to create two standard journaled file systems in *rootvg* with the following characteristics (these are the same as in Part 1).
   - Size of file systems: 16 MB (65536 512-byte blocks)
   - Mount points:
     - File system 1: */home/jupiter*
     - File system 2: */home/mars*

___ 15. What are the corresponding logical volume names that have been created for the file system?
   Logical volume for */home/jupiter*: _______________
   Logical volume for */home/mars*: _______________

___ 16. Mount the */home/jupiter* and */home/mars* file systems, and add a few files to each.

___ 17. At this stage, the following problems will come up when you import the *datavg* volume group:
   - The *lv_raw, lv00* and *lv01* logical volumes already exist in *rootvg*
   - The */home/jupiter* file system already exists in *rootvg*
   - The */home/mars* file system already exists in *rootvg*
Let's see how `importvg` will react to this situation.
Import the `datavg` volume group into the system.

__18. Write down the new logical volume names that are created for `datavg` during the import.__

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

__19. Another problem that you should see at this stage, is that the `/home/jupiter` and `/home/mars` file systems already exist in `rootvg`.

To fix this problem, first unmount the `/home/jupiter` and `/home/mars` file systems from `rootvg`.

__20. Mount the file systems from `datavg` over the corresponding mount points. Use the new logical volume names that have been created. You have to specify the log device that is part of `datavg`.

Write down the commands you executed.

________________________________________________________________________
________________________________________________________________________

__21. Check the files you have created in `/home/jupiter` and `/home/mars`. They should exist in these directories.

__22. At the end of this exercise, all four file systems should be mounted at the same time.

Start with unmounting `/home/jupiter` and `/home/mars`.

__23. Create two new directories, `/datavg/jupiter` and `/datavg/mars`. These will be the new mount points for the file systems from `datavg`. 
24. Create two new stanzas in /etc/filesystems that describe the file systems from datavg. You must use the new logical volume names that have been created during the import of datavg.

25. Mount the /datavg/jupiter and /datavg/mars file systems.

26. Verify you can access all the files.

27. Unmount the /datavg/jupiter and /datavg/mars file systems.

28. Varyoff the datavg volume group.

29. Export the datavg volume group.

30. Remove the /home/jupiter and /home/mars file systems from the rootvg volume group.

End of exercise
Exercise 8. Saving and Restoring a User Volume Group

What This Exercise Is About

This exercise provides an opportunity to back up a non-rootvg volume group and then restore the data to simulate a failure of a complete volume group.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Use the savevg command
- Change volume group characteristics
- Use the restvg command

Introduction

All instructions in this exercise require root authority. There must be enough free space on the other disk to back up the user volume group. This disk will probably be the rootvg disk. If there are two students using the system, they must work on this exercise together.

NOTE: It is imperative that Exercise 7: Exporting and Importing Volume Groups was completed. This exercise assumes a user volume group (datavg) was created.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

1. Check that your user volume group, datavg, is defined and varied on. Also, check that the /home/jupiter and /home/mars file systems are mounted. If not, import datavg and mount the file systems. (If you did the optional exercise in Exercise 7: Exporting and Importing Volume Groups, Part 2 - Analyze import messages (Optional) datavg should have been exported in the last exercise.)

2. What is the physical partition size of datavg? _________

3. How many partitions are allocated for /home/jupiter? ______

4. Execute the mkvgdata command to create a control file for the savevg command. Write down the command you used.

5. Before saving the volume group datavg, change the control file that is used during the restore process. Edit the file and change the number of logical partitions that are allocated for /home/jupiter to 4.

6. Back up your user volume group, datavg, to a file image. Do not use SMIT to save the volume group. Write down the command you used.

7. Unmount all file systems from datavg. Write down the commands you used.

8. Varyoff the volume group datavg. Write down the command you used.

9. Export the volume group from the system. Write down the command you used.

10. Execute the restvg command and restore the volume group from your backup image. Write down the command you used.
11. Write down the number of partitions that are allocated for /home/jupiter:

12. Using SMIT, save the volume group datavg again. Specify the same backup file image as before. Write down the command that SMIT executes.

13. Execute the same steps as before (umount, varyoffvg, exportvg) to remove the complete volume group datavg from the system.

14. Using SMIT, restore the volume group, datavg, from the file image. In SMIT, specify a bigger physical partition size, for example 64 MB. Write down the command that SMIT executes.

15. After restoring the volume group, check the physical partition size of datavg.

16. How many partitions are allocated for /home/jupiter?

   Do you still have four partitions for /datavg/jupiter?

End of exercise
Exercise 9. Error Log and syslogd

What This Exercise Is About

This exercise has two parts. In the first part, you will work with the AIX error logging facility. In the second part, you will work with the syslogd daemon and the ODM error notification class errnotify.

Two different lab sessions will be devoted to this exercise. So, you should stop after completing the first part of the exercise and not continue with the second part of the exercise.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Determine what errors are logged on your machine
- Generate different error reports
- Start concurrent error notification
- Identify errors and warnings sent by the syslogd daemon
- Create and maintain the /etc/syslog.conf file
- Automate error logging with errnotify
- Redirect syslogd messages to the error log

Introduction

In “Part 1” of this exercise, you will work with the AIX error logging facility. You should do this part of the exercise during the first lab session allotted to this exercise.

In “Part 2” of this exercise, you will work with the syslogd daemon and the ODM error notification class errnotify. You should do this part of the exercise during the second lab session allotted to this exercise.

You will need root authority to complete this exercise.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Working with the error log

__ 1. Generate a summary report of your system’s error log. Write down the command that you (or SMIT) used:

____________________________________________________________

__ 2. Generate a detailed report of your system’s error log. Write down the command that you (or SMIT) used:

____________________________________________________________

__ 3. Using SMIT, generate the following reports:

• A summary report of all errors that occurred during the past 24 hours. Write down the command that SMIT executes:

____________________________________________________________

• A detailed report of all hardware errors. Write down the command that SMIT executes:

____________________________________________________________

__ 4. If you are using local lab machines, this instruction requires that either: (a) a graphical desktop, such as CDE, is active or (b) that you have a windowing workstation where you can have a telnets from multiple windows. In this environment, start two windows.

If you are using a remote server, open an additional network connection (using a tool such as telnet) and log in as root, in order to have two windows to work with. In one window startup concurrent error logging, using the errpt command. Write down the command that you used:

____________________________________________________________

In the other window, execute the errlogger command to generate an error entry. Write down the command you used:

____________________________________________________________

Is the complete error text shown in the error report?

____________________________________________________________
Stop concurrent error logging.

5. Write down the characteristics of your error log:

   LOGFILE: _______________________________________________________

   Maximum LOGSIZE: _____________________________________________

   Memory BUFFER SIZE: _________________________________________

   What command have you used to show these characteristics?
   _______________________________________________________

6. List the entries that have an error class of operator.

7. Clean up all error entries that have an error class of operator. Write down the command, you (or SMIT) used:

   _______________________________________________________

8. Verify that the operator entries are now gone.

End of Part 1

(If you are doing “Part 1” of this exercise, stop here. Do not go on to “Part 2.”)
Part 2

If you are doing "Part 2" of this exercise, start here.

Section 1: Working with syslogd

7. Edit the /etc/syslog.conf file and configure the syslogd daemon to log all daemon messages to a file with the name /tmp/syslog.debug.
   Write down the line that you added to /etc/syslog.conf:

8. Execute the touch command and create the file /tmp/syslog.debug.

9. Refresh the syslogd daemon so it will pick up the changes. Write down the command that you used:

10. Stop the inetd daemon and restart it in debug mode. Use the appropriate System Resource Controller command to start the inetd daemon in debug mode (-d flag). Write down the commands that you used:

11. Use the telnet command to telnet back to your own system, log in, and then log back out of the telnet session. This step is performed to log several debug messages. Use your login name when you telnet to your system.

12. Stop the inetd daemon and restart it without debug mode. Use the appropriate System Resource Controller command to start the inetd daemon. Write down the commands that you used:

13. Analyze the content of the file /tmp/syslog.debug. Many debug messages from the inetd daemon processes are shown.

14. Change your /etc/syslog.conf. All messages should be directed to the AIX error log. Write down what you have changed:

15. Refresh the syslogd subsystem. Write down the command that you used:
16. Generate a **syslogd** message, for example, use an invalid password during a login. Check that the message is posted to the error log.

---

**Section 2: Error Notification with errnotify**

This part of the exercise demonstrates how to automate working with the error log.

17. Create an **errnotify** object that mails a message to **root**, whenever an **operator message** is posted to the **errlog**. Write down the stanza that you added:

---

18. Execute the **errlogger** command and create an entry in the **errlog**. Write down the command that you used:

---

19. After a short time, check the mail for the **root** user. The mail processing is batched and it could take more than a minute before the mail is delivered; using **sendmail -q** may help to expedite this.

---

End of Part 2

End of exercise
Exercise 10. Diagnostics

What This Exercise Is About

This exercise describes how to use diagnostic routines in several different modes.

What You Should Be Able to Do

At the end of the lab, you should be able to:

• Execute hardware diagnostics in the following modes:
  - Concurrent
  - Maintenance
  - Service (standalone)

Introduction

Only one person per machine can execute these commands.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Specifically, it requires either a local machine were access does not depend upon network access or a remote LPAR which is accessible using a virtual terminal (HMC) with a physical Ethernet adapter.

1. Determine if your system has a physical Ethernet adapter and whether it is configured.

2. If your physical Ethernet adapter is not configured, then configure it with a private address which will not conflict with any existing lab subnets. For example, you might assign it 192.168.252.<your team number>. Check with the instructor if you are unsure.

3. Start up diagnostic routines in concurrent mode and test a communication adapter that is in use on your system. What happens?

4. Return to the FUNCTION SELECTION menu. Then, select Diagnostic Routines. What is the difference between System Verification and Problem Determination?

5. Return to the FUNCTION SELECTION screen. Using Task Selection, query the vital product data of one of your physical Ethernet adapters.

6. Return to the TASKS SELECTION LIST screen.

   f. Who will be notified when a hardware error is posted to the error log?

   g. If root was not in the notification list, return to the AUTOMATIC ERROR LOG ANALYSIS AND NOTIFICATION SERVICE AID screen and add root to the notification list.

7. Start up diagnostic routines in single user mode using the following steps:
a. You will need to be at the system console to do this. If you are using a remote LPAR, then first open a virtual terminal to your system from the HMC.

b. Shutdown your system to single user mode. (Note: In this particular case, it would have been sufficient to detach the interface related to the network adapter, rather than having to shut down to single user mode. But, there will be other situations where one may need to run diagnostics from single user mode, maintenance mode, or even booting with a diagnostic routine provided on CD or over the network.)

c. At your system console, login to single user mode using root’s password.

d. Start the diagnostics facility.

8. Test the communication adapter again in maintenance mode. What happens now?

9. Exit the diagnostic utility.

10. Start up the diagnostic utility in service mode from the hard drive using the following steps:
   a. Shut down AIX and power off your machine or logical partition:
   
   b. Boot your system to diagnostics using service mode off the hard drive.

11. Test the communication adapter again in maintenance mode. What happens now?

12. Exit the diagnostic utility.

13. Boot your system in normal (multi-user) mode.

14. When AIX finishes booting, log in as root.
    View the contents of the diagnostics log using both the summary format and the detailed format. Did you find any errors?
End of exercise
Exercise 11. System Dump

What This Exercise Is About

This exercise allows you to become familiar with the AIX dump facility. In addition, you'll use the `snap` command to collect system data that is needed to analyze the system dump. During this exercise, you will also use the `kdb` command, but only at a very introductory level.

What You Should Be Able to Do

After completing this exercise, you should be able to:

- Initiate a dump
- Use the `snap` command

Introduction

In this exercise you will create a dump and use the `kdb` command to look at that dump.

You will need `root` authority to complete this exercise.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Note: All users must perform this exercise together if there is more than one user on your system.

Working with the AIX Dump Facility

1. Record the following dump-related settings for your system:
   - primary dump device _____________________________
   - secondary dump device ___________________________
   - copy directory ___________________________________
   - dump compression (ON or OFF) _____________________

2. Execute the command to display the estimated size of a dump and record the estimate you obtain:

3. Verify that the dump copy directory is large enough to hold the dump size reported on the previous command.
   *If there is not enough space, you must increase the size of the corresponding file system.* (If necessary, use the `chfs` command to increase the size of the appropriate file system, typically `/var`.) After increasing the size, reverify that the filesystem is large enough.

4. Set the value of the `autorestart` attribute for `sys0` to `true`. (If `autorestart` is set to `true`, the system will reboot after a crash.)

5. Use the command `sysdumpstart -p` to start a dump to the primary dump device.
   - What LED code appears for several minutes after this command is entered? This is referred to as an Operator Panel Value (pre-HMCv7) or as the Reference Code (HMCv7) in the HMC display across from your LPAR name.

6. After the system reboots, determine and write down the size, uncompressed size, and filename for your system dump:

7. Uncompress the dump file (for example, `/var/adm/ras/vmcore.0.BZ`). When doing the dump-uncompress, keep the original compressed file. Note, based on the
reported *Uncompressed Size* just reported, that you may need to further increase the size of */var* to accommodate the size of the uncompressed dump (in addition to the already created compressed dump).

Then, execute the **kdb** command on the uncompressed dump that was created. Write down the commands you used:

_______________________________________________
_______________________________________________

__8._ Use the **kdb stat** and **status** subcommands to show the system name and time of the dump, and the processes/threads running when the dump occurred. Leave the **kdb** command afterwards.

__9._ Remove the uncompressed dump, but keep the original compressed dump. (This will ensure proper processing of the system dump by the **snap** command, which you will use in a subsequent lab step.)

__10._ Check to see how much free space is currently available in */tmp*.

If necessary, increase your */tmp* file system so that there is at least 32 MB of free space. We need this space in the next lab step.

Write down the commands you used:

_______________________________________________
_______________________________________________

__11._ Run the command **snap -a**. (Note that this command will take approximately 10 minutes to run.)

Review the output of this command. This output will include a list of various directories (in */tmp/ibmsupt*) to which the **snap** command writes its output.

In these directories, you will find files with names that end in *.snap*, which are ASCII files. Review the content of a few of these files.

*End of exercise*
Exercise 12. Basic Performance Commands

What This Exercise Is About

The purpose of this exercise is to provide basic performance commands.

What You Should Be Able to Do

At the end of the lab, you should be able to:

• Use `ps` to identify CPU and memory-intensive programs
• Execute a basic performance analysis
• Implement a Korn shell job queue
• Work with `nice` and `renice` to change the priorities of processes

Introduction

All instructions in this exercise should be executed with `root` authority.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Part 1 - Working with ps, nice, and renice

1. Implement an alias `top` that shows a sorted output from `ps aux` according to the CPU usage. Write down the alias definition.

2. Execute `top` and identify the process that consumes the most CPU.

3. Start the program, `/home/workshop/ex12_prog1`, in background. Use the `ps` command to identify the assigned priority and nice value.

   Priority: 
   Nice value: 

4. Stop `ex12_prog1` and restart it in background with a very low priority. Write down the command that you used.

   Priority: 
   Nice Value: 

5. Without restarting `ex12_prog1`, increase the priority of the process. Write down the command you used.

   Check that the priority has been increased.

   Priority: 
   Nice Value: 

6. Stop the program `ex12_prog1`.


Part 2 - Basic Performance Analysis

__ 7. Start the program /home/workshop/ex12_cpu in background. Execute the sar command to analyze CPU usage on your system. Set it up to collect the data at two second intervals for five times. Write down the command that you used to monitor CPU usage.

From the output, what can you conclude?

__ 8. Use the ps command to check that the priority that has been assigned to the process. Is the priority high or low?

__ 9. Stop the program ex12_cpu.

__ 10. Create two JFS file systems on one of your hard disks then mount them. Use the following commands (these commands assume rootvg is on hdisk0, your disk name may be different):

```
# mklv -y lv1 -t jfs -u 1 rootvg 1 hdisk0
# crfs -v jfs -d lv1 -m /fs1
# mklv -y lv2 -t jfs -u 1 rootvg 1 hdisk0
# crfs -v jfs -d lv2 -m /fs2
# mount /fs1; mount /fs2
```

__ 11. Start the program /home/workshop/ex12_io -c -w -t 120 in the background. (The value used for the -t option specifies how long to run this program in seconds.) Execute the iostat command to analyze your disk I/O while ex12_io is running. Look at iostat disk information for two second intervals five times. Write down the command that you used to monitor disk I/O.

From the output, what can you conclude?

__ 12. The ex12_io program should stop after 2 minutes (120 seconds). If it has not ended, stop the program.

__ 13. Start the memory intensive process /home/workshop/ex12_memory in the background. Execute the vmstat command to analyze your memory utilization. Run vmstat at five second intervals. Write down the command that you used to measure memory.

From the output, what can you conclude?
Part 3 - Working with a Korn Shell Job Queue

14. Create a Korn shell job queue as shown in your student notes. Write down the definitions for the queue and the queue device:

15. Bring down the queue. Write the command you used.

16. Put the job /home/workshop/ex12_job into the ksh queue. Write down the command you used.

17. Verify that the job is queued. Write down the command you used.

18. Bring up the queue. Write down the command you used.

What happens?

End of exercise
Exercise 13. Performance Diagnostic Tool

What This Exercise Is About

The purpose of this exercise is to give students an opportunity to use the Performance Diagnostic Tool.

What You Should Be Able to Do

After completing this exercise, students should be able to use the Performance Diagnostic Tool (PDT) for ongoing data capture and analysis of critical system resources.

Introduction

This exercise deals with the Performance Diagnostic Tool (PDT) for on-going data capture and analysis of system resources.
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

1. Verify that PDT is loaded on your exercise system. Start PDT to enable default data collection and reporting.

2. The _adm_ user is needed to run this procedure. _su_ to the _adm_ user and change the _crontab_ entry so PDT will collect data within 10 minutes from now and run the reports five minutes later. Make sure you check the system date first so you will know what hour and minutes to put in the _crontab_ entries.

   After you modify the _crontab_ entry, continue onto the next step. The next step needs to run while PDT is collecting information.

3. Run the script _ex13_perf_ located in _/home/workshop_. You need to run this as _root_. This script will create some items that should be reported when PDT runs in 10 minutes.

4. After the time frame is over in which the report should have been created (based on your entries in the _crontab_ file), view the report.

5. To change the severity level to severity level 2 and the user to whom the report is mailed, execute the _pdt_config_ program. Once you are finished making the changes, exit the program.

6. Run PDT again, this time from the command line. Do it twice. Once to see a severity level 2 report. The other time to do a severity level 3 report.

End of exercise
Exercise 14. Authentication and ACLs

What This Exercise Is About

This exercise will familiarize you with three security features: the login.cfg file, authentication, and access control lists (ACLs).

What You Should Be Able to Do

After completing this exercise, students should be able to:

- Customize the login.cfg file
- Add an additional primary authentication method for a user
- Implement access control lists (ACLs)

Introduction

This exercise consists of three parts:
1. Customizing the login.cfg file
2. Adding a primary authentication method
3. Access control lists (ACLs)

Requirements

- Program /home/workshop/ex14_login
Exercise Instructions

Preface

Two versions of these instructions are available; one with hints and one without. You can use either version to complete this exercise. Also, please do not hesitate to ask the instructor if you have questions.

All exercises of this chapter depend on the availability of specific equipment in your classroom.

Setting a new login herald

1. Log in as root and edit /etc/security/login.cfg. Change the herald message to read:

   *Restricted Access*
   Authorized Users Only
   Login:

2. Use the login command to log in over yourself. You must be at the command line login to see your changes. If you are using the CDE graphical login, click the options button and select Command line login.

   Does it look correct? If not, try step one again.

3. Log in as root.

4. Review the failed login attempts made on your machine.

5. Review the su activity on your machine.

6. Review all the logins on your system.

7. Review all root logins on your system.

Adding a primary authentication method

In /home/workshop you find a procedure with the name ex14_login, which implements an additional primary authentication method. This method restricts a user to one login session on a system.

8. With root authority, change to /home/workshop and analyze the procedure ex14_login. Which statement indicates a valid or invalid login?

   __________________________________________
   __________________________________________

   Check that ex14_login is executable.

9. Install the procedure ex14_login as an additional authentication method on your system. You will need to add a stanza in /etc/security/login.cfg. Write down the stanza you add and the name of the file in which you place the stanza:
10. Check to be sure that team01 is a defined user, with a password set and no ADMCHG flag to force password reset. The password should be the same as the user name.

11. Install the additional authentication method for user team01 in /etc/security/user. Write down the stanza definition for team01:

12. Working in a graphical environment, open two windows and execute the login command in both of them. Login as team01. The second login should fail.

13. Remove the additional authentication method from team01.

Access control lists

14. Use the login command to log in over yourself as team01 and switch user to root. Create two new users named michael and sarah. Assign each new account a password that is the same as the login name.

15. Return to your team01 user ID.

16. Create a shell script named sample in your home directory with the following content:

```bash
tput clear
banner We love AIX
print End of Program
```

Set the base permissions for sample to 700 and verify that the script works.

17. Use the login command to log-in over yourself as michael. Change directory to /home/team01.
18. Try to display the sample script, using the cat command. Try to execute sample. You should not be able to do either.

19. Use the login command to log in over yourself team01. Set and export the EDITOR variable to /usr/bin/vi in your .profile.
   Use the login command to log in over yourself as team01. The alternative is to execute the .profile you just created (without launching a subshell).

20. Use the acledit command to change the extended permissions of the sample script so that michael has rwx access to the script, and sarah has only r-x access to the script. (Use the AIXC ACL type for this exercise.) Apply the modified ACL.

21. Execute ls -e and check that extended permissions are set for sample.

22. Use the login command to log in over yourself as michael. Change to the /home/team01 directory and test the extended permissions by trying to add the date command to the end of the script. Execute the sample script afterwards.

23. Use the login command to log in over yourself as sarah. Change to /home/team01 and try to change the sample script by removing the date command. Does it work?

24. Use the login command to log in over yourself as team01. Change the base and extended permissions to the sample script so that members of the staff group can read and execute the script, except for michael.
25. Use the `login` command to log in over yourself as `michael`. Issue the `groups` command to ensure you are part of the `staff` group. Change directory to `/home/team01`. Can you execute the `sample` script?

26. Use the `login` command to log in over yourself as `team01`. Create a new file named `sample2`. Type a couple of lines in the file. Use `aclget` to see that no extended permissions are set on `sample2`.

27. Using `aclget` and `aclput`, copy the access control information of the file `sample` to the new file `sample2`. Verify that the ACLs were copied over to `sample2`.

28. Execute a `chmod 700` on `sample`. Execute `acledit` on `sample`. What is different?

End of exercise
Appendix A. Auditing

What This Exercise Is About

This exercise is an introduction to the use of the AIX auditing subsystem to trace and record security-relevant information.

What You Should Be Able to Do

At the end of the lab, you should be able to:

- Audit objects and application events
- Create audit classes
- Audit users
- Set up auditing in bin and stream mode

Introduction

Completion of this exercise requires root authority.

Requirements

- /home/workshop/exappa_job
Exercise Instructions

*Bin mode auditing*

__1. Answer the following question first: Where do you specify file system objects that should be audited?__

____________________________________________________________________________________

__2. Set up auditing of the program `/usr/bin/passwd`. When you are finished, whenever a user calls `passwd`, you should get an audit record. Add this object to the corresponding audit configuration file.__

____________________________________________________________________________________

____________________________________________________________________________________

Write down the event name you have created:

____________________________________________________________________________________

__3. In which file do you have to specify the format definitions for your new event?__

____________________________________________________________________________________

__4. Add the format definition for your new event to the corresponding audit configuration file.__

____________________________________________________________________________________

____________________________________________________________________________________

__5. In which file do you specify the start mode for the auditing subsystem?__

____________________________________________________________________________________

__6. Create a directory `/var/myaudit`. We want to use this directory to collect all audit-related files.__

__7. Change the corresponding configuration file to start up the auditing subsystem in *bin mode*. Specify the following bin files:

```
bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2
trail = /var/myaudit/trail
```

____________________________________________________________________________________

____________________________________________________________________________________

__8. Start the auditing subsystem. Write down the command you used.__

____________________________________________________________________________________
9. Log in as **team01**. Use the password **team01**. When prompted to change password, set it back to **team01**. If you use a graphical environment, execute the `login` command in a separate window.

10. Execute the `passwd` command and change the password for **team01**.

11. With **root** authority, stop the auditing subsystem. Write down the command you used.

12. Change to `/var/myaudit` and display the audit records that have been recorded. Write down the command you used.

---

**Stream mode auditing**

13. In which file do you configure **audit classes** and **audit users**?

14. Change this configuration file in the following way:
   - The auditing subsystem starts up in **stream mode**.
   - Create an audit class **kill**, that contains an audit event whenever a process gets killed.
   - Remove the **root** user in the **users** stanza.
   - The user **team01** should be audited for the audit classes **kill** and **tcpip**.

15. In which file do you configure the **auditstream** daemon?

16. Before starting the auditing subsystem in stream mode, change the configuration file for the **auditstream** daemon. All audit records shall be written to file `/var/myaudit/stream.out`. Be sure to terminate the command with a & sign.

17. Start your auditing system.
18. Use the `touch` command and create an empty file `/var/myaudit/stream.out`. Use the `tail` command in a separate window to display the audit records real-time.

19. Log in as `team01` and trigger the events that you are auditing for this user:

- Execute the `ftp` command. Use your local host as destination host. You should see the corresponding audit records in file `/var/myaudit/stream.out`.

- Start the program `/home/workshop/exappa_job` in `background`. Kill the started program afterwards. You should see audit records for the `kill` audit class you have created.

20. Stop the auditing subsystem.

End of exercise
Exercise Instructions With Hints

**Bin mode auditing**

1.  Answer the following question first: Where do you specify file system objects that should be audited?

   ________________________________

   **Hint:** /etc/security/audit/o...

2.  Set up auditing of the program /usr/bin/passwd. When you are finished, whenever a user calls `passwd`, you should get an audit record. Add this object to the corresponding audit configuration file.

   ________________________________

   ________________________________

   ________________________________

   **Hint:** Add an `x-event` for the program `/usr/bin/passwd`

   Write down the event name you have created:

   ________________________________

3.  In which file do you have to specify the format definitions for your new event?

   ________________________________

   **Hint:** /etc/security/audit/e...

4.  Add the format definition for your new event to the corresponding audit configuration file.

   ________________________________

   ________________________________

   **Hint:** Specify the event name and add a `printf` definition.

5.  In which file do you specify the `start mode` for the auditing subsystem?

   ________________________________

   **Hint:** /etc/security/audit/c...

6.  Create a directory `/var/myaudit`. We want to use this directory to collect all audit-related files.

7.  Change the corresponding configuration file to startup the auditing subsystem in `bin mode`. Specify the following bin files:

    ```
    bin1 = /var/myaudit/bin1
    bin2 = /var/myaudit/bin2
    trail = /var/myaudit/trail
    ```
__ 8. Start the auditing subsystem. Write down the command you used.

_HINT: Execute audit...

__ 9. Log in as team01. Use password team01. When prompted to change password, set it back to team01. If you use a graphical environment, execute the login command in a separate window.

__ 10. Execute the passwd command and change the password for team01.

__ 11. With root authority, stop the auditing subsystem. Write down the command you used.

_HINT: Execute audit...

__ 12. Change to /var/myaudit and display the audit records that have been recorded. Write down the command you used.

__ 13. In which file do you configure audit classes and audit users?

_HINT: /etc/security/audit/c...

__ 14. Change this configuration file in the following way:

   • The auditing subsystem starts up in stream mode.
   • Create an audit class kill, that contains an audit event whenever a process gets killed.
   • Remove the root user in the users stanza.
   • The user team01 should be audited for the audit classes kill and tcpip.

_HINT: You must change the start, classes and users stanzas.
15. In which file do you configure the `auditstream` daemon?

Hint: `/etc/security/audit/stream`...

16. Before starting the auditing subsystem in stream mode, change the configuration file for the `auditstream` daemon. All audit records shall be written to file `var/myaudit/stream.out`. Be sure to terminate the command with a `&` sign.

Hint: Change the `auditpr` command.

17. Start your auditing system.

18. Use the `touch` command and create an empty file `var/myaudit/stream.out`. Use the `tail` command in a separate window to display the audit records real-time.

Hint: `# tail -f /var/myaudit/stream.out`

19. Log in as `team01` and trigger the events that you are auditing for this user:

- Execute the `ftp` command. Use your local host as destination host. You should see the corresponding audit records in file `var/myaudit/stream.out`.
- Start the program `/home/workshop/exappa_job` in `background`. Kill the started program afterwards. You should see audit records for the `kill` audit class you have created.

Hint: Execute `ftp localhost`. Use the `kill` command to kill the command that runs in background.

20. Stop the auditing subsystem.

End of exercise
Exercise Instructions With Solutions

Bin mode auditing

1. Answer the following question first: Where do you specify file system objects that should be audited?
   
   /etc/security/audit/objects

2. Set up auditing of the program /usr/bin/passwd. When you are finished, whenever a user calls passwd you should get an audit record. Add this object to the corresponding audit configuration file.
   
   # vi /etc/security/audit/objects

   /usr/bin/passwd:
   x = "X_EVENT"

   Write down the event name you have created:
   
   X_EVENT

3. In which file do you have to specify the format definitions for your new event?
   
   /etc/security/audit/events

4. Add the format definition for your new event to the corresponding audit configuration file.
   
   # vi /etc/security/audit/events

   X_EVENT = printf "%s"

5. In which file do you specify the start mode for the auditing subsystem?
   
   /etc/security/audit/config

6. Create a directory /var/myaudit. We want to use this directory to collect all audit-related files.
   
   # mkdir /var/myaudit

7. Change the corresponding configuration file to start up the auditing subsystem in bin mode. Specify the following bin files:
   
   bin1 = /var/myaudit/bin1
   bin2 = /var/myaudit/bin2
   trail = /var/myaudit/trail
   
   # vi /etc/security/audit/config

   bin:
   trail = /var/myaudit/trail
   bin1 = /var/myaudit/bin1
bin2 = /var/myaudit/bin2
binsize = 10240
cmds = /etc/security/audit/bincmds

__ 8. Start the auditing subsystem. Write down the command you used.

    # audit start

__ 9. Log in as team01. Use password team01. When prompted to change password, set
it back to team01. If you use a graphical environment, execute the login command
in a separate window.

    # login

__ 10. Execute the passwd command and change the password for team01.

    $ passwd

__ 11. With root authority, stop the auditing subsystem. Write down the command you
used.

    # su
    # audit shutdown

__ 12. Change to /var/myaudit and display the audit records that have been recorded.
Write down the command you used.

    # cd /var/myaudit
    # auditpr -v < trail

    You should see the audit event X_EVENT that has been created, triggered by the
execution of passwd.

Stream mode auditing

__ 13. In which file do you configure audit classes and audit users?

    /etc/security/audit/config

__ 14. Change this configuration file in the following way:
• The auditing subsystem starts up in stream mode.
• Create an audit class kill, that contains an audit event whenever a process gets killed.
• Remove the root user in the users stanza.
• The user team01 should be audited for the audit classes kill and tcpip.

    # vi /etc/security/audit/config

start:
    binmode = off
    streammode = on

classes:
kill = PROC_Kill

users:
  team01 = kill, tcpip

15. In which file do you configure the **auditstream** daemon?

/etc/security/audit/streamcmds

16. Before starting the auditing subsystem in stream mode, change the configuration file for the **auditstream** daemon. All audit records shall be written to file /var/myaudit/stream.out. Be sure to terminate the command with a & sign.

```
# vi /etc/security/audit/streamcmds

/usr/sbin/auditstream | auditpr -v > /var/myaudit/stream.out &
```

17. Start your auditing system.

```
# audit start
```

18. Use the **touch** command and create an empty file /var/myaudit/stream.out. Use the **tail** command in a separate window to display the audit records real-time. (Depending on your environment, you may want to start AIXwindows for the next step (# xinit).)

```
# touch /var/myaudit/stream.out

# tail -f /var/myaudit/stream.out
```

19. In a separate window log in as **team01** and trigger the events that you are auditing for this user:

```
# login
```

- Enter login name as **team01**. Provide **team01**'s password.
- Execute the **ftp** command. Use your local host as destination host. You should see the corresponding audit records in file /var/myaudit/stream.out.

```
$ ftp localhost
Name: team01
Password: team01

ftp> bye
```

- Start the program /home/workshop/exappa_job in background. Kill the started program afterwards. You should see audit records for the **kill** audit class you have created.

```
$ /home/workshop/exappa_job &
$ kill %1
```

20. Stop the auditing subsystem.

```
# audit shutdown
```
End of exercise