AIX 6 System Administration II: Problem Determination

(Course code AU16)

Student Exercises with hints

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

Contents

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

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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:

_____

» # **lsvg**

__ b. The physical volumes for your system:

_____

» # **lspv**

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

_____

_____

_____

» # **lsvg -l rootvg**

__ d. All paging space areas for your system:

_____

» # **lsps -a**

__ e. Real memory on your system:

_____

» # **bootinfo -r**

-or-

prtconf -m

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

_____

» # **bootinfo -p**

-or-

getconf MACHINE_ARCHITECTURE

-or-

prtconf | grep "Model Architecture"

__ g. Processor type of your system:

_____

» # **prtconf | grep "Processor Type"**

__ 2. Identify the logical volumes that reside on your **hdisk0**.

Write down the command you used:

_____

» # **lspv -l hdisk0**

From the fact that the number of LPs is equal to the number of PPs, what can you conclude?

_____

» No mirroring

Using the Service Update Management Assistant (SUMA)

__ 3. Display the **man** page for the **suma** command.

» # **man suma**

Locate the examples illustrating use of this command. How many such examples are there?

_____

» On the systems used during the preparation of this exercise, the **man** page included 19 such examples.

__ 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.

» # **suma -s "30 3 15 * *" -a RqType=Fileset**
-a RqName=bos.rte.install -a RqLevel=latest -a Repeats=y
Task ID <Task ID number assigned by system> created.

__ 5. Enter a **suma** command that will list information regarding the SUMA task you just created.

```
» # suma -l <Task ID number returned in previous lab step>
```

End of exercise

Exercise 2. The Object Data Manager (ODM)

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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?

_____

» # **lsdev -P -H | pg**

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

_____

» **PdDv**

__ 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?

_____

» # **lsdev -C -c disk -H**

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

_____

» **CuDv**

__ 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: _____

» This information can be obtained from the output of the command:

`lsdev -C -c disk -F "name status location physloc"`, but answers (particularly the location information) may vary.

__ 4. Use the ODM command line interface and list the ODM object that describes this disk device. Also, use the ODM command line interface to list the ODM object that give the physical location code as part of the Vital Product Data information.

What command(s) did you used?

_____

»# **`odmget -q name=hdisk0 CuDv`**

»# **`odmget -q name=<parent-adapter> CuVPD`**

From the output complete the following list for disk **hdisk0**:

Status: _____

Chgstatus: _____

Parent: _____

Location: _____

Connwhere: _____

PdDvLn: _____

Physical Location: _____

» This information can be obtained from the output of the command **`odmget -qname=hdisk0 CuDv`**, but answers (particularly the parent and location information) may vary. For some devices, such as virtual devices, there may not be any AIX location code; in that case the physical location code provides the location information. You may have noted that the connwhere (used to locate a device once we know the parent adapter port) is often part of either the AIX location code or the physical location code.

__ 5. From this output please answer the following questions.

What is the meaning of the descriptor `chgstatus`?

_____

» A value of `2` (the expected result) indicates that the status of the disk device has not changed since the last reboot.

The **`lsdev`** command provides a description field, which is not part of ODM class **CuDv**. Where does the description come from?

_____

_____

_____

_____

» The description of a device is obtained using the descriptors `setno`, `msgno` and `catalog` from the object class **PdDv**. The attribute `PdDvLn=disk/scsi/scsd` in **CuDv** is a reference to the object in class **PdDv**.

__ 6. Execute the **`lsattr`** command and identify the *physical volume identifier* for your **hdisk0**.

What is the command you used?

_____

» # **`lsattr -El hdisk0`**

Write down the physical volume ID of the disk:

pvid: _____

» This value can be obtained from the output of the command **`lsattr -El hdisk0`**, but answers will vary. On the two systems used to test this exercise for the September 2005 revision of this course, the `pvid` values obtained were `0009330f2d01c69f0000000000000000` and `00cee60e58b2d39a0000000000000000`. Note that while these are 32-digit values, the last 16 digits are zeros.

__ 7. Use the ODM command line interface, and list the ODM object that stores the *physical volume identifier*:

What is the command you used?

_____

» # **`odmget -q"name=hdisk0 and attribute=pvid" CuAt`**

-OR-

`odmget CuAt | grep -p hdisk0 | grep -p pvid`

__ 8. The /**dev** directory contains the special files to access the devices. Write down the major and minor number of the special file for **hdisk0**.

Major number: _____

Minor Number: _____

» # **`ls -l /dev/hdisk0`**

Which ODM class is used to identify the major number and minor number for the device driver?

_____

» **CuDvDr**

Enter the following command to see the relevant entry in the **CuDvDr** object class:

`odmget -q value3=hdisk0 CuDvDr`

__ 9. List all your logical volumes that are part of the **rootvg**.

What is the command you used?

_____

» # **lsvg -l rootvg**

Query the ODM class **CuDep** and identify all logical volumes that belong to **rootvg**.

What is the command you used?

_____

» # **odmget -qname=rootvg CuDep**

-OR-

odmget -q parent=rootvg CuDv

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.

_____

» # **lsdev -P -c disk**

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.

_____

» # **odmget -q"type=osdisk and subclass=scsi" PdDv**

» OR

» # **odmget -q"type=vdisk and subclass=vscsi" PdDv**

From the output complete the following:

type: ___osdisk or vdisk_____

class: ___disk_____

subclass: ___scsi or vscsi_____

prefix: ___hdisk_____

Device Driver: ___scdisk or scsidisk_____

Configuration Method: ___/etc/methods/cfgscdisk or cfgscsidisk_____

__ 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.

_____

» # **odmget -qparent=scsi0 CuDv**

-OR-

lsdev -C -c disk -F "name status location physloc description"

From the output, write down the SCSI IDs which are in use:

SCSI IDs in use: _____

» The SCSI addresses are given on the `connwhere` line in the output of the command **odmget -qparent=scsi0 CuDv**. (You only need to write down the number before the comma.) They are also given at the end of the entry in the `location` column of the output of the command **lsdev -Cs scsi -H**.

Choose a free SCSI ID and write it down in the table. You need to specify this ID later.

Free SCSI ID: _____

» Answers will vary. Either the ID 3 or the ID 5 is often available.

__ 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.

_____

» # **mkdev -d -c disk -s scsi -t osdisk -p scsi0 -w x,0**

Note regarding **-w x,0** (last portion of command): **x** is the free SCSI ID identified in step 12

What device name has been assigned to the disk?

Device name: _____

» # **lsdev -Cs scsi**

__ 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:

_____

» If necessary, substitute the appropriate device name for **hdisk2** in the command given below:

odmget -qname=hdisk2 CuDv

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

_____

_____

» If necessary, substitute the appropriate device name for **hdisk2** in the command given below:

mkdev -l hdisk2

The **mkdev** command will call the configuration method of the disk. The name of the configuration method is stored in **PdDv**. The configuration method detects that no disk has been attached to the SCSI adapter and will fail. The disk will still be in the *defined* state.

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

Write down the command you used:

_____

» If necessary, substitute the appropriate device name for **hdisk2** in the command given below:

rmdev -l hdisk2 -d

Examine your **CuDv** object class. Did you find the removed disk in this object class?

_____

» If necessary, substitute the appropriate device name for **hdisk2** in the command given below:

```
# odmget -qname=hdisk2 CuDv
```

You will not find the removed disk in this class. The object has been removed from the ODM.

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.

» # **mkdir /tmp/odm**

```
# cd /tmp/odm
```

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

```
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:

_____

_____

» # **odmcreate parts.cre**

```
# odmshow parts
```

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

_____

» **parts.c** and **parts.h**

What do you think is the purpose of these files?

_____

_____

_____

» As an administrator, you use the ODM command line interface to access the ODM database files. For accessing the ODM from applications or system programs, an ODM application programming interface (API) exists. These programs need to include the files that have been created by **odmcreate**.

Where does the ODM class **parts** reside?

_____

» **/etc/objrepos** (if you did not change the ODMDIR variable)

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

Part Number	Description	Warehouse	Contained In
10001	Wheel	a12	50001
10003	Frame	a19	50001
10005	Saddle	a01	50001
10006	Front wheel brake	a03	50001
10007	Rear wheel brake	a03	50001
50001	City Bike Easy Rider	x99	

» **# vi parts.add**

Insert the data from the preceding table into **parts.add**, using the format shown below:

```
parts:
part_number = "10001"
part_description = "Wheel"
warehouse = "a12"
contained_in = "50001"


parts:
part_number = "10003"
part_description = "Frame"
warehouse = "a19"
contained_in = "50001"
```

. . . (Add all the information from the table, using the same format.)

 # odmadd parts.add

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

_____

» **# odmget -qcontained_in=50001 parts**

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

» Extract the object and place it in a file.

```
# odmget -qpart_description=Wheel parts > part_change
```

- Edit the file to change the `warehouse` location.

```
# vi part_change
...
warehouse="b10"
...
```

- At this point, you have two options:

Option 1:

- Remove the old record.

```
# odmdelete -qpart_description=Wheel -oparts
```

- Add the modified record.

```
# odmadd part_change
```

-OR-

Option 2:

```
# odmchange -qpart_description=Wheel -oparts part_change
```

- Verify the change.

```
# odmget -qpart_description=Wheel parts
```

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

_____

» `# odmdrop -o parts`

End of exercise

Exercise 3. System Initialization Part 1

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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? _____

» `# bootlist -m normal -o`

__ 2. Does your model support a customized **service** bootlist? _____

- What command did you use? _____

» `# bootlist -m service -o`
Your lab system should supports a service bootlist.

- 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:_____

» The customized service bootlist should have been already customized. If in a local machine environment, then it should have the CD drive listed first. If in a remote system environment, then it should have the network adapter listed first (in order to boot using a NIM server).

__ 3. Identify which disks are contained within the **rootvg**:

_____

- What command did you use? _____

» **`# lsvg -p rootvg`**

- Which disk is the bootable disk? (That means the disk that contains the boot logical volume **hd5)**: _____

  What command did you use? _____

» **`lspv -l hdisk0`** (for example) or **`lslv -m hd5`**

- What is the **logical volume type** of **hd5**? _____

  What command did you use? _____

» **`# lsvg -l rootvg`**

» TYPE: boot

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.

» `# bootlist -m normal hdisk0`

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

rootvg VGID	
First disk PVID	
Second disk PVID	

- What command did you use to determine the **rootvg** VGID? _____

»`# lsvg rootvg`

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

»`# lspv`

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

»`odmget -q "name=hdisk0 and attribute=pvid" CuAt`

- What difference do you see with the ID value?

» The ODM stores physical volume IDs in a 32-number field, and adds 16 zeros to the ID of the disk. **lspv** just shows 16 bytes.

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 straight forward:

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".

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.

»**# /home/workshop/ex3pro1**

You have successfully broken your machine!
Now, run **shutdown -Fr** to attempt a reboot.

»**# shutdown -Fr**

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

» On an AIX system that is not an LPAR system:

Your systems shows 20EE000B on the console display, a series of 4 digit "E" codes on the operator panel LED/LCD and continues with reboot attempts. The boot record at the beginning of your boot disk pointing to **hd5** has been removed.

» On an LPAR system (using the HMC console):

- Access the HMC and locate your partition as described in the previous section.
- The state of the partition will depend on the level of code. On older POWER5 systems you might see a state of `open firmware`. On newer systems you might see a state of "starting" with a reference code of AA06000D.

  The boot record at the beginning of your partition has been removed. When an LPAR is unable to locate a boot image, its behavior depends on the firmware level. On older firmware levels it automatically booted to SMS, which is a menu front end to the system firmware. At newer firmware levels, it repeatedly retries the bootlist and displays a message of:

  "No OS image was detected by firmware

  At least one disk in the bootlist was not found yet

  Firmware is now retrying the entries in the bootlist

  Press Ctrl-C to stop retrying"

  Pressing Ctrl-C would then boot to SMS.

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

» On an AIX system that is not an LPAR system:

- Follow the procedure in the previous section to boot to maintenance mode. Since you do not have an AIX command prompt, you will need to power off the machine rather than using the **shutdown** command. You may use either the default bootlist (**F5/5**) or the service bootlist (**F6/6**).
- The console should eventually display the **Installation and Maintenance** menu.

» On an LPAR system (using an HMC):

- If you do not already have a virtual console window for your LPAR, then open a virtual terminal at this point. The procedure varies depending upon the version of HMC you are using.
  - If pre-HMCv7, then right-click your partition icon and select **Open Terminal Window** to obtain a virtual console window.
  - If HMCv7 or later, then select your partition and when the menu icon appears, click the menu icon. Click the “Console Window” item and then click the “open terminal connection” item in the sub-menu.
- You should see a multi-boot SMS menu.
- Type **x** to exit out of SMS. This will cause the system to start booting again.
- Periodically, press the numeric **6** key to cause the use of the customized service bootlist. If you again get the “No OS Image was detected by firmware” message, respond by pressing Ctrl-C and again repeatedly pressing the numeric 6 key.

  There are alternate ways we could have used to get to maintenance mode:

  We could alternately have used SMS to control select the boot device and, in a network boot, provided SMS with the networking details.

  We also could have shut down the partition and, from that state, booted to a maintenance mode following the procedure given earlier.
- The console should eventually display the **Installation and Maintenance** menu.

__ 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.

» Select your terminal.

» Select your language.

» If booting from media you will see: **Welcome to the Base Operating System Installation and Maintenance**. From here, select **Start Maintenance Mode for System Recovery**.

» From the **Maintenance** menu, select option **1**, **Access a Root Volume Group**.

» Type **0** to continue.

» The **Access a Root Volume Group** screen displays. Select the option for the root volume group whose logical volume information you want to display. The **Access a Root Volume Group** screen lists all of the volume groups (**root** and otherwise) on your system. After entering your selection, the **Volume Group Information** screen displays. Note: Reviewing the disk PVID and location code information on the **Volume Group Information** screen enables you to determine whether the volume group you selected was the root volume group disk you examined earlier. You can return to the **Access a Root Volume Group** screen if the choice you made was not the correct root volume group disk. You cannot continue beyond the **Volume Group Information** screen without selecting a disk; and if you select the wrong disk you will either get serious errors attempting to treat the disk as a rootvg disk or you will be working with a different instance of an operating than you intended (such as in a alternate disk install environment).

» Select option **1**, **Access this volume group and start a shell**. Selecting this choice imports and activates the volume group and mounts the file systems for this root volume group before providing you with a shell and a system prompt.

__ 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.

» # bootlist -o -m normal

» # lsvg -l rootvg

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

» **# bosboot -ad /dev/hdisk0** (for example)

»**# sync**

»**# sync**

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

» Do a proper shutdown and reboot:

 # shutdown -Fr

You may see a series of error messages related to workload partitions (wpar), such as messages involving corral. We do not have any workload partitions configured and you may safely ignore these error messages.

End of exercise

Exercise 4. System Initialization Part 2

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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.

» `# lsvg -l rootvg`

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

»`# /home/workshop/ex4pro1`

__ 3. Power on your system.

» if you are using a remote LPAR, follow the instructions in the step *Start your partition to multiuser mode* in the Exercise 3, Part 3.

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

_____

_____

» LED, LCD or HMC **Operator Panel Value** will show `0557` and, if your system has a two line LED display, will also show `ROOT MNT FAILED`). The mount of /**dev**/**hd4** (root file system) failed.

__ 5. Boot your machine in maintenance mode.

» If not using a remote LPAR, power your machine off and back on again, booting to maintenance mode as described in Exercise 3, Part 3 (*Booting to maintenance mode*).

» If using a remote LPAR, use your HMC to shut down your partition and then activate your partition to maintenance mode as described by the procedures described in Exercise 3, Part 3 (*Booting to 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.

» Select your terminal.

» Select your language.

» If booting from media you will see: **Welcome to the Base Operating System Installation and Maintenance**. From here, select **Start Maintenance Mode for System Recovery**.

» From the **Maintenance** menu, select option **1**, **Access a Root Volume Group**.

» Type **0** to continue.

» The **Access a Root Volume Group** screen displays. Select the volume group that is causing the problem.

» Select option **2**, **Access this volume group and start a shell before mounting file systems**. Notice the error messages while **rootvg** is varied on. These provide more clues to the problem:

```
Importing Volume Group...
imfs: can't find log for volume group rootvg
rootvg
Checking the / filesystem.
```

```
fsck: Cannot find the vfs value for file system /dev/hd4
.Checking the /usr filesystem.

fsck: Cannot find the vfs value for file system /dev/hd2
.Exit from this shell to continue the process of accessing the
root volume group.
```

__ 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`**.

» If it is a JFS2 file system:

```
# logform -V jfs2 /dev/hd8
logform: Destroy /dev/hd8 (y)? y
# fsck -y -V jfs2 /dev/hd1
# fsck -y -V jfs2 /dev/hd2
# fsck -y -V jfs2 /dev/hd3
# fsck -y -V jfs2 /dev/hd4
# fsck -y -V jfs2 /dev/hd9var
# fsck -y -V jfs2 /dev/hd10opt
# fsck -y -V jfs2 /dev/hd11admin
```

» If it is a JFS file system:

```
# logform -V jfs /dev/hd8
logform: Destroy /dev/hd8 (y)? y
# fsck -y -V jfs /dev/hd1
# fsck -y -V jfs /dev/hd2
# fsck -y -V jfs /dev/hd3
# fsck -y -V jfs /dev/hd4
# fsck -y -V jfs /dev/hd9var
# fsck -y -V jfs /dev/hd10opt
# fsck -y -V jfs /dev/hd11admin
```

__ 8. Shut down your system and reboot your system in normal mode.

»**`# shutdown -Fr`**

» If you are unable to shutdown from the command prompt, then stop and start your system:

- For non-LPAR systems, use the front panel power control.
- For remote LPAR systems, use your HMC to shut down your partition and then activate your partition to normal mode as described in Exercise 3.

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?

_____

_____

_____

» The system stops with LED/LCD or HMC Operator Panel Value of `0553`
This is an indication for a corrupted **/etc/inittab**.
Another possible symptom would be for a prompt to appear on the AIX system console which asks for a run level; when responding with a multi-user run level of “2”, the system may simply hang with no message or error code.

__ 10. Reboot the system to maintenance mode.

» If not using a remote LPAR, power your machine off and back on again, booting to maintenance mode as described in Exercise 3, Part 3 (*Booting to maintenance mode*).

» If using a remote LPAR, use your HMC to shut down your partition and then activate your partition to maintenance mode as described by the procedures described in Exercise 3, Part 3 (*Booting 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.

» Select your terminal.

» Select your language.

» If booting from media you will see: **Welcome to the Base Operating System Installation and Maintenance**. From here, select **Start Maintenance Mode for System Recovery**.

» From the **Maintenance** menu, select option **1**, **Access a Root Volume Group**.

» Type **0** to continue.

» The **Access a Root Volume Group** screen displays. Select the volume group that is causing the problem.

» Select option **1**, **Access this volume group and start a shell**. Selecting this choice imports and activates the volume group and mounts the file systems for this **root** volume group before providing you with a shell and a system prompt.

» If you are using an LFT terminal, issue the command:
```
# export TERM=lft
```

» The corrupted file is /**etc**/**inittab**.

__ 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.

» Notice that the file has a semi-colon instead of a colon as the first delimiter. Correct this by manually editing /**etc**/**inittab**.

» **# vi /etc/inittab**
:%s/;/:/g
:wq!

» Shut down your system and reboot your system in normal mode. Your machine should boot now without any boot failure.

```
# sync
# sync
# shutdown -Fr
```

End of exercise

Exercise 5. LVM Tasks and Problems

(with hints)

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/rvgrecover**

Exercise Instructions with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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).

»**# extendvg rootvg hdisk1**

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

» Specify the name of an existing volume group (such as **rootvg**) on the first SMIT screen that appears when you enter **smit mklv**. Then, enter the values shown below on the "main" screen that appears next:

Logical Volume NAME	**mirrorlv**
Number of LOGICAL PARTITIONS	**2**
Number of COPIES of each logical partition	**2**
Allocate each logical partition copy on a SEPARATE physical volume?	**yes****

Note: Set this value to **no if you only have one physical volume in your volume group.

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:

LP	PP1	PV1	PP2	PV2
0001				
0002				

» Use the following command to obtain the information required to complete the table:

```
# lslv -m mirrorlv
```

Finally, remove the logical volume **mirrorlv**.

» # rmlv 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.

» # **lspv -p hdisk0**

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.

» Enter the values shown below on the main **smit mklv** screen:

Logical Volume NAME	**lvtmp1**
Number of LOGICAL PARTITIONS	**1**
PHYSICAL VOLUME names	**hdisk0**
POSITION on physical volume	***

*** Select a region that has free PPs. You determined which regions have space available by using **lspv -p hdisk0** in the previous lab step.

» or

» **# mklv -y lvtmp1 -a "***" rootvg hdisk0** (where *** is either ie, im, c, m, or e)

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

» To check the position of **lvtmp1:**

 # lspv -p hdisk0

__ 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?

_____

»**# chlv -a ie lvtmp1**

» Use **lspv -p hdisk0** to determine whether the partitions have been moved to another region. The expected result is that the partitions have NOT moved.

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:

_____

» # **reorgvg rootvg lvtmp1**

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

» The partitions have been moved to the location specified earlier.

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

_____

» Answers will vary. In the hints that follow, we will assume the partitions identified are partitions 82 and 83 on **hdisk0**.

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
```

» Create the map file:

```
# vi /tmp/lvtmp2map
```

- Add the free partitions that you identified into the allocation file. For example,

```
hdisk0:82-83
```

- Next, use **smit mklv** to create lvtmp2 with 2 logical partitions on the selected disk and modify the screen to use your map file:

```
File containing ALLOCATION MAP /tmp/lvtmp2map
```

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

» # **lspv -p hdisk0**

__ 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
```

» Refer to the table on the visual entitled “Volume Group Limits.” This **mkvg** command creates a big volume group with a maximum number of 4064 partitions on a disk (**-t 4** indicates 4*1016 partitions). The maximum number of disks is 32.

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.

Disk name	PVID	Volume group

» # lspv

__ 8. Execute **lsvg -p** to list all physical volumes that are part of your **rootvg**.

Complete the following table:

PV_NAME	PV STATE	TOTAL PPs

» # lsvg -p rootvg

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

_____

» # odmget -q "name like hdisk? and attribute=pvid" CuAt

PV_Name	PVID

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.

_____

_____

_____

_____

_____

_____

_____

name =
attribute = “pvid”
value =
type = “R”
generic = “D”
rep = “s”
nls_index = 11

__ 10. Execute the program **/home/workshop/ex5_corrupt_pvid**.

» # **/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.

Disk name	PVID	Volume group

» # **lspv**

The value PVID value is “none,” and the Volume group value is “None” for each entry.

__ 12. Repeat the **lsvg -p** command you used earlier to list the physical volumes in **rootvg**.

What is the output from the command?

_____

_____

_____

_____

» # **lsvg -p rootvg**

You get a number of error messages (the number of messages depends on the number of disks in **rootvg**) like the following:

```
0516-304 lsvg: Unable to find device id 00008371b5969c35 in the Device
Configuration Database
```

The same information is shown as that which was obtained when the command **lsvg -p rootvg** was executed earlier, but instead of disk names the PVIDs are shown.

__ 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:

_____

» Physical volume objects

__ 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*.

_____

» # **odmget -q "name like hdisk?" CuDv**
odmget -q "name=hdisk0" CuAt
odmget -q "name=hdisk1" CuAt

The PVIDs for all disks in object class **CuAt** are missing.

__ 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?

» # **lqueryvg -p hdisk0 -At** (or another disk in the appropriate volume group instead of **hdisk0**)

Look for the label "Physical." The values given after this label are the missing PVIDs.

__ 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.

_____

_____

» # **vi fix.add**

Example objects (*Use your information!*):

```
CuAt:
   name = "hdisk0"
   attribute = "pvid"
   value = "00008371b5969c3500000000000000000"
   type = "R"
   generic = "D"
   rep = "s"
   nls_index = 2

CuAt:
   name = "hdisk1"
   attribute = "pvid"
   value = "002106699b1dd44400000000000000000"
   type = "R"
   generic = "D"
   rep = "s"
   nls_index = 2
```

- # **odmadd fix.add**

__ 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` valve?

» # **lspv**

lsvg -p rootvg

The output from these commands should indicate that your fix worked.

- *If your fix does not work*:

```
# odmdelete -o CuAt -q"attribute=pvid"
```

Fix the typo, and add the fixed objects:

```
# odmadd fix.add
```

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

__ 18. Execute the program **/home/workshop/ex5_corrupt_odm**.

» # **/home/workshop/ex5_corrupt_odm**

__ 19. Verify the following information:

__ a. Check whether your volume groups are ok. Use **lsvg**.

» # **lsvg**

__ b. Check whether your physical volumes are ok. Use **lspv**.

» # **lspv**

__ c. Check whether your logical volumes are ok. List all logical volumes that are part of your **rootvg**. Use **lsvg -l rootvg**.

» # **lsvg -l rootvg**

What happens?

_____

_____

_____

» Typically for this problem, the TYPE information for some logical volumes is not shown. (The string ??? is shown instead.)

The logical volume type is stored in **CuAt**; so, this result indicates that there might be a problem with logical volume objects in the **CuAt** object class in the ODM.

__ 20. Display information for logical volume **hd2**. Use **lslv hd2**.

» # **lslv hd2**

What happens?

_____

_____

_____

» The following error message is displayed:

```
0516-306 lslv: Unable to find hd2 in the Device
        Configuration Database.
```

__ 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?

_____

_____

_____

» Use the following commands:

- `# odmget -q "name=hd2" CuDv`
  `# odmget -q "name=hd4" CuDv`

  The results indicate that the logical volumes are missing in **CuDv**.

- `# odmget -q "name=hd2" CuAt`
  `# odmget -q "name=hd4" CuAt`

  The results indicate that the LVIDs are missing in **CuAt**.

__ 22. Examine the **/home/workshop/rvgrecover** 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/rvgrecover**. Ignore the messages. This may take up to 1 minute, depending upon the speed of your lab system.

» `# /home/workshop/rvgrecover`

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

» `# lsvg -l rootvg`
`# lslv hd2`

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

» The two main steps are:

- Deleting all **rootvg** ODM objects
- Importing new ODM objects by reading the information from the VGDA and LVCB on the boot disk.

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

» `# rmlv lvtmp1`
`# rmlv lvtmp2`

End of Part 2

End of exercise

Exercise 6. Mirroring rootvg

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

Exercise: Mirror and Unmirror the Complete rootvg

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

_____

_____

» # **lspv**

The third column of output will indicate which disks are currently in **rootvg**.

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?

_____

» # **lspv -l hdiskx**

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:

_____

_____

_____

_____

_____

_____

_____

_____

_____

» The following hints assume that your **rootvg** originally contained only **hdisk0** and that **hdisk1** is available and is not currently assigned to a volume group.

- **Solution A:**

```
# extendvg rootvg hdisk1 (if hdisk1 is not already in rootvg)
# chvg -Qn rootvg        (This step is to disable quorum.)
# mirrorvg -s rootvg
```

-OR-

- **Solution B:**

```
# extendvg rootvg hdisk1 (if hdisk1 is not already in rootvg)
# chvg -Qn rootvg        (This step is to disable quorum.)
# mklvcopy hd1 2 hdisk1
# mklvcopy hd2 2 hdisk1
# mklvcopy hd3 2 hdisk1
# mklvcopy hd4 2 hdisk1
# mklvcopy hd5 2 hdisk1
# mklvcopy hd6 2 hdisk1
# mklvcopy hd8 2 hdisk1
# mklvcopy hd9var 2 hdisk1
# mklvcopy hd10opt 2 hdisk1
# mklvcopy  hd11admin 2 hdisk1
```

Do the same for any other **rootvg** logical volumes.

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

» `# 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:

_____

» `# syncvg -v rootvg`

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

» # **lsvg rootvg**

__ 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**?

» No. The value YES is shown for only one of the two disks.

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

Write down the commands you executed:

_____

_____

» # **bosboot -a**
(**bosboot** will default to the disk used on the last boot when the **-d** flag is not used.)

» If you get an error of “not enough space to create”, then check and then increase the filesystem space:

»**# lsfs**

»**# chfs -a size=+1 /tmp**

» # **bootlist -m normal hdisk1 hdisk0** (This example specifies both **hdisk1** and **hdisk0** as boot devices available for normal boots.)

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

» # **ipl_varyon -i**

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**?

» Yes. The value YES is now shown for both of the two disks.

__ 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**.

» # **lslv -m hd5**

__ 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.

_____

_____

» # **shutdown -Fr**
bootinfo -b

__ 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.

» # **errclear 0**
/home/workshop/ex6_diskfailure
touch /tmp/data1
touch /tmp/data2

__ 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:.

» # **smitty errpt**

- Select **1 filename (defaults to stdout)**
- Select **1 no** (in answer to question: CONCURRENT error reporting?)
- Specify **summary** for Type of 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.

» # **smitty errpt**

- Select **1 filename (defaults to stdout)**
- Select **1 no** (in answer to question: CONCURRENT error reporting?)
- Specify **detailed** for Type of Report.

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:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

» You should see a number of error log entries like the following:

1. LABEL: LVM_SA_STALEPP
...
Description
PHYSICAL PARTITION MARKED STALE

2. LABEL: LVM_SA_PVMISS
...
Description
PHYSICAL VOLUME DECLARED MISSING

__ 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?

_____

_____

_____

» # **lspv hdisk0**
lspv hdisk1

One of the disks is in the removed state and has stale partitions.

__ 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.

» The removed state is shown on the visual entitled “Physical Volume States.”

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

_____

» # **varyonvg rootvg**

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

_____

_____

» Several error messages are displayed.

The disk still has the `removed` state.

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

_____

» # **`chpv -v a hdisk1`** (Substitute the appropriate disk name for **hdisk1**.)

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

_____

» # **`lspv hdisk1`** (Substitute the appropriate disk name for **hdisk1**.)

The disk is back in the `active` state.

__ 18. Check whether your **rootvg** still contains *stale* partitions.

_____

_____

» # **`lsvg rootvg`**

The **rootvg** still contains stale partitions.

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

_____

» # **`varyonvg rootvg`**

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

» This should update the stale partitions, but note that the synchronization takes some time. (The **`varyonvg`** command starts **`syncvg`** in the background.) You can monitor the process by executing **`lsvg rootvg`** periodically and watching the value of `STALE PPS` decrease.

__ 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**:

_____

» # **`unmirrorvg rootvg hdisk1`** (We're using **hdisk1** as an example.)

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

_____

» You get a recommendation to execute **chpv -c <diskname>** to clear the boot record and avoid a potential boot off an old boot image.

Follow this recommendation and clear the boot record.

_____

» # **chpv -c hdisk1** (We are using **hdisk1** as an example.)

__ 21. Check that all logical volumes have been removed from the disk and remove the "empty disk" from **rootvg**.

_____

_____

» # **lspv -l hdisk1** (We're using **hdisk1** as an example.)

The output should show that **hdisk1** is empty now.

reducevg rootvg hdisk1 (We're using **hdisk1** as an example.)

__ 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:

_____

_____

_____

_____

» # **bosboot -ad /dev/hdisk0** (using **hdisk0** as an example)

bootlist -m normal hdisk0

shutdown -Fr

bootinfo -b

End of exercise

Exercise 7. Exporting and Importing Volume Groups

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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:

_____

» To find a disk that is empty:

```
# lspv
```

» You may need to remove **hdisk1** from **rootvg**. If so, use:

```
# reducevg rootvg hdisk1
```

» To create the **datavg** volume group:

```
# mkvg -s 16 -y datavg hdisk1
```

If you get the following error message:

```
0516-1398 mkvg: The physical volume hdisk1, appears to belong
to another volume group. Use the force option to add this
physical volume to a volume group.
0516-862 mkvg: Unable to create volume group.
```

Then, use this command:

```
# mkvg -f -s 16 -y datavg hdisk1
```

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

_____

» List the active volume groups:

```
# lsvg -o
datavg
rootvg
```

__ 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

» **# smit mklv**

__ 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.

»**# smit jfs**

__ 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**:

»

```
# lsvg -l datavg
datavg:
LV NAME   TYPE     LPs   PPs   PVs   LV STATE      MOUNT POINT
lv_raw    jfs      1     1     1     closed/syncd  N/A
loglv00   jfslog   1     1     1     closed/syncd  N/A
lv00      jfs      1     1     1     closed/syncd  /home/jupiter
lv01      jfs      1     1     1     closed/syncd  /home/mars
```

LV NAME	TYPE	MOUNT POINT

An example of the table filled in based on the results above are:

LV NAME	TYPE	MOUNT POINT
lv_raw	jfs	**N/A**
loglv00	jfslog	**N/A**
lv00	jfs	**/home/jupiter**
lv01	jfs	**/home/mars**

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

```
»# mount /home/jupiter
»# mount /home/mars
»# cd /home/jupiter
»# touch j1 j2 j3
»# cd /home/mars
»# touch m1 m2 m3
```

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

_____
_____
_____

```
»# cd
»# umount /home/jupiter
»# umount /home/mars
»# varyoffvg datavg
»# exportvg datavg
```

__ 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**.)

```
» # lsfs
```

/home/jupiter and **/home/mars** do not exist on the system.

```
» # more /etc/filesystems
```

/etc/filesystems contains no reference to a file system that has been exported.

__ 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:

_____

```
»# importvg -y datavg hdisk1
```

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

```
» # lsvg -o
```
(datavg should be varied on)

Check to see if the file system information is back.

```
»# lsfs
»# more /etc/filesystems
»# mount
```

References are there but file systems are not mounted.

__ 11. Mount the /**home**/**jupiter** and /**home**/**mars** file systems. Check that no files have been lost.

```
»# mount /home/jupiter
»# mount /home/mars
»# ls /home/jupiter
»# ls /home/mars
```

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.

```
»# cd
»# umount /home/jupiter
»# umount /home/mars
»# varyoffvg datavg
»# exportvg datavg
```

__ 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

```
» # smit mklv
```

__ 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**

```
»# smit jfs
```

__ 15. What are the corresponding logical volume names that have been created for the file system?

```
»# lsfs
```

Logical volume for /**home**/**jupiter**: _____/**dev**/**lv00** (for example)_____

Logical volume for /**home**/**mars**: _____/**dev**/**lv01** (for example)_____

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

```
»# mount /home/jupiter
»# mount /home/mars
»# cd /home/jupiter
»# touch j20 j21 j22
»# cd /home/mars
»# touch m20 m21 m22
```

__ 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.

```
»# importvg -y datavg hdisk1
        0516-530 synclvodm: Logical volume name lv_raw changed to fslv00.
        0516-530 synclvodm: Logical volume name loglv00 changed to loglv01.
        0516-530 synclvodm: Logical volume name lv00 changed to fslv01.
        0516-530 synclvodm: Logical volume name lv01 changed to fslv02.
        imfs: mount point "/home/jupiter" already exists in
        /etc/filesystems
        imfs: mount point "/home/mars" already exists in /etc/filesystems
        datavg
```

__ 18. Write down the new logical volume names that are created for **datavg** during the import.

_____

_____

_____

» **lv_raw** has been changed to **fslv00** (Example names)

» **loglv00** has been changed to **loglv01**

» **lv00** has been changed to **fslv01**

» **lv01** has been changed to **fslv02**

» Mount point /**home**/**jupiter** already exists in /**etc**/**filesystems**

» Mount point /**home**/**mars** already exists in /**etc**/**filesystems**

__ 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**.

```
»# cd
»# umount /home/jupiter
»# umount /home/mars
```

__ 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.

_____

_____

```
»# mount -o log=/dev/loglv01 -V jfs /dev/fslv01 /home/jupiter
»# mount -o log=/dev/loglv01 -V jfs /dev/fslv02 /home/mars
```

__ 21. Check the files you have created in /**home**/**jupiter** and /**home**/**mars**. They should exist in these directories.

```
»# ls /home/jupiter
»# ls /home/mars
```

__ 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**.

```
»# umount /home/jupiter
»# umount /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**.

```
»# mkdir -p /datavg/jupiter
»# mkdir -p /datavg/mars
```

__ 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**.

```
»/datavg/jupiter:
                dev = /dev/fslv01
                vfs = jfs
                log = /dev/loglv01
                mount = false
                options = rw
                account = false

»/datavg/mars:
                dev = /dev/fslv02
                vfs = jfs
                log = /dev/loglv01
                mount = false
                options = rw
                account = false
```

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

```
»# mount /datavg/jupiter
»# mount /datavg/mars
»# mount /home/jupiter
»# mount /home/mars
```

__ 26. Verify you can access all the files.

```
»# ls /datavg/jupiter
»# ls /datavg/mars
»# ls /home/jupiter
»# ls /home/mars
```

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

```
»# umount /datavg/jupiter
»# umount /datavg/mars
```

__ 28. Varyoff the **datavg** volume group.

```
»# varyoffvg datavg
```

__ 29. Export the **datavg** volume group.

```
»# exportvg datavg
```

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

```
»# umount /home/jupiter
»# umount /home/mars
»# rmfs /home/jupiter
»# rmfs /home/mars
```

End of exercise

Exercise 8. Saving and Restoring a User Volume Group

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

__ 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.)

```
»# lsvg -o
```

If its not defined, use

```
# importvg -y datavg hdisk1
# mount /home/jupiter
# mount /home/mars
```

__ 2. What is the physical partition size of **datavg**? ___16 MB_____

```
» # lsvg datavg
```

__ 3. How many partitions are allocated for /**home/jupiter**? ___1___

```
»# lsvg -l datavg
```

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

```
»mkvgdata datavg
```

__ 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.

```
»# vi /tmp/vgdata/datavg/datavg.data
```

```
lv_data:
        ...
        LPs= 4
        ...
        MOUNT_POINT= /home/jupiter
        ...

fs_data:
        FS_NAME= /home/jupiter
        FS_SIZE= 131072 (4 times original value; 4 x 32768 = 131072)
         ...
```

__ 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.

» **# savevg -f /tmp/datavg.img datavg**

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

» **# umount /home/mars**
» **# umount /home/jupiter**

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

» **# varyoffvg datavg**

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

» **# exportvg datavg**

__ 10. Execute the **restvg** command and restore the volume group from your backup image. Write down the command you used.

» **# restvg -f /tmp/datavg.img hdisk1**

» The utility will first display the vgname, target disk and allocation policies. As prompted, enter “y” to continue.

» You may see error messages about the upper bound being outside the allowable limits. This will not stop your restore from being successful and can safely be ignored. The **restore** utility will make adjustments for the problem and an acceptable upper bound will be automatically established.

__ 11. Write down the number of partitions that are allocated for /**home/jupiter**:

```
»# lsvg -l datavg
```

(Should be 4 partitions now)

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

```
»# smit savevg
        Backup DEVICE or FILE                    [/tmp/datavg.img]
        VOLUME GROUP to back up                  [datavg]
```

The command SMIT executes is:
/usr/bin/savevg -f /tmp/datavg.img -i datavg

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

```
»# umount /home/mars
»# umount /home/jupiter
»# varyoffvg datavg
»# exportvg datavg
```

__ 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.

```
»# smit restvg
        Restore DEVICE or FILE                   [/tmp/datavg.img]
        Physical partition SIZE in megabytes     [64]
```

The command SMIT executes is:
/usr/bin/restvg -q -f /tmp/datavg.img -P 64

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

```
»# lsvg datavg
```

The physical partition size is now 64 MB.

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

```
»# lsvg -l datavg
```

It should be 1 partition now, because the physical partition size was increased.

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

» No. When the physical partition size is not the same as specified in **vgname.data**, each logical volume will be altered with respect to the new physical partition size.

End of exercise

Exercise 9. Error Log and syslogd

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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:

_____

» # **errpt**

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

_____

» # **errpt -a**

__ 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:

  _____

» # **smit errpt**

After generating the report, press **F6** or **<Esc-6>** to see the command used.

The command SMIT executes is **errpt -s 'mmddhhmmyy'** (where **mmddhhmmyy** were the month, day, hour, minute, and year 24 hours ago).

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

  _____

» # **smit errpt**

After generating the report, press **F6** or **<Esc-6>** to see the command used.

The command SMIT executes is **errpt -a -d H**.

__ 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:

_____

» # **errpt -c**

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

_____

» # **errlogger "This is a test."** (The text shown here is just an example.)

Is the complete error text shown in the error report?

_____

» No. Only the description OPERATOR NOTIFICATION is shown.

Stop concurrent error logging.

» **<Ctrl-c>**

__ 5. Write down the characteristics of your error log:

LOGFILE: _____

Maximum LOGSIZE: _____

Memory BUFFER SIZE: _____

What command have you used to show these characteristics?

_____

» # **smit errdemon**

-OR

/usr/lib/errdemon -l

Notice that the labels for these characteristics used by SMIT are somewhat different from the labels used when you execute the **errdemon** command directly.

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

» # **errpt -d O**

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

```
» # errclear -d O 0
```

__ 8. Verify that the operator entries are now gone.

```
» # errpt -d O
```

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**:

_____

» `daemon.debug /tmp/syslog.debug`

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

_____

» `# touch /tmp/syslog.debug`

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

_____

» `# refresh -s syslogd`

__ 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:

_____

_____

» `# stopsrc -s inetd`

`# startsrc -s inetd -a "-d"`

__ 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.

_____

» `# telnet <name of your host>` (Use the name of your own host.)

`# exit`

__ 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:

_____

_____

» # **stopsrc -s inetd**

startsrc -s inetd

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

_____

» # **pg /tmp/syslog.debug**

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

_____

» *.debug errlog

__ 15. Refresh the **syslogd** subsystem. Write down the command that you used:

_____

» # **refresh -s syslogd**

__ 16. Generate a **syslogd** message, for example, use an invalid password during a login. Check that the message is posted to the error log.

_____

_____

» # **login** (Use an invalid password.)
After three bad attempts, return to your command prompt and check the error log.
errpt | more

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:

_____

_____

_____

_____

_____

_____

_____

```
» # vi notify.add
        errnotify:
        en_name="sample"
        en_persistenceflg=0
        en_class="O"
        en_method="errpt -a -l $1 | mail -s ERRLOG root"
        # odmadd notify.add
```

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

_____

```
» # errlogger test entry in the log
```

__ 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.

_____

```
» # mail
  ? t
```

End of Part 2

End of exercise

Exercise 10.Diagnostics

(with hints)

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 with Hints

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 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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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

» `# lsdev -Cc adapter | grep ent`

» Look for adapters which are not virtual adapters. For the physical adapters, determine if the corresponding ethernet interface is configured.

»`# netstat -in`

__ 2. If your physical Ethernet adapter is not configured, then configure it with an 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.

»`# smitty chinet`

or

»`# chdev -l en1 -a netaddr=192.168.252.5 -a state=up`

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

» `# diag`

- At the **FUNCTION SELECTION** screen, select **Diagnostic Routines**
- You may be asked for your terminal type if it has not been defined.
- In the **DIAGNOSTIC MODE SELECTION** screen, select **System Verification**.

- Select your configured physical communications adapter (e.g. **ent1**) by using cursor control to position the cursor on the adapter and pressing enter.
- Press **F7** or **<esc-7>** to commit

» The communications adapter was not able to be tested. You should get the message "No trouble was found. However, the resource was not tested because the device driver indicated that the resource was in use."

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

» Press the **Previous Menu** key (**F3** or **<Esc-3>**) until you see the **FUNCTION SELECTION** screen. Then, select **Diagnostic Routines**.

» **System Verification** tests a resource and does not analyze the error log.

Problem Determination tests a resource and analyzes the error log.

Problem Determination should not be used after a hardware repair unless the error log has been cleaned up.

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

» Press the **Previous Menu** key (**F3** or **<Esc-3>**) until you see the **FUNCTION SELECTION** screen.

- Select **Task Selection**.
- Select **Display Hardware Vital Product Data**.
- Select **ent1** (or whatever physical adapter is available on your system)
- Press **F7** or **<esc-7>** to commit.

» If you have only virtual adapters, then there is not much information. If you have a physical adapter (or disk) the VPD information can be fairly extensive.

__ 6. Return to the **TASKS SELECTION LIST** screen.

» Press the **Previous Menu** key (**F3** or **<Esc-3>**) until you see the **TASKS SELECTION LIST** screen.

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

» Select **Automatic Error Log Analysis and Notification**.

» You will see the screen:

```
AUTOMATIC ERROR LOG ANALYSIS AND NOTIFICATION SERVICE AID
```

```
This task controls the error notification mailing list for both
Periodic Diagnostics and Automatic Error Log Analysis. The error
notification mailing list can consist of system users and email
addresses of the form user@domain. Also, this task allows
automatic error log analysis to be disabled or enabled. By
default automatic error log analysis is enabled.


To continue, press 'Enter'.
```

» Press **Enter.**

» Select **Display the error notification mailing list**.

» By default, the list is empty.

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.

» Press the **Previous Menu** key (**F3** or **<Esc-3>**) until you see the **AUTOMATIC ERROR LOG ANALYSIS AND NOTIFICATION SERVICE AID** screen.

» Select **Add to the error notification mailing list**.

» You will see the following menu. Add **root**.

```
ADD TO THE ERROR NOTIFICATION LIST                          802103

Type in an email address (including the user name and domain) or
a system user to be notified of hardware problems, then press
'Commit' to add it to the mailing list.

  email address or system user                    [root]          +
```

» Press **F7** or **<esc-7>** to commit.

» Press **F10** or **<esc-0>** to exit **diag.**

__ 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.

»Follow the instructions on how to start a virtual terminal provided in Exercise 3, Part 3 for your HMC environment

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.)

»**# shutdown -F -m**

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

»INIT: SINGLE USER MODE

Password: ***root's password***

d. Start the diagnostics facility.

»**# diag**

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

» At the **Function Selection** menu, select the **Diagnostic Routines**.

» You may be asked for your terminal type if it has not been defined:

- If you are at a graphics console, set the terminal type to `lft` (low function terminal).
- If you are using a remote virtual console, set the terminal type to `vt320`.

» In the **DIAGNOSTIC MODE SELECTION** screen, select **System Verification**.

» Select your communications adapter (ex. **ent1**).

» Press **F7** or **<esc-7>** to commit.

» The communications adapter was able to be tested. Hopefully, you got the message `No trouble was found`.

__ 9. Exit the diagnostic utility.

» Press **F10** or **<Esc-0>**

__ 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:

»**# shutdown -F**

- For a local server, when you see `halt completed` use the front panel to power off the machine.

- For a remote server, when you see `halt completed` close the virtual console window:
  - The HMC window should show a partition state of **shutting down** and eventually **not activated**.
  - If the state stays at **running**, then use the HMC to shut down the partition. Use the procedure for your HMC provided in Exercise 3, Part 3.
  - If not using HMCv7, you may need to also close the virtual terminal from the HMC window in order to start it again later; right-click your partition and select **Close Terminal Connection** and confirm when prompted.

b. Boot your system to diagnostics using service mode off the hard drive.

» For a local server:

- Make sure that there is no bootable media in the CD drive.
- Power on the machine from the front panel.
- When the console displays the list of discovered devices (memory, keyboard, network, scsi, speaker), press **F5** (or **5**).

» For a remote server which is NOT using NIM:

- Ensure that the service bootlist has the hard drive listed first (DO NOT MAKE THIS CHANGE FOR NIM ENVIRONMENTS).

```
•# bootlist -m service hdisk0
```

- In the HMC window, activate your LPAR to service mode using the default bootlist. Follow the *Activate your LPAR into maintenance mode* procedure in Exercise 3. Due to the service bootlist change, this will boot to diagnostics mode.

» For a remote server which IS using NIM:

- In the HMC window, activate your LPAR to service mode using the default bootlist. Follow the *Activate your LPAR into maintenance mode* procedure in Exercise 3, except use the **5** key (default bootlist) instead of the **6** key (stored bootlist). Since there should not be at bootable CD, the system will boot off the hard drive into diagnostic mode.

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

» At the **Function Selection** menu, select the **Diagnostic Routines**.

» You may be asked for your terminal type if it hasn't been defined:

- If you are at a graphics console, set the terminal type to `lft` (low function terminal).
- If you are using a remote virtual console, set the terminal type to `vt320`.

» In the **DIAGNOSTIC MODE SELECTION** screen, select **System Verification**.

» Select your communications adapter (ex. **ent1**).

» Press **F7** or **<esc-7>** to commit.

» The communications adapter was able to be tested. Once again, you should see the message `No trouble was found`.

__ 12. Exit the diagnostic utility.

» Press **F10** or **<Esc-0>**

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

- For a local server, when you see `halt completed` power off and back on from the front panel.
- For a remote server, when you see `halt completed`.
  - Close the virtual console window.
  - The HMC window should show a partition state of **shutting down** and eventually **not activated**.
  - If the state stays at **running**, then use the HMC to shut down the partition. Use the procedure for your HMC provided in Exercise 3, Part 3.
  - When the HMC shows your LPAR state as **not activated**, activate the partition to a multi-user mode (using the normal bootlist).
  - If not using HMCv7, you may need to also close the virtual terminal from the HMC window in order to start it again later; right-click your partition and select **Close Terminal Connection** and confirm when prompted.

__ 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?

» **# /usr/lpp/diagnostics/bin/diagrpt -r | more**

» **# /usr/lpp/diagnostics/bin/diagrpt -a | more**

End of exercise

Exercise 11.System Dump

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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`) _____

» Use the command **`sysdumpdev -l`**. Note that, if your system is running AIX 5L V5.3 or later, the value shown for `dump compression` should be `ON`. (This is the default for AIX 5L V5.3 or later. In AIX 6.1, this cannot be changed.)

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

_____

» # **`sysdumpdev -e`**

» On a system with approximately 1 GB of memory that was used in testing this exercise, the value obtained was approximately 157 MB.

__ 3. Verify that the dump `copy directory` is large enough to hold the dump size reported on the previous command.

» # **`df -m`**

You could also use the command **`/usr/lib/ras/dumpcheck -p`** to check if the size of the copy directory is large enough. If no message is sent to **stdout**, then the size is sufficient.

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.

» # **chfs -a size=+##M /var**
where ## represents the number of megabytes that /**var** must be increased by to hold the dump.

On a system with approximately 1 GB of memory that was used in testing this exercise, the value used for ## was 168.

Note that **chfs** will now accept **M** (Megabytes) and **G** (Gigabytes) unit identifiers for file system size specifications. In our example, the command **chfs -a size=+168M /var** can be used to indicate that the size of /**var** should be increased by 168 MB.

 # /usr/lib/ras/dumpcheck -p

__ 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.)

» # **chdev -l sys0 -a autorestart=true**

__ 5. Use the command **sysdumpstart -p** to start a dump to the primary dump device.

» # **sysdumpstart -p**

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.

_____

» 0c2

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

» # **sysdumpdev -L**

Sample output is given below:

```
Device name:            /dev/hd6
Major device number:    10
Minor device number:    2
Size:                   74921984 bytes
Uncompressed Size:      634635938 bytes
Date/Time:              Tue Nov 27 15:23:04 2007
Dump status:            0
Type of dump:           traditional
dump completed successfully
```

Dump copy filename: /var/adm/ras/vmcore.0.BZ (if it's the first dump)

Note that the value shown for `Uncompressed Size` is much larger than the value shown for `Size`. Also note that the .BZ extension means that the compressed dump cannot be uncompressed using the **uncompress** command.

__ 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:

_____

_____

» **# chfs -a size=+##M /var**

The additional number of megabytes should be greater than the *Uncompressed Size* you have recorded.

» # **dmpuncompress -p /var/adm/ras/vmcore.0.BZ** (if it's the first dump)
kdb /var/adm/ras/vmcore.0

__ 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.

» Sample output is shown below:

```
(0)> stat
SYSTEM_CONFIGURATION:
CHRP_SMP_PCI POWER_PC POWER_6 machine with 2 available CPU(s)
(64-bit registers)

SYSTEM STATUS:
sysname... AIX
nodename.. rt1s3vlp2
release... 1
version... 6
build date Oct  5 2007
build time 21:34:35
label..... 0741A_610
machine... 00C35BA04C00
nid....... C35BA04C
```

```
time of crash: Tue Nov 27 15:23:04 2007
age of system: 6 day, 4 hr., 35 min., 32 sec.
xmalloc debug: enabled
FRRs active... 0
FRRs started.. 0
(0)> status
CPU     TID  TSLOT     PID  PSLOT  PROC_NAME
  0   10E027    270   480B2     72  sysdumpstart
  1    12025     18    D01A     13  wait
  2-127   Disabled

(0)> q
```

__ 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.)

» **# rm /var/adm/ras/vmcore.0** (if it's the first dump)

__ 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:

_____

_____

» # **df -m**

If necessary, you can increase the size of /**tmp** using a command similar to the following:

chfs -a size=+64000 /tmp

(The unit used for **size** is specified, by default, in terms of 512-byte units/blocks.)

__ 11. Run the command **snap -a**. (Note that this command will take approximately 10 minutes to run.)

» # **snap -a**

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.

» # **cd /tmp/ibmsupt**
  # **ls**

```
# cd <subdirectory>
# view *.snap
```

End of exercise

Exercise 12.Basic Performance Commands

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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.

»**# alias top="ps aux | tail +2 | sort -nr -k 3,3"**

» Check how ENV is defined. If ENV=$HOME/.kshrc, add the alias to the file **$HOME/.kshrc**.

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

»**# top**

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

»**# /home/workshop/ex12_prog1 &**

» **# ps -elf | grep ex12_prog1**
or
 # ps -e -F "pid,pri,ni,args" | grep ex12_prog1

Priority: ___68_____

Nice value: __24_____

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

»**# kill %1**

»**# nice -n 15 /home/workshop/ex12_prog1 &**

» **# ps -elf | grep ex12_prog1**
or
 # ps -e -F "pid,pri,ni,args" | grep ex12_prog1

» Again write down the nice value and priority that have been assigned to the process:

Priority: ____98_____

Nice Value: ___39_____

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

» **# renice -n -10 3688** (use your corresponding PID)

Check that the priority has been increased.

» **# ps -elf | grep ex12_prog1**
or
 # ps -e -F "pid,pri,ni,args" | grep ex12_prog1

Priority: ____78_____

Nice Value: ___29_____

__ 6. Stop the program **ex12_prog1**.

»**# kill %2 (or whatever job number was assigned to the background job)**

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.

» **# /home/workshop/ex12_cpu &**

» **# sar -u 2 5**

From the output, what can you conclude?

» After starting **ex12_cpu**, one CPU is active the whole time. The program is very CPU-intensive, and causes the system to be CPU bound.

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

» **# ps -elf | grep ex12_cpu**
or
 # ps -e -F "pid,pri,ni,args" | grep ex12_cpu

» The priority is very low (the PRI value is high). As the process consumes a lot of CPU time, the system protects itself by calculating a low priority to the process.

__ 9. Stop the program **ex12_cpu**.

» **# kill %1**

__ 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.

```
»# /home/workshop/ex12_io -c -w -t 120 &
»# iostat 2 5
```

From the output, what can you conclude?

» After starting **ex12_io**, the disk activity is relatively high. Depending on the system you work on, the CPU has to wait for outstanding I/Os. In this case your system is I/O bound.

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

```
»# kill %1
```

__ 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.

```
»# /home/workshop/ex12_memory &
»# vmstat 5
```

From the output, what can you conclude?

» After starting **ex12_memory** the system begins paging. If paging takes place the system approaches its limits, because the real memory is not sufficient.

If you do not see any increase in the pi and po values (they stay at zero), use the **rmss** command to simulate a smaller memory size. **rmss** is a utility that simulates a system with a smaller amount of memory than it really has. To see the current amount of memory the system has, use the **rmss -p** command. To change the amount of memory the system thinks it has, use the command **rmss -c *simulated-memory-size*.** Once you are finished, be sure to set the memory back to its original size with the **rmss -r** command.

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:

```
»# vi /etc/qconfig

        ksh:
            device = kshdev
            discipline = fcfs

        kshdev:
            backend = /usr/bin/ksh
```

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

```
»# qadm -D ksh
```

or

```
»# disable ksh
```

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

```
»# qprt -P ksh /home/workshop/ex12_job
```

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

```
»# lpstat
```

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

```
»# qadm -U ksh
```

or

```
»#enable ksh
```

What happens?

» The program **/home/workshop/ex12_job** will be executed.

End of exercise

Exercise 13.Performance Diagnostic Tool

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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

» `# lslpp -L bos.perf.diag_tool`

» `# /usr/sbin/perf/diag_tool/pdt_config`

» Select Item number **4) modify/enable PDT collection** and then item number **7) exit pdt_config**.

__ 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.

»`# su adm`

»`$ date`

» Change the entries for `/usr/sbin/perf/diag_tool/Driver_ daily` and `/usr/sbin/perf/diag_tool/Driver_ daily2` so that the collector step will run 10 minutes from now and reporters step will run 15 minutes from now:

```
$ crontab -e
```

(If you have problems here, check your `EDITOR` variable)

```
10 11 * * * /usr/sbin/perf/diag_tool/Driver_ daily
15 11 * * * /usr/sbin/perf/diag_tool/Driver_ daily2
```

(This example assumes the current time is 11:00 am. Modify the hour and minutes accordingly, depending on your current system time.)

__ 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.

»**# /home/workshop/ex13_perf**

__ 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.

»**# pg /var/perf/tmp/PDT_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.

»**# /usr/sbin/perf/diag_tool/pdt_config**

» From the menu, select **2) modify/enable PDT reporting**.
When prompted for recipient, enter new user to send reports.
When prompted for severity level, enter **2**.
Then, exit by selecting **7) exit pdt_config**.

__ 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.

»**# /usr/sbin/perf/diag_tool/pdt_report 2 | more**

»**# /usr/sbin/perf/diag_tool/pdt_report 3 | more**

End of exercise

Exercise 14.Authentication and ACLs

(with hints)

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 with Hints

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.

The output shown in the answers is an example. Your output and answers based on the output may be different.

All hints are marked with a >> sign.

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:
```

» # **vi /etc/security/login.cfg**

In the `default` stanza add the herald information.

```
default:
...
herald = "\n\n\n\n* Restricted Access *\n\rAuthorized Users
Only\n\rLogin: "
```

Note: Do not use the **<ENTER>** key in your herald string. You must use **\n** for new lines and **\r** for return. When you are editing the line, if you reach the end of the line, let it wrap to the next line. Again, do *not* use the **<ENTER>** key.

__ 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.

» # **who /etc/security/failedlogin | more**

__ 5. Review the **su** activity on your machine.

» # **more /var/adm/sulog**

__ 6. Review all the logins on your system.

» # **last | more**

-OR-

who /var/adm/wtmp | more

__ 7. Review all root logins on your system.

» # **last root**

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?

_____

_____

» The statement **exit 0** will be executed in the case of a valid login.

The statement **exit 1** will be executed in the case of an invalid login, that is, if a user tries to open a second login session.

Check that **ex14_login** is executable.

» The expected result is that this script is executable. If it is not, use the **chmod** command to make it 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:

_____

_____

_____

» Add the following stanza:

```
COUNT:
program = /home/workshop/ex14_login
```

__ 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.

»# **grep -p team01 /etc/passwd**

 # grep -p team01 /etc/security/passwd

» If necessary, properly define the team01 user:

```
mkuser team01
# passwd team01
Changing password for "team01"
team01's New password: team01
Re-enter team01's new password: team01
# pwdadm -c team01
```

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

_____

_____

» Add the following stanza:

```
team01:
auth1 = SYSTEM,COUNT
```

__ 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**.

_____

» # **vi /etc/security/user**

Remove the attribute `auth1` in the **team01** stanza.

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.

_____

_____

_____

_____

» $ **su**
mkuser michael
mkuser sarah
passwd michael
passwd sarah

__ 15. Return to your **team01** user ID.

» # **exit**

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

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

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

_____

_____

» $ **vi sample**
(Insert above statements.)
$ **chmod 700 sample**
$ **sample**

__ 17. Use the **login** command to log-in over yourself as **michael**. Change directory to **/home/team01**.

»# login

Log in as **michael**.

$ **cd /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.

_____

_____

» $ **cat sample**
cat: cannot open sample
$ **sample**
ksh: sample: cannot execute

__ 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).

_____

_____

» $ **vi .profile**

Insert the following line in your **.profile**:
export EDITOR=/usr/bin/vi

$ **login**

Log in again as **team01**.

__ 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.

_____

_____

_____

_____

» $ **acledit sample**

Replace the word `disabled` with `enabled` under the `extended permissions` line.

Add the following two lines to the file, under the word `enabled`:

```
permit rwx u:michael
permit r-x u:sarah
```

```
:wq
```

```
Should the modified ACL be applied? (yes) or (no) yes
```

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

_____

_____

» $ **ls -e sample**
$ **exit**

__ 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.

_____

_____

» $ **cd /home/team01**
$ **vi sample**
...
date
$ **sample**

__ 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?

_____

» No. The user **sarah** has read and execute, but no write permission.

__ 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**.

_____

_____

_____

_____

» $ **acledit sample**

Change the base permissions for group (staff) from --- to **r-x**

Delete the two permit lines under extended permissions.

Add the following line:
deny rwx u: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?

_____

_____

» $ **groups**
staff
$ **cd /home/team01**
$ **sample**
ksh: sample: cannot execute

==> The keyword deny denies the access to the file.

__ 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**.

_____

_____

» $ **vi sample2**
(Type in a couple of lines.)
$ **aclget 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**.

_____

_____

» $ **aclget sample | aclput sample2**
$ **aclget sample2**

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

_____

_____

» $ **chmod 700 sample**
$ **aclget sample**
The extended permissions have been disabled.

End of exercise

Appendix A. Auditing

(with hints)

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
```

_____

_____

Hint: You must change the `start` and `bin` stanzas.

__ 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.

_____

_____

Use the **`auditpr`** command and query the *trail*.

Stream mode auditing

__ 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/str...

__ 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