Linux System Administration I: Implementation

(Course code LX03)

Student Exercises

ERC 7.0

Authorized

IBM Training

Trademarks

IBM® and the IBM logo are registered trademarks of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®	AS/400®	DB2®
GPFS™	PowerPC®	Tivoli®

VMware® and the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the "Marks") of VMware, Inc. in the United States and/or other jurisdictions.

PS/2® is a trademark or registered trademark of Lenovo in the United States, other countries, or both.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft® and Windows® 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.

Other product and service names might be trademarks of IBM or other companies.

March 2011 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® and the IBM logo are registered trademarks of International Business Machines Corporation.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®	AS/400®	DB2®
GPFS™	PowerPC®	Tivoli®

VMware® and the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the "Marks") of VMware, Inc. in the United States and/or other jurisdictions.

PS/2® is a trademark or registered trademark of Lenovo in the United States, other countries, or both.

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Microsoft® and Windows® 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.

Other product and service names might be trademarks of IBM or other companies.

Exercises description

In exercise 1, you install your own Red Hat Enterprise Linux and SuSE Enterprise Server system, on which you perform exercises 2 through 14. Exercises 2 through 14 are designed so that you can practice various system administration tasks.

These exercises teach you how to install and configure Linux. The exercises are largely independent of each other, except that exercise 3 and up all depend on the successful completion of exercise 1.

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

Exercise Instructions — This section contains what it is you are 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 unit Student Notebook, your past experience, and maybe a little intuition.

Exercise Instructions With Hints — This section is an exact duplicate of the Exercise Instructions section except that in addition, specific details and/or hints are provided to help step you through the exercise. A combination of using the Instructions section along with Instructions With Hints section can make for a rewarding combination providing you with no hints when you do not want them and hints when you need them.

Distribution naming conventions

This course deals with multiple distributions of the Linux operating system.

The following acronyms will be used throughout the course.

- Red Hat Enterprise Linux Entry Server
- RHEL: Applies to all versions of Red Hat Enterprise Linux
- RHEL6: Applies to Red Hat Enterprise Linux 6
- rhel60s: Applies to Red Hat Enterprise Linux Server 6
- SUSE Linux Enterprise Server
- SLES: Applies to all versions of SUSE Linux Enterprise Server

- SLES11: Applies to SUSE Linux Enterprise Server 11

Exercise 0. Pretest

What this exercise is about

This exercise provides you to opportunity to take an assessment of your knowledge related to material presented in this course.

What you should be able to do

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

- Rate your knowledge coming into this course

Introduction

In this exercise, you will answer a series of questions related to material that is going to be presented during the course. Once completed, the instructor will review the questions and provide the correct answer for your self-evaluation.

Requirements

- This workbook

Exercise instructions

Preface

- Answer the following series of questions. Once completed, the instructor will provide the correct answers.
- IBM does not track the results of your self-evaluation. It is meant as a learning tool to improve your understanding of the material presented in this course.

Questions

__ 1. True/False: A network install server needs to be a Linux system.

__ 2. Which of the following install methods does not require a network server?

a. NFS

b. SMB

c. FTP

d. CD-ROM

__ 3. What are some possible locations where a RHEL/Fedora Kickstart or SLES Linux AutoYaST file can be stored?

__ 4. Name the four steps that form the startup order of a Linux system:

__ 5. How would you select a graphical login screen (**xdm**, **kdm**, or **gdm**)?

__ 6. The RHEL_____ tool provides a menu-based interface for various tools used during a text-based installation.

__ 7. True/False: RHEL provide separate tools that start with **system-config** to administrate the system with a GUI interface.

__ 8. SUSE provides a tool called _____ as a GUI interface/text menu tool to be used for various system administration tasks.

__ 9. What is the default port number to connect with the Webmin administration tool using a Web browser?

__ 10. Which basic modes of operation does **rpm** have?

__ 11. Which command can I use to verify that the permissions of `/etc/sendmail.cf` are still correct?

__ 12. From the list provided, check all software maintenance operations that the **rpm** command provides:

____ Installation of a RPM package

_____ Installation of a tar ball archive

_____ Removal of seldom used packages

_____ Updating a package

_____ Verification of package installation

__ 13. What directory structure gives you access to the kernel runtime parameters?

__ 14. True/False: 2.6 kernel modules end with a .o suffix.

__ 15. True/False: The command make oldconfig will generate a new .config file.

__ 16. The command _____ loads a module and modules that it depends on.

__ 17. True/False: RAID volumes can be used as physical volumes in an LVM setup.

__ 18. Mirroring is offered by RAID level:

a. Linear

b. Zero

c. One

d. Four

e. Five

__ 19. What command is used to create a RAM disk?

__ 20. Assuming a blocksize of 1024, how many i-nodes and data blocks do you need for a file on an ext2 file system?

a. With size 0?

b. b. With size 1?

c. With size 2000?

d. With size 12289 (12 K+1)?

__ 21. What are the two methods of copying a file to a (not yet mounted) MS-DOS floppy?

__ 22. What files are important with respect to quotas?

__ 23. Which file in `/proc` shows current system-wide memory performance statistics?

__ 24. List two commands that provide system memory status:

__ 25. What is the difference between a paging partition and a paging file? Which is more efficient?

__ 26. What does the command **top** do?

__ 27. What command can be used to look at your crontab jobs?

__ 28. How do you regulate the use of the **crond** and **atd** daemon?

__ 29. What is the difference between A and B?

a. **`find /home/francis -print cpio -ov >/dev/rmt0`**

b. `find . -print cpio -ov >/dev/rmt0`

__ 30. Which one of the following commands supports multilevel incremental backups?

a. tar

b. dump

c. cpio

__ 31. True/False: An incremental backup will always back up the operating system files.

__ 32. True/False: It is not necessary to use the dash (**-**) with the option in the **tar** command.

__ 33. What is a User Private Group?

a. A group for users who need privacy

b. A group which has the same name as the user; this user has this group as its primary group

c. A group which is used for sharing files between the members of this group

d. The “staff” group

__ 34. Where are the passwords of users stored?

__ 35. True/False: The user “root” can log in anywhere, anytime.

__ 36. True/False: PAM is the subsystem responsible for user authentication.

__ 37. The _____receives all logging requests and forwards it to the right destination, depending on priority and facility.

__ 38. What does the **logger** command do?

__ 39. The **logrotate** command:

a. Creates new log files

b. Rotates and cleans up log files

c. Deletes log files

__ 40. True/False: One of the advantages of queues is that each user can have a different default queue set up for them.

__ 41. Can any user bring the print queue down? Name a few people who can.

__ 42. True/False: Once the printer is down, no more jobs can be submitted to the queue.

__ 43. Can users delete all their print jobs in a specific queue? If so, how?

END OF EXERCISE

Exercise review/wrapup

This exercise provided the students a chance to check their understanding of material presented in this course.

Exercise 1. Advanced Linux installation

What this exercise is about

This exercise lets you install Linux through the network.

What you should be able to do

After completing this exercise, you should have experience with installing Linux through the network.

Introduction

You will install Linux on your classroom workstation in this exercise. This system load will then be used for all following exercises, so please follow the instructions as given (to assure the proper setup).

Verify with your instructor what Linux distribution should be loaded.

Requirements

To complete this exercise, you will need the following:

- A set of network install diskettes for your distribution
- The IP address and NFS export name of the install server

Your instructor will provide you with this.

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended but not required.
- Always replace the variable *yourname* in exercise steps with your name.

Exercise instructions (bootp)

If your system is set up for Bootp (Network Boot), your machine must be able to boot through PC LAN. You might have to interrupt the normal boot process by pressing **1** (maybe **F12** or **F1**, depending on the BIOS) and enter the SMS panel. Ask your instructors for information regarding the network install server.

You will then see the IBM Linux Netboot menu. You have to make a choice within 60 seconds before the system will choose **local boot** by default. If you type a number, your install will be a network install from 10.0.0.1:/export/<*distro*>. If you type the distribution name, your installation will allow you to enter the Network Install Server IP Address and Server Directory name.

Note: To turn off the 60 second timer, press the **Spacebar**.

Exercise instructions (RHEL)

Installing through the network

__ 1. If you are booting through bootp, wait for the boot menu to display, then enter <*distro*> `askmethod`. Skip to Step 4.

__ 2. Boot/reboot your PC with the CD #1. You might have to press **1** (or some other F-key) to access the alternate boot device.

__ 3. If you are installing RHEL, type `linux askmethod`

__ 4. Choose the language for the installation process and choose **OK**.

__ 5. Choose your keyboard model, and choose **OK**.

__ 6. If you are using RHEL, select **NFS image** as install media.

__ 7. Depending on the classroom network setup, you might be able to use Dynamic IP, or need to use Static IP addressing. Your instructor can tell you which choice to make, and what IP addresses and hostnames to use. Disable **IPv6 support**. Choose **OK**.

__ 8. Enter the IP address of the NFS and the directory that is exported and contains your distribution data.

__ 9. The installation process now contacts the NFS server and starts the X-based installer (Anaconda). This takes a minute or so.

__ 10. You will see the RHEL welcome screen. Click **Next**. RHEL6 will ask for an installation number. Select **Skip entering Installation Number**, then **OK**, then **Skip** on the next dialog box.

__ 11. If you are prompted to initialize the drive, select **Yes**. If a previous installation of Linux was found, you might be prompted to upgrade. Override this option by selecting the **Install** option. Click **Next** to continue.

__ 12. Click the **Remove Linux partition...** box and choose **Create custom layout**, then **Next**.

__ 13. You are now looking at your partition table. If Windows is installed on this PC, you might be able to create a “dual boot” process. Highlight and click **Delete** on all non-Windows type partitions.

__ 14. You can now create your Linux partitions by clicking **New** to begin adding these partitions:

- A / (root) partition of 1500 MB
- A /usr partition of 6000 MB
- A /boot partition of 100 MB
- A /home partition of 1250 MB
- A /tmp partition of 1250 MB
- A /var partition of 1500 MB
- A swap partition of 512 MB

All partitions, except the swap partition, should be formatted as ext3.

Do not make partitions larger than the size indicated since you need free, unpartitioned space in other exercises.

Click **Next**. If you get a popup window that some values are less than what your distribution suggests, ignore the problem and click **Next**.

__ 15. The next screen allows you to set up your boot loader. Accept all defaults and click **Next**.

__ 16. Check to see if the network parameters are correct. Then click **Next**.

__ 17. Now select your Time Zone, and uncheck the **UTC** box, then click **Next**.

__ 18. For convenience in the class, set the root password to `ibmlnx`. Click **Next**.

__ 19. On the Software screen, select the **Customize now** radio button, then click **Next**. On the next screen, make the following selections:

Desktop Environment: KDE (K Desktop Environment)

Development: Development Tools

KDE Software Development

X Software Development

Servers: Network Servers

Printing Support

Server Configuration Tools

Base System: System Tools

X Window System

Click **Next.**

__ 20. Note the location of the log file, `/root/install.log`, and click **Next**.

__ 21. The installation process now formats and installs your selections. This might take anywhere from one to 15 minutes, depending on the number of packages to install, the network bandwidth available and the speed of the computer.

While installing, you can see what is going on in detail by switching to the third virtual terminal with **Ctrl+Alt+F3**. Switch back with **Alt+F7**. Also, take a look at other virtual screens (1 through 6).

__ 22. After the installation finishes, the system will reboot. On first boot, a Welcome screen will appear, with additional configuration tasks. Answer the questions in the following manner:

Accept the License Agreement

Disable the firewall Disable SELinux RHEL: Disable Kdump

Set the correct date and time

RHEL: Do not register for software updates

Create a user account (use your name)

RHEL6: Do not install any additional CDs

RHEL Kickstart installation

__ 23. After reboot, log in as root and take a look at the `anaconda-ks.cfg` file in your home directory. Do you recognize the choices you made during the installation process?

__ 24. Verify the following is set up in the anaconda-ks.cfg file:

- The installation should be done through NFS from the class server.
- All partitions will be deleted and recreated according to the scheme previously set up. (**Hint**: The partition directives need to be uncommented)
- The following lines are added to the %post section to create users tux1 and tux2:

The `anaconda-ks.cfg` file should now look similar to this:

```
# Kickstart file automatically generated by anaconda.

install
nfs --server=XX.XX.XX.XX --dir=/export/
rhel51s lang en_US.UTF-8
keyboard us
xconfig --resolution 1024x768 --depth 16 --startxonboot --defaultdesk
top=GNOME
network --device eth0 --bootproto dhcp
# network --device eth1 --onboot no --bootproto dhcp
rootpw --iscrypted $1$KpLhOdnJ$Sot2rtAvD5CeGtDLsznNE.
firewall --disabled
authconfig --enableshadow --enablemd5
selinux --disabled
timezone America/New_York
bootloader --location=mbr
# The following is the partition information you requested
# Note that any partitions you deleted are not expressed
# here so unless you clear all partitions first, this is
# not guaranteed to work
clearpart --linux --drives=hda

part /boot --fstype ext3 --size=100
part /usr --fstype ext3 --size=6000
part swap --size=1024
part /var --fstype ext3 --size=1500
part / --fstype ext3 --size=1500
part /tmp --fstype ext3 --size=1250
part /home --fstype ext3 --size=1250

%packages<o:p\
@ admin-tools
@ base
@ base-x
@ core
@ dialup
@ development-libs
@ editors
@ games
@ gnome-desktop
@ gnome-software-development
@ kde-desktop
@ graphical-internet
```



```
@ graphics
@ java
@ legacy-software-development
@ network-server
@ office
@ printing
@ sound-and-video
@ text-internet
@ development-tools
@ x-software-development
bsf
comps-extras
cracklib-
dicts gnome-
mime-data
iso-codes
rmt
tzdat
a

%post
adduser -c "Tux the Penguin 1" tux1
echo penguin | passwd --stdin tux1
adduser -c "Tux the Penguin 2" tux2
```

__ 25. Ask your instructor what to do with the kickstart file. There are two options:

__ a. Put the kickstart file on a blank floppy, which your instructor provides.

__ b. Upload the file, under your own name, to the instructor server.

__ 26. Reboot your system. If the system boots from CD, make sure that you start a kickstart install.

__ 27. When the installation is finished, click **Reboot**.

END OF EXERCISE

Exercise instructions (SUSE Linux Enterprise Server)

Installing SUSE Linux Enterprise Server (SLES) through the network

__ 28. If you are booting through bootp, wait for the boot menu to display, then enter `sles11`. Skip to Step 32.

__ 29. Boot/reboot your PC with the CD #1. You might have to press **F9** (or some other F-key to access the alternate boot device).

__ 30. The system should boot from CD, and you see the SUSE Linux boot menu (there is a 20 second timer, so press the **Spacebar** to stop this timer). Change the video setting by pressing **F3**, and use **F4** to enter the type of boot, which in our exercise will be NFS. Enter your NFS server (xx.xx.xx.xx) and directory (`/export/sles11`), and then press **Enter**.

__ 31. Choose the language for the installation process and click **Next**.

__ 32. Answer **Yes** to the License Agreement, and click **Next**.

Note: If an existing instance of SLES is already installed, you will be prompted to either perform a New Installation or an Update. If this occurs, click **New Installation**, and select **Next**.

__ 33. Select the proper Time Zone and Hardware Clock setting, and click **Next.**

__ 34. Under the Expert tab, review the installation settings, and make the following changes (you will need to select **Create Custom Partition Setup** to accomplish this step):

- Partitioning: Partition your system so that you get the following partitions:
  - A / (root) partition of 1500 MB
  - A /boot partition of 100 MB
  - A /usr partition of 6 GB
  - A /var partition of 1500 MB
  - A /opt partition of 1 GB
  - A /home partition of 1250 MB
  - A /tmp partition of 1250 MB
  - A swap partition of 512 MB

All logical partitions should be formatted as ext3 except for the swap partition.

Do not make partitions larger than indicated since we will need free, unpartitioned space in the rest of this course.

When finished, click **Finish**.

__ 35. Now click **Software** and check the boxes next to **KDE Desktop, File Server, Web and Lamp**, and **C/C++ Compiler**. Click **Accept** and **Accept** again. If you see another pop-up regarding dependencies, click **Continue**, then **Install**.

__ 36. From the Confirm Installation menu, select **Install** to proceed.

Note: The installation will proceed at this point. Monitor your screen for any messages that require your assistance. You should see on the right side of your

screen a list of tasks to be completed. The task currently operating is noted with an arrow.

Remove the CD-ROM at this time.

__ 37. SLES automatically reboots during the installation. This is normal and the installation process will continue automatically. If any pop-up windows appear, choose **Continue**.

__ 38. Enter the password for the root user. For convenience in class, use `ibmlnx`.

__ 39. Your instructor will indicate whether to use automatic network configuration through DHCP or manual configuration and provide the appropriate settings. For manual configuration, you will need a hostname, IP address, netmask, default gateway, and name server address.

__ 40. Under the Network Configuration menu, click **Enabled** to disable the Firewall, then click **Disable IPv6**, then **Next**.

Once complete, click **Next**.

__ 41. Answer **No, Skip This Test** under the Test Internet Connection menu. Click **Next** to proceed.

__ 42. On the CA Management screen, change setting to **Skip Configuration** option. Click **Next** to proceed.

__ 43. User Authentication should remain at the default setting, Local (/etc/passwd). Click **Next** to proceed.

__ 44. Configure a user account for yourself, and then click **Next**.

__ 45. Read the release notes if you wish, and then click **Next**.

__ 46. Check the hardware configuration. Pay particular attention to getting a correct X setup, since you will need X in the next few steps. When done, click **Next**.

__ 47. Make sure the box next to **Clone This System for Autoyast** is marked, then click **Finish** to complete the installation.

SLES AutoYaST installation

__ 48. Log in as root.

__ 49. Start the Autoinstallation module in YaST2, and load in the `autoinst.xml` file.

__ 50. Go through all menus to create a configuration file that is more-or-less identical to the system you are working on now.

Make sure under **Hardware -> Partitions -> Configure** that you use the partitioning scheme as suggested earlier. Note that the size you specify is in bytes, so you need to specify 150M to get a 150 MB partition.

Follow **Software -> Package Selection -> Configure -> Filter -> Patterns** and add the following package categories:

- GNOME Desktop system
- KDE Desktop system
- File Server
- C/C++ Compiler and Tools

Make sure the root password is set to `ibmlnx`. (**Note**: This is hidden far within the menu structure. Go to **Security and Users -> User Management**. Click **Configure** and **Set Filter** to **System Users**. You can now click **Edit** to modify the root account.)

Add two user accounts, tux1 and tux2, with full names `Tux the Penguin (1)` and `Tux the Penguin (2)`, and passwords penguin.

Last, make sure that **confirm installation** (within **System** -> **General Options**) is set to **No**. Please double-check *all* settings before you save the file.

__ 51. Save the autoyast configuration file using your own surname as file name, and make sure to add the extension xml. Then exit yast.

__ 52. Take a look at the autoyast configuration file. Do you recognize the configuration choices you made?

__ 53. Upload the file to the instructor server.

__ 54. Reboot your system. When the system boots from CD, make sure you start an autoyast install.

END OF EXERCISE

Exercise 2. Startup and shutdown

What this exercise is about

This exercise illustrates the startup and shutdown process of a Linux system.

What you should be able to do

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

- Choose between a graphical and a text-based login screen by changing the run levels of a system
- Boot a Linux system in single-user mode
- Use run level editors

Introduction

In this exercise, you will modify the run level configuration of the system.

Requirements

- This workbook
- A workstation with Fedora, RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

The GRUB boot loader (x86)

__ 1. Log in as root.

__ 2. Change the `/boot/grub/menu.lst` configuration file so that a password is required when you try to alter the various boot options. Make sure the password is encrypted.

For convenience in class, use the password ibmlnx.

Note: When starting grub, it starts probing devices to guess BIOS drives. This might take a long time - up to several minutes, depending on your hardware. Wait until this is done before typing the **md5crypt** command within grub.

__ 3. Reboot your machine. Use the shutdown command to perform the reboot. Then try to alter the GRUB boot sequence without and with supplying the password.

The YABOOT boot loader (ppc)

__ 4. Log in as root.

__ 5. Change the `/etc/yaboot.conf` configuration file so that a password is required when you try to alter the various boot options. Make sure the password is encrypted.

For convenience in class, use the password ibmlnx.

__ 6. Reboot your machine. Use the shutdown command to perform the reboot. Then try to proceed through the boot sequence without and with supplying the password.

Retrieving kernel messages

The messages which are displayed by the kernel can be a helpful tool in problem determination. That is why you might want to retrieve them, even after your system has booted.

__ 7. Log in as root.

__ 8. View the kernel messages and the log file `/var/log/messages`.

Setting the default runlevel

The default runlevel can be altered to configure your system for your situation. In this exercise, you set the default to 3 so that you do *not* get a graphical login prompt. Note, however, that depending on your hardware, kickstart or autoyast might not have configured X correctly. Thus, you need to test and, if needed, configure X first.

__ 9. Verify the current run level of the system. The current run level is _____.

__ 10. If your current runlevel is 5, change it to 3.

__ 11. Edit the file `/etc/inittab` so that the default runlevel will be 5. (Your default might already be 5. That is okay.)

__ 12. Depending on your hardware, kickstart (RHEL/Fedora) or autoyast (SLES) might not have configured X correctly. Start X with the X command. (This only starts the XFree86 server.) If X comes up correctly (black or grey background with an X-shaped mouse cursor, nothing more, nothing less), stop the X server with **Ctrl+Alt+Backspace** and continue with the next step. If X does not come up correctly, run system-config-display (RHEL/Fedora) or sax2 (SLES) to configure X. You should be in runlevel 3 to do this.

__ 13. Reboot your system, this time using **Ctrl+Alt+Delete**.

__ 14. When GRUB appears, do nothing. After a few seconds, GRUB should automatically boot your default operating system.

__ 15. When the graphical login prompt appears, switch to the first virtual terminal. Then switch back to **VT 7**.

__ 16. Log in as root, then start a terminal screen (from menus).

__ 17. Verify the current run level of the system. The current run level is _____.

Configuring services

__ 18. Make a long list of files in the directories `/etc/rc.d/rc3.d`, `/etc/rc.d/rc4.d`, `/etc/rc.d/rc5.d`, and `/etc/rc.d/init.d` (RHEL/Fedora) or `/etc/init.d` (SLES).

__ 19. Create a list of services with **chkconfig**, and check its output with the output from the previous commands.

__ 20. Verify that the **portmap** service is enabled in your current runlevel. Disable this service, and then check the symbolic links in `/etc/rc.d/rc5.d` again.

__ 21. Check whether the **portmap** daemon is currently running. Then reboot the system.

__ 22. Log in and check whether the **portmap** daemon is running now.

__ 23. Enable the **portmap** service again, and start the service manually.

Using single-user mode

Single-user mode is very convenient for system maintenance.

__ 24. Log out, reboot the system with **Ctrl+Alt+Delete**, and boot in single-user mode.

__ 25. Look at the list of running processes.

__ 26. Reboot the system.

END OF EXERCISE

Exercise 3. System administration tools

What this exercise is about

This exercise gives you some practical experience in using the system administration tool on your distribution.

What you should be able to do

At the end of the lab, you should be able to use and configure the system administration tool on the distribution used.

Introduction

In this exercise, you will utilize various system administration tools found on the distribution installed on your system.

Requirements

- This workbook
- A workstation with RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.
- This exercise also includes an optional exercise that deals with the installation of Webmin from an installation source.

Working with various system administration tools: SLES11

__ 1. Login as root with the password of ibmlnx.

__ 2. From a terminal window, issue the **yast** command to bring up the ncurses menu.

```
# yast
```

__ 3. Navigate to the Software configuration module and select the **Installation Source** sub-menu. List the first two installation sources available to your system:

Name:_____ URL: _____

Name: _____ URL: _____

__ 4. Exit the yast interface.

__ 5. List the available fast paths that can be used with yast/yast2, by issuing the following command:

```
# yast -l
```

__ 6. Using the fast path of **host** with the **yast** command, create a /etc/hosts entry with the following values:

```
# yast2 host
IP Address: xx.xx.xx.xx
Hostname: service.example.com
Alias: service
```

__ 7. Exit the yast interface.

__ 8. Verify the entry was created:

```
# grep service /etc/hosts
```


__ 9. From a terminal window, issue the **yast2** command to bring up GUI menu.

```
# yast2
```

__ 10. Navigate to the **System** configuration module and select the **System Services (Runlevel)** sub-menu. Click the **Expert Mode** radio button in the top portion of the GUI menu. In what run-level is the **acpid** service enabled? _____

__ 11. Enable the **apache2** service to be enabled in run-level 5 only:

```
# chkconfig apache2 off
# chkconfig apache2 --level 5 on
```

__ 12. In a separate window, verify the **apache2** service was enabled by using the **chkconfig** command:

```
# chkconfig --list apache2
```

__ 13. Search the yast2 interface and find the configuration module that has the sub-menu V**iew System Log**. Which configuration module is it? _____

__ 14. Exit the yast2 interface.

Working with various system administration tools: RHEL

__ 15. Login as root with the password of ibmlnx.

__ 16. From a terminal window, issue the **system-config-packages** command to bring up the GUI menu.

```
# system-config-packages
```

__ 17. Select **Edit -> Repositories** from the menu. Click the **ADD** button. Type the following information into the dialog box:

Name: `RHEL5 Server`

Description: `Red Hat Enterprise Linux 6 Server`

Location: `http://xx.xx.xx.xx/rhel60s/Server`

Click **OK** when finished.

Exit the application and restart it. (Re-run the command `system-config-packages`.)

__ 18. Locate the **Application -> Editors** package group and answer the following questions:

What are the total amount of packages available to be installed? _____

How many of the packages are currently installed on the system? _____

List the package names that are a part of the Editors package group:

Note: Your package list might not match this list exactly.

- _____
- _____
- _____
- _____
- _____

__ 19. Exit the system-config-pages interface.

__ 20. Using the system-config-network interface, click the **Hosts** tab. Create a /etc/hosts entry with the following values:

IP Address: xx.xx.xx.xx

Hostname: service.example.com

Alias: service

__ 21. Exit the system-config-network interface.

__ 22. Verify the entry was created:

```
# grep service /etc/hosts
```

__ 23. According to the system-config-services interface, in what run-level is the **acpid** service enabled? _____

__ 24. Enable the **httpd** service to be enabled in run-level 5 only:

```
# chkconfig httpd off
# chkconfig httpd --level 5 on
```

__ 25. In a separate window, verify the **httpd** service was enabled by using the **chkconfig** command:

```
# chkconfig --list httpd
```

__ 26. Exit the system-config-services interface.

__ 27. Determine which system-config interface should be used to modify group assignments for users. Which system-config interface is it?

Configuring a printer

__ 28. Modify the CUPS configuration file /etc/cups/cupsd.conf so that the directive **FileDevice** is set to allow printing to a file:

```
# vi /etc/cups/cupsd.conf
```

(Change)

```
#FileDevice No
```


(To)

`FileDevice Yes`

__ 29. Start the CUPS printer daemon and make sure it is started on system boot.

__ 30. Start a browser and enter the location http://localhost.631. Log in as root and configure a printer with the following attributes:

- Name: `lx03`
- Location: `Classroom`
- Description: `LX03 printer for this server`
- Device: `AppSocket/HP JetDirect`
- DeviceURI: `file:/tmp/lx03.prn`
- Make: `Generic`
- Model: `Generic text-only printer (en)`

__ 31. Review the contents of the file `/etc/cups/printers.conf` and verify it looks similar to the following output:

```
# cat /etc/cups/printers.conf
# Printer configuration file for CUPS v1.1.22rc1
# Written by cupsd on Wed 21 Jun 2006 11:27:22 AM PDT
<Printer lx03>
Info LX03 printer for this server
Location Classroom
DeviceURI file:/tmp/lx03.prn
State Idle Accepting Yes JobSheets none none QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
```

__ 32. Select the **Printers** menu option. What is the state of the printer lx03?

_____

__ 33. Submit the file `/etc/passwd` to the printer using the **lpr** command and see if the page is printed.

__ 34. Select the **Printers** menu option. For the lx03 printer, click **Reject Jobs**. What is the state of the printer?

_____

__ 35. Submit the file `/etc/passwd` to the printer using the **lpr** command and see if the page is printed. Did the file print?

_____

__ 36. Select the **Printers** menu option. For the lx03 printer, click **Accept Jobs**. What is the state of the printer?

_____

CUPS

__ 37. Modify the CUPS configuration file `/etc/cups/cupsd.conf` so that the directive FileDevice is set to allow printing to a file:

```
# vi /etc/cups/cupsd.conf
```

(Change)

```
#FileDevice No
```

(To)

```
FileDevice Yes
```

__ 38. Start the CUPS printer daemon and make sure it is started on system boot.

__ 39. Start a browser and enter the location **http://localhost:631**. Log in as root and configure a printer with the following attributes:

- Name: `lx03`
- Location: `Classroom`
- Description: `LX03 printer for this server`
- Device: `AppSocket/HP JetDirect`
- DeviceURI: `file:/tmp/lx03.prn`
- Make: `Generic`
- Model: `Generic text-only printer (en)`

__ 40. Review the contents of the file `/etc/cups/printers.conf`, and verify it looks similar to the following output:

```
# cat /etc/cups/printers.conf

# Printer configuration file for CUPS v1.1.22rc1
# Written by cupsd on Wed 21 Jun 2006 11:27:22 AM PDT
<Printer lx03>
Info LX03 printer for this server
Location Classroom
DeviceURI file:/tmp/lx03.prn
State Idle

Accepting Yes
JobSheets none none
QuotaPeriod 0
PageLimit 0
KLimit 0
</Printer>
```

__ 41. Select the **Printers** menu option. What is the state of the printer lx03?

_____

__ 42. Submit the file `/etc/passwd` to the printer using the **lpr** command and see if the page is printed.

__ 43. Select the **Printers** menu option. For the lx03 printer, click **Reject Jobs**. What is the state of the printer?

_____

__ 44. Submit the file `/etc/passwd` to the printer using the **lpr** command and see if the page is printed. Does the file print?

_____

__ 45. Select the **Printers** menu option. For the lx03 printer, click **Accept Jobs**. What is the state of the printer?

_____

END OF EXERCISE

Optional Exercise: Installing Webmin

Preface

- This optional exercise deals with the installation of Webmin from an installation source.

Downloading and installing Webmin

__ 1. Create a directory `/mnt/files`. Mount the directory from the installation server.

Note: If you cannot mount the files directory from the installation server, then check whether the **portmap** daemon is running, and check whether you do not have any **iptables** rules that prevent NFS mounts.

__ 2. Locate and install the **`webmin`** RPM file.

__ 3. Access Webmin by starting a Web browser and connecting to localhost and port 10000 as the directions on the screen imply.

END OF OPTIONAL EXERCISE

Exercise review/wrapup

This exercise covered utilization of various system administration tools found on the distribution installed on your system.

Exercise 4. Packaging tools

What this exercise is about

This exercise gives you some practical experience in using the packaging tool on your distribution.

What you should be able to do

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

- Install, upgrade and deinstall packages
- Query packages
- Verify the authenticity of packages
- Create simple packages

Introduction

In this exercise, you will be performing various package management activities.

Requirements

- This workbook
- A workstation with Fedora, RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

Retrieving information about installed packages

__ 1. Make a list of all packages that are installed on the system.

__ 2. Find out which package installed the `/etc/inittab` file.

__ 3. List the information of that package.

__ 4. List all files in that package.

__ 5. Verify whether all files in that package are still the same. Which file has changed and in what respect? Why?

Installing packages

__ 6. Create a directory `/mnt/server`. Mount the installation directory from the installation server on `/mnt/server`. List all the package files that are available on the install server.

__ 7. Add the public key from the distribution to your keyring. Your public key is located in your installation directory `/mnt/server`.

__ 8. Locate the **mtools** package. Verify that the package **mtools** is not installed. Verify the package on the installation server, and install **mtools** and any package dependencies. Then verify that it installed, and list the files in the package. You can also execute mtools to ensure that it works.

__ 9. Uninstall the **mtools** package.

Using a package management frontend

__ 10. If necessary, start X, then start the RPM frontend tool for your distribution (system-config-packages for Fedora/RHEL, yast2 for SLES). Use the tool's search function to find the package **mtools** and install it.

Creating RPMs

__ 11. Check with your instructor if the file `hello-1.2` files are available somewhere on the network and download it into `/root`. If they are available, then create a directory `/root/hello-1.2` and copy the files into it. If not, you need to create the directory `/root/hello-1.2` and create the following files in it:

hello.c:

```
#include <stdio.h>
main()
{
printf("Hello, World!\n");
 }
```

Makefile:

```
all: hello
hello: hello.c
gcc -o hello hello.c
install: hello
install -d $(DESTDIR)/usr/bin
install -s -m 0755 -o root -g root hello $(DESTDIR)/usr/bin
 clean:
rm -f hello
```

Note: The lines containing commands are indented with a tab, not with spaces!

README:

```
(c) Copyright IBM 2005
This program is licensed under the GPL.
This program prints the text "Hello, World!" on your screen. This is
an excellent way to start your day - some people even consider it
better than getting a random fortune cookie every morning!
To build, simply type make
To install, simply type make install
```

Then, go back to your `/root` directory and make a tar file of all these files, and gzip it.

__ 12. Copy the `hello-1.2.tar.gz` to `/usr/src/redhat/SOURCES` (RHEL/Fedora) or `/usr/src/packages/SOURCES` (SLES)

__ 13. Copy the `hello.spec` file, as shown into the visual, into the correct directory.

The file should look like this:

```
#
# SPEC file for hello world program
#
Summary: Hello, World program
Name: hello
Version: 1.2
Release: 1
License: GPL
Group: Applications/Useless
Source: hello-1.2.tar.gz
Distribution: Useless Linux 1.2
Vendor: IBM IT Education Services
Packager: Wanda C. Results <wcr@qlx.ibm.com>
BuildRoot: /var/tmp/hello-1.2

%description
This program prints the text "Hello, World!" on your screen.
This is an excellent way to start your day - some people even
consider it better than getting a random fortune cookie every
morning!

%prep
%setup

%build
make

%install
make install DESTDIR=${RPM_BUILD_ROOT}

%files
%doc README
/usr/bin/hello
```

__ 14. Start the RPM prep stage and watch the results.

__ 15. Start the RPM build stage and watch the results.

__ 16. Start the RPM install stage and watch the results.

__ 17. Create the binary RPM and watch the results.

__ 18. Create the source RPM and watch the results.

Documenting your changes

__ 19. Test first to see if there is a hello program installed. Since there is not, install your brand new hello RPM file using the rpm command. Test it, uninstall it, and then test it again.

END OF EXERCISE

Exercise 5. X Window System

What this exercise is about

This exercise lets you use the X Window System.

What you should be able to do

After completing this exercise, you should have experience with:

- Configuring X
- Starting and stopping X
- Window managers
- Running X applications over a network

Introduction

In this exercise, you will be performing various activities using the X Window System.

Requirements

- This workbook
- A workstation with RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is not required.

Configuring Xorg

Do this section if your distribution uses Xorg.

__ 1. If you are in a runlevel that automatically starts X, switch to a runlevel that does not start X. If you started X manually, stop it.

__ 2. Make a backup copy of the `xorg.conf` file.

__ 3. Try to configure your `xorg.conf` file using **system-config-display.** If `system-config-display` yields a sufficiently good `xorg.conf` file (test this with **startx**), then make a backup of this file called `xorg.conf.system-config-display`.

__ 4. Try to configure your `xorg.conf` file using **Xorg -configure**. If **Xorg -configure** yields a sufficiently good `xorg.conf` file (test this with **startx**), then make a backup of this file called `/etc/X11/xorg.conf.xorg`.

__ 5. Select the `xorg.conf` file that worked the best for you and rename it to `/etc/X11/xorg.conf`. Then, start X or switch to the runlevel that starts X for you.

Running X applications

In this section, you will see that the X environment is made up of more than one layer. We will begin in runlevel 3, then build an X windows environment one layer at a time.

__ 6. First, you want to be on VT-1 in runlevel 3 with no X server running. Kill any X server that is running with **Ctrl+Alt+Backspace**, and then change to run-level 3.

__ 7. Start the X server and switch to VT-7. You will see a blank screen with a cursor in the shape of an “X,” and that is all you will see. This indicates the X server is running.

__ 8. The X server is not very helpful at this point. If you attempt to run an **xterm**, it will fail because there is no place to “DISPLAY” the application. Log in on VT-2 as root, set the *DISPLAY* variable, and try again, this time running **xterm** in the background.

__ 9. Now you have a term, or window, where you can type commands. However, if you want to open a new application, it covers up the old and you cannot move the display.

Note: Some of these x-applications might not exist on *your* distribution.

__ 10. In order to gain control of the situation, you need a program to manage your X application windows, so let's start an X window manager. Go to VT-2 and start the TWM X Window Manager. Change to VT-7, and now you can move and resize your application windows.

__ 11. This is helpful but does not look like our “normal” X windows environment. You can now start the **startkde** or **gnome-session** program of your choice and watch how the X window environment changes.

__ 12. Now you have a full windows environment. When you are done, you can shut down each application that you started, but you should do that in reverse order - otherwise known as “backing out gracefully” or you can kill the X server which will kill all applications that are dependent upon it. Let's press **Ctrl+Alt+Backspace**. This should take you back to VT-1 automatically. When you get there, check your runlevel.

__ 13. Finally, let's get back to runlevel 5. This will force the session manager to ask you to log back in.

Running applications over a network (optional)

Since applications use a TCP/IP connection to communicate to the X server, you can also run applications from another server.

__ 14. Using the **gdmsetup** tool, modify the configuration to allow remote logins.

Under the Security tab, check the box for **Enable XDMCP**, and uncheck the box for **Deny TCP connections...**

When finished, exit the tool, and restart the login manager.

__ 15. Ask a fellow student to open a terminal window for you.

__ 16. Set the display variable to point to your own screen. The command for that will look like this: **export DISPLAY=*1.2.3.4*:0.0**, where *1.2.3.4* is your own IP address.

__ 17. Now try to start an **xterm**. You should get an error message: `Could not open display`. This is a safety feature of X: it does not automatically accept incoming connections.

__ 18. Go back to your own system and enter the command **xhost +** This will enable incoming connections.

__ 19. Go back to the system of your fellow student and retry the **xterm** command. This time it should succeed.

__ 20. In your graphical screen, you should see a new xterm. Try the **hostname** command in this screen to verify that the **xterm** application is actually running on the other system.

__ 21. Close the xterm and do an **xhost**.

__ 22. Now try to open another **xterm** from your partner's system, but this time use **xauth** authentication.

Note: This only works if your system's hostname has been set properly and can be resolved through DNS.

Running X-sessions over a network

You cannot only run a single application over a network, but you can run your whole X-session over a network as well. In this case, it is not only the application that is running on a remote system, but the window manager as well. In fact, the only program that needs to be running locally is your X server.

__ 23. Using the **gdmsetup** tool, modify the configuration to allow remote logins.

Under the Security tab, check the box for **Enable XDMCP**, and uncheck the box **Deny TCP connections...**

When finished, restart your login manager.

__ 24. Start a second X-server, this time telling X to get its login manager from your partner system. Make sure there is a space before the "**:1**".

__ 25. Stop the second session and start it again, but do an indirect broadcast for a login manager. You should get a chooser which allows you to login to any system running a display manager on the network. Make sure there is a space before the "**:1**".

__ 26. If time permits, perform step 22 and 23 again, but this time use **Xnest** to start an X server within your current X environment. Make sure there is a space before the "**:1**".

Note: You might need to install **Xnest** first.

END OF EXERCISE

Exercise review/wrapup

This exercise covered various basic operations dealing with the X Window System.

Exercise 6. Logging

What this exercise is about

This exercise lets you use the logging daemon.

What you should be able to do

At the end of the lab, you should have experience with:

- The **syslogd** daemon
- The `/etc/syslog.conf` configuration file
- The **logger** command
- The **logrotate** command
- Logfile analysis

Introduction

In this exercise, you will create events that the logging process will capture and then review the log files to identify these events. During this lab exercise, you will use the Linux distribution that you have loaded onto your lab system. Please disregard any exercise steps that reference a different Linux distribution.

Requirements

To complete this exercise, you will need the following:

- This workbook
- A workstation with RHEL or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

Configuring logging

__ 1. Look at the file `/etc/syslog.conf` and try to find out to which files the logging actually goes. Take a look at these files as well.

__ 2. Try to log in as a valid user (for instance, tux1) but with an invalid password, and try to log in as an invalid user. Then view the log entries.

__ 3. Change the syslog configuration file and make sure that everything that has to do with authentication is logged in `/var/log/secure`. Then restart the **syslog** daemon.

__ 4. Again, try to log in as a valid user but with an invalid password, and try to log in as an invalid user. Then view the log entries again.

Using the logger command

__ 5. Look at your `/etc/syslog.conf` file, and try to use the **logger** command so that all the different methods in the `/etc/syslog.conf` file are used once.

__ 6. Do the same, but this time when logged in as tux1.

Using the logrotate command

__ 7. Review the file `/etc/logrotate.conf` and all files in `/etc/logrotate.d`.

__ 8. Make a long, recursive listing of all files in the `/var/log` directory.

__ 9. Manually run the **logrotate** command.

__ 10. Again make a long, recursive listing of all files in the `/var/log` directory. Do you see any differences? Why not?

__ 11. Force a **`logrotate`** with the **`-f`** option.

__ 12. Again make a long, recursive listing of all files in the `/var/log` directory. Do you see any differences? Why?

Using Logwatch (RHEL only)

__ 13. On a Red Hat system, **logwatch** is installed and activated automatically. The results of this analysis will be mailed to root. So check your mail to see the results.

END OF EXERCISE

Exercise review/wrapup

This exercise dealt with the logging function under Linux. It is a relatively short exercise, so the student should not need excessive time to complete.

Exercise 7. Block devices, RAID, and LVM

What this exercise is about

This exercise lets you work with block devices, RAID, and LVM.

What you should be able to do

After completing this exercise, you should have experience with:

- RAM disks
- Software RAID
- Logical Volume Management

Introduction

In this exercise, you will create and modify Linux storage devices, both physical and logical. During this lab exercise, you will use the Linux distribution that you have loaded onto your lab system. Please disregard any exercise steps that reference a different Linux distribution.

Requirements

- This workbook
- A workstation with Fedora, RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

Notice

In this unit you are going to create and use various block devices. Block devices as such are not really useful, though, until you create a file system in there. For this, you are going to use the **`mke2fs`** command to create a file system, and the **`mount`** command to mount the file systems. These commands are covered in-depth in the next unit.

Working with RAM disks (optional)

__ 1. Verify the size of the RAM disks that the kernel uses.

__ 2. Format the ramdisk with **mke2fs**.

__ 3. Create a directory `/mnt/ramdisk`, and mount the RAM disk on that mountpoint.

__ 4. Run the **df** command to see how much space is available on this RAM disk.

__ 5. Copy some files onto this ramdisk, and verify that they are indeed there.

__ 6. Unmount the ramdisk and mount it again. Is the file still there?

__ 7. Reboot your system and then try to mount the RAM disk again. Does this work?

Working with partitions

If you installed your system correctly, then you should have at least 450 MB in unpartitioned space left. We're going to create four partitions in here, 150 MB each, which are going to be used in the subsequent exercises.

__ 8. Start the **fdisk** program and create four additional partitions. The partition size should be 150 MB each. Save the partition table to disk. Write down the partition numbers you created, because you will be using these in the rest of this exercise.

__ 9. Ensure the changes to the partition table have been modified in the running kernel by executing the **partprobe** command.

__ 10. Format the first of the four partitions with the **mke2fs** command. Create a mountpoint `/mnt/partition` and mount this partition on this mountpoint.

__ 11. Run the **df** command to see how much space is available on these partitions.

__ 12. Copy some files onto these partitions and verify that they are indeed there.

__ 13. Reboot your system and then try to mount the partition again. Does this work?

__ 14. Unmount the partition you just mounted.

Preparing for RAID

__ 15. With fdisk, change the partition types of the four partitions to 0xfd (Linux RAID autodetect). Use **partprobe** to re-read the partition table and, if necessary, reboot your system afterwards.

Working with RAID

__ 16. Create a RAID array using three partitions in a linear array. Use the **mdadm** command to initialize the array, and then create a file system on it. Mount the array on a mountpoint called `/mnt/raid`. Run the **df** command to find out how much space this partition has, and view the status of the RAID array. Then, unmount the partition and stop the RAID array.

Notice: In order to watch what happens during the steps below, such as the --**create** or --**manage** options, open another terminal and type:

```
watch -n1 -d cat /proc/mdstat
```

Now adjust your other terminal window so you can type in it and see the results of the watch commands.

__ 17. Create a new RAID array using four partitions in a RAID-0 array. Use the **mdadm** command to initialize the array, and then create a file system on it. Mount the array on a mountpoint called `/mnt/raid`. Run the **df** command to find out how much space this partition has, and view the status of the RAID array. Then, unmount the partition and stop the RAID array.

__ 18. Create a new RAID array using three partitions in a RAID-1 array and one more used as a spare. Use the **mdadm** command to initialize the array, and then create a file system on it. Mount the array on a mountpoint called `/mnt/raid`. Run the **df** command to find out how much space this partition has, and view the status of the RAID array. Set one or two disks in the array to faulty and watch the array recover itself using the spare disk. Then, unmount the partition and stop the RAID array.

__ 19. Create a new RAID array using three partitions in a RAID-5 array and one more used as a spare. Use the **mdadm** command to initialize the array, and then create a file system on it. Mount the array on a mountpoint called `/mnt/raid`. Run the **df** command to find out how much space this partition has, and view the status of the

RAID array. Set one or two disks in the array to faulty and watch the array recover itself using the spare disk. Then, unmount the partition and stop the RAID array.

Working with LVM

__ 20. With fdisk, change the partition types of the four partitions to 0x8e (Linux LVM). Try using partprobe and, if necessary, reboot your system afterwards.

__ 21. Initialize each of the four partitions that you used earlier as physical volumes. If you get the error message `/etc/lvmtab` does not exist, then run the **vgscan** command to create it.

__ 22. Create a volume group vg00 consisting of only the first two LVM partitions. Use a PE size of 4 MB.

__ 23. Create a logical volume lv00 of 50 MB in the volume group vg00.

__ 24. Format the logical volume with the **mke2fs** command. Create a mount point `/mnt/lv00` and mount the logical volume. Then, run the **df** command to see how much space is available.

__ 25. Use the various LVM commands to retrieve information about the physical volumes, the volume group, and the logical volumes.

__ 26. Take a look at the files that were created in `/etc/lvm`. Can you read these files?

__ 27. Add the third and fourth LVM partitions to the volume group vg00, and migrate all data onto these physical volumes. Then, reduce the volume group so that the volume group only contains the third and fourth partition. Do you need to unmount the `/dev/vg00/lv00` logical volume first?

__ 28. Add the first two LVM partitions back to your volume group, and create three more logical volumes in this volume group, called lv01 through lv03. Each logical volume needs to be 50 MB as well. We will need these in the next exercise.

__ 29. Take a look at the output of the **pvscan** command. Does this seem helpful?

END OF EXERCISE

Exercise 8. File systems

What this exercise is about

This exercise lets you work with file systems.

What you should be able to do

After completing this exercise, you should have experience with:

- Creating, mounting, and unmounting Linux file systems
- Mounting and unmounting non-Linux file systems
- Automatically mounting file systems at system startup
- Configuring user quota

Introduction

In this exercise, you will create and modify Linux file systems. During this lab exercise, you will use the Linux distribution that you have loaded onto your lab system. Please disregard any exercise steps that reference a different Linux distribution.

Requirements

- This workbook
- A workstation with Fedora, RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

Creating Linux file systems

You now create various file systems in the logical volumes we made earlier. Note that /dev/vg00/lv00 already contains an ext2 file system.

__ 1. Create an ext3 file system in `/dev/vg00/lv01`.

__ 2. Create a mount point, `/mnt/lv01`, for this file system.

__ 3. Mount the file system on this mount point.

__ 4. Issue the **mount** and **df** commands to see the mounted file systems. Can you see how much space the journals are taking? Also, list the root directory of the mounted file systems. Are the journals visible?

__ 5. Create some files and/or directories on these mounted file systems.

Automatically mounting file systems at system startup

File systems that you want to mount automatically at system startup should be listed in `/etc/fstab`.

__ 6. Edit the `/etc/fstab` file to include all the previously created file systems.

__ 7. Issue the **sync** command to ensure that all data is written to disk. Wait five seconds, and then turn off your computer with the power off button. (Do *not* do a proper shutdown!) Turn on your computer and watch the boot messages. Which file system was checked the fastest?

Changing file system size

__ 8. Using **lvextend** and **resize2fs**, increase the size of each file system in vg00 to 100 MB. When complete, verify the new file system size using **df**.

Note: The ext2 file system does not support online resizing, so it will have to be unmounted. The ext3 file system can be resized online (while mounted).

Working with files with multiple names

As seen in the lecture, a file can have multiple names, as long as each name refers to the same i-node. An example of this situation is `/bin/gzip`, `/bin/gunzip`, and `/bin/zcat`.

__ 9. Verify that `/bin/gzip`, `/bin/gunzip`, and `/bin/zcat` indeed have the same i-node number.

__ 10. Go to `/root` and create a file named one, using **vi**. Put some text in the file, so that you can identify that file for yourself.

__ 11. Verify the file name and the contents.

__ 12. Now create a second link to the file (a second file name) named two. Verify the file name and the contents.

__ 13. Delete the file one.

__ 14. Verify the file name and contents of two. Note that the link count has dropped.

__ 15. Create a third link, called three, in the / directory. It should again point to the original file. Verify the file name and contents of this file.

__ 16. Now try to create a fourth link, named four, on one of the file systems you just created.

__ 17. Create a symbolic link from `/root/two` to `/mnt/lv00/four`. Read the manual page for ln if necessary. Verify the name and contents of the file.

__ 18. Now delete /`root/two`, and verify the contents of both `/tmp/three` and `/mnt/lv00/four`.

Configuring user quota

__ 19. Remount the `/home` file system so that quota are enabled, and ensure that quotas are enabled when the system reboots.

__ 20. Perform a recalculation of current quotas and turn quota checking on.

__ 21. Set the soft limit of users tux1 and tux2 to 4 MB and the hard limit to 5 MB.

__ 22. On another virtual terminal, log in as tux1, and create five files of 1 MB each. Watch what happens.

__ 23. Still logged in as tux1, check the quota.

__ 24. Switch back to your root session and check the quota of all users.

END OF EXERCISE

Exercise 9. Memory management

What this exercise is about

This exercise demonstrates memory management in Linux and how to create additional paging space volumes. It also demonstrates virtualization using Xen.

What you should be able to do

After completing this exercise, you should have experience with:

- The most important tools in memory management
- Creating and activating paging partitions and paging files
- Creating virtual machines using Xen

Introduction

In this exercise, you will be performing various memory management and virtualization activities.

Requirements

- This workbook
- A workstation with a supported distribution of Linux installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

If you have more than 256 MB of memory

In this exercise, you are going to look at memory management in Linux. As part of this exercise, we are going to exhaust the available memory. Experience has shown that if you have more than 256 MB of memory, parts of this exercise will take too long. (Yes, there is a disadvantage to having a lot of memory!) Therefore, we are going to limit the amount of memory that Linux detects to 256 MB, just for this exercise. You will not be able to run the graphical environment with 256 MB of RAM, so also specify that the machine should boot into text mode (runlevel 3).

__ 1. Reboot your system and pause at the GRUB menu. Enter your GRUB password to access the maintenance functions.

__ 2. Select the **modify the kernel arguments** option, and add the following to the end of the line: **mem=256M 3**.

Using top

__ 3. Start **top** and identify the following pieces of information:

- The time, the up time, the number of users, and the load information on the first line
- The number of processes on the second line
- The CPU breakdown on the third line
- The real memory breakdown on the fourth line
- The swap space breakdown on the fifth line
- The processes (sorted by CPU-time) on the next lines

Verify that Linux detected only 128 MB.

__ 4. **top** automatically refreshes itself after 10 seconds. To increase this to one second, enter the command **s1**.

__ 5. **top** is by default does not show the amount of swap space used by each processes. To show this amount too, call up the Field Order screen with the **f** command, and enable the swap space display.

__ 6. To sort processes in a different order use the **M, P,** or **T** command.

__ 7. Leave **top** running in a separate window.

Checking swap space

__ 8. There should be one active swap space (check the fifth line of top). Deactivate this swap space. (If you do not know which partition is used as swapspace, check `/proc/swaps`.)

__ 9. Go back to your **top** window and check what happened.

__ 10. Reactivate the swap space.

Using the cache

__ 11. Caching is done automatically. If you have performed the exercises so far, your cache usage should be about 20 MB. Let's see what happens if we really start accessing the disk. Start a very disk-intensive program.

__ 12. Watch what happens to the amount of cached data in **top**. This number should increase until your cache occupies almost all of your real memory. Only about 4 MB will be left over for application use. However, also note that virtually no processes will be swapped out to disk.

Using the swap space

The swap space is only used if there is a real shortage in memory. First the amount of cached data decreases to about zero before processes are swapped to disk. Therefore, you need to have a program which uses a lot of real memory. The easiest solution is to write that ourselves.

__ 13. Create the directory `/root/bin` and **cd** into it.

__ 14. Create a program called **usemem**, with contents:

```
#!/usr/bin/perl
print "Allocating $ARGV[0] megabytes of memory...\n";
$big = "";
for( $i=0; $i<$ARGV[0]; $i++ )
{
$big .= "1234567890"x104858;
}
print "Press ENTER to release memory...\n";
<STDIN>;
```

When you start this program, it will gradually allocate the amount of memory you specify on the command line and then wait for you to press **Enter**. When you press **Enter**, the program exits and thus will deallocate its memory.

Note: Your instructor might have this program available already.

__ 15. Do a **chmod 755** of this program and start it with parameter **1**. This will ensure that the program allocates about 1 MB of memory.

__ 16. Now, go back to your top screen and select the **sort by memory usage** mode by issuing the **M** command. Note how much memory is used by the **usemem** program.

__ 17. Stop **usemem** and start it with about half the amount of memory which is in your system. Thus, if you have 256 MB of real memory or decreased the amount to 256 MB, use **99** as parameter. Watch what happens in **top**. First you should see the amount of cached data decrease, and maybe see some processes already being swapped out.

__ 18. Now start **usemem** with about 80% of your memory and watch what happens.

__ 19. Stop **usemem** and watch what happens. Note that processes swapped out are not automatically swapped in as soon as memory comes available. They will be swapped in when needed.

__ 20. Disable your swap space with **swapoff** and watch what happens.

__ 21. Enable swap space again, and run **usemem** with about 90% of the total amount of memory (real + swap). If you exhaust your real memory, and exhaust the swap space, you will see that **usemem** is automatically killed when it tries to allocate even more memory.

Creating a swap file

If you suddenly need more swap space on a running system but you have no more empty partitions to spare, you can use swap files. Using swap files, however, is less efficient than swap partitions, so use this only in an emergency. It is even possible to put your swap file on an NFS-mounted directory, but there are easier ways of bringing down a network.

__ 22. First, find a file system where you have room for a large swap file (at least 64 MB free). Locate a suitable directory on this file system.

__ 23. Create the large file to be used as swap file.

__ 24. Convert this file into a swap file.

__ 25. Activate it.

__ 26. Go to your **top** window and check whether the swap space has increased. Also, view the `/proc/swaps` file. What do you think is the meaning of the Priority field, and why is this different from the swap partition? Now try the **usemem** command that failed last time again.

__ 27. Add the swap file to your `/etc/fstab` file so that it is activated next time you reboot.

__ 28. Reboot your system to make sure that you are working with the correct amount of memory again.

Configuring and managing Xen

Xen SLES

This exercise consists of three parts. First, you will be installing the Xen packages and configuring domain0. Second, you will be installing a Xen guest domain using YaST. Finally, you will be modifying the memory allocation of your Xen guest.

Part 1: Installing Xen

__ 29. If you are not already logged in to the system, log in as a normal user.

__ 30. Start the YaST Control Center.

__ 31. Select **Software -> Software Management.**

__ 32. Select **View Packages: in patterns.**

__ 33. Expand the **Server Functions** option in the left pane, and click **Xen Virtual Machine Host Server** to highlight.

__ 34. Go back to Server Functions, select **Search,** and type xen in the search field. You will see a number of Xen packages already selected in the previous step. Select the **kernel-xen** package from the list and click **Accept** when done.

The Xen packages will now be installed on your system, using the preconfigured repository (NFS server or local CD/DVD). If using local media, you might be prompted to insert the appropriate disks.

__ 35. Reboot the machine, and select the **Xen** GRUB option.

__ 36. Once the machine comes up, log in as a user.

__ 37. Open up a terminal window and su - to the root user.

__ 38. Enter the command **xm list.** You should see one domain (Domain-0) with the status running.

Part 2: Installing a guest domain

__ 39. Open the YaST Control Center.

__ 40. Select **System -> Virtual Machine Management.**

__ 41. Click **Add.**

__ 42. Select **Run an OS Installation Program,** and then click **Next**.

__ 43. Click **Next**.

At this point, a terminal window opens with a standard SLES installation menu. Select this window with the mouse.

__ 44. Press **Alt+N** to advance to the next menu.

__ 45. Agree to the license agreement, and press **Alt+N** to continue.

__ 46. Continue through the installation screens, selecting the defaults and pressing **Alt+N** to advance.

__ 47. Start the installation with **Alt+I**.

__ 48. Once the installation has completed, click **Continue** in the Installation Complete dialog box.

__ 49. In the Domain Configuration overview, click **Next**.

__ 50. In the Virtual Machine Started message box, click **Finish**.

__ 51. Switch to the terminal of the virtual domain.

__ 52. Click **Next** to continue.

__ 53. For the root password, type `ibmlnx`**.** Accept the password warnings.

__ 54. Press **Alt+N** to continue.

__ 55. Skip the Internet connection tests, and press **Alt+N** to continue.

__ 56. Create a user account `student` with a password of `student`.

__ 57. Continue through the installation process until it is complete.

Part 3: Changing memory allocation

In this exercise, you will learn how to change the memory allocation of your Xen guest domain by changing the configuration file.

__ 58. Open a terminal window if not already open, and su - to the root user.

__ 59. Enter the command **xm list.**

_____

__ 60. What is the current guest domain's memory allocation set to?

__ 61. Switch to the Xen domain terminal and halt the guest by typing `halt`.

__ 62. Return to the root terminal and run **xm list** again. Is the guest domain running?

_____

__ 63. Edit the file `/etc/xen/vm/vm1` with an editor.

__ 64. Look for the **memory** parameter, and change the value to 172. Save and close the file.

__ 65. Start the guest domain again. Use the **xm** command to do this.

__ 66. Allow the guest to boot up to the login prompt.

__ 67. Switch back to your root terminal, and run **xm list** again. What is the memory allocation size?

Xen RHEL

This exercise consists of three parts. First, you will be installing the Xen packages and configuring domain0. Second, you will be installing a Xen guest domain. Finally, you will be modifying the memory allocation of your Xen guest.

Part 1: Installing Xen

__ 68. If you are not already logged in to the system, log in as a normal user.

__ 69. Select **Applications -> Add/Remove Software**. When prompted, type in the root password.

__ 70. In the left pane, select **Base System,** then in the right pane, select the checkbox for V**irtualization.**

__ 71. Click the **Optional Packages** button. Make sure all packages are selected (have check marks in the boxes), and then click **Close**.

__ 72. Click **Apply,** then **Continue.** If you see the Dependencies Added dialog box, click **Continue** to install all dependent software. You might also see an Import Key dialog box; if so, select **Import Key**.

__ 73. When the installation is complete, exit the Package Manager tool and reboot the system. When you see the GRUB auto-boot menu, press any key to get the GRUB menu.

__ 74. Select the kernel with Xen support (fc8xen in the description) and press **Enter** to boot.

__ 75. At the login prompt, log in as your user ID, and then switch to root using `su -`.

__ 76. Open a terminal window and type `xm list`. You should see an entry for the Xen Domain-0, with the status of running (under State, you should see an “r”).

Part 2: Installing a guest domain

__ 77. Select **Applications -> System Tools -> Virtual Machine Manager** from the menus. When prompted, enter the root password.

__ 78. You should see the Virtual Machine Manager (virt-manager) start up. In the display output, there will be one entry for localhost, with a status of Disconnected**.** Right-click the localhost entry, and select **Connect** from the menu.

__ 79. The localhost entry should now have a status of **Active,** and you should see an entry for Domain-0 with a status of **Running.**

__ 80. On the localhost line, to the right of the Memory Usage column, there is an icon. Click the icon, and the Create a new virtual system dialog should start.

__ 81. For the system name, use `rhel6-guest` for RHEL6.

__ 82. For the virtualization method, select **Paravirtualized**. Click **Forward** to continue.

__ 83. For the install media URL, use the following URLs, depending on your distribution:

RHEL6: http://<classroom server>/rhel6

__ 84. At the Assigning Storage Space dialog box, select the **Simple File** radio button. The values for the File Location and File Size fields remain at their defaults. Uncheck the box for **Allocate entire virtual disk now?** and click **Forward** to continue.

__ 85. At the Connect to host network dialog box, make sure **Virtual network** option is selected (the default), and click **Forward** to continue.

__ 86. At the Allocate memory and CPU dialog box, set VM Max Memory to **500 MB**, and VM Startup Memory to **500 MB**. Set the VCPUs value to **1**, and click **Forward** to continue.

__ 87. Verify your settings at the Ready to begin installation screen, and click **Finish** when done.

The installation of the guest should start. You should see the new guest domain get created and appear in the Virtual Machine Manager window. You will also see the console for your new guest appear and the Fedora installation program start.

__ 88. In the Virtual Machine console, select the correct install language and keyboard.

__ 89. At the Configure TCP/IP screen, leave the default values and click **OK**.

The guest will DHCP an address from the network and begin the installation program.

__ 90. At the splash screen, click **Next** to continue.

__ 91. A warning dialog will appear stating the device “xvda” has an invalid partition table. Click **Yes** to initialize the drive.

__ 92. At the partitioning screen, leave the default values, and click **Next** to continue.

__ 93. At the network configuration screen, leave the default values and click **Next** to continue.

__ 94. Select the correct time zone and city, and click **Next** to continue.

__ 95. Type `ibmlnx` for the root password, and click **Next** to continue.

__ 96. At the package selection screen, de-select the **Office and Productivity** package category, and click **Next** to continue.

__ 97. Click **Next** at the final installation confirmation screen.

The guest should start to install. The installation will take 10-15 minutes to complete.

__ 98. When the installation is complete, click **Next** to reboot the guest. The guest will shut down, and the console will give you a message `Guest Not Running`. Click the **Run** button on the menu bar to restart the guest.

__ 99. Go through the Fedora Welcome screens, selecting the appropriate options. Your guest will then present you with the graphical login program.

Part 3: Changing memory allocation of a guest machine

In this section, you will experiment with dynamically changing the memory allocated to your Xen virtual machine.

__ 100. Log in to the Xen guest domain you just installed as the root user.

__ 101. Start up a terminal emulator, and within it, run the **top** application.

__ 102. Press **Ctrl+Alt** to release the mouse. Select the Virtual Machine Manager window.

__ 103. Right-click the entry for your guest machine in the VMM, and select **Details** from the drop down menu.

__ 104. Select the **Hardware** tab.

__ 105. Select the **Memory** option in the left window pane.

__ 106. Change the memory allocated to the client to `256 MB` and click **Apply**.

__ 107. Switch back to the Virtual Console of the guest and look at the **top** output. What do you notice?

__ 108. Switch back to the guest Machine Details window, and increase the memory of the guest machine to its previous value, and click **Apply** to finish.

__ 109. Switch back to the Virtual Console and look at the **top** output. Does the allocated memory increase?

END OF EXERCISE

Exercise review/wrapup

This exercise covered various memory management activities.

Exercise 10. Scheduling

What this exercise is about

This exercise provides the student with knowledge about scheduling in Linux. Students will set up scheduling and start their own jobs through scheduling.

What you should be able to do

After this exercise, students should be able to:

- Name the commands and daemons that make up scheduling on Linux
- Start and stop jobs through scheduling

Introduction

In this exercise, you are only going to work with **cron** and **at**. You will not be working with **batch** because it is dependent on the load of the system, and it is too hard to reliably create a situation where **batch** can be demonstrated, given the wide range of configurations on which this course needs to run.

Requirements

- This workbook
- A workstation with RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

Working with cron *and* at

__ 1. Log in at **Ctrl+Alt+F1** as tux1 and **Ctrl+Alt+F5** as root.

__ 2. Go to VT 5 (where root is logged in).

__ 3. Check if the daemons necessary for scheduling are running.

__ 4. View the contents of your crontab file. Where are the crontab files for the users stored?

__ 5. List the contents of the `/var/spool/cron` and `/var/spool/at` directories. What do you see?

__ 6. Edit your crontab file to make sure that your system runs the **df** command every five minutes.

__ 7. Check the contents of the `/var/spool/cron` directory again. What do you see now?

__ 8. Start a job that will run in 10 minutes. The command that should be executed in 10 minutes is **who**. At what time will the command run according to **at**?

__ 9. Check if there are at jobs listed in the contents of the `/var/spool/at` directory. What do you see?

__ 10. Look at the contents of the file. What do you see?

__ 11. Enter the next command:

- **`# at 9am`**
- at> **`ps aux`**
- at> **Ctrl+D**

When will this command run?

__ 12. Switch to the VT where tux1 is logged in.

__ 13. Look at the crontab file of tux1.

__ 14. Make a crontab file for tux1 that executes the **ls $HOME** | **wc -l** command every hour. Check if the crontab file for tux1 exists afterwards.

__ 15. Remove the crontab file.

__ 16. Switch back to the VT 5. Set up the **cron** daemon so that tux1 is not allowed to use crond anymore.

__ 17. Go to VT 1 and try to create a crontab file again. What happened?

__ 18. As root, check your mail. What kind of mail did you receive?

__ 19. Remove the crontab files of root. (Do not use the **rm** command.)

__ 20. Take a look at the system crontab files and directories.

END OF EXERCISE

Exercise 11. Backup and restore

What this exercise is about

The purpose of this exercise is to familiarize you with backup and restore techniques.

What you should be able to do

After completing this unit, students should be able to:

- Perform system, full, and incremental backups
- Use **tar** and/or other commands to perform backup and restore

Requirements

- This workbook
- A workstation with RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

Using tar

__ 1. Check the **man** pages for the discussed backup commands: **tar, cpio.**

__ 2. Create in your `/home` directory three subdirectories, called subdir1, subdir2, and subdir3, and create five files, called `testfile`*x*, in each subdirectory.

__ 3. Create an extra logical volume in your volume group. Name this logical volume backuplv. Its size should be about 10 MB.

__ 4. Use the **tar** command to back up your `/home` directory to the partition you just created.

__ 5. Verify the contents of the backup you just created.

__ 6. Delete your `/home/subdir`*n* directories and all the files in them.

__ 7. Restore the backup you created.

__ 8. Produce a list (/tmp/1daytar.list) of all the files in `/home` that were accessed in the last day.

__ 9. Create an incremental backup using the **tar** command.

__ 10. Restore the backup you created in the previous step.

Using cpio

__ 11. Produce a list /tmp/todaycpio.list of all the files in `/home` that were modified in the last day.

__ 12. Create an incremental backup using **`cpio`**. Use the same partition you used earlier.

__ 13. Restore the backup you created in the previous exercise.

Using dump

__ 14. Verify that the **dump** program is installed. If not, use your distribution's package manager tool to install it.

__ 15. Dump the `/home` file system to the backup partition.

__ 16. Write down the dates of all files in `/home/subdir*`. Update all files in `/home/subdir1` and make an incremental backup to floppy disk.

__ 17. See what files are dumped to the backup LV and the floppy disk.

__ 18. Unmount the `/home` file system and reformat it. Then, restore both dumps in the correct order. Verify that the dates on the files are correct.

END OF EXERCISE

Exercise 12. User administration

What this exercise is about

This exercise teaches students how to add users and groups and where to find information about users and groups in their system.

What you should be able to do

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

- Add and delete users and groups
- Specify the files involved in user administration

Introduction

In this exercise, you will create and modify Linux users, and groups. During this lab exercise, you will use the Linux distribution that you have loaded onto your lab system. Please disregard any exercise steps that reference a different Linux distribution.

Requirements

To complete this exercise, you will need the following:

- This workbook
- A workstation with Fedora, RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

User environment

__ 1. Log out every user.

__ 2. Log in as tux1 at VT 1 and as root at VT 5.

__ 3. Look at the following files:

- `/etc/profile`
- `/etc/bashrc` (RHEL/Fedora) or `/etc/bash.bashrc` (SLES)
- `$HOME/.bash_profile` (RHEL/Fedora) or `$HOME/.profile` (SLES)
- `$HOME/.bashrc`
- `$HOME/.bash_logout` (RHEL/Fedora only)

What do these file do in your system? What information can be found in there?

User administration

These hints describe the command line tools. Do not forget that you also have the system administration tools to do user and group administration.

__ 4. List all users defined in your system at this moment.

__ 5. Does every user have a password?

__ 6. As root, using the command line, add a user with the username tux3 and full name Tux the Penguin (3).

__ 7. Look at the contents of `/etc/shadow`. What is the password of the new user?

__ 8. Give tux3 the password penguin. Then take a look at the `/etc/shadow` file again.

__ 9. List the contents of the `/etc/passwd` file. What is stored in this file?

__ 10. What is the login group of the user you added?

__ 11. Add the group penguins to the system.

__ 12. Place the user tux3 in the group penguins.

__ 13. Locate this change in `/etc/group`

__ 14. What is the syntax of the group file?

__ 15. Which users are specified in the last field of the `/etc/group` file?

Communicating with the users

__ 16. As root, modify the `/etc/issue` file. Write in this file that the system is for authorized users only. Remove all distribution and kernel information.

__ 17. As root, create the `/etc/motd` file. Write in this file your system name and the name of the distribution you are using.

__ 18. As tux1, log out and log in again. Look at the information shown on your screen.

__ 19. As tux1, create the file `.hushlogin` in your home directory.

__ 20. Log out and then log in. Again look at the information showing up on your screen. Do you notice any difference with the output from step 16?

END OF EXERCISE

Exercise 13. User-level security

What this exercise is about

This exercise focuses on the security concepts in a Linux system. Students will learn how file permissions work, where information about logins is stored, and where user data for user administration is stored.

What you should be able to do

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

- Locate the files involved in security on the system
- Explain the different file permissions and when to use them

Introduction

In this exercise, you will be performing various activities related to user level security.

Requirements

- This workbook
- A workstation with Fedora, RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

Security logs

__ 1. Log out as root.

__ 2. Try to log in as root, but use a wrong password.

__ 3. Try to log in as a non-existing user (for instance: teacher).

__ 4. Log in as root with the correct password.

__ 5. Look at the contents of the following files and try to find the records of the previous steps 2, 3 and 4.

- `/var/log/lastlog`
- `/var/log/messages`
- `/var/log/secure` (RHEL/Fedora)
- `/var/log/wtmp`
- `/var/run/utmp`

__ 6. At virtual terminal 1 (VT 1), log in as tux1 and at Virtual Terminal 2 (VT 2), as tux2. Log in as root on VT 5.

__ 7. As root, look at what users are currently logged on. Look also at the contents of `/var/log/wtmp`.

__ 8. Log out as user tux2.

__ 9. As root, find the last times that user tux1 and tux2 logged in. Also list the users that logged in at virtual terminal 1.

__ 10. As tux1, check what groups you are a member of.

File permissions

__ 11. Log in as root at VT 5 (if you are not already logged on as root at VT 5). Be sure that tux1 is logged in at VT 1 and tux2 at VT 2.

__ 12. As root, check if the SUID is set. Then remove the SUID bit from `/usr/bin/passwd`. Be sure to leave the execute permissions in place.

__ 13. As tux1, run the **passwd** command. Answer all the questions and watch the output. What is the error message, and why do you get this message?

__ 14. Set the SUID bit again on `/usr/bin/passwd`, and try to change the password now.

__ 15. As tux1, make a directory in your home directory with the name exercise. In this directory, make a file with:

df > df.out

Set the permissions on df.out as rwx------ (700), and make your home directory readable and executable for group members and others.

__ 16. As tux2, try to open `df.out` and read the contents of the file. Do you succeed? Also, try to add an extra line to `df.out`.

__ 17. As tux1, set the permissions on `df.out` as rwxr--r- (744).

__ 18. As tux2, try to open the file again. Do you succeed this time?

__ 19. As tux2, try to alter the file. Do you succeed?

__ 20. As tux1, set the permissions on `df.out` as rwxrwxrwx (777).

__ 21. As tux2, try to write to the file again. Do you succeed this time?

__ 22. Can you remove `df.out`? Why or why not? Is the data in `df.out` safe with these permissions set on the file?

__ 23. As tux1, set the permissions on your exercise directory as rwxrwxrwx (777).

__ 24. As tux2, try to remove `df.out`. Is this possible and why is it or why not?

Team directories

__ 25. As root, create a team directory, `/groups/penguins`, for all members of the penguins group.

__ 26. Check which users are currently member of the penguins group.

__ 27. As tux1, try to create and view files in the `/groups/penguins` directory. Does this work?

__ 28. As tux3, try to create and view files in the `/groups/penguins` directory. Does this work?

__ 29. Add tux1 and tux2 to the penguins group. Then try to create and delete files in the `/groups/penguins` directory again. Also, try to modify files created by other users in the same group. Does this work?

Using the su command

__ 30. As tux1, go to your home directory and run the **su** command. Write down the following information:

- The current working directory
- The contents of the $**PATH** variable

__ 31. Become yourself (tux1) again.

__ 32. As tux1, run the **su -** command and write the following information down:

- The current working directory
- The contents of the $**PATH** variable

__ 33. Compare this information with the information written down earlier. What is the difference between **su** and **su -** ?

__ 34. As tux1, try to become tux2. Does this work?

Using the sudo command

__ 35. As tux1, try to execute the /**sbin**/**ifdown eth0** command. Does this work?

__ 36. As root, edit the `/etc/sudoers` file so that tux1 is allowed to execute the /**sbin**/**ifdown eth0** and /**sbin**/**ifup eth0** commands.

__ 37. As tux1, try to execute the /**sbin**/**ifdown eth0** command again, this time with the help of sudo. Does this work? If it works, bring the Ethernet adapter back up again as well.

Note: The first time a user runs the **sudo** command, he or she receives a warning about responsible behavior. To confirm having read this, the user needs to type his or her own password, not the root password!

Security files

__ 38. As tux2, look at the content of the following files:

- `/etc/passwd`
- `/etc/group`
- `/etc/profile`

END OF EXERCISE

Exercise 14. Troubleshooting

What this exercise is about

This exercise lets you troubleshoot some Linux problems and use some techniques to solve these problems.

What you should be able to do

After completing this exercise, you should have experience with:

- Rescue mode

Introduction

In this exercise, you will be performing various activities related to troubleshooting.

Requirements

- This workbook
- A workstation with RHEL, or SLES installed

Exercise instructions

Preface

- All exercises of this chapter depend on the availability of specific equipment in your classroom.
- The hints provided for locating documentation on particular Web pages were correct when this course was written. By nature, Web pages tend to change over time, so ask your instructor if you have trouble navigating the Web sites.
- A computer system with a connection to the World Wide Web and a Web browser is recommended by not required.

Install problem creation scripts

__ 1. Log in to the system as root.

__ 2. Create a mount point for the NFS server named /mnt/problem.

__ 3. Mount the NFS server directory `/export/files/problem` to the /mnt/problem mount point.

NFS server name/address: _____

__ 4. Install the problem creation script rpm problem-1.0-0.i386.rpm.

```
# rpm -ivh /mnt/problem/problem-1.0-0.i386.rpm
```

Problem #1

__ 5. Log in to the system as root.

__ 6. Create the problem by issuing the following command: /**root**/**prob**/**p1create**. Note the problem:

__ 7. Return the system to normal operation. How was the problem resolved?

__ 8. Have the instructor validate the problem has been resolved.

Problem #2

__ 9. Log in to the system as root.

__ 10. Create the problem by issuing the following command: /**root**/**prob**/**p2create.** Note the problem:

__ 11. Return the system to normal operation. How was the problem resolved?

__ 12. Have the instructor validate the problem has been resolved.

Problem #3

__ 13. Log in to the system as root.

__ 14. Create the problem by issuing the following command: /**root**/**prob**/**p3create.** Note the problem:

__ 15. Return the system to normal operation. How was the problem resolved?

__ 16. Have the instructor validate the problem has been resolved.

Problem #4

__ 17. Log in to the system as root.

__ 18. Create the problem by issuing the following command: /**root**/**prob**/**p4create.** Note the problem:

__ 19. Return the system to normal operation. How was the problem resolved?

__ 20. Have the instructor validate the problem has been resolved.

Problem #5

__ 21. Log in to the system as root.

__ 22. Create the problem by issuing the following command: /**root**/**prob**/**p5create.** Note the problem:

__ 23. Return the system to normal operation. How was the problem resolved?

__ 24. Have the instructor validate the problem has been resolved.

END OF EXERCISE

Exercise review/wrapup

This exercise introduced a number of problems onto the students system that required the student to troubleshoot and resolve the issue.