Linux Basics and Installation

(Course code LX02)

Student Exercises

ERC 7.0

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® Notes®

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

PostScript is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel is a trademark or registered trademark 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, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

February 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® Notes®

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

PostScript is either a registered trademark or a trademark of Adobe Systems Incorporated in the United States, and/or other countries.

Intel is a trademark or registered trademark 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, Windows and Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

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

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

Exercises description

This course includes the following exercises:

- Installing Linux
- Using the system
- Working with files and directories
- File and directory permissions
- Linux documentation
- Editing files
- Shell basics
- Working with processes
- Linux utilities
- Shell scripting
- The Linux GUI
- Customizing the user environment
- Basic system configuration
- Integrating Linux in a Windows environment
- End-of-course challenge exercise (optional)

In the exercise instructions you will see each step prefixed by a line. You may wish to check off each step as you complete it to keep track of your progress.

Most exercises include required sections which should always be completed. These might be required before performing later exercises. Some exercises may also include optional sections that you may wish to perform if you have sufficient time and want an additional challenge.

The objective of the exercises is to let you become familiar with installing and running Linux on your personal workstation. To achieve this, a variety of real-world exercises are performed, aimed at simulating real-world tasks.

Each exercise unit consists of two parts:

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, the online documentation 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 don't want them and hints when you need them.

In this last section, multiple ways to accomplish the same task are often provided. Where this has been done, the various methods are separated by an **-OR-**

All exercises and hints apply both to Fedora, RHEL, and SLES equally unless mentioned.

Exercise 1. Installing Linux

What this exercise is about

This exercise covers installing Linux.

What you should be able to do

After completing this exercise, you should be able to:

- Prepare a system for installation
- Partition a system
- Install Linux

Requirements

- A set of installation CDs or a network capable boot CD for your distribution

Exercise instructions

Preface

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

Note

The exercises in this course material have been designed for and tested on the following two distributions:

- Red Hat Enterprise Linux (RHEL) 6.0 Enterprise Client
- SUSE Linux Enterprise Server (SLES) 11

If you are using one of these three distributions, follow the instructions below that apply to your distribution. If you are using another distribution, or another version of one of the three distributions above, then your instructor gives you additional information.

Optional: Depending on the circumstances, your instructor might have to loan you a full set of CDs for each distribution, so that you can perform a CD-based install, or your instructor might have to loan you an installation CD so that you can perform a network-based install.

If you need to perform a network install, your instructor will give you additional information, specifically:

- The install method: NFS (FTP or HTTP is also available)
- The IP address that is to be used for your workstation, if DHCP is not used
- The name or IP address of the install server
- The path to the installation images on the install server

Installing Red Hat Enterprise Linux 6.0

__ 1. Turn on or reboot the computer.

__ 2. Your systems needs to boot to the network instead of from the hard drive or CD. Depending on your hardware, you might need to press F12 or F9 or F1 to force a network boot or SMS screen. You see a screen with a text version of the IBM logo that gives you different installation options. You will simply type an appropriate number and then press **Enter**.

- For RHEL 6.0 Client, type 13 and press **Enter**.

Note

If your system is not booting to the network, ask your instructor for additional assistance.

__ 3. Choose the language for the installation process. Click **OK**.

__ 4. Choose your keyboard model and layout, click **OK**.

__ 5. On the TCP/IP screen, you want to disable **IPv6 Support**, click **OK**.

__ 6. At the initial graphic screen, click **Next**.

__ 7. If you are installing RHEL 6.0, you are asked to input an “Installation Number” -- choose **Skip Installation Number**; then click **OK**.

__ a. Next, click in the pull-down that says **Remove Linux partitions** and choose Create custom layout; then click **Next**.

__ b. The Disk Druid screen displays and shows the current layout of your disks. You first need to **Delete** all partitions manually. You can then start adding Linux partitions. Make sure you create three additional partitions:

i. One partition is used as root partition. Its Mount Point should be “/”, the File System Type should be **ext3**, and the size of this partition should be 6 GB (**6000 MB**).

ii. Add a boot partition. Its Mount Point should be /**boot**, the File System Type should be **ext3,** and the size of this partition should be **100 MB**.

iii. The last partition is used as swap space, which does not have a mount point. The size should be equal to the amount of real memory, with a maximum of 1000 MB, and the File System Type should be **swap** (the Mount Point shows <Not Applicable>)

__ 8. Let the instructor check your partition configuration before you save it! After the instructor has checked your partition configuration, click **Next**.

__ 9. The installation program now allows you to configure your boot loader. You can accept all defaults here, then click **Next**.

__ 10. Configure your network adapters. Your instructor should tell you whether to use **DHCP** or will provide you with the IP Address, Netmask, Network and Broadcast addresses, with the Hostname, Gateway and DNS addresses. Enter these values, double-check them, and click **Next**.

__ 11. Now select your **Time Zone** and clear the **UTC** check box; then click **Next**.

__ 12. In the next screen you need to set the root password. For convenience in the class, set the root password to **ibmlnx**; then click **Next**.

__ 13. At the Software screen, select **Customize now**. Click **Next** and add the **KDE Desktop Environment** group. Also, click **Development** and add **Development Tools**; then click **Next**.

__ 14. Note the location of the log file and click **Next**.

__ 15. The installation program now formats the file systems and installs Linux. This might take anywhere from 5 minutes to an hour, depending on the number of packages to install, 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 to the graphical installation screen with **Ctrl+Alt+F7**. Also, take a look at other virtual screens (1 through 6).

__ 16. When your installation is complete -- click **Reboot** to reboot your system.

__ 17. When your Linux system boots for the first time, the RHEL Setup Agent is started. Click the **Forward** button.

__ 18. Read the License Agreement, if asked; then select **Yes, I agree** and click **Forward**.

__ 19. The next screen allows you to configure firewall rules. Choose **Disabled** from the **Firewall** list and click **Forward** and then **Yes**. Then **Disable** the SELinux Setting the same way, and click **Forward** and then click **Yes**.

__ 20. RHEL 6.0 will ask you if you want **Kdump** enabled -- you do *not*, so click **Forward**.

__ 21. Check the date and time. If the network has an NTP server, configure it here as well. Click **Forward**.

__ 22. RHEL 6.0 Client will ask if you want to **Set Up Software Updates**. Click **No> Forward> No thanks> Forward**.

__ 23. Answer No if a confirmation screen is displayed.

__ 24. Add a personal user account for yourself, with a password you make up yourself, then click **Forward**.

__ 25. Verify that your sound card has been detected and is configured correctly, by playing a test sound. Then click **Yes**, then **Finish**.

__ 26. RHEL 6.0 Client will ask if you want to install any additional CDs, click **No> Finish> OK**.

__ 27. The installer might ask that you reboot your machine at this time -- do so and the installation is complete.

Installing SUSE Linux Enterprise Server 11

__ 28. Turn on or reboot the computer.

__ 29. Your systems needs to boot to the network instead of from the hard drive or CD. Depending on your hardware, you might need to press F12 or F9 or F1 to force a network boot or SMS screen. You see a screen with a text version of the IBM logo that gives you different installation options. You will simply type an appropriate number and then press **Enter**. For **SLES 11**, type **15** and press **Enter**.

Note

If your system is not booting to the network, ask your instructor for additional assistance.

__ 30. Select the language and keyboard map (if asked) for the network configuration process and click **Next**.

__ 31. On the License Agreement screen, choose **Yes, I agree** and click **Next**.

__ 32. If your system has already been installed with Linux, then a window might open stating this. Select **New installation** and click **Next**.

__ 33. Browse through the autodetected installation settings, and make changes if required:

__ a. Make sure your **Time Zone** and **UTC/Local** choices are correct.

__ b. Click **Partitioning;** then click **Create Custom Partition Setup**, then **Next**. Now click **Custom Partitioning** and **Next**. Delete all partitions that you see. Make these choices to create three partitions:

i. Click> **Primary**> **OK**. Format as **ext3**, highlight the number in the **End** box and change it to **6GB**, enter `/` in the **Mount Point** field, then click **OK**.

ii. Click> **Primary**> **OK**, Format as **ext3**, highlight the number in the **End** box and change it to **100MB**, enter `/boot` in the **Mount Point** field, then click **OK**.

iii. Click **Primary**> **OK**. Format as **Swap**, highlight the number in the **End** box and change it to **1GB**, and then click **OK**> **Finish**.

__ c. Click **Software**, make sure that you add **KDE** and **C/C++ Compiler and Tools**

Click **Accept** (on any and all pop-up screens), and click **Install**. SLEs 11 now installs itself. This takes five minutes to an hour, depending on the speed of your computer.to the default selection of software; then click **Accept**.

__ 34. Note that SLES 11 might automatically reboot midway through the installation process. This is normal. When the initial boot screen appears, do nothing so that the system boots from hard disk. The installation process should continue automatically.

__ 35. Next, you need to enter the root password. For convenience in class, use **ibmlnx** as the root password and click **Next**.

__ 36. On the **Hostname and Domain Name** screen, select the box next to **Change Hostname via DHCP** check box and then click **Next**.

__ 37. The **Network Configuration** screen allows you to configure your network. Make sure all detected values are okay. If necessary, consult your instructor for IP addresses and such. Then, click the word **Enabled** next to **Firewall** to toggle the firewall setting; then click **Disable IPv6** and click **Next**.

__ 38. Even if you have an Internet connection, click **No, skip the test** for your Internet connection test and click **Next**.

__ 39. Select **Local (/etc/passwd)** as **User Authentication Method** screen. Click **Next**.

__ 40. Add a local user account for yourself, using a secret password. Do *not* select **Automatic Login**. Then click **Next**.

__ 41. SuSEConfig now executes several configuration scripts. This might take several minutes.

__ 42. If you feel like it, read the **Release Notes** for this version. Then click **Next**.

__ 43. Check your Hardware Configuration, (You may change the Graphics Card/Monitor settings if you know what they should be.) Then click **Next,** and then **Finish**.

__ 44. Select the **Clone for Autoyast** check box and log in when the system is ready.

End of exercise

Exercise 2. Using the system

What this exercise is about

The purpose of this exercise is to become familiar with Linux, the command syntax, and some basic commands. The exercise also serves to show some mult-iuser concepts.

What you should be able to do

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

- Switch between virtual terminals
- Log in to a Linux system and change passwords
- Execute basic commands
- Use the wall and write commands to communicate with other users
- Use keyboard control keys to control command line output
- Use the mouse to copy and paste commands
- Use the command history
- Lock a Linux system
- Log out of a Linux system

Exercise instructions

Preface

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

Logging in on a virtual terminal

In this section, you are going to log in to the system using both text and graphical virtual terminals.

__ 1. If the install went correctly then you should now see a graphical login prompt. If this is not the case, ask your instructor to fix this. (You learn how to do this yourself later in the course.)

__ 2. Verify that you indeed have seven different virtual terminals. Cycle through them by pressing Alt+Fn, where n is the terminal number you want to access. Use Ctrl+Alt+Fn when you are in a graphical terminal.

__ 3. In your first virtual terminal (tty1), log in to the system with your own username, which you also configured when installing the system.

__ 4. In your second virtual terminal (tty2), log in to the system as root. After having logged in, look at the command prompt. Do you notice anything different from the command prompt in the other virtual terminals?

__ 5. In your seventh virtual terminal (tty7), log in to the system with your own username and password.

__ 6. Open a terminal window. Take a look at the command prompt. Does it differ from the command prompt on tty1? Why or why not?

Basic commands

In this section, we are going to execute some basic commands, in order to familiarize yourself with the command syntax of Linux, and the fact that you are currently on a multiuser, multi-tasking system.

All commands in this section are executed on virtual terminal seven (the graphical login prompt where you are logged in as yourself), using the terminal window you just opened, unless specified otherwise.

__ 7. Change your password. Memorize this password because no one can find out your password if you forget it.

__ 8. Display the system's date.

__ 9. Display the whole calendar for the year 2011.

__ 10. Display the month of January for the year 1999 and 99. Are 1999 and 99 the same?

__ 11. Generate a list of all users present on your system.

__ 12. Display your login name.

__ 13. Display the login information of your own user account, and of root.

__ 14. Clear your screen.

__ 15. Print the text Out to lunch on your display.

__ 16. Make sure you are willing to receive messages.

__ 17. Write the message Out to lunch to the display of root. Check whether root got the message.

__ 18. Write the message Out to lunch to the display of all users. Check whether everybody on your system got the message.

Keyboard and mouse tips

__ 19. The bash shell has a command history function. View some of the commands you have entered. Try to alter one of these commands; then run the command again.

__ 20. Your terminal has a buffer that keeps track of the output of your commands. View the output of the previous commands.

__ 21. Bash supports command and filename completion with the TAB character. Try to use this feature, both on commands and on filenames.

__ 22. Both in a text terminal and an emulated terminal in the graphical desktop, try to re-execute commands by scrolling up a little, selecting the command with the left mouse button, and then pasting it onto the same terminal again with the middle mouse button. Also try this across different text and graphical terminals.

Note

SuSE does not enable gpm by default; therefore your mouse will not work in a text terminal when you are using SuSE.

Using the history

__ 23. Use the history command to view the last 20 commands you typed.

__ 24. Execute one of the commands from the history list.

__ 25. Execute the echo command again, this time changing the word lunch to dinner.

__ 26. Bash also supports searching in the history. Try this feature as well.

Locking terminals

Note

Note: Not all distributions install vlock and xlock by default. If vlock and xlock are not installed, then you learn how to do that in the basic system configuration exercise.

__ 27. Lock a virtual terminal. Can you switch to another virtual terminal while this one is locked? Unlock the terminal.

__ 28. Lock the console. Can you switch to another virtual terminal now? Unlock the console.

__ 29. Lock the graphical environment and then unlock it again.

Logging off

__ 30. Log off all users that are logged in at any TTY.

End of exercise

Exercise 3. Working with files and directories

What this exercise is about

This exercise provides you with the opportunity to begin working with directories and the files they contain.

What you should be able to do

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

- Work with directories
- Work with files
- Work with files and directories recursively
- Work with binary files

Exercise instructions

Preface

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

Working with directories

__ 1. If you are not logged in as yourself at **tty7**, log in now. Make sure you've got a terminal window open.

__ 2. Check the directory you are placed in. What directory is this? _____

__ 3. Change your current directory to the root directory (/).

__ 4. Verify that you are in the root directory and then execute both a simple and a long listing of the files in that directory.

__ 5. List all files in the current directory and list all files in the current directory and below. **Note:** This command provides extensive output. When you have seen enough, end the command with the correct **<Ctrl>** sequence.

__ 6. Return to your home directory and list its contents including hidden files.

__ 7. Create a directory in your home directory called mydir. Then, issue the command to view a long listing of your home directory and the ~/mydir directory. (Do not show the contents of the directories.) What is the size of each directory? _____

__ 8. Change to the mydir directory. Create two zero-length files called `myfile1` and `myfile2`.

__ 9. Issue the command to view a long listing of the contents of the `mydir` directory. What are the sizes of `myfile1` and `myfile2`?_____

__ 10. Return to your home directory and use the `ls -R` command to view your directory tree.

__ 11. Try to remove the `mydir` directory. Does it work?

__ 12. Go to the `mydir` directory once more and delete the two files in that directory; then go back up to your home directory and delete the `mydir` directory.

Working with files

__ 13. Look at the contents of the `/etc/passwd` file. The `/etc/passwd` file contains a list of all the users authorized to use the system.

__ 14. Copy the `/etc/passwd` file to your home directory, and rename it to `usersfile`.

__ 15. Split the `usersfile` into a number of smaller files of 200 bytes each.

__ 16. Make a long listing of all files in your home directory.

Working with files and directories recursively

__ 17. Create a directory sub1 and create a directory sub2 in sub1. Do this all with one command.

__ 18. Go to the sub2 directory and create a file called myfile.

__ 19. Go back to your home directory. Then make a copy of the whole sub1 directory tree by the name of tree1. Make a recursive listing of all files and directories in sub1 and tree1.

__ 20. You now have two directory trees, named sub1 and tree1. Move the directory tree tree1 into the sub1 subdirectory.

__ 21. List the contents of your home directory. Make a recursive listing of all files and directories in the sub1 directory.

Working with binary files

__ 22. List the content of the file `/bin/ls` using **od** or **hexdump**.

__ 23. List all strings in the /bin/ls program.

End of exercise

Exercise 4. File and directory permissions

What this exercise is about

This exercise provides you with the opportunity to work with file and directory permissions.

What you should be able to do

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

- Apply file and directory permissions

Exercise instructions

Preface

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

Creating user accounts

To demonstrate permissions in full, you need to create a few additional users, tux1 and tux2, who both are members of the penguins group. For this, you need to execute a few commands that have not been covered in the course and which normally do not need to be executed by a regular user. They are covered in full in LX03.

__ 1. On tty3, log in as root.

__ 2. Execute the following series of commands:

```
# groupadd penguins
# useradd -m -g penguins -c "Tux the Penguin (1)" tux1
# useradd -m -g penguins -c "Tux the Penguin (2)" tux2
# passwd tux1
New password: penguin1
Retype new password: penguin1
# passwd tux2
New password: penguin2
Retype new password: penguin2
```

__ 3. On tty1, log in as **tux1** with password **penguin1**, and on tty2, log in as **tux2** with password **penguin2**.

File and directory permissions

__ 4. Switch to VT 1, where you are logged in as tux1, and look at the permissions on your home directory.

__ 5. Switch to VT2, where you are logged in as tux2. Try to change to the home directory of tux1, or read the contents of the home directory of tux1. Does this work?

On a Fedora or Red Hat system, both commands fail, because the default permissions on a user's home directory are set to rwx------. On a SuSE system, both commands succeed, because the default permissions are set to rwxr-xr-x.

__ 6. Fedora/Red Hat only: Switch to tty1. Change the permissions on the home directory of tux1 so that other users are allowed to read and access it. Then try to access the directory again as tux2. Does this work now?

__ 7. As tux2, try to create and delete files in tux1s home directory. Does this work?

__ 8. Switch once again to tty1. Create a bin directory (Fedora/Red Hat only) and copy the file /bin/ls in there, renaming it to my_ls in the process.

__ 9. Set the permissions on my_ls to **rw-r-----**, and then try to execute it both as tux1 and tux2. Does this work? Why not?

__ 10. Now set the permissions to **rwxr-xr-x**, then try to execute it once more, both as tux1 and tux2. Does this work now?

__ 11. Try to execute my_ls as tux1, as tux2, and as yourself, but now with permissions **rw-------**, **rw-rw----**, **rwx------**, **rwx--x---**, and **rwx--x--x** as well. What permissions are required, at a minimum, for tux1 to execute my_ls? What permissions are required for tux2? What permissions does your own user account require?

End of exercise

Exercise 5. Linux documentation

What this exercise is about

The purpose of this exercise is to give you the opportunity to explore and experiment with the **man** and **info** commands. You will also read the FAQ and HOWTO documentation.

What you should be able to do

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

- Use the **man** command
- Use the **info** command
- Locate and use other Linux documentation

Exercise instructions

Preface

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

Manual pages

__ 1. If you are not already logged on, log in as tux1.

__ 2. Display the manual page for the **man** command.

Read the information presented to obtain a better understanding of the functionality of the **man** command.

__ 3. Search for the string *PAGER*.

__ 4. Use the <**Q**> key to end the **man** command.

__ 5. Display the man page of the **ls** command. Move through the manual pages.

__ a. Go to the last page.

__ b. Go to the the previous page.

__ c. Go to the first page.

__ 6. Close the **man** command.

__ 7. Find out which manual pages deal with *passwd*. Then view each page, giving the correct section number.

Note

If the **man -k** or **apropos** commands do not work, then you need to run the **makewhatis** command as user root. Normally, the **makewhatis** command is automatically executed nightly, but because your system is freshly installed, this might not yet have occurred.

Info command

__ 8. View the info documentation for the **finger** command.

Are you actually reading info documentation now?

__ 9. Move through this page by using the <space> and <backspace> keys.

__ 10. Read the help for the **info** command.

__ 11. End the **info** command.

__ 12. Read the info documentation of the **info** command.

__ 13. **info** has a nice built-in tutorial. If you have spare time during this course, look at the tutorial to see some of the advanced features of info.

Other documentation

__ 14. Make a listing of all directories in the `/usr/share/doc` directory.

Browse some of these directories to see what sort of information is available.

__ 15. If the classroom systems have an internet connection, look at the http://www.tldp.org website. This is the main documentation website for Linux.

Note that in some classrooms, some additional configuration of your web browser might be needed because the classroom could be behind a socks or proxy-based firewall. In this case, your instructor gives you additional instructions.

Find the HOWTO for *IPv6*.

__ 16. Visit http://www.google.com/linux. Search for *IPv6*. Does Google find the same HOWTO?

End of exercise

Exercise 6. Editing files

What this exercise is about

The purpose of this exercise is to give you the opportunity to create and edit files using the most common UNIX editor, Vi, and to try out a number of other editors that might be available.

What you should be able to do

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

- Use Vi to create and edit files
- List a few other editors that are available on your system

Introduction

In this exercise, you will be exploring the common file editor, Vi.

Requirements

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

Exercise instructions

Preface

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

Working with Vi

__ 1. If you aren't already logged in as tux1 at **tty1**, log in now.

__ 2. Ensure that you are in your home directory. Create a file in your home directory named `vitest` using Vi.

Type the following text and the marine alphabet into the `vitest` file. Adding the alphabet is an easy way to fill a couple of screens of information needed for later use. This is a training session about the usage of the Vi editor. We need some more lines to learn the most common commands of the editor.

```
a alpha
b bravo
c charlie
...
(the rest of the marine alphabet)
x x-ray
y yankee
z zulu
```

__ 3. Return to command mode. Write and quit the file. Notice that as soon as you press the colon (**:**), it appears below the last line of your input area. When the buffer is empty and the file is closed, you see a message giving the number of lines and characters in the file.

Cursor movement keys

__ 4. Open `vitest` file again. Notice that the bottom line of the screen indicates the name of the file and number of characters.

__ 5. Using the H, J, K, and L keys, practice moving the through the file.

__ 6. Use the appropriate Vi commands to move through the text.

__ a. Move forward one page.

__ b. Move back one page.

__ c. Move the cursor to the first line on the screen.

__ d. Move the cursor to the last line in the file.

__ e. Move the cursor to the first line in the file.

__ f. Move the cursor to line 5 of the file.

__ g. Move the cursor to the end of the line.

__ h. Move the cursor to the beginning of the line.

__ 7. Change the file `vitest` so that after each letter of the alphabet a common first name is added that starts with that letter. Make sure you use different methods for switching from command mode to insert mode.

__ 8. Practice some more with all the commands that are listed on your cheat sheet.

__ 9. Save the file but do not exit Vi.

Using set to customize the editing session

__ 10. Turn online numbering and set your tab stop to 4.

Global search and replace

__ 11. Replace all spaces in your file with tabs.

Hint

colon percent s slash space slash <Tab> slash g <Enter>)

__ 12. Save your file.

Working with other editors

__ 13. Your system has various other text mode and graphical editors available as well. Start some of these to get acquainted with them.

Note

All editors listed in the course material might not be available or installed on your distribution.

End of exercise

Exercise 7. Shell basics

What this exercise is about

This exercise provides an opportunity to get to know the basic features of the Linux shell (Bash).

What you should be able to do

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

- Use wildcards for file name expansion
- Redirect standard in, standard out, and standard error
- Use pipes to provide the output of one process as input to another process
- Perform command grouping and line continuation

Introduction

In this exercise, you will be exploring the common file editor, Vi.

Requirements

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

Exercise instructions

Preface

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

Wildcards

__ 1. If you are not logged in as **tux1** at **tty1**, log in now.

__ 2. Go to the `/etc` directory and make a list of all files here.

__ 3. Use **ls** with wildcards to list file names:

__ a. That end with *conf*

__ b. That begin with a *d* or *D*

__ c. That contain an *o* in the fifth position

__ d. That contain the word *tab* (in any combination with capitals and lowercase characters)

__ e. That end with a number

__ f. That do not end with a number

(Note that wildcard expansion is done by the shell. If one of the file names that matches is a directory name, then **ls** by default lists the contents of that directory instead of the file name itself. To prevent this, use the -**d** option.)

__ 4. What happens if you execute the command `ls -d ?[!y]*[e-g]?` What would the shortest file name be that can match? Execute this command to verify your answer.

__ 5. Return to your home directory.

Redirection

__ 6. Use the **cat** command and redirection to create a file called `junk` containing a few lines of text. When you have typed a few lines, end your input to the **cat** command and return to the shell prompt. Then view the contents of the file you just created.

__ 7. Append more lines to the `junk` file using redirection. Then view the contents of the file `junk` and check if all the lines you saved in this file are there.

Pipes, tees, and filters

__ 8. Count the number of files in your current directory. Use a pipe. Do not count the files manually.

__ 9. Does `ls > tempfile ; wc -l tempfile ; rm tempfile` do the same thing as the pipe you made in the previous command? Why or why not?

__ 10. Use the **ls** command and save the output in a file called `tempfile2` before you count the files.

__ 11. Use the **sed** command to alter the output of the `ls -l /etc/` command so that it looks like you own all files in `/etc`. Execute this both with and without the global option. What is the difference?

__ 12. Use the **awk** command to display the first and ninth column of the output of the `ls -l /etc/` command.

__ 13. Use the **tac** command to display the output of the **ls** command in reverse order.

__ 14. Use the **nl** command to number the lines of `tempfile2`.

__ 15. Use the **pr** command to format `tempfile2` for the printer.

__ 16. Combine all `usersfile` parts from the file and directory permissions exercise into one big file, called `usersfile5`. Check to see if this file is identical to the original `usersfile`.

Command grouping

__ 17. On the same command line, display the current system date and all the users that are logged in, together with some explaining comments, and save all this to one file after numbering the lines. Check your output.

Process environment

__ 18. Display all your variables that are defined in your current process environment. Also display all variables that are currently exported.

__ 19. Create a variable **x** and set its value to **10**. Check the value of the variable. Again, display all your current variables.

__ 20. Create a subshell. Check to see what value variable **x** holds in the subshell. What is the value of **x**? _____ List the subshell's current variables. Do you see a listing for **x**? _____

__ 21. Set the value of **x** to **500** and go back to your parent process. What is the current value of **x**? _____ Why?_____

__ 22. Make sure that child processes inherit the variable **x**. Verify this by creating a subshell and checking the value of variable **x**. After this, exit your subshell.

End of exercise

Exercise 8. Working with processes

What this exercise is about

This exercise familiarizes you with process manipulation and process control.

What you should be able to do

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

- Monitor processes
- Change and understand the process environment
- Control jobs
- Terminate processes

Introduction

In this exercise, you will be executing and manipulating Linux processes.

Requirements

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

Exercise instructions

Preface

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

Listing processes

__ 1. Log in at **tty1** as **tux1**.

__ 2. Check the PID of your log in environment and then create a subshell by entering **bash**. What is the process ID of the subshell? Is it different from your login process?

_____

_____

__ 3. Enter the command `ls -R / >outfile 2>/dev/null &` and then show the processes that you are running in the system. Which processes are running?

Note: This command is explained in full in the next units.

_____

_____

__ 4. While the **ls** command is still running, run the **pstree** command. (It might be necessary to restart the **ls** command.)

__ 5. Log in as **tux2** on **tty2** and run `vi tux2_file`.

__ 6. Go back to tty1 and show all the processes in your system. If necessary, look in the man pages and info to find the correct options to show all processes running in your system. Look for your own processes as well as the processes of tux2.

__ 7. Again, run the `ls -R / >outfile 2>/dev/null &` command and then exit your current process. List the processes you are running. What happens to processes if you kill their parent process?

Job control

__ 8. Using Vi or another editor, create the file named `myclock` in your bin directory with the following contents:

```
while true
do
   date
   sleep 10
done
```


Make the script executable.

__ 9. Run the script `myclock`. Run this script in the foreground.

__ 10. Suspend the job you just started.

__ 11. List all the jobs that you are running on the system and restart the above job in the background.

__ 12. List all users that are logged in. Bring the job back to the foreground, wait until you get a timestamp, and then exit the job.

Terminating a process

__ 13. Execute the `myclock` script again, this time in the background. **Hint**: Take note of the PID.

__ 14. List all your processes and kill the sleep process. What happened?

__ 15. Now stop the shell script `myclock`.

End of exercise

Exercise 9. Linux utilities

What this exercise is about

The purpose of this exercise is to become familiar with some of the many helpful tools available with Linux.

What you should be able to do

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

- Search for files that meet specific criteria
- List specific columns of a file
- Search text files for lines that match a pattern
- Sort lines in a file
- Display the first or last few lines of a file
- Find out where executables are located
- Compress files and decompress them

Introduction

In this exercise, you will be exploring various Linux utilities.

Requirements

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

Exercise instructions

Preface

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

Working with find and locate

__ 1. Log in as tux1 at tty1, if you aren't already.

__ 2. Find and display all files and directories in your home directory.

__ 3. Find all files in your system that begin with the string *abc* and have `ls -l` automatically executed on each file name found. Discard all errors.

__ 4. Repeat the previous command, but interactively prompt the user to display the long listing on each file. Do not discard errors, since stderr is used to display the prompt.

__ 5. Find all files starting from `/usr` that are owned by the user root.

__ 6. Modify the last command to count the number of files on the whole system owned by root. Now alter the command so that you don't get error messages on your screen.

__ 7. Find all directories in your system and save this list in the file `all.directories`. The error message can be sent to the bit bucket. Execute this command in the background.

__ 8. Fedora/Red Hat only: Use the **locate** command to locate all files that match the string *passwd*.

Note: SuSE does not install the **locate** command by default. You learn how to install the **findutils-locate** in the basic system configuration exercise.

Working with cut

__ 9. Display the contents of the `/etc/passwd` file.

__ 10. Only show the user name and the home directory of the users listed in `/etc/passwd`.

__ 11. Show the name and the members of all groups listed in `/etc/group`.

__ 12. List only the type, size, and name of files in the current directory.

Working with grep

__ 13. Find all lines in the `/etc/passwd` that begin with the letter *s*.

__ 14. Repeat the search in the previous instruction, but this time display only the number of lines that contain the pattern.

__ 15. Find all processes running on the system, owned by user tux1 or tux2.

Working with sort

__ 16. Display the contents of the `/etc/passwd` file in alphabetical order.

__ 17. Display the contents of /etc/passwd again but now sorted on the home directory field.

Working with head and tail

__ 18. Display the first 10 lines of the `/etc/passwd` file.

__ 19. Display the last 6 lines of the `/etc/passwd` file.

__ 20. The **tail** command is also handy for stripping out header information from the output of a command. First, list the processes currently running on your system. Notice the headings. Next, display the processes running on your system excluding the header information.

Working with type, which, and whereis

__ 21. Find out where the **passwd** command is stored. Locate the manual pages and source code for this command.

Working with gzip, gunzip, and zcat

__ 22. Create a big file named `big` in your home directory, for instance by capturing the output of the `ls -lR /` command. What is the size of `big`?

__ 23. Make the file twice as large.

__ 24. Note the size of `big`._____ Compress the `big` file. What is the new size of the file and what is its new name? _____

__ 25. Look at the contents of the `big.gz` file.

__ 26. Restore the old `big` file. What is the size of `big` and what is its name? _____

End of exercise

Exercise 10.Shell scripting

What this exercise is about

After you have been using Linux for a while, you find certain characteristics of your environment that you would like to customize along with some tasks that you execute regularly that you would like to automate.

This exercise introduces you to some of the more common constructs used to help you write shell scripts to customize and automate your computing environment.

What you should be able to do

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

- List common constructs used in writing shell scripts
- Create and execute simple shell scripts

Introduction

You need no programming experience to perform this exercise. Refer to the unit in the Student Notebook for help with the syntax of constructs when creating the shell scripts in this exercise.

Requirements

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

Exercise instructions

Preface

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

Working with positional parameters

__ 1. If you are not logged in as tux1 at tty1, log in now.

__ 2. Create a shell script named `parameters` that echoes the five parameters that follow using predefined special variables set by the shell to fill in the blanks. Execute the script using the positional parameters 10 100 1000.

Conditional execution

__ 3. Using conditional execution, create a shell script named `checkfile` that checks to see if the file named `parameters` exists in your directory. If it exists, use a command to show the contents of the file. Execute the script.

__ 4. Modify the `checkfile` script and change the name of the file from `parameters` to `noname` (check to ensure that you do not have a file by this name in your current directory). Also, using conditional execution, if the **cat** command was not successful, display the error message, `The file was not found.` Execute the script.

__ 5. Modify the `checkfile` script to accept a single parameter from the command line as input to the **ls** and **cat** commands. Execute the script twice, once using the file named `parameters` and again using the file named `noname`.

__ 6. Execute the `checkfile` script again but this time with no parameters. What happens? Modify the script so that this does not happen again.

Loops

__ 7. Using the **for** loop, modify the `checkfile` script to accept multiple files as input from the command line instead of just one. If the files are found, display all of them. If the files are not found, display an error message showing all file names that were not found. Look in your directory and note a few valid file names that you can use as input. Execute the script using valid and invalid file names.

__ 8. Now do the same thing, but use a **while** loop in combination with the **shift** command.

Arithmetic

__ 9. From the command line, display the results of multiplying 5 times 6.

__ 10. Now create a shell script named `math` to multiply any two numbers when entered as input from the command line. Execute the script multiplying 5 times 6. Experiment with any other two numbers.

Integration exercise

__ 11. Use the knowledge you gained in this course to write a script that accepts a directory name as a parameter and calculate the total size of the files in this directory.

Note: The column numbers might need to be adjusted a little.

End of exercise

Exercise 11. The Linux GUI

What this exercise is about

This exercise provides you with an opportunity to get acquainted with the two main Linux desktop environments: KDE and GNOME.

What you should be able to do

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

- Start X
- Work with GNOME
- Work with KDE
- List and compare various applications within GNOME and KDE

Requirements

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

Exercise instructions

Preface

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

Starting the GUI

__ 1. Log in as **root** on **tty4**.

__ 2. Edit the `/etc/inittab` file and make sure the default runlevel is 3.

__ 3. Reboot your system. Does the graphical environment get started?

__ 4. Log in as tux1 on tty1 and start X with the **startx** command.

__ 5. End your X environment, then log out and log in as **root**.

__ 6. Edit the `/etc/inittab` file again and set the default runlevel to 5. Then reboot the system again. Did the graphical environment start?

Working with GNOME and KDE

__ 7. Log in to the graphical environment using your own name.

__ 8. Both the GNOME and KDE project have delivered various applications, such as word processors, file managers, text editors, and so forth, as standard part of the codebase. These applications are typically direct competitors of the corresponding applications on a Microsoft Windows platform.

A default Linux installation installs a lot of these applications, and you can download more from the GNOME and KDE Web sites.

Browse around in both the GNOME and KDE desktop environments and try to identify the name of the application that fulfills a certain function. (You can retrieve the name of the application by opening a terminal window and executing the **ps** command.) Some names have already been filled in as an example.

To switch between KDE and GNOME, use your display managers (login prompt) menu.

Function	GNOME	KDE
Window manager	sawfish, metacity	kwin
File manager	nautilus	konqueror
Text editor(s)		
Internet dialer		
Email client		
Web browser		
CD Player		
MP3 Player		
Sound mixer		
Word processor		
Spreadsheet		
Presentation package		
Photo/bitmap editor		
Vector oriented graphics editor		
Clipboard		

__ 9. In both desktop environments, explore the themes capabilities. After setting a theme in KDE, start a GNOME application, and vice versa. Does this work?

__ 10. In KDE, try to start a GNOME application and vice versa. Does this work? Try to cut and paste between KDE and GNOME applications. Does this work?

End of exercise

Exercise 12. Customizing the user environment

What this exercise is about

When users log in, they generally prefer their environment to be customized to meet their specific needs. In this exercise, you will customize your environment with some very useful functions that are invoked every time you log in.

What you should be able to do

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

- Customize the `.bash_profile` and `.bashrc` files
- Set alias definitions
- Alter umask values

Requirements

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

Exercise instructions

Preface

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

Customizing the shell environment

__ 1. If you are not logged in, log in as **tux1** at **tty1**.

__ 2. Change the appropriate file to change your environment each time you log in. Make sure that you have the following functions when you log in:

__ a. Change the primary prompt to show you the complete path of the current directory.

__ b. Display a message stating your login name and the date you logged in.

__ c. Define an alias **num** that shows you how many users are logged in at that moment.

__ d. Set the variable **cheese** to **gouda**.

__ 3. Log out and log in again. Check if the functions you defined in step 1 are activated.

__ a. Does your prompt show complete path of the current directory? _____

__ b. Did your message display? _____

__ c. Can you use the **num** command? _____

__ d. Is the variable **cheese** set to **gouda**? _____

__ 4. If all the previous questions are answered with yes, continue with step 5. Otherwise, try steps 2 and 3 again to fix the problems.

__ 5. Start a subshell and answer the following questions.

- Does your prompt show the complete path of the current directory? _____
- Did your message display? _____
- Can you use the **num** command? _____
- Can you use the command history with **Vi**? _____
- Is the variable **cheese** set to **gouda**? _____

__ 6. If the settings are also available in subshells, continue with step 9. Otherwise, continue with step 7.

__ 7. Most settings, with the exception of system variables, apply only to the current environment and are not passed to subshells (child processes). There is a configuration file in your system that makes settings available in subprocesses too. Which file is this? _____

__ 8. Edit the `.bash_profile` and `.bashrc` files so that the correct settings are in the correct configuration file. What settings should be in `.bash_profile` and what settings should be in `.bashrc`?

_____

_____

__ 9. Log out and log in again and see if your settings are set in your login environment. Also check if the settings are set in a subshell.

__ 10. In the previous steps, you altered configuration files and then logged out and in to activate the new settings. How can you activate settings in an altered customization file without logging out and in again?

__ 11. If you are not in your login shell, return there now.

__ 12. Remove the **num** alias from your environment without editing the `.bashrc` or `.bash_profile` file. Then display the list of aliases currently set and try to execute the **num** alias.

__ 13. Add the **num** alias to your environment and check if **num** is there again.

Customizing the X environment

__ 14. Switch to virtual terminal 7 and log in using your own name. Open a few applications, change some themes, and log out. While logging out, select the **Save session** check box. Then log in again. Do your applications and settings appear again?

End of exercise

Exercise 13.Basic system configuration

What this exercise is about

This exercise provides you with an opportunity to become familiar with basic system configuration that might be needed on a workstation.

What you should be able to do

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

- Install and uninstall RPMs
- Configure a printer
- Configure a sound card
- Configure the network interface

Requirements

- This workbook
- A set of installation CDs for your distribution or access to the NFS installation media directories on the classroom server

Exercise instructions

Preface

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

The RPM Package Manager

__ 1. Log in as root in your graphical environment. Open a terminal window.

__ 2. Make a list of all packages that are installed on the system.

__ 3. List the information of the bash package.

__ 4. List all files in the bash package.

__ 5. List all the package files that are available on the distribution CD-ROMs or network install server.

Note

<*server*> is 10.0.0.1.

<*dir*> is `/export/fedo7` or `/export/rhel51c` or `/export/sled10`.

__ 6. Remember the **vlock** command that we tried to use in a previous exercise? We could not do that exercise because **vlock** was not installed. Now that you know how to install an RPM, install the **vlock** RPM, and try to perform that particular exercise once more.

Before you install the **vlock** RPM, list the information of the RPM, and list all files in the RPM.

The **vlock** RPM is in `/export/rhel51c` or `/export/sles10`; however, if you are using Fedora 7, you will not find the **vlock** RPM file in `/export/fedo7`. You will have to access the **vlock** RPM from the instructor's `/export/files` directory. Ask for help.

__ 7. Verify that the application **vlock** is indeed installed by performing the exercises from the using the system exercise.

__ 8. Uninstall **vlock** and verify that it indeed is no longer available.

Configuring a printer (optional)

If a printer is available in your classroom (either locally attached to your system or remotely through the network), your instructor will provide you with the information about this printer. If no printer is available, skip this exercise.

__ 9. Use your browser to configure your printer.

__ 10. Print the `/etc/passwd` file.

Configuring a sound card (optional)

__ 11. Use the sound card configuration tool that came with your distribution and configure your sound card. Then try to play some audio.

Configuring your network (optional)

In most classrooms, it is not possible to alter the network configuration since this might lead to network problems which might also affect other classes that are currently running. If it is safe to play with network settings, your instructor will give you additional exercises to perform.

__ 12. Browse the files where the network configuration for your system is stored.

__ 13. If the classroom uses DHCP to configure your network card, then take a look at the current configuration with the **ifconfig** and **route** commands.

__ 14. Ask your instructor for permission to modify the current network settings. This is a safety issue because a wrong network configuration might lead to problems for other students -- even students in other classrooms! If you obtained permission, start the configuration tool that is appropriate for your distribution and configure static networking, using the IP address, netmask, and default gateway that you saw in the previous exercise.

End of exercise

Exercise 14.Integrating Linux in a Windows environment

What this exercise is about

This exercise provides you with an opportunity to become familiar with the different options when integrating Linux in a Windows environment.
Note: Because VMWare require a commercial license, it cannot be demonstrated in this class.
WINE requires extensive configuration and is, therefore, not included in the exercises.

What you should be able to do

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

- Access Windows file systems
- Access Windows servers

Requirements

- This workbook
- The NetBIOS name of a Windows server, and the name and password of a user account/home directory on that server

Exercise instructions

Preface

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

Accessing Windows file systems

__ 1. Make a list of all partitions that exist on your system with the `fdisk -l /dev/hda` or `fdisk -l /dev/sda` (SCSI) command. Remember which one you have. The rest of the exercise refers to `/dev/hda`. Use the correct designation.

__ 2. List all file systems that are currently mounted with the **mount** command. Compare this list with the output of the previous command. This should give you a list of Windows file systems that are not mounted yet.

__ 3. Create mountpoints under /mnt for all Windows file systems that you want to mount. Mount these file systems manually, using the **mount** command. Verify that the file system was indeed mounted and list the contents of the file system.

__ 4. Add a line to the `/etc/fstab` file so that this file system is mounted automatically when the system boots, and reboot the system to verify that this worked.

Accessing Windows servers

Your instructor configures a Windows server or a Samba server so that you can access this using the Samba client software, which is part of Linux. He or she provides you with the following information about this server:

- Netbios name
- Share name
- User name
- Password

__ 5. Use the **smbclient** program to retrieve information from the Windows or Samba server. Then use it to access the share ftp-style. Upload and download the `/etc/passwd` file to test if things are working.

__ 6. Create a mount point for this Windows share called `/mnt/share`.

__ 7. Mount the share on this mountpoint using the **smbmount** command. Verify that the mount succeeded.

__ 8. Edit the `/etc/fstab` file and add an entry for this share. Then reboot the system and verify that the share was mounted after the reboot.

OpenOffice

__ 9. Start OpenOffice and try to create and save various types of documents: text documents, presentations, spreadsheets, and so forth. Try to save them in Microsoft-compatible formats.

__ 10. Start an Internet browser and use Google to search for and obtain various documents in Microsoft formats (search, for instance, for test.doc, test.ppt, and test.xls). See if you can open, modify, and save these files.

End of exercise