MODULE - I

Linux Introduction

MODULE 1

Linux Introduction

Module Description

The main goal of studying Linux introduction is to begin your tour to UNIX/Linux universe. UNIX had a turbulent background. Knowing this background will help to understand the objectives that guided its development. This module explains the unique features of UNIX and its architecture. Ubuntu and its features are also discussed in this module as it is the most popular Linux distribution. To start your tour to UNIX/Linux universe, commands related to login, file system and basic commands are introduced.

In this module, students will learn basic features of UNIX/Linux operating systems that make them to consider UNIX as one of the favourite operating system of the world. Students also would be familiar with various flavours of Linux.

At the end of this module a student would be able to login to a UNIX/Linux machine, create and manipulate files and check disk status as well.

Chapter 1.1
Fundamentals of Linux

Chapter 1.2
Exploring Linux Flavours and Commands.

Chapter Table of Contents

Chapter 1.1

Fundamentals of Linux

Aim

To introduce fundamental Linux concepts to students

Instructional Objectives

After completing this chapter, you should be able to:

- Describe the multi-user nature of UNIX
- Explain the evolution of UNIX along with its versions
- Describe the features and benefits of UNIX
- Explain UNIX system architecture
- Demonstrate how to login and logout UNIX machine in different terminals
- Illustrate how to create and view files using cat command
- Perform file comparison and view the required contents of a file
- Describe the commands for checking disk free space and disk used space

Learning Outcomes

At the end of this chapter, you are expected to:

- Outline the advantages of multi user system
- Summarise the growth of UNIX over the years
- List the features and benefits of UNIX
- Describe the three major components of UNIX system
- Perform login and logout in UNIX machine
- Demonstrate file creation and view files using cat command
- Demonstrate the commands to check disk free space and disk used space

1.1.1 Introduction

UNIX is a reliable, multiuser and secure operating system. UNIX implemented the concept of cross platform standardisation, due to which you can find UNIX on massive mainframes, on distributed clusters, on PCs, on Apples, on tablets, on smartphones, on wristwatches, in automobiles etc. UNIX continues to dominate web and application hosting landscape.

While UNIX is adored by academicians, Linux has a great commercial domination. Linux is open source unlike other UNIX clones. Linux is recognised as one of the best operating system in the world because of its simplicity, its ability to run on different machines, open standard design. If we compare Linux with Windows, Linux is more reliable, secure and cheap. In this chapter we will take an overview of UNIX/Linux operating system.

1.1.2 Introduction to Multiuser System

Let us start our journey of an operating system (i.e. Linux) who has been popular for last two decades due to its multi-user, multi-tasking environment, stability, portability and powerful networking capabilities. Linux is a multiuser system. In a multiuser system, resources like hard disk/memory/ ram/ application programs are accessible to multiple users at same time. In a multiuser system, all computers are connected to main computer whose resources are availed by all the users. This kind of setup is highly economical when same data is to be shared by all users. The following figure 1.1.2 shows a multiuser system setup.

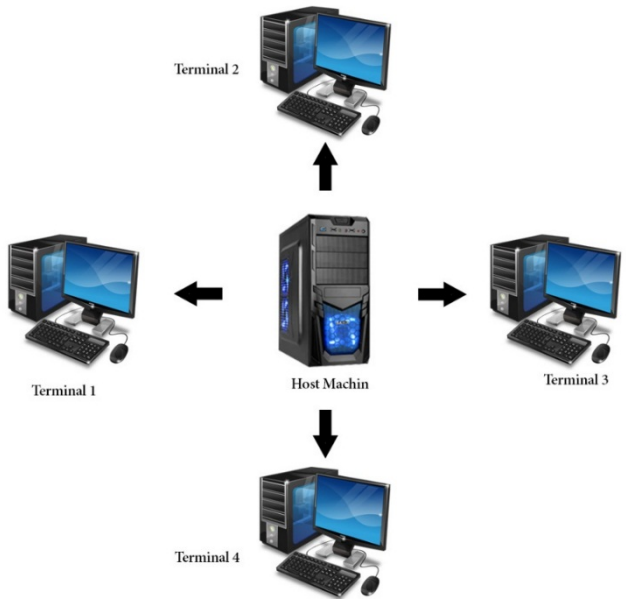

Figure 1.1.1 Multiuser System

As you can see above, the heart of this multiuser system is a host machine, which is also known as a server or console. Various terminals can be connected to host machine. The number of terminals that can be connected to the host machine depends on the number of ports that are present in its controller card.

Advantages of Linux Multiuser System:

- In a multiuser system, a user not only uses the connected computer but also uses the peripherals attached to it, say for instance a printer. Printing jobs in the office or library can be best handled by multi user operating system.
- Each user can access same data on their own computer. For example if one computer has a file/song/movie then other computers attached with it can play that song or movie on their computer also.
- If one computer in the network gets error then other computers not get affected and that system handles this efficiently.
- Multiple users can run multiple programs each at the same time without interfering with each other or crashing the system.

1.1.3 History and Versions of UNIX

Linux operating system was modelled on the UNIX operating system. UNIX was developed by Dennis Ritchie and Ken Thompson at AT&T (American Multinational Company). The main objective to develop UNIX was to run the largest network in world i.e. the AT&T telephone system. The UNIX operating system was designed to be scalable, reliable, modular, secure and portable with network extensibility.

Journey of UNIX system started in 1965 with Bell Telephone Laboratories, the General Electric Company (GE) and Massachusetts Institute of Technology (MIT). Objective was to develop a new multiuser/multitasking operating system. As a result of this project Multiplexed Information and Computing System (MULTICS) was developed. This operating system was not successful.

A simple version of MULTICS (It was developed only for 2 users) was written by Dennis Ritchie and Ken Thompson, in 1969 at AT& T.

Thus, in 1969, a simpler version Uniplexed Information and Computing System (UNICS) was developed on a PDP-7(a minicomputer introduced in 1965). During this time, memory and CPU power were considered as premium resources, short commands were used to minimize the space needed to store them and the time needed to decode them. Later, in 1971 UNIX was ported to PDP-11 computer. Again, this version of UNIX was not portable as its assembly code was machine dependent.

Figure 1.1.2 Ken Thompson and Dennis Ritchie, working on PDP-11 and UNIX in 1972

(Source-http://history-computer.com/ModernComputer/Software/Unix.html)

Ken Thompson created a new language 'B' to overcome the issue of portability. Denis Ritchie fixed the problems of new language 'B' and modified it to a new language 'C'. Later on, Ritchie and Thompson completed the task of rewriting the UNIX code in 'C' language.. This was a big step towards the system's portability. In 1974, Fifth Edition of UNIX was released for universities.

Now UNIX had come a long way from its PDP-7 days, and was soon licensed to quite a number of Universities, companies and other commercial institutions. With the time, UNIX

capabilities were enhanced. The Seventh Edition, released in 1978, marked a split in UNIX development into two main branches: SYSV (System 5) and BSD (Berkeley Software Distribution). BSD arose from the University of California at Berkeley where Ken Thompson spent a retreat year. Its development was continued by students at Berkeley and other research institutions. SYSV was developed by AT&T and other commercial companies. UNIX flavours based on SYSV have traditionally been more conservative, but better supported than BSD-based flavours.

The latest versions of SYSV (SVR4 or System 5 Release 4) and BSD Unix are actually very similar.

By mid-eighties UNIX had more than a hundred thousand installations. UNIX was so much endorsed by Academia and industry that still it has the record of being soul of most of computer networks than any other OS.

By the mid-eighties there were more than.

In 1991, Linus Torvalds developed a UNIX like operating system i.e. Linux but unlike UNIX, it is a free open source.

Linux incorporates the features of SYSV as well as pure BSD. Linux also supports IEEE POSIX (Portable Operating System Interface).Following Figure 1.1.3 shows the evolution of Linux operating system.

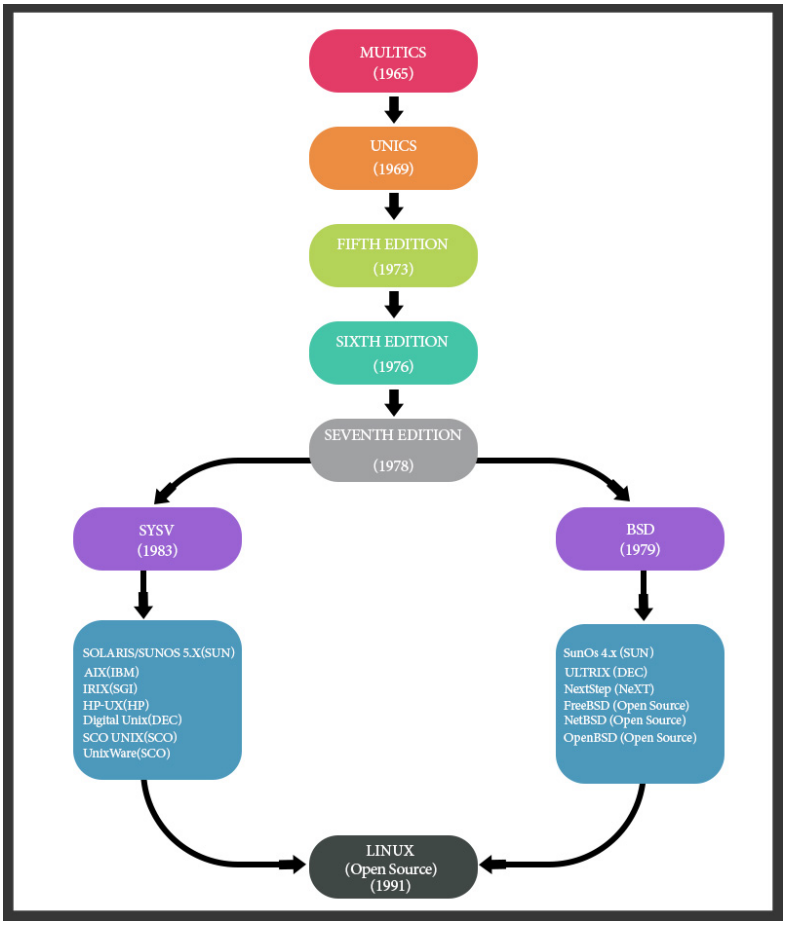

Figure1.1.3 Evolution of Linux

1.1.4 Features and Benefits of UNIX

Back in 1969, UNIX started out as a small and limited operating system. Throughout its journey, it evolved as a featured-packed operating system that is utilized by many businesses and companies for everyday business operations. Even though Windows is the preferred option among desktop computer users all over the world, UNIX is capable of running computers utilized in various functions such as electronic commerce, management of phone systems and even the Internet. Following are some of the important features of UNIX Operating System:

- **Portable** –UNIX is has gained popularity as it is highly portable. UNIX is portable as 70% of its code is written in 'C' language which guarantees hardware transparency.
- **Multi-User** –In UNIX system multiple users can access system resources like memory/ ram/ application programs at same time.

- **Multiprogramming** – UNIX system allows execution of multiple applications atthe same time. It allows you to type a document in one of its editor while some printing work is going on and you are listening some online music.
- **Hierarchical File System** - UNIX system treats everything as file including the terminal and all devices. All utilities, applications, data in UNIX is stored as files. The UNIX file system resembles an upside down tree. Root directory is on the top of this hierarchy. There are many sub directories under root like bin, lib, usr, etc, tmp and dev.
- **Shell** It is a user interface which is used to execute commands of the operating system.
- **Communication** – UNIX system allow its users to communicate with fellow users who may be on same machine or could be on other network. By using this communication system, users can easily exchange mail, data, and program through network.
- **Security** - UNIX provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data. UNIX allow sharing of data, but with security. UNIX has three inherent ways for protecting the data. The first way is provided by assigning the passwords and login names to users. At file level, there are read, write and execute permissions to each file which decide who can access file, who can modify it and who can execute it. Lastly, there is file encryption. This utility encodes your file into an unreadable format.

UNIX provides all above features while using a minimal amount of memory. UNIX makes a great platform for utility servers such as Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP) and Web servers. UNIX also provides a stable and secure platform for applications that do not require direct user interaction with the operating system.

1.1.5 UNIX System Architecture

" Success of UNIX lies not so much in new inventions but rather in the full exploitation of a carefully selected set of fertile ideas, and especially in showing that they can be keys to the implementation of a small and yet powerful operating system."- According to Thompson and Ritchie. UNIX is no longer a small system, but it is certainly a powerful one. Before we examine the UNIX features, we need to understand its software architecture as shown in figure 1.1.4.

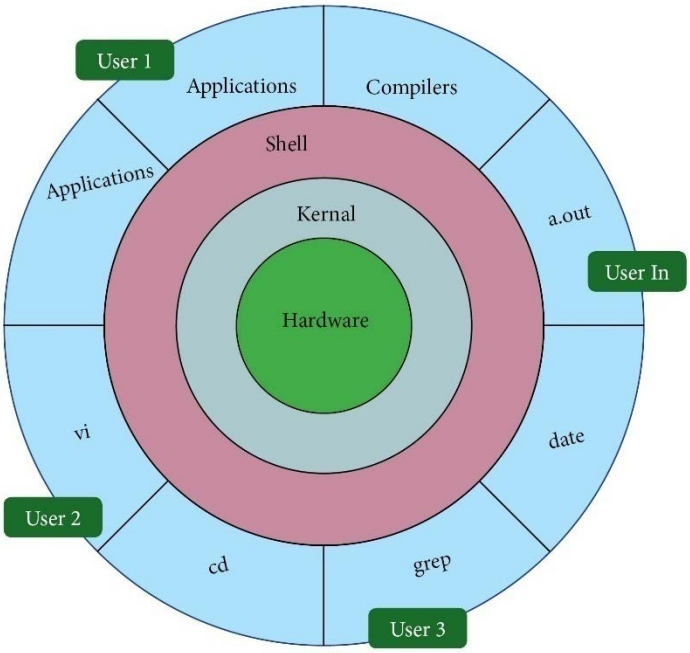

Figure 1.1.4 UNIX System Architecture

UNIX System Architecture is consists of following layers

- Hardware layer
- Kernel
- Shell
- Utilities

Among the "fertile idea" is division of work between kernel and shell. Kernel interacts with machine hardware, and the shell interacts with the user.

Hardware layer -This layer is not part of UNIX operating system. This layer consists of all peripheral devices like RAM, hard disk, CPU.

Kernel –Kernel is the core component of Operating System where actual code and functionality of operating system lies. The kernel interacts with the hardware layer and provides services to user through upper layer. If a user program/ application need to access the hardware, it uses the services of kernel, which performs the job on user behalf. The user programs access the kernel through a set of functions called system calls. The kernel architecture support functions in two categories namely, functions for process management

and functions for file management (files include device files). Process management entails allocation of resources including CPU, memory, and offers services that processes may need. The file management in itself involves handling all the files required by processes, communication with device drives and regulating transmission of data to and from peripherals.

In summary, we can say that the kernel handles the following operations:

- It is responsible for scheduling running of user and other processes.
- It is responsible for allocating memory.
- It is responsible for managing the swapping between memory and disk.
- It is responsible for moving data to and from the peripherals.
- It receives service requests from the processes and honours them.

All these services are provided by the kernel through a call to a system utility.

Shell - An interface for the user through which he can interact with the operating system. A shell environment allows you to run commands, programs and shell scripts. There are different types of shells. In UNIX there are two major types of shells:

- Bourne Shell
- C Shell

The default prompt for Bourne shell is $ character and for C shell is % character. Bourne shell was the first shell on UNIX operating system and generally installed as /bin/sh on most of the systems.

Utilities - Utility programs are the commands that perform a single task like printing date and time or searching a file in given directory. Different commands can be combined through pipelines (|). Some examples of UNIX utilities are sh, cp, ls, ws, mv, rm, wc,vi.

The kernel program is usually stored running in a file called 'unix' whereas the shell program is in a file called 'sh'. For each user working with UNIX at any time different shell programs are running. Thus at a particular point in time there may be several shells running in memory but only one kernel. This is because, at any instance UNIX is capable of executing only one program as the other programs wait for their turn. As kernel executes the program one kernel is

sufficient. However, different users at different terminals are trying to seek kernel's attention. As the users interacts with the kernel through the shell different shells are necessary.

Kernel Mode vs. User Mode

Kernel mode is used to perform lower level functionalities on the other side in user mode, the code has no access to hardware or lower level resources like memory. Code running in user mode delegate the task to system APIs to access lower level resources.

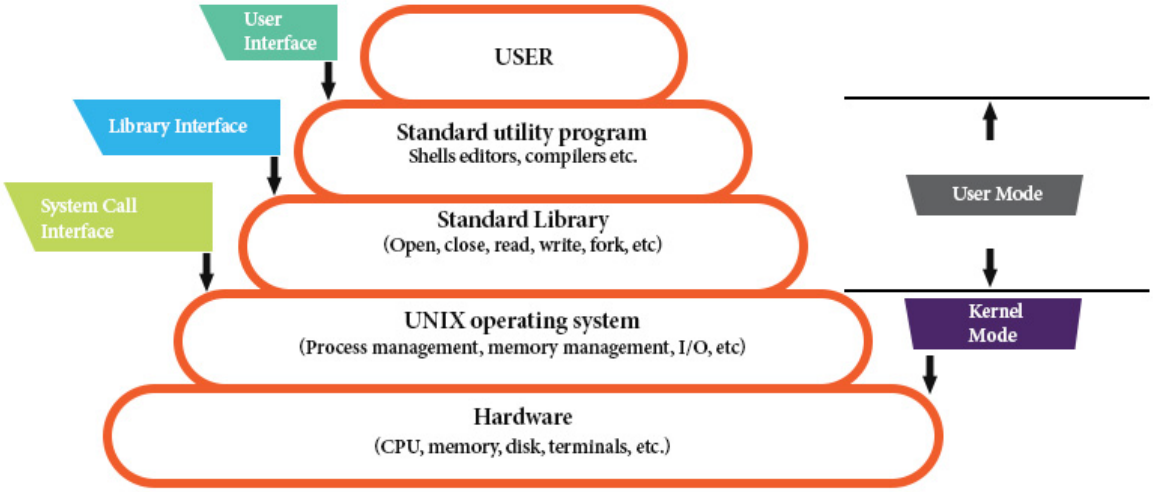

Figure 1.1.5: Kernel mode Vs User mode

1.1.6 Getting Started (Login/Logout)

UNIX is security conscious, and all the users are required to maintain an account with computer system. The list of accounts is maintained separately in the computer. System administrator is the person who grants the authority to use the system. He opens an account for you with a username and password. If you are running UNIX on your desktop, then you are the administrator of your machine.

Suppose you have a username "XYZ". When you connect to a UNIX computer remotely or when you log in locally, you will see the prompt:

```
login:
```

At this prompt, type in your usename and press the enter/return/↵ key. Remember that UNIX is case sensitive (i.e. xyz, xYZ and xyZ are all different logins).

You should then be prompted for your password:

```
login: XYZ
password:
```

Type your password in at the prompt and press theenter/return/↵ key. Note that your password will not be displayed on the screen as you type it in.

If you mistype your username or password you will get an appropriate message from the computer and you will be presented with the login: prompt again. Otherwise you should be presented with a shell prompt which looks something like this:

```
$
```

This is a typical UNIX prompt, and many UNIX systems use the $ as the default prompt string. For some users, you might see the % instead of the $ and the administrator generally use # symbol. UNIX allows you to customise the prompt.

Following screenshot (figure1.1.6) gives you an example of Login scenario on a Ubuntu 14.04 system:

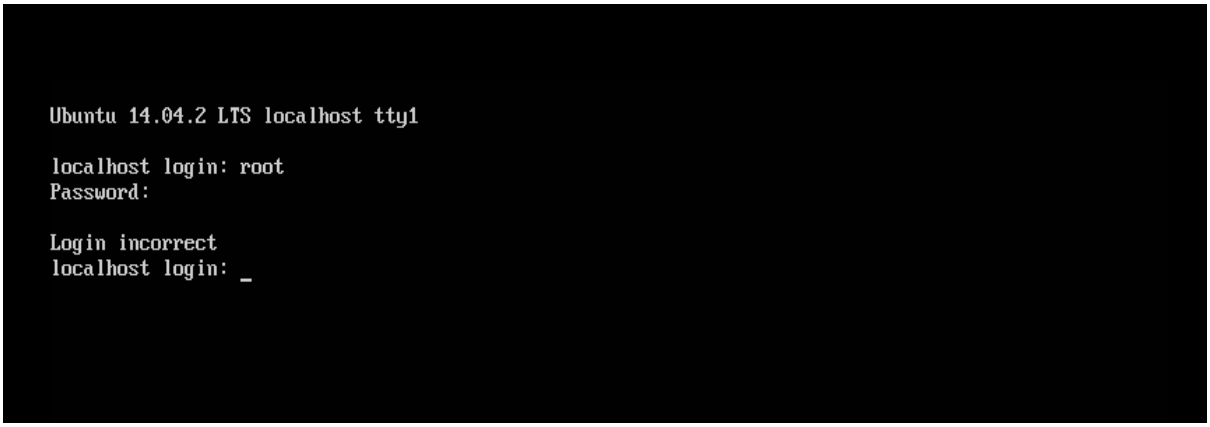

Figure 1.1.6: login screen in Ubuntu 14.04

To log out of a text-based UNIX shell, type "exit" at the shell prompt (or if that doesn't work try "logout"; if that doesn't work press ctrl-d).

1.1.7 Manipulating Files

A file system is a logical method to organise and store data. The UNIX filesystem is organised in a tree structure. The top-level directory is known as the root (designated by a slash '/). In a tree structure, a directory can have many child directories. This tree structure has only one root directory. Figure 1.1.7. Illustrates this layout.

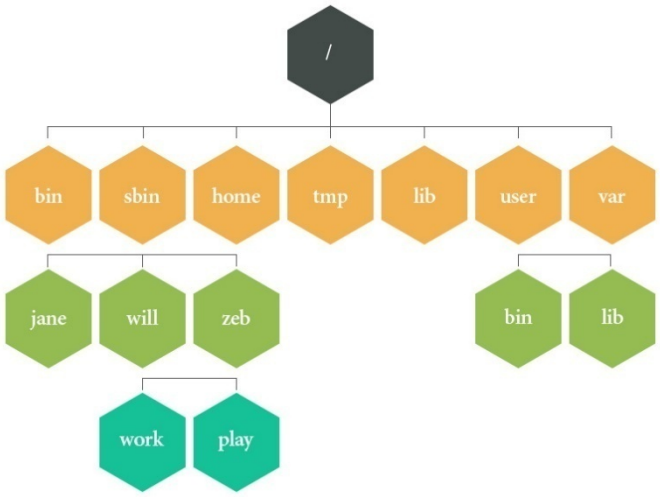

Figure 1.1.7: UNIX directory structure

As UNIX is a multiuser system, many users can login to UNIX system at a time and work together. Each user is assigned a home directory. It is the home directory of a user where he is placed after login to UNIX system. To identify location of a user in a file or /directory, we must specify a path through the tree.

The absolute path to a location can be defined by a complete list of the names of the directories that are on the route from the root directory to that file or directory, with each directory separated by a "/".

The relative path to a location uses "." to specify current directory and ".." for the parent directory. For example, the absolute path to the directory "abc" is /home/XYZ/abc, while the relative path to this directory is../XYZ/abc.

- **Creating and displaying files using cat command**
- **File comparisons**
- **Viewing Files**

`cat` is one of the most frequently used commands on UNIX. The three basic functions of cat are: displaying files, combining copies of them and creating new ones.

`cat's` general syntax is

```
cat [options] [filenames] [-] [filenames]
```

The square brackets indicate that the enclosed items are optional.

Displaying Files/Reading Files

The most common use of cat is to read the contents of files, and cat is often the most convenient program for this purpose. To open a text file for reading on the display monitor, type the word cat followed by a space and the name of the file and then press the ENTER key. For example, the following will display the contents of a file named *file1*:

```
$ cat file1
```

The standard output for cat, is generally the monitor screen. However, it can be redirected from the screen, to another file to be written to that file or to another command to use as the input for that command.

In the following example, the standard output of cat is redirected to *file2* using the *output redirection operator* (which is represented by a rightward pointing angular bracket i.e.'>'):

```
$ cat file1 > file2
```

Here, the output from cat is written to *file2* instead of being displayed on the monitor screen.

The *standard input* for cat is generally the keyboard. That is, if no file is specified for it to open, cat will read whatever is typed in on the keyboard.

Typing the command *cat* followed by the output redirection operator and a file name on the same line, pressing ENTER to move to the next line, then typing some text and finally pressing ENTER again causes the text to be written to that file. Thus, in the following example the text that is typed on the second line will be written to a file named *Tutorial*:

```
$ cat> tutorial
$ This is a CAT command.
```

The program is terminated and the normal command prompt is restored by pressing the CONTROL and d keys simultaneously.

Repeating the above example without using a redirection operator and specifying a destination file, i.e.

```
$ cat
$ This is a CAT command.
```

causes the text to be sent to standard output, i.e., to be repeated on the monitor screen.

Concatenation

The other role of cat is *concatenation* of the contents of files. (This is the source of cat's curious name.) Because the concatenation occurs only to the copies, there is no effect on the original files.

Simply put, buyback of stock is the reduction or cancellation of share capital. It is a repurchase of outstanding shares by company itself to reduce the shares in the market. For example, the following command will concatenate copies of the contents of the three files *file1*, *file2* and *file3*:

```
$ cat file1 file2 file3
```

The contents of each file will be displayed on the monitor screen starting on a new line and in the order that the file names appear in the command. This output could just as easily be redirected using the output redirection operator to another file, such as *file4*, using the following:

```
$ cat file1 file2 file3 > file4
```

File Creation

The third use for cat is file creation. For small files this is often easier than using *vi*, *gedit* or other text editors. It is accomplished by typing *cat* followed by the output redirection operator and the name of the file to be created, then pressing ENTER and finally simultaneously pressing the CONTROL and *d* keys. For example, a new file named *file1* can be created by typing

```
$ cat> file1
```

then pressing the ENTER key and finally simultaneously pressing the CONTROL and *d* keys.

If a file named *file1* already exists, it will be *overwritten* by the new, empty file with the same name. Thus the cautious user might prefer to instead use the *append operator* (represented by two successive rightward pointing angular brackets) in order to prevent unintended erasure. That is,

```
$ cat>> file1
```

That is, if an attempt is made to create a file by using cat and the append operator, and the new file has the same name as an existing file, the existing file is, in fact, preserved rather than overwritten, and any new text is added to the end of the existing file.

Text can be entered at the time of file creation by typing it in after pressing the ENTER key. Any amount of text can be typed, including text on multiple lines.

cat can also be used to simultaneously create a new file and transfer to it the data from an existing file. This is accomplished by typing *cat*, the name of the file from which the output will come, the output redirection operator and the name of the file to be created. Then pressing ENTER causes the new file to be created and written to. For example, typing the following and then pressing ENTER creates a new file named *file2* that contains a copy of the contents of *file1*:

```
cat file1 > file2
```

1.1.8 Disk Related command

How do I check free disk space in UNIX operating system?

UNIX offers two commands for checking out free disk space:

(a) df command : Report file system disk space usage.
(b) du command : Estimate file space usage.

Type df -h or df -k to list free disk space:

$ df -h

OR

$ df –k

Outputs:

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sdb1	20G	9.2G	9.6G	49%	/
varrun	393M	144k	393M	1%	/var/run

```
varlock            393M     0   393M   0% /var/lock
procbususb          393M   123k   393M   1% /proc/bus/usb
udev               393M   123k   393M   1% /dev
devshm              393M     0   393M   0% /dev/shm
lrm                393M    35M   359M   9% /lib/modules/2.6.20-15-generic/volatile
/dev/sdb5           29G   5.4G    22G  20% /media/docs
/dev/sdb3           30G   5.9G    23G  21% /media/isomp3s
/dev/sda1          8.5G   4.3G   4.3G  51% /media/xp1
/dev/sda2           12G   6.5G   5.2G  56% /media/xp2
/dev/sdc1           40G   3.1G    35G   9% /media/backup
```

The df utility displays statistics about the amount of free disk space on the specified file system or on the file system of which file is a part. Values are displayed in 512-byte per block counts. -H option is called as “Human-readable” output. It use unit suffixes: Byte, Kilobyte, Megabyte, Gigabyte, Terabyte and Petabyte in order to reduce the number of digits to four or fewer using base 10 for sizes i.e. you see 30G (30 Gigabyte).

du command examples

du shows how much space one ore more files or directories is using, enter:

$ du -sh du shows how much space one ore more files or directories is using, enter:

$ du–sh

$ du -sh

Did you Know?

Top companies using Linux are:

Google- an American based multinational company, the services of which includes search, cloud computing and online advertising technologies runs on Linux.

Twitter- famous online social networking and micro-blogging site that is Powered by nix.

Facebook-one of the most famous and most widely used Social Networking services runs on the same platform.

Amazon-An American based international company which deals with International Online Retailing is in the list of Linux powered Company.

IBM- (International Business Machine Corporation) the American based company which for sure don't requires any introduction, is again powered by nix.

McDonalds-The world's largest chain of hamburger fast foot restaurant uses GNU/Linux (Ubuntu) too.

Submarines-The submarines in the United State Navy are controlled by same platform.

NASA-National Aeronautical and Space Administration, The United nation's Space program widely uses Linux in many of their programmes.

Watches-Most of you would not be knowing that there are Linux Powered Watches in the market, already. The watch developed by IBM running Linux.

Mobile Devices-True, you all know that Linux is powering Mobile Phones, Tablets and Kindle.

Self-assessment Questions

1) UNIX was developed by:
   a) Bell Labs
   b) Berkley Software Group
   c) California University
   d) American Defence Academy

2) Pick the incorrect statements:
   a) Shell is a command interpreter.
   b) Shell is the interface between user and kernel
   c) System can't work without a shell...
   d) Shell is a program

3) Which combination of keys is used to exit from terminal?
   a) Ctrl + t
   b) Ctrl + z
   c) Ctrl + d
   d) Ctrl + e

4) Which command is used to list all the files in your current directory (including hidden)?
   a) ls –l
   b) ls –t
   c) ls -a
   d) ls –i

5) Which command is/are used to remove directory in Linux?
   a) rmdir
   b) rm–r
   c) Only b
   d) Both a and b

6) Which of the following command is used to create file in Linux?
   a) Touch
   b) Cat
   c) Echo
   d) All of the above

7) In Linux everything stored as a
   a) File
   b) Directory
   c) Executable
   d) None of the above

Summary

- Linux is a UNIX implementation that is constantly growing.
- Several users can use system together (multiuser), and a single user can also run multiple jobs concurrently (multitasking).
- Kernel addresses the hardware directly.
- The shell interacts with the user. It processes a command, scans it and rebuild in a form that kernel can understand.
- UNIX is security conscious and can be used only by those persons who maintain an account with computer system.
- In UNIX everything is treated as a file.

Terminal Questions

1. What is UNIX? List the features and benefits of UNIX Operating System.

2. Summaries the growth of UNIX over years. Also discuss about various versions of UNIX.

3. Describe UNIX architecture. Explain the role played by the kernel and shell.

Answer Keys

Self-assessment Questions	
Question No.	Answer
1	a
2	c
3	c
4	c
5	d
6	d
7	a

Activity

Activity Type: Online/Offline **Duration: 30 Minutes**

Description:

Do an online study and prepare a list of duties performed by a system administrator.

Bibliography

e-References

1) This website was referred on 3rd May 2016 while developing content for Introduction to Unix http://www.linuceum.com/Distros/osLinuxDistros.php

2) This website was referred on 3rd May 2016 while developing content for Introduction to Unix http://tldp.org/LDP/intro-linux/html/sect_01_05.html

3) This website was referred on 3rd May 2016 while developing content for Introduction to Unix http://www.ubuntu.com/server

External Resources

- Maurice J. Bach, The Design of Unix Operating System, (2010) Pearson Education
- S. Prata, Advance UNIX, a Programmer's Guide, (2011), BPB Publications, and New Delhi,
- B.W. Kernighan & R. Pike, The UNIX Programming Environment, (2009) Prentice Hall of India.
- Jack Dent Tony Gaddis, Guide to UNIX Using LINUX, (2010) Vikas/ Thomson Pub. House Pvt. Ltd.

Video Links

Topic	Link
The Linux File System	https://www.youtube.com/watch?v=2qQTXp4rBEE
Directory structure of the UNIX file system	https://www.youtube.com/watch?v=PEmi550E7zw

Notes: