GCA(S1) 02-01

PC Software

SEMESTER - 1

BACHELOR OF COMPUTER APPLICATION

Block 1



KRISHNA KANTA HANDIQUI STATE OPEN UNIVERSITY

Subject Experts

Prof. Anjana Kakati Mahanta, Gauhati University
Prof. (Retd.) Pranhari Talukdar, Gauhati University
Dr. Jyotiprokash Goswami, Assam Engineering College

Course Co-ordinator :

Dr. Sanjib Kr. Kalita, KKHSOU
Dr. Tapashi Kashyap Das, KKHSOU
Sruti Sruba Bharali, KKHSOU

SLM Preparation Team

UNITS	CONTRIBUTORS
1	Dr. Tapashi Kashyap Das, KKHSOU
2	Gitanjali Patowary, IDOL, Gauhati University
3, 4 & 5	Dr. Kshirod Sarmah, NERIM

Editorial Team

Content : Manoj Sarma, Royal Global University
Language : Prof. (Retd.) Robin Goswami, Cotton College
Structure, Format & Graphics: Dr. Tapashi Kashyap Das, Sruti Sruba Bharali, KKHSOU

June, 2018

This Self Learning Material (SLM) of the Krishna Kanta Handiqui State University is made available under a Creative Commons Attribution-Non Commercial-ShareAlike4.0 License (International): http.//creativecommons.org/licenses/by-nc-sa/4.0.

Printed and published by Registrar on behalf of the Krishna Kanta Handiqui State Open University.

Head Office : Patgaon, Rani Gate, Guwahati-781 017; Web: www.kkhsou.in
Sub-Head Office : Housefed Complex, Dispur, Guwahati-781 006

The university acknowledges with thanks the financial support provided by the **Distance Education Bureau, UGC** for the preparation of this study material.

BACHELOR OF COMPUTER APPLICATION

PC SOFTWARE

DETAILED SYLLABUS

Block-1

UNIT 1: Disk Operating System **Pages: 7-27**

Definition of DOS; DOS Prompt; Files and Directories: Root Directory, Path, Naming Files and Directories, Types of File; DOS Commands: Internal Commands, External Commands; Wildcards

UNIT 2: UNIX Operating System **Pages: 28-41**

Features of UNIX; Booting Process of UNIX; Run Levels; UNIX Architecture: Kernel, Shell, Utilities; File System

UNIT 3: LINUX Operating System **Pages: 42-72**

LINUX Operating System: Login into Linux, Using the Desktop, Files and Folders; Concept of Paths

UNIT 4: LINUX Commands **Pages: 73-101**

Linux Commands: Internal Commands, External Commands; Directory Handling Commands; File Handling Commands; Utility Commands; File Permissions; Text Editor *vi*

UNIT 5: MS Windows Operating System **Pages: 102-118**

Windows Operating System; Components of Windows: The Desktop, Icons, Taskbar, My Computer; Working with Files and Folders: Creating Files and Folders, Naming Files and Folders, Renaming Files and Folders, Deleting Files and Folders; Cut, Copy, Paste

COURSE INTRODUCTION

There are two broad fields in computer science. The first one is the hardware and the second one is software. Hardware cannot work independently without software. Software's are also categorized into two categories. The first one is called the system software and the second one is called the application software. To start a computer we should have at least one system software that is operating system. There are many operating systems some are open source and the others are license. License software's are required to purchase and the open source software's are freely downloadable. The main purposes of operating system is to control the computer i.e. memory management, process management, file management and I/O management. The operating system installed in a computer performs these operations.

The popularity and increasing demand of information technology is due to its easiness of operations. Different types of applications are performed with the help of different types of application software's. One of such most commonly used application software (licensed) package is MS office. To handle these packages users does not require any programming or technical knowledge. This package is mainly used to prepare office documents, balance sheet, seminar presentations etc.

This course mainly concern about the operating system and MS office. After completion of this course learners will be able to handle a computer with different operating system installed. Moreover, they will be confident to prepare office documents including financial and presentation document.

There are two block in this course.

Block 1 deals with the fundamental concept of different operating systems namely DOS, UNIX, Linux, MS Windows. This block discusses about the working of the computer.

Block 2 concentrates on some of the most important application packages like MS word, MS excel & MS PowerPoint. The main objective of this block is to make familiar with the basic application packages.

BLOCK INTRODUCTION

This is the first block of the course. This course contains thirteen units in two blocks. Block A contains 5 units. These units discuss the concept of operating systems.

This block comprises of the following five units :

Unit 1: deals with the Disk Operating System (DOS). This unit introduces the learners about the concept of directories, files, commands etc.

Unit 2: introduces about UNIX operating system. This unit discusses about UNIX file system, booting process, UNIX architecture etc.

Unit 3: discusses Linux operating system. The unit discusses the basic understanding of LINUX system like booting, different shells, ownership and file permission etc.

Unit 4: introduces us to the different LINUX commands. Directory handling, file handling and utility commands are discussed in this unit.

Unit 5: discusses the most commonly used operating system called Windows operating system. Most of the commonly used menus and toolbars are discussed in this unit.

The structure of block 1 is as follows:

UNIT 1: Disk Operating System

UNIT 2: UNIX Operating System

UNIT 3: LINUX Operating System

UNIT 4: LINUX Commands

UNIT 5: MS Windows Operating System

While going through a unit, you will notice some along-side boxes, which have been included to help you know some of the difficult, unseen terms. Again, we have included some relevant concepts in ‘‘LET US KNOW’’ along with the text. And, at the end of each section, you will get “CHECK YOUR PROGRESS” questions. These have been designed to self-check your progress of study. It will be better if you solve the problems put in these boxes immediately after you go through the sections of the units and then match your answers with “ANSWERS TO CHECK YOUR PPROGRESS” given at the end of each unit.

UNIT 1: DISK OPERATING SYSTEM

UNIT STRUCTURE

1.1 Learning Objectives

1.2 Introduction

1.3 Definition of DOS

1.4 DOS Prompt

1.5 Files and Directories

- 1.5.1 Root Directory
- 1.5.2 Path
- 1.5.3 Naming Files and Directories
- 1.5.4 Types of File

1.6 DOS Commands

- 1.6.1 Internal Commands
- 1.6.2 External Commands

1.7 Wildcards

1.8 Let Us Sum Up

1.9 Answers to Check Your Progress

1.10 Further Reading

1.11 Model Questions

1.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- work with Disk Operation System
- use DOS prompt for typing commands
- handle Files and Directories in DOS
- work with root directory
- give the concept of path
- use Internal and External DOS commands
- define Wildcard characters.

1.2 INTRODUCTION

In this unit, learners will be acquainted with DOS - the operating system that controls computer's hardware. This unit discusses the concept of files and directories, how to create them as well as some basic DOS commands.

1.3 DEFINITION OF DOS

DOS is the acronym for ***D**isk **O**perating **S**ystem.*It is a non-graphical user interface operating system. It works as an interpreter which translates the commands given by user so that they can be interpreted by the computer. Thereby it makes a link between the user and the computer.

Microsoft Disk Operating System in short MS-DOS is the most commonly used member of the DOS family of operating systems created for IBM compatible computers that was first introduced by Microsoft in 1981.This operating system has been upgraded many times, incorporating new features by *Microsoft.*

1.4 DOS PROMPT

Once DOS is loaded, it displays a Prompt on the screen, indicating that it is ready to accept instructions. **C :\>** is known as***DOS prompt*** or ***command prompt***. The first character indicates the disk drive that is currently active. In DOS, drive C is, by default, the current active drive.The flashing underscore next to the command prompt is called the **cursor**.

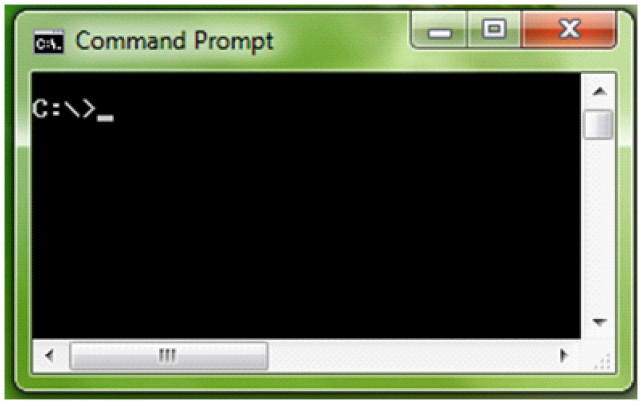


Fig. 1.1: The Command Prompt

Typing a Command in DOS Prompt: In DOS, you can type a command in either uppercase or lowercase letters. Following are the steps to type a command at the command prompt:

1) First you type the command at the command prompt. Any mistake in typing can be corrected or renamed by pressing ***Backspace*** key.
2) Then, press ***Enter****key*. You must press ***Enter*** after every command you type.
3) If you type an invalid command that DOS cannot recognize, the following message will be displayed:

Bad command or file name

Cancelling a DOS command : In some situation, it becomes very necessary to cancel a command after issuing it. A DOS command can be cancelled by pressing **Ctrl+C** simultaneously. When this command is applied, ^C will be viewed on the DOS prompt, and it cancels the current command.

1.5 FILES AND DIRECTORIES

On a hard disk any information in the form of a program, document, picture, game etc., is stored in some physical locations.Each of these separate locations in a computer disk is commonly called a ***file***.

A ***directory*** in DOS represents an area on the hard disk that acts like a file cabinet or file box. DOS allows to create a large number of directories. Everything inside a directory is either a file or another directory. The directory lying inside a directory is referred to as a **subdirectory**. The outer directory is called the **parent directory.** Thus, a directory can be both a parent directory and a subdirectory.

1.5.1 Root Directory

The ***root directory*** is the first or top-most directory in a hierarchy. All other directories are branches of this directory. The root directory is designated by a backslash (\).

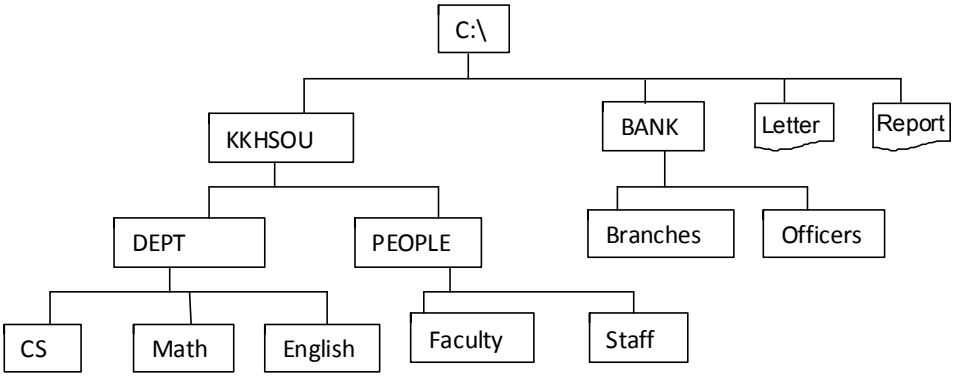


Fig. 1.2: Directories and Files

For instance, in (Fig. 1.2), the root directory is **C:**which contains two subdirectories, namely **KKHSOU** and **BANK,** and two files namely **Letter** and **Report** respectively. The **KKHSOU** subdirectory also contains two more subdirectories **DEPT** and **PEOPLE**; Under **DEPT**, there are three subdirectories **CS**, **Math** and **English** respectively. Inside the **PEOPLE** subdirectory, there are two subdirectories **Faculty** and **Staff**. Similarly, **BANK** subdirectory also contain subdirectories inside it namely, **Branches** and **Officers**.

1.5.2 Path

A ***path*** is defined as the address where the directory or the file exists. It is basically the list of directories, subdirectories starting with the root directory. Each directory name is separated by a backslash (\). To get to a particular file, you must trace the path from the root directory to the directory that contains the file. For example, to reach the directory ***English*** (in Fig. 1.2), the path would be:

C:\KKHSOU\DEPT\English

Again, the path to reach the directory***Officers*** would be like this:

C:\BANK\Officers

1.5.3 Naming Files and Directories

To differentiate various directories and files, we usually label them on the basis of its contents. DOS does not allow us to use the identical name more than once inside the same parent directory, i.e., each file/directory inside a directory should have a unique name. If we try to put two files with identical names into the same directory, DOS erases one of them without warning.



DOS Fill Name
BCA. TXT
MCA.TXT

A file name in DOS has two parts:

- **First name :** It can be 1 to 8 characters long.
- **Extension name :** It is optional and can be 1 to 3 characters long.

The *First* and the *Extension* names are joined with a dot (.) in between them.

There are certain definite rules for naming files in DOS. These are :

1) The combination of the following characters can be used for naming a file.
   - *Alphabets* A to Z, both capital and small letters, are allowed.
   - *Digits* 0 to 9 are allowed.
   - *Special characters* like & # @ $! % ‘ () - {} _ ~ are allowed.
2) *Space* is not allowed in between two characters in a file name.
3) *Dot* (.) can be used only once in a file name, i.e. just to give extension name.
4) Files with extension names like .EXE, .COM and .BAT are given a special status by DOS. Those extensions are for some special program files.
5) *Reserved words* like CON, COM1, COM2, LPT1, PRN, NUL etc. are not allowed in a file name.

1.5.4 Types of Files

We can create different types of files in a computer system according to our usage.DOS supports a wide variety of files. The

type of the file can be identified with the extension of the file in the filename. Some of them are given below:

- *Program files* often have **.EXE**, **.COM**, or **.BAT** extension. If we try to view one on screen, we will observe that it is filled with some funny characters.
- The **.TXT** extension is usually used for files that consist of unformatted text.
- The **.DOC** extension file means that the file was created using word processing software.
- The **.BAK** is used by programs and users to make a copy of an existing file for safety and/or archival purposes.
- The **.TIF**, **.BMP**, **.JPEG** etc. are extensions for image files.
- The **.BIN** extension is for binary files.
- The **.HLP** extension means that the file contains help information.



CHECK YOUR PROGRESS

Q.1: Fill in the blanks :

a) DOS stands for .....

b) Extension name in a file name can be ..... characters long.

c) .JPEG, .TIF, .BMP are extensions for ..... files.

d) ..... files generally have extensions .exe, .com.

1.6 DOS COMMANDS

By now you must have been familiar with some basic concepts of DOS. In order to work with your computer in DOS, it is necessary to learn DOS commands.DOS commands are divided into two categories:

- **Internal Commands**
- **External Commands**

1.6.1 Internal Commands

Internal commands are automatically loaded into the computer's memory during the ***booting*** process. These commands are executable immediately after getting the DOS prompt. Internal commands are always available until we switch off the computer. These commands are also called ***memory-resident commands***. A few examples of internal commands are: DIR, MD, CD, COPY, DATE, TIME etc. We will discuss some important internal commands with their proper format later in this section command com file is responsible for internal command.

At this, we would like to mention that throughout this section, you will come across the term 'syntax'. ***Syntax*** is the writting rule of a command. Again, whenever you come across the sambol ↵ after a command, it means that you have to press the **Enter** key from the keyboard after typing the command.

Let us discuss some commonly used ***internal commands***.

1) **DATE command :** It shows the current date and enables you to reset it. The syntax in the command prompt is:

C:\>DATE

The command will display the following two messages in the screen as shown in Figure 1.3.

The current date is: Sat 05/10/2014

Enter the new date: (mm-dd-yy)

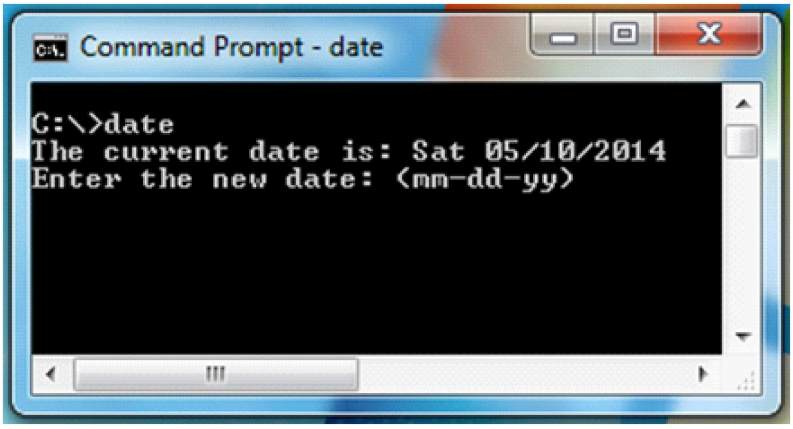


Fig. 1.3: Command Prompt Showing DATE Command



Booting: Process that starts up a computer is called **booting.** It checks for proper functioning of all the peripheral devices attached with the system. It searches for the operating system and, when located, loads it into the main memory.

If you do not want to change the date, press **Enter** to leave the date as it is.

If you want to change the date, type in the new date using the format shown.

2) **TIME Command:** This command displays current time of system clock and provides a way for you to reset the time. The syntax in command prompt is:

C:\>TIME↵

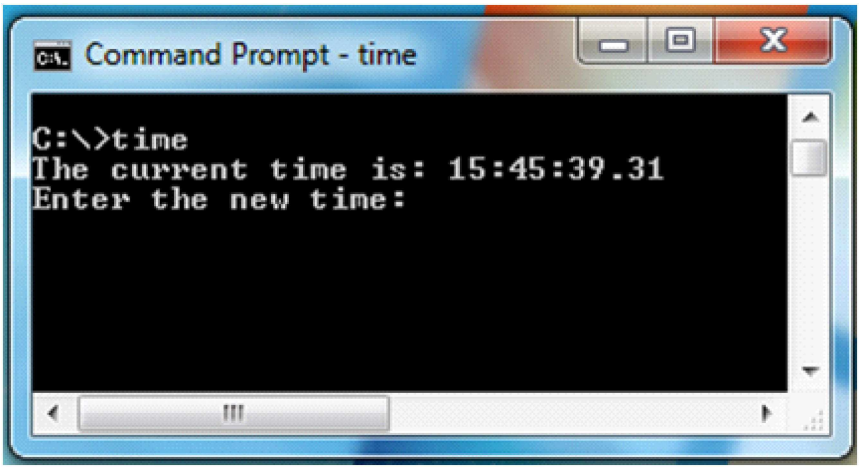


Fig. 1.4: Command Prompt Showing TIME Command

If you do not want to change the time, press **Enter** key and if you want to change it, type the correct time.

3) **Clearing the Screen:** The **CLS** command (CLear Screen) clears any text on the screen and displays the system prompt at the upper left corner of the screen. The syntax in command prompt is:

C:\>CLS↵

4) **The Directory command (DIR) :** By default, the **DIR** command produces a list of all the files and subdirectories, under the directory that is currently active with the details like size in bytes, time and date of creation etc.

We can display the files of some other directory or subdirectory or drive from the current directory through a *path* specification. For example, suppose we are now present in the root directory (C:\) and we are required to view the contents of

mydata directory which is inside the root directory. For this, the command in the command prompt will be:

C:\>DIR \mydata ↵

or **C:\>DIR mydata**

We can add various***options*** with DIR command to display the list in different ways. A few of them are listed below:

Commands with Options	Functions
C:\>DIR /P	Displays the file names one full screen at a time. As soon as a screen full of file names gets displayed, it pauses displaying a message "*Press any key to continue... ... ...*". You can continue to see the rest of the file listing by pressing any key on the keyboard.
C:\>DIR /W	Displays the directories and file names width-wise.
C:\>DIR /B	Displays only file names.
C:\>DIR /AH	Displays the list of hidden files.
C:\>DIR /AS	Displays the list of *system files*.
C:\>DIR /ON	Lists the directories and files in alphabetical order (i.e., A to H)
C:\>DIR /O-N	Lists the directories and files in a reverse alphabetical order (i.e., Z to H)

Wildcard characters (? and *) which will be discussed later in this unit, can also be used with DIR command to specify groups of files.For example,

- to display only files with the **TXT** filename extension on the current drive that begin with the letters K, we have to enter the command:

  DIR K*.TXT↵

- To display only files on drive D that have file extension **.DOC**, we have to enter the command:

  DIR d:*.DOC↵

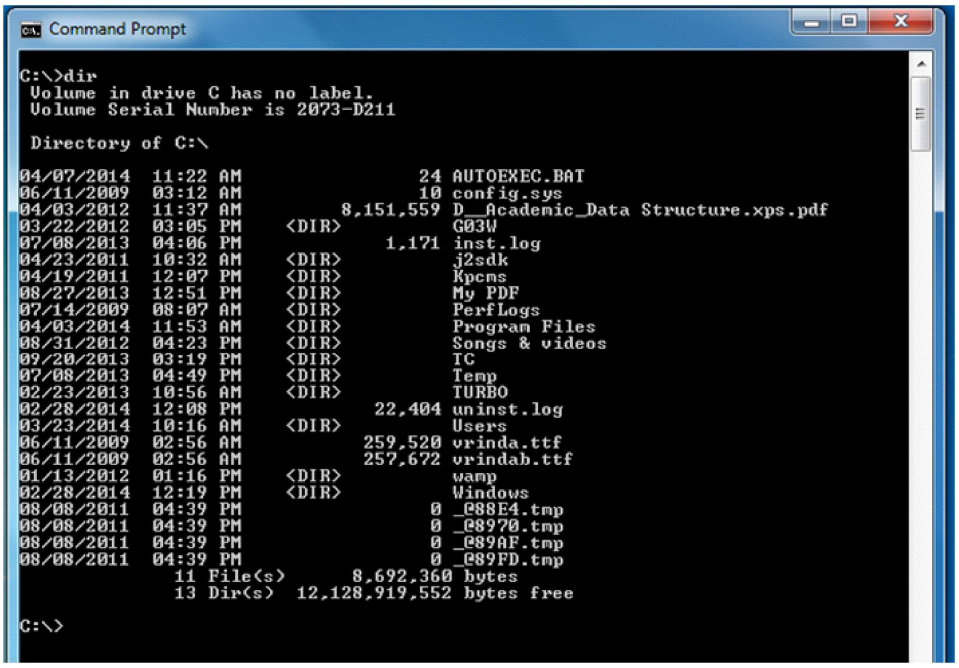


Fig. 1.5: Output or DIR command of current drive C of a particular machine

5) **Changing the Drive:** If you are working in C drive and want to change to D drive, the command will be :

C:\>D:↵

6) **Making and Naming a Directory:** The directory structure is like a tree but viewed upside down, i.e., the root is at the top and the branches point downwards.The **MD**command is used to make as well as name a directory or a subdirectory. The syntax is:

C:\>MD Name

where *Name* can be any directory name. For example:

C:\>MD KKHSOU

C:\>MD Data

To create subdirectory “Academics” inside “University” directory at a time, the command will be:

C:\>MD \University\Academics

7) **Changing Directory:** You may move or change from directory to directory using the **CD**command.If you want access to the files of a particular directory, you have to move into that directory.

For this, first you have to type CD, then leave a blank space and then type the name of the directory to which you wish to reach and press ***Enter*** in the keyboard. The syntax is:

C:\>CD DirectoryName↵

For example, to change to the KKHSOU directory (Fig.1.2) from the root directory, the command is:

C:\>CD KKHSOU↵

The prompt will look like C:\KKHSOU>

Again, to move to the **DEPT** directory which is inside **KKHSOU** directory, the command will be:

C:\KKHSOU>CD DEPT↵

The prompt will look like C:\KKHSOU\DEPT>

8) **Going Back to Root Directory Directly:** The command **CD** will help you to go back to root directory directly. If you are in the DEPT directory (Fig.1.2) and you wish to go back to the root directory, type the command **CD** in the prompt **C:\KKHSOU\DEPT>** as follows:

C:\KKHSOU\DEPT>CD

This will bring you to root directory C:\.

9) **Going Back to Root Directory Stepwise:** For this, we have to type CD followed by two dots (**..**) and press ***Enter*** key.

C:\KKHSOU\DEPT>CD..↵

The prompt will becomeC:\KKHSOU>

To come out of **KKHSOU** directory, again type **CD..**as follows:

C:\KKHSOU>CD..↵

The prompt will become **C:\>**

Now you are in root directory **C:\>** . At this stage, if you again type **CD.** DOS will display the message *'Invalid directory'*.

10) **Removing a Directory:** In DOS, the **RD** or **RMDIR** command removes a directory. The syntax is:

C:\>RD DirectoryName↵

Before giving the RD command, the following points are to be remembered:

- All the files inside the directory are to be deleted before, i.e. there should be no files or sub-sub-directories inside the directory we wish to delete.
- We have to be out of that directory which you wish to delete.

11) The MOVE Command: To rename a directory, we use **MOVE** command. For example, to change the name of a directory say, BOOK to MEGAZINE, the command will be:

C:\>MOVE BOOK MEGAZINE

This will rename the directory BOOK to MEGAZINE.

MOVE command can also be used to move one or more files to a desired location. If you want to move two files named Report1 and Report2 from the current directory to the \DATA directory on drive D, enter the following command:

C:\>MOVE Report1, Report2 D:\DATA

12) Creating a File: We can create a file in DOS using the **COPY CON** command. COPY and CON should be separated by a blank space. While using this command, the following points are to be kept in mind :

- Blank space must be left between COPY CON and file name.
- Something should be written in the file after creating it.
- Contents should be saved using Ctrl+Z or F6.
- Extension name can also be included while creating the file. The syntax is as follows:

C:\>COPY CON mydata.txt

This will create a text file with file name **mydata**. After writing your content, you should press ***Enter*** and save the content with F6 or Ctrl+Z.

13) Displaying the Content of a File: To see the content of a file, use **TYPE** command. For example, we can see the contents of the file ***mydata*** created earlier in this way:

C:\>TYPE mydata

14) Copying Files: When we copy a file, it creates a duplicate of the original file and does not remove the original file. This is

useful for many purposes. In DOS, we can copy a single file and a group of files using the **COPY** command. When we use the **COPY** command, we must include *source* and *destination.*

- *Source* is the location and name of the file which we want to copy.
- *Destination* is the location to which we want to copy the file.

Source and destination should be separated by a space. The COPY command follows this pattern:

C:\>COPY source destination

To copy a file **Data.txt** from C drive to D drive, the command will be:

C:\>COPY Data.txt D:↵

To copy a file Data.txt from D drive to E drive under a new name Mydata.txt, the command will be:

C:\>COPY D:\Data.txt E:Mydata.txt↵

Copying a File from one Directory to another Directory: To copy a file from one directory to another, we have to specify the path of the source and the destination directory in the COPY command. For example, to copy the file COMMAND.COM from root directory to KKHSOU directory which is already in C drive, the command will be

C:\>COPY COMMAND.COM \KKHSOU

15) Deleting Files: The **DEL** command is used to delete a file. The syntax is :

C:\>DEL filename↵

Note: Del Command delete files permanently so, be careful.

If the file is present on the same drive on which you are working, then it is optional to give the drive name with the DEL command. For example, If you want to delete a file LETTER.DOC on drive D, the command will be:

C:\>DEL D:LETTER.DOC↵

If you want to delete all files on C drive which have the extension **.txt**, the command is:

C:\>DEL *.txt↵

16) **Renaming Files:** You can change the name of any file with the help of **REN** command. When you use the REN command, you must include two parameters.First is the file you want to rename and the second is the new name for the file. You separate the two names with a space. The REN command follows this pattern:

REN oldname newname↵

For example, to rename the LETTER.DOC file to REPORT.DOC, type the following at the commandprompt:

C:\>REN LETTER.DOC REPORT.DOC↵

We can rename a group of files at a time using *wildcard* characters with the REN command. For example, the command:

C:\>REN *.BAK *.TXT

will change all the files with extension .BAK into .TXT extension. Wildcards will be discussed later in this unit.

17) **The EXIT Command:** The **EXIT** command is an internal command and used to exit from the currently running application and the DOS session. The syntax in the command prompt is:

C:\>EXIT



CHECK YOUR PROGRESS

Q.2: Write down the commands to create the directories and subdirectories shown in Figure 1.2.

Q.3: View the files of C drive with a page pause.

Q.4: Display all the files with:

a) EXE extension

b) B as the first character of the file name

1.6.2 External Commands

External commands are used for some special purpose. They are loaded into main memory only when called. External commands are also termed as ***disk-resident commands***. A few examples of external commands are: CHKDSK, FORMAT, DELTREE, XCOPY etc.

Let us discuss some important ***external commands***:

1) **The FORMAT Command:** The **FORMAT** command formats a disk so that computer can read it. For that, first we type FORMAT, and then leave one blank space and type the drive name which you want to format. It is necessary to put a colon(:) after the drive name. Care should be given while issuing FORMAT command as it erases all the existing data of the particular disk drive which has formatted and makes it ready to use for new data.The syntax is:

C:\>FORMAT drive:↵

For example, we can format a floppy with the following command:

C:\>FORMAT A:

Options can be included with FORMAT command. For example,

C:\>FORMAT A:/Q/performs a quick format.

C:\>FORMAT A:/S copies system files of DOS on floppy disk.

2) **The CHKDSK Command:** The **CHKDSK** command is used for getting information about the available space in the disk and memory. It checks a disk and displays a status report. CHKDSK may be used without any parameters, in which case the current drive is checked. The syntax is:

C:\>CHKDSK↵

To check the D drive, the command is:

C:\>CHKDSK D:↵

3) **The DELTREE Command:** This external command is available in DOS 6.0 onwards. DELTREE deletes a directory and all the subdirectories and files inside that directory at a time. For example, to delete the entire directory BOOK with all its contents, we have to type the following :

C:\>DELTREE BOOK↵

4) **The XCOPY Command:** This external command is available in DOS 3.2 and its letter versions. The XCOPY command acts

quickly and helps you copy entire directory or sub-directory with all files excluding system and hidden files. The syntax is:

XCOPY source destination

where *source* specifies the file to copy and *destination* specifies the location and/or name of new files.

5) **The TREE Command:** With the use of **TREE** command we can see the list of directories and files in the form of a tree.

To view the tree structure of the current drive, the command is:

C:\>TREE

To view the directories and subdirectories of drive D, the command is:

C:\>TREE /D:

If you want to display the files altogether, the command is:

C:\>TREE /F

To display the list pagewise, just add MORE like this:

C:\>TREE /F| MORE

6) **The ATTRIB Command:** This external command is used to attribute some features to your file. You can make a file hidden or set into “Read only” form using the ATTRIB command. The syntax is:

C:\>ATTRIB Data/+H

After having executed this command, you cannot find the above file ***Data*** using the ***DIR*** command as it has been made hidden. Some other attributes which can be added are as follows:

–H To cancel the hidden attribute of the file and it is seen again in the displayed list.

+R To set the file into “Read only” mode

–R To turn the file back into normal mode from “Read only” mode.

7) **HELP Command:** HELP command is an external command and it displays information regarding the given command. For example,

C:\>HELP DIR

displays information related of DIR command on the command window.

1.7 WILDCARDS

When you want to work on group of files at one time, you use wildcard characters. There are two types of wildcard characters :

*(Star)

and ? (Question mark)

The star wildcard (*)will match all characters. The notation *.* means that the file can have any file name with any extension. Suppose in your C drive, you have the following files Letter.DOC, REPORT.DOC, Menu.DOC, RESUME.DOC, RESULT.XLS

If you give the file specification *.DOC, it will include the following files with extension.DOC

Letter.DOC

REPORT.DOC

Menu.DOC

RESUME.DOC

The file specification *.DOC specifies files with any first name and the extension .DOC

If you give the file specification as **R*.***, it would include all files that start with the letter R in the first name and with any extension. The following files would be included in this group:

REPORT.DOC

RESUME.DOC

RESULT.XLS

The question mark wildcard (?) matches any single character. **?data.DOC** will include all files that start with any character, but the remaining letters in the filename mush be **data**, and the file must have the extension DOC.*Wildcard characters can be used in different combinations to control the files included in a group.*



CHECK YOUR PROGRESS

Q.5: Write down the commands to copy all files of C drive with the first letter L and the extension DOC to D drives.



1.8 LET US SUM UP

In this unit, we have briefly discussed the DOS along with few internal and external commands. What we have learned in this unit can be summarized as follows:

- DOS is an acronym for Disk Operating System.
- DOS works as an interpreter that establishes a link between hardware and software.
- DOS offers two types of commands. Some commands are *internal* and built into COMMAND.COM. Others are *external* commands stored on disk.
- DATE, TIME, CLS, DIR, MD, CD, RD, MOVE, COPY CON, TYPE COPY, DEL, REN, EXIT etc. are some important internal commands.
- DATE command displays and/or sets the system date.
- TIME command displays current time setting of system clock and provides a way for you to reset the time.
- DIR command is used for viewing the available files and directories on the disk.
- MD command is used for creating a directory.
- CD command is used for changing the current directory.
- RD command is used for removing the current directory, provided the directory is empty.
- MOVE command moves one or more files to the location you specify. It can also be used to rename directories.
- COPY CON command allows us to create a file through the command prompt.

- TYPE command displays the content of a file.
- COPY command is used for making a duplicate copy of a file.
- DEL command is used to delete a file.
- REN command is used for renaming a file.
- EXIT command quits command prompt program.
- FORMAT, CHKDSK, DELTREE, XCOPY, TREE, ATTRIB, HELP etc. are few examples of external commands.
- FORMAT command is used to format a disk.
- CHKDSK command is used for getting information about the space available in the disk and memory.
- DELTREE deletes a directory including all files and subdirectories that are in it.
- XCOPY command copies directories, subdirectories, and files.
- TREE command displays directory paths and files in each subdirectory.
- ATTRIB command displays or sets the Read-only, archive, system, and hidden attributes of a file or directory.
- HELP command displays information about a DOS command.
- Wildcards can be used to designate a group of files in a DOS command string. There are two wildcards in DOS:

  The *is a wildcard character that represents any group of characters in a filename.

  The?is a wildcard character that replaces a single character in a filename.



1.9 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: a) Disk Operating System; b) three; c) image; d) Program files

Ans. to Q. No. 2: In the diagram, there are two subdirectories under the root directory **C:** namely **KKHSOU** and **BANK**. To create these two subdirectories, the series of commands will be:

C:\>MD KKHSOU ↵

C:\>MD BANK ↵

To create the subdirectories under **KKHSOU**, the command will be:

C:\>MD \KKHSOU\DEPT ↵

C:\>MD \KKHSOU\PEOPLE ↵

To create three subdirectories under **DEPT** and two subdirectories under **PEOPLE**, the command will be:

C:\>MD \KKHSOU\DEPT\CS ↵

C:\>MD \KKHSOU\DEPT\Math ↵

C:\>MD \KKHSOU\DEPT\English ↵

C:\>MD \KKHSOU\PEOPLE\Faculty ↵

C:\>MD \KKHSOU\PEOPLE\Staff ↵

In the same manner, **Branches** and **Officers** subdirectories can be created inside **BANK** under root directory **C**. The commands are as follows:

C:\>MD \BANK\Branches ↵

C:\>MD \BANK\Officers ↵

Ans. to Q. No. 3: C:\>DIR /P ↵

Ans. to Q. No. 4: a) DIR *.EXE↵

b) DIR B*.* ↵

Ans. to Q. No. 5: C:\> L*.DOC D:\ ↵



1.10 FURTHER READING

1) Gookin, D. (2011): *DOS for Dummies*; John Wiley & Sons.

2) Murdock, E. (2008): *DOS the Easy Way;* EasyWay Downloadable Books.



1.11 MODEL QUESTIONS

Q.1: What is DOS prompt?

Q.2: What is Microsoft DOS? Explain its functions.

Q.3: What are the rules for naming a file in DOS?

Q.4: Give some examples of DOS file types with their purpose.

Q.5: What is root directory?

Q.6: What are DOS commands? Give some examples with their syntax.

Q.7: What are internal and external commands in DOS? Give suitable examples.

Q.8: What can be done using DIR command?

Q.9: Explain MD and CD command with appropriate example.

Q.10: Differentiate between the functions of MOVE and REN commands.

Q.11: What is the difference between COPY and COPY CON command?

Q.12: What is the difference between DEL and RD command?

Q.13: What are wildcards in DOS? Explain with suitable examples.

Q.14: What is the purpose of the following commands:
TREE, HELP, CHKDSK, XCOPY, FORMAT, ATTRIB, CLS.

Q.15: Write down the steps to copy all files from directory AUTHOR of BOOK directory into another directory KKHSOU under UNIVERSITY directory.

*** ***** ***

UNIT 2: UNIX OPERATING SYSTEM

UNIT STRUCTURE

2.1 Learning Objectives

2.2 Introduction

2.3 Features of UNIX

2.4 Booting Process of UNIX

2.5 Run Levels

2.6 UNIX Architecture

- 2.6.1 Kernel
- 2.6.2 Shell
- 2.6.3 Utilities

2.7 File System

2.8 Let Us Sum Up

2.9 Answers to Check Your Progress

2.10 Further Reading

2.11 Model Questions

2.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- learn the basics of Unix Operating System
- describe the concept of Unix boot process
- learn about the UNIX file system
- understand the concept of Unix Architecture.

2.2 INTRODUCTION

In this unit we will discuss the basics of UNIX operating system. Features of Unix Operating System are explained in the first section. We will learn the concept of UNIX booting process in the next section. We shall also discuss about the Unix Architecture and UNIX file system.

2.3 FEATURES OF UNIX

This section contains a brief overview of the main features of UNIX. UNIX operating system was first developed in the 1960s, and has been under constant development ever since. An operating system can be defined as a control program that helps the user to communicate with computer hardware. It is written in high-level language, 'C', making it easy to port to different configurations as per the requirements.

Unix Features:

a) **Portability or Hardware Independence:** UNIX Operating System software can be easily moved from one hardware system to another. It hides the machine architecture from the user, making it easier to write programs that run on different hardware implementation.

b) **Multi-User:** One important part of the UNIX OS is multi-user support. More than one user can use the system at a time supported via terminals. It does this by time-slicing the computer processor at regular intervals among the various users.

c) **Multitasking:** UNIX is a multitasking Operating system i.e. more than one program can be run at a time.

d) **Hierarchical File System:** It uses a hierarchical file system that allows easy maintenance and efficient implementation.

e) **Open Source:** Source code of UNIX is freely available and it is community based development project.

f) **Security:** UNIX provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

2.4 BOOTING PROCESS OF UNIX

The process of bringing a computer system to life and making it ready for use is known as Bootstrapping. The boot process can be initiated automatically or manually, and it can begin when the computer is powered on or as a result of a reboot command from a running system. When the power comes on or the system is reset, the BIOS starts the master boot

program, located in the first 512 bytes of the active partition on the system disk. It then loads the kernel.

Once control passes to the kernel, it will perform the complete hardware diagnostics, install the loadable drivers for the various hardware devices present on the system and/or loads the kernel modules as required. The kernel then starts the init process, which in turn starts system processes (daemons) and initializes all active subsystems. When everything is ready, the system begins accepting user logins.

In Unix-based operating systems, init (initialization) is the first process started during booting of the computer system and continues running until the system is shut down.

Based on the above explanation we can classify the UNIX boot process in the following five stages–

i) System Startup or Hardware Initializations
ii) MBR (Master Boot Record) Loading
iii) GRUB Loader
iv) Kernel
v) Init Process

i) System Startup or Hardware Initializations: When you power on/ restart the machine the power is supplied to SMPS (switched-mode power supply) and then it will supply to all the devices connected to CPU such as Motherboard HDD's, CD/DVD-ROM, mouse, keyboard etc. BIOS (Basic Input/Output System) is initiated by the CPU and performs mainly two tasks:

a) POST (Power On Self Test) Operations and
b) Selection of first boot devices.

POST operations will check whether all the hardware devices attached are working properly or not and report if any faulty hardware is found. Once the POST is completed BIOS will have the list of bootable devices available and gives back the control to processor (CPU).

ii) MBR (Master Boot Record): It is located in the 1st sector of the bootable disk having size 512 bytes. It contains information about

GRUB (or LILO in old systems). The main function of MBR is to load and execute the GRUB boot loader.

iii) **GRUB Loader (Grand Unified Boot Loader):** GRUB displays a splash screen, waits for a few seconds, if we don't enter anything. It loads the default kernel image as specified in the grub configuration file (/boot/grub/grub. conf). GRUB just loads and executes Kernel and initrd images and then passes control on to the kernel. (initrd stands for Initial RAM Disk).

iv) **Kernel:** Once the control is given to kernel, it will act as a mediator of hardware and software components. Once kernel is loaded into to RAM it always resides on RAM until the machine is shutdown. Once the Kernel starts its operations the first thing it does is executing INIT process.

v) **Init Process:** init (initialization) process is the parent process of all the processes which run under Linux/Unix. The first process it runs is a script file at /etc/rc.d/rc.sysinit which check all the system properties, hardware, display, SElinux, load kernel modules, file system check, file system mounting etc. Based on the appropriate run-level, scripts are executed to start/stop various processes to run the system and make it functional.

Once the initialization process completes the mandatory run level and reaches to default run level set in /etc/inittab, init process runs one more file /etc/rc.local which is the last command run in initialization process or even booting process. Once everything is completed the control is given back to the kernel.

Once the kernel gets the control it starts multiple instances of "getty" and gives the user prompt to login.

2.5 RUN LEVEL

We can define run level as the state of machine after boot and it will identify what system services are operating. System administrators use run levels to define which subsystems are working, e.g., whether X is running, whether the network is operational or not and so on.

Run levels are identified by numbers and seven run levels exist, numbered from zero to six. Each run level has a certain number of services stopped or started, giving the user control over the behavior of the machine.

The /etc/inittab file is used to set the default run level for the system. This is the run level that a system will start up on upon reboot.

ID	Description
0	Halt the system.
1	Single-user mode (for special administration).
2	Local multiuser with networking but without network service (like NFS).
3	Full multiuser with networking.
4	Not used.
5	Ful multiuser with networking and X Windows (GUI)
6	Reboot.



CHECK YOUR PROGRESS

Q.1: How do you relate the word "Portability" with UNIX?

Q.2: What is the purpose of Run levels in UNIX OS?

Q.3: Write the full form of GRUB and MBR.

Q.4: Where is MBR located in disk?

2.6 ARCHITECTURE OF UNIX OPERATING SYSTEM

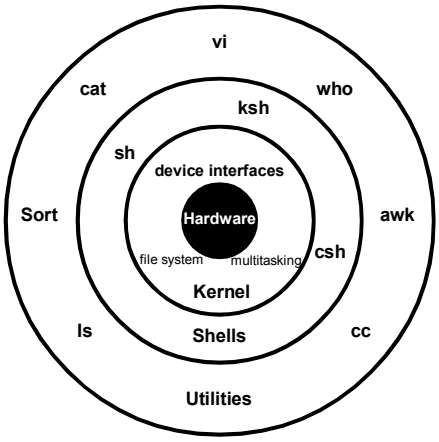


Fig. 2.1: Architecture of UNIX Operating System

UNIX operating system consists of three main components:

- Kernel
- Shell
- Utilities

2.6.1 Kernel

Kernel is the core or key component of UNIX operating system and is directly related to the computer hardware. It sits on top of hardware and acts as an interface between user activities and the hardware. Kernel programs are mostly written in C.

It handles memory management, input and output requests, process scheduling for time-shared operations, resource allocation etc.

2.6.2 Shell

A shell is a program that resides on the kernel and acts as an agent or interface between users and the kernel. A shell can be termed as the command-line interpreter. It is a program that accept commands from the keyboard and gives them to the operating system to perform.

When a user logs in, the login program checks the user name and password, and then starts another program called the shell.

Shell Types: In UNIX there are two major types of shells:

a) The Bourne shell.

If you are using a Bourne-type shell, the default prompt is the $ character.

b) The C shell.

If you are using a C-type shell, the default prompt is the % character.

There are again various subcategories for Bourne Shell which are listed as follows:

- Bourne shell sh
- Korn shell ksh
- Bourne again shell bash
- POSIX shell sh

The different C-type shells follow:

- C shell-csh
- TENEX/TOPS C shell tcsh

a) Different Bourne shell:

- **Bourne shell (sh):** Bourne shell is the default UNIX shell for many UNIX operating systems. This shell is developed by Stephen Bourne of AT&T Bell Laboratories. The Bourne shell was the first shell to appear on UNIX systems, thus it is referred to as ‘‘the shell’’. This shell is used as a scripting language rather than an interactive shell. The Bourne shell is considered the primary shell in scripts. It is still the most widely available shell on UNIX systems, providing a language with which to script programs and basic user functionality to call other programs. The only problem with the Bourne shell is that it has fewer features for user interaction than some of the more modern shells.
- **Korn shell (ksh):** Korn shell was developed by David Korn at Bell Labs in the early 1980s. The Korn shell is an interactive command and programming language that provides an interface to the Unix and other systems. The Korn shell offers compatibility with the Bourne shell, while providing a more robust programming language and command interpreter. Programs written for the Bourne shell can run under the Korn shell without modification. Programs written in the Korn shell can run faster than similar programs written in the Bourne or C shells. The Korn shell is the most advanced of the shells that are "officially" distributed with UNIX systems.
- **Bourne Again Shell (bash):** Bourne again shell was written for the GNU project by Brian Fox. It is called Bourne again

shell for many reasons, the first being that it is the open-source version of the Bourne shell and the second as a pun on the concept of being born again. Bourne again shell (Bash) is a free Unix shell that can be used in place of the Bourne shell. Bash is a sh-compatible shell that incorporates useful features from the Korn shell (ksh) and C shell (csh).

- **POSIX shell:** The full form of POSIX is ‘‘Portable Operating System Interface’’. POSIX shell is based on the standard defined in Portable Operating System Interface (POSIX) - IEEE P1003.2. POSIX defines the application programming interface (API), along with Unix command line shells and utility interfaces. This ensures software compatibility with flavors of Unix and other operating systems. The POSIX standard is designed to be used by both application programmers and system administrators.

b) Different C-type shell:

- **C shell:** The C shell was created at university of California, Berkeley in late 1970s. The C shell is a command interpreter with a syntax similar to the C programming language. The primary purpose of csh is to translate command lines typed at a terminal into system actions, such as invocation of other programs. The C shell program name is csh, and the shell prompt (the character displayed to indicate readiness for user input) is the % symbol. The C shell was invented for programmers who prefer syntax that are similar to C programming language.
- **TCSH Shell:** The full form of TCSH is ‘‘tee cee ess aitch’’. tcsh is an enhanced but completely compatible version of the Berkeley UNIX C shell.). It is a command language interpreter usable both as an interactive login shell and a shell script command processor. It includes a command-line editor, programmable word completion, spelling correction, a history mechanism, job control and a C-like syntax.

2.6.3 Utilities

UNIX utilities are commands or specialized software tools that, generally, perform a single task. It may be as simple as printing the date and time, or a complex as finding files that match many criteria throughout a directory hierarchy. Example: ls, cp, grep, awk, bc, wc, more, rm, mkdir.

Files and Directories: These commands allow you to create directories and handle files.

Command	Example	Description
mkdir	mkdir kkhsou	Make a directory called kkhsou
rmdir	rmdir kkhsou	Remove directory *kkhsou* provided it is empty
cd	cd kkhssou cd ... cd ~sim/materials	Change directory to *kkhsou* Move back one directory Move into *slm's materials* directory
ls	ls ls ~alF	Lists files in current directory List in long format
cp	cp file1 ~slm/materials	Copy file into directory
rm	rm file1.bak rm *.jpg	Remove or delete file Remove all file with .jpg extension
mv	mv old1.html new1.html	Move or rename files
man	man ls	Online manual (help) about the command
pwd	pwd	Print current working directory
cat	cat old1.html	Display File Contents
chmod	chmod 754 myfile	*Change the permissions of files or directories.

* Numerical permissions for chmod.

#	Permission	rwx
7	read, write and execute	rwx
6	read and write	rw-
5	read and execute	r-x
4	read only	r--
3	write and execute	-wx
2	write only	-w-
1	execute only	--x
0	none	---

Some other important UNIX commands are:

Command	Example	Description
vim/vi	vi/vim *filename*	Opens vi/vim text editor
sort	sort *data.txt*	Sorts the contents of a text file, line by line
wc	wc *myfile.txt*	Count words, lines and characters about the file **myfile.txt**
grep <str><files>	grep “bad word”*	Find which files contain a certain word
passwd	passwd	Change passwd
tar	tar -cf subdir.tar subdir tar -xvf subdir.tar	Create an archive called subdir.tar of a directory Extract files from an archive file
gzip <file>	gzip bigfile gunzip bigfile.gz	Compress file Uncompress file
who	who	Lists who is logged on your machine
history	history	Lists commands you’ve done recently
date	date	Printout current date
top	top	Print system usage and top resource hogs
logout (exit)	logout or exit	Quit a UNIX shell
ping	Ping 192.168.0.3	See if machine is alive

2.7 FILE SYSTEM

A file system can be defined as the logical collection of files on a partition or disk. Everything in UNIX is considered to be a file; for example, physical devices like DVD-ROM, USB device configurations and working functions are written as files. All the files in the UNIX file system are organized into a multi-level hierarchy.

At the very top of the file system is single directory called ‘‘root’’ which is represented by a / (slash). All other files are ‘‘descendents’’ of root (/).

Directory Structure: The directories have specific purposes and generally hold the same types of information for easily locating files. Following are the directories that exist on the major versions of UNIX–

Directory	Description
/	This is the root directory which should contain only the directories needed at the top level of the file structure.
/bin	This is where the executable files are located. They are available to all users.
/dev	These are device drivers.
/etc	Supervisor directory commands, configuration files, disk configuration files, valid user lists, groups, ethernet, hots, where to send critical messages.
/lib	Contains shared library files and sometimes other kernel-related files.
/boot	Contains files for booting the system.
/home	Contains the home directory for users and other accounts.
/mnt	Used to mount other temporary file systems, such as cdrom and floppy for the CD-ROM drive and floppy diskette drive, respectively.
/proc	Contains all processes marked as a file by process number or other information that is dynamic to the system.
/tmp	Holds temporary files used between system boots.
/usr	Used for miscellaneous purpose, or can be used by many users. Includes administrative commands, shared files, library files, and others.

/var	Typically contains variable-length files such as log and print files and any other type of file that may contain a variable amount of data.
/sbin	Contains binary (executable) files, usually for system administration. For example *fdisk* and *ifconfig* utilities.
/kernel	Contains kernel files.



CHECK YOUR PROGRESS

Q.5: Write the command to sort the lines of data in a file in reverse order?

Q.6: What will be the command to display the directory attributes rather than its contents?

Q.7: What is the purpose of /etc directory in UNIX OS?

Q.8: Write one difference between /bin and /sbin folder.



2.8 LET US SUM UP

- UNIX was one of the first operating systems to be written in a high-level programming language, namely C.
- UNIX is capable of handling activities from multiple users at the same time.
- The source code of UNIX OS is freely available.
- GRUB (GRand Unified Bootloader) is a boot loader package from the GNU Project. GRUB loads its configuration at start up, allowing boot-time changes, such as selecting different kernels or initial RAM disks.
- init is the first process started during booting of the computer system and continues running until the system is shut down.
- runlevel as the state of machine after boot and it will identify what system services are operating. /etc/inittab file, defines what each configured runlevel does in a given system.
- Kernel interacts with hardware and performs tasks like memory management, task scheduling and file management.

- Shell is the utility that processes your requests, when you type in a command at your terminal, the shell interprets the command and calls the program that you want.
- Everything in UNIX is considered to be a file.



2.9 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: Portability means that Unix OS can be installed on any hardware.

Ans. to Q. No. 2: Run levels are used to identify which subsystems are working, e.g., whether X is running, whether the network is operational or not.

Ans. to Q. No. 3: MBR-Master Boot Record. GRUB-Grand Unified Bootloader

Ans. to Q. No. 4: It is located in the 1st sector of the bootable disk having size 512 bytes.

Ans. to Q. No. 5: sort -r

Ans. to Q. No. 6: ls -l -d

Ans. to Q. No. 7: Contains the various system related configurations files.

Ans. to Q. No. 8: "sbin" contains binary (executable) files, usually for system administration and "bin" –this is where the executable files are located and are available to all user.



2.10 FURTHER READING

1) Kanetkar, Y. P. (2003): *Unix Shell Programming*; B. P. B. Publications.
2) Peek, J. D., Todino, G., & Strang, J. (1998): *Learning the UNIX Operating System*; O'Reilly.



2.11 MODEL QUESTIONS

Q.1: Explain the following commands with syntax

a) mkdir b) cp c) mv d) rm

Q.2: What is the difference between multi-user and multi-tasking?

Q.3: What is a UNIX shell? Briefly mention the various types of shell.

Q.4: How is UNIX different from Linux?

Q.5: How can you find out what a command does?

Q.6: Explain the booting process of UNIX Operating System.

Q.7: Briefly explain the UNIX Architecture.

*** ***** ***

UNIT 3: LINUX OPERATING SYSTEM

UNIT STRUCTURE

3.1 Learning Objectives

3.2 Introduction

3.3 LINUX Operating System

- 3.3.1 Login into Linux
- 3.3.2 Using the Desktop
- 3.3.3 Files and Folders

3.4 Concept of Paths

3.5 Let Us Sum Up

3.6 Answers to Check Your Progress

3.7 Further Reading

3.8 Model Questions

3.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- describe the Linux operating system
- login into Linux
- describe files and directory structure
- learn about the concept of path.

3.2 INTRODUCTION

We have already discussed the MS DOS and Unix operating system. In the previous units, we have also learned how to create files and directories and the various operations on files and directories like cut, copy, paste, rename etc. in both the operating systems i.e. MS DOS and Unix.

In this unit, we will introduce you to the LINUX operating system. We will discuss how to login into LINUX, customizing the desktop and file operations.

3.3 LINUX OPERATING SYSTEM

What is Linux ?

Linux (pronounced “LIH-nucks”) is a 32-bit operating system (OS) that can be used on virtually every computer hardware platform. Originally designed for the Intel x86 platform, it is now available for Intel, Mac, Sparc, and Alpha, of embedded devices and many more platforms. Linux is a UNIX-like OS that is very stable, reliable, and flexible. It is used on PCs, professional workstations, servers, routers, and practically on every other computing platform. Linux works well in all these environments, and since it is multi-user and multi-tasking, it can perform virtually any job or group of jobs quickly and efficiently. Linux is as powerful as any other operating system, and most importantly, it is free. Linux was created to be a free alternative to UNIX and has become a competitor for all operating systems available today. This fact has enabled Linux to greatly increase its user base, and to be supported and used by the largest corporations and governments in the world. Because of its open nature and technical strengths, Enterprise use of Linux is best illustrated by its large-scale use on the Internet– it powers a majority of Internet applications, including the Apache Web server. Linux is a direct competitor with other high profile operating systems, such as UNIX and Microsoft Windows.

Who Started Linux?

Linus Torvalds started the Linux project while he was a student at the University of Helsinki. He worked to create a UNIX-like operating system equipped with more features than Minix.Torvalds created a Linux kernel that could work with UNIX applications, and in 1991, he released his first kernel for the Intel x86 platform, which was widely distributed over the Internet. A *kernel* is the core of any operating system, and the Linux kernel was built to work like UNIX but doesn’t use any of the UNIX code– this is why Linux is not UNIX. The kernel is what allows most softwares to access the hardware that it is installed on. The key to Linux is its kernel because the kernel allows other programmers to refine it, add to it, and incorporate most of the features and applications of UNIX. Most standard UNIX

applications have been ported to Linux, such as windows managers, Internet utilities, program development utilities, and just about every other tool used in computing today.

Current Application of Linux systems

Today **Linux** has joined the desktop market. Linux developers concentrated on networking and services in the beginning, and office applications have been the last barrier to be taken down. We don't like to admit that Microsoft is ruling this market, so plenty of alternatives have been started over the last couple of years to make Linux an acceptable choice as a workstation, providing an easy user interface and MS compatible office applications like word processors, spreadsheets, presentations and the like.

Today more than 97% of the world's supercomputers (including the complete top 10), more than 80% of all smartphones, many millions of desktop computers, around 70% of all web servers, a large chunk of tablet computers, and several appliances (dvd players, washing machines, dsl modems, routers, self-driving cars, space station laptops...) run **Linux**. Linux is to far the most commonly used operating system in the world.

On the server side, Linux is well known as a stable and reliable platform, providing database and trading services for companies like Amazon, the well known online bookshop, US Post Office, the German army etc. Especially, Internet providers and Internet service providers have grown fond of Linux as firewall, proxy and web server, and you will find a Linux box within reach of every UNIX system administrator who appreciates a comfortable management station. Clusters of Linux machines are used in the creation of movies such as "Titanic", "Shrek" and others. In post offices, they are the nerve centers that route mail and in large search engine, clusters are used to perform internet searches. These are only a few of the thousands of heavy duty jobs that Linux is performing day to day across the world.

It is also worth noting that modern Linux not only runs on workstations, ''mid and high" end servers, but also on "gadgets" like PDA's, mobiles, a shipload of embedded applications and even on experimental wrist watches. This makes Linux the only operating system in the world covering such a wide range of hardware.

Linux kernel version 4.0 was released in April 2015, which is contributed to by hundred of commercial companies Red Hat,Intel, Samsung, Broadcom, Texas Instruments, IBM, Novell, Qualcomm, Nokia, Oracle, Google, AMD and even Microsoft (and many more).

3.3.1 Login into LINUX

In order to start using your system you will usually have to perform what is known as a user login. This procedure is necessary to identify yourself to the system. The system allows multiple users to use it concurrently and so it has to be able to identify a user in order to grant them the necessary privileges and rights to use the system and applications. Each user, upon successful login, will be assigned to his home directory (folder).

Some system may have been set up so that upon power-on, a default user is logged on automatically. If this is so, then you will not be presented with a login screen or prompt as described in the section below.

Another important thing you will have to know is that there is a special user called the **root** or **superuser** (this user is usually created during the system installation) which has unlimited access and rights to all the system files and resources. You only need to login as root if certain system level administrative tasks are to be carried out. Otherwise there is usually no need for a normal user to login as root. This is to prevent accidentally damaging the system by deleting or modifying important system files.

Depending on how you have set up your system, you will either have a graphical login screen or a text-based login prompt for you to perform the login process. To login, first enter the username followed by the ENTER key and when the password prompt appears, enter the password followed by the ENTER key.

Figure 3.1 and Figure 3.2 depicts the GUI based login form and test based login form respectively.

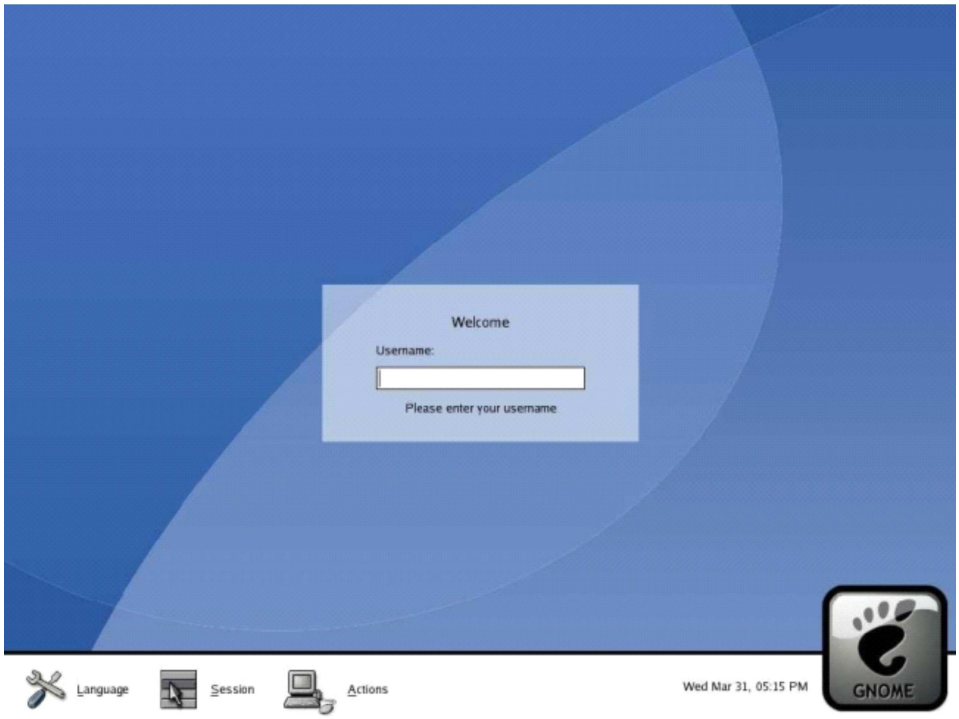


Fig. 3.1: Graphical User Interfull

Fedora Core release 1

Kernel 2.4.22-1 on an i686

localhost login:

Fig. 3.2: Text-based Login Prompt

Starting The Graphical Desktop: If you have logged in from the graphical login screen, the graphical desktop will be started automatically. The graphical desktop presents a Graphical User Interface (GUI) for the user to interact with the system and run applications. If you have used the text-based login screen, you will hav to start the graphical desktop manually by entering the command **startx** followed by the ENTER key.

```
[anita@localhost anita]$ startx
```

Fig. 3.3: Starting the Graphical Desktop

The graphical desktop that we will be using throughout most of this unit is called the GNOME desktop. There is another desktop environment in popular use on Linux systems– the KDE desktop.

Using the Mouse on the Desktop: Most Linux graphical desktops are designed for use with a 3-button mouse. If you are using a 2-button mouse, during installation, it should have been configured to emulate the middle-button of a 3-button mouse by pressing both buttons simultaneously.

To click on a mouse the left button is depressed. (A mouse configured for a left-handed user will need to have its right button depressed.) The right (or left button for a left handed mouse) and middle buttons are usually used to invoke special or specific features of the GUI and instructions to do so will be explicitly given. The term "clicking on the mouse" means that you click on the mouse once. The term "double clicking" means that you click twice in succession on the mouse. Figure 3.4 shows a typical view of desktop in Linux environment.

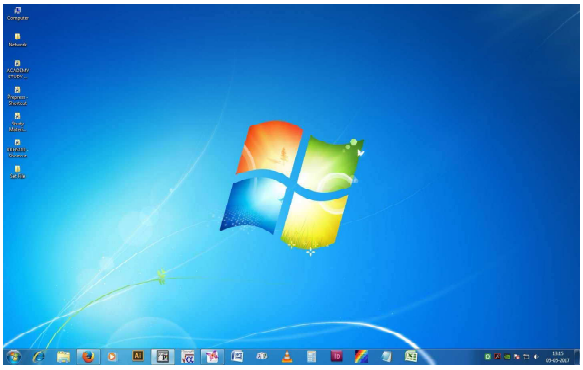


Fig. 3.4: Typical view of Desktop

To "select" an item means clicking it once with the mouse.

The term "drag and drop" means that you will have to click on an item and continuing to hold the mouse button down, drag the item to another place and on reaching its destination drop it by releasing the mouse button.

Main Components of the Desktop: The figure below shows a typical view of the graphical Linux desktop.

There are 3 main components on the desktop:

- the Menu System
- the Panel
- the Desktop itself

The Menu System: The main menu can be accessed by clicking on the Main Menu button located at the extreme lower left hand corner of the desktop. This may be portrayed by one of several icons depending on the desktop environment used.



Fig. 3.5: The Menu System

Generic GNOME Main Menu button

Fedora/Red Hat Main Menu button



Clicking on this will bring up the menu system as shown in the figure below. From the Menu System you can start many of the applications installed on your Linux system. Note that the menu system consists of a Main Menu panel and sub-menu panels. Each entry in the menu system which has an arrow on its right means that it is an entry point to a sub-menu, and there can be sub-menus within each sub-menu. In this way applications in the menu Ssystem can be organised and categorised for easy reference and access.



Fig. 3.6: Main Menu and Sub-Menu

The Panel: The long bar across the bottom of the desktop screen is called the panel. The panel contains the main menu icon, the application launcher icons, a notification area and applets. Installed by default are several application launcher icons on the Panel. Clicking on one of these will run an application. Commonly accessed applications can be added to the panel and those that are less frequently used can be taken off.

The notification area holds alert icons so that the user can be alerted to critical messages.

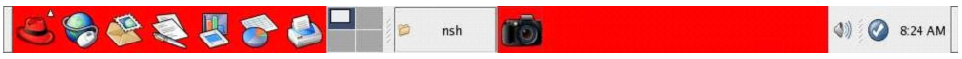


Fig. 3.7: The Panel

Applets are small applications that run on the panel. These usually perform useful and informative tasks like setting the sound level of the soundcard, monitoring whether the system software needs an update, etc. By default the following applets are run.

The Workspace Switcher: The graphical desktop can be regarded as a workspace drawing an nalogy with the working area on a real physical tabletop. Programs are run, documents displayed and files opened on the workspace. To cut down on workspace clutter and to enable the user to organise his workspace more efficiently, the graphical desktop environment allows the usage of multiple workspaces. Each workspace can be considered as a virtual desktop.

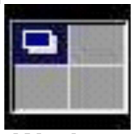


Fig. 3.8: The Workspace Switcher

By default the user has 4 desktop workspace areas to work on. The workspace switcher represents each workspace as a small square and shows the applications running in each of them. To access a workspace click on the square with the mouse.

The Taskbar: The taskbar applet is located next to the workspace switcher and shows the titles of all the running applications in a virtual desktop (a workspace).

The Desktop Space: The desktop space refers to the rest of the screen. It contains icons which are graphical representations of shortcuts to application launchers, file folders, files and peripheral devices like floppy disks, CD-ROM drives and printers. Double-clicking on an icon representing an application will launch or execute the application. Commonly used applications and files are usually placed on the desktop space.

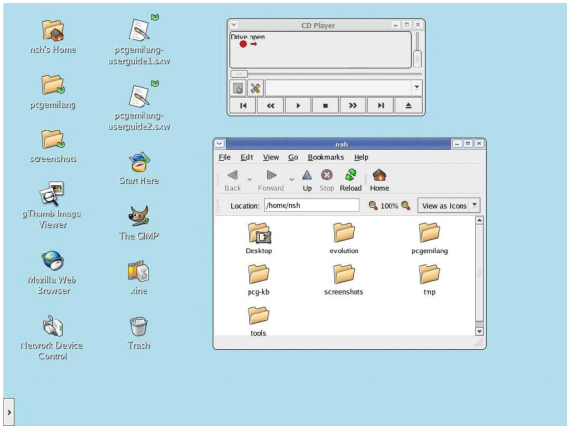


Fig. 3.9: Desktop Space

Using the Window Manager: An interactive application that is run on the graphical desktop, is usually displayed inside a window. This window can be accessed and manipulated using the window manager.

The Titlebar: When an application is started an application window opens and a frame as shown in Figure 3.10 is placed around the window of the application. The top edge of the frame has a titlebar that contains the title of the application.

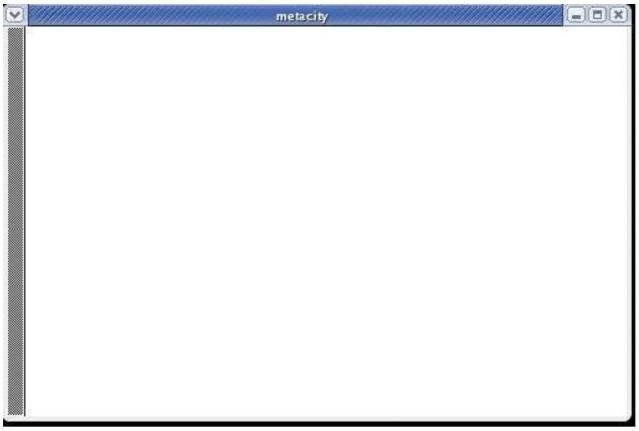


Fig 3.10: An Application Window Frame

At the left hand corner of the titlebar is the Window Menu button.

Clicking on this button will open up the Window Menu.



Fig 3.11: The Window Titlebar

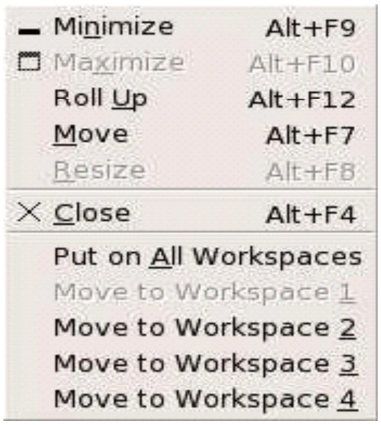


Fig 3.12: The Window Menu

You can perform operations such as minimize, maximize, roll up, move and resize the window from this menu. At the right

edge of the titlebar there are three buttons which allow you to minimise, maximise and close the window.

To perform an operation in the Window Menu, open the menu by clicking on the Window Menu button and then select the desired operation.

Logout: When you have finished working on the system, you will need to logout. Logging out will inform the system that you are no longer using the system's resources. All the files opened and programs run by you will be closed and/or stopped unless you have specifically informed the system to keep them open or running for you.To logout, at the Main Menu select:

Main Menu → Logout

and at the dialog window select logout and click OK.

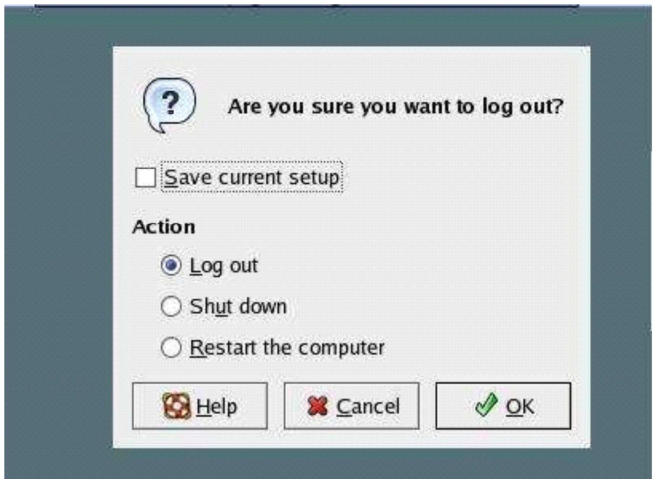


Fig 3.13: Logout Dialog Window

Shutting Down the System: When you have finished using the computer and want to power it off, you will have to perform a system shutdown.

It is very important that a proper system shutdown is performed. You should not just turn off the power switch of the computer to shut it down. Failure to observe this may lead to system software and data corruption and failure.

To shutdown, at the Main Menu, perform a logout. Then at the graphical login screen select the "Shut down" option at the bottom of the screen. Sometimes (depending on the login screen chosen)

the Shutdown option is available as a sub-option under the Actions option at the bottom of the screen.

Alternatively on some systems, you may be able to perform a shut down by performing a log out operation as described in the previous section but instead of selecting “Log out”, select the “Shut down” option from the Logout dialog.

3.3.2 Using the Desktop

Basic Desktop Customization: The desktop can be customised to your preferences and tastes in a variety of ways. Here we will explore some basic customisations. More details can be obtained by running the “Help” application on the Main Menu. After running the Help application choose:

Desktop → Basic Preferences

Desktop Background: The background image displayed on the desktop can be changed by running the Background application from the Main Menu (this is in the Preferences sub-menu).

Main Menu → Preferences → Background

To change the background image, click on the square marked “Select picture”. An image selector dialog is displayed. Choose an image from the dialog. If you want to choose an image from another directory, click the Browse button. When you have chosen an image, click OK.

Desktop Themes: The desktop theme specifies the visual appearance of the panels, applets, and windows. The desktop theme may also specify the appearance of interface items in applications. For example, the theme affects the appearance of buttons, scrollbars, check boxes, and so on in the applications.

The theme used by the Desktop can be changed by running the Theme application from the Main Menu (this is in the Preferences sub-menu).

Main Menu → Preferences → Theme

An theme selector dialog is displayed. To change the theme, click on a new theme. The screenshot below shows a sample of some of the themes which may be available. Note that the actual themes available can vary from system to system.

Default Fonts: The default fonts used to display applications and the desktop background can be changed by running the Fonts application from the Main Menu (this is in the Preferences sub-menu).

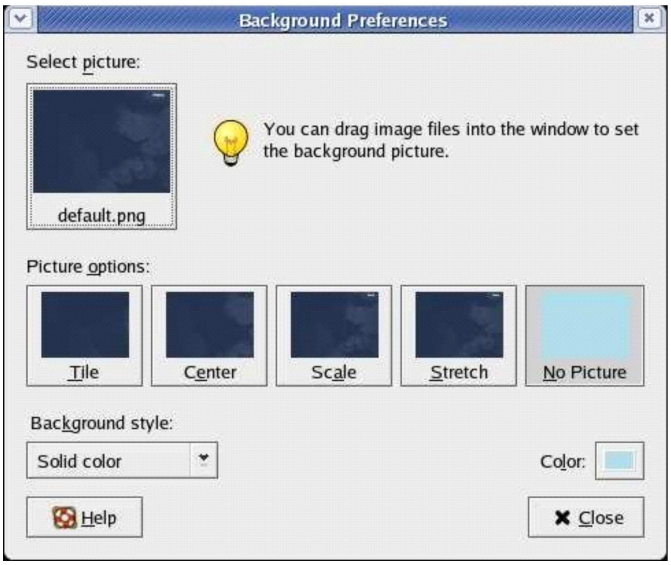


Fig. 3.14: Selecting New Background Image

Main Menu → Preferences → Font

A selector dialog for the application, window title, dialog and terminal fonts are displayed. To change the font for each category of usage, click on the space listing the font.

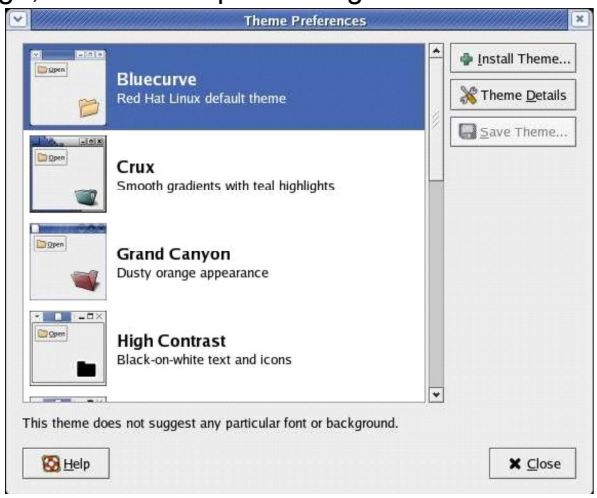


Fig. 3.15: Selecting a New Theme

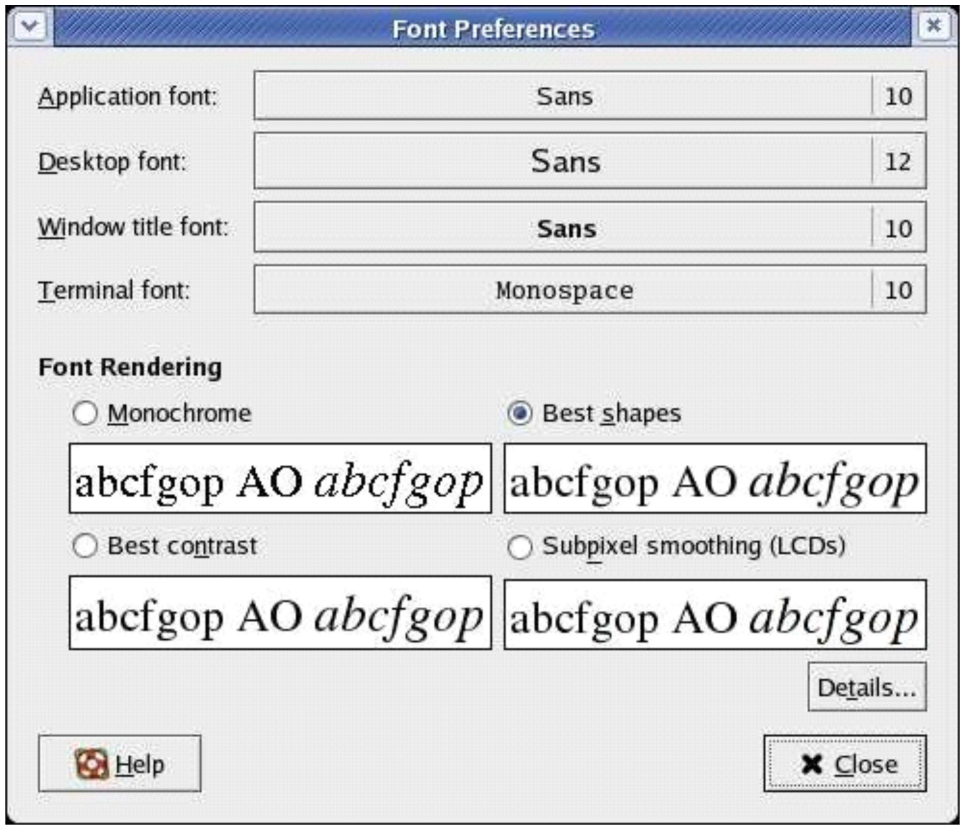


Fig. 3.16: Selecting New Fonts

Menus and Toolbars: The Menus & Toolbars tool is used to customise the appearance of menus, menubars, and toolbars. Again this can be run from the Preferences of sub-menu in the Main Menu.

Main Menu → Preferences → Menus & Toolbars

Experiment on each of the settings to get the ones preferred.

Using the Desktop: The Desktop provides a useful metaphor for a modern office work environment. So on the Desktop we find that the applications that we want to run can easily be started or launched and the information and data that we need can easily be located.

Running Applications: Applications that we can run from the Desktop are to be found either from the Main Menu (and sub-menus therein), or as icons on the Panel and the Desktop itself.

To run an application from the Main Menu, open up the menu (or sub-menu) and click on the application listed in the menu bar.

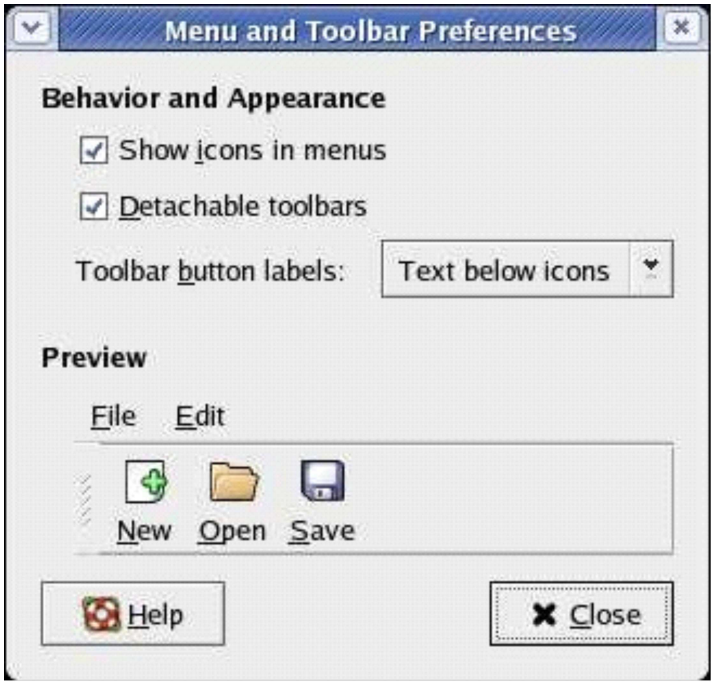


Fig. 3.17: Customising Menus and Toolbars

To run an application from the Panel, click once on the icon representing the application.

To run an application from the Desktop itself, double-click on the icon representing the application.

Sometimes it is more handy to have the application as an icon on the Panel or Desktop where you can launch it more easily by just clicking on it, rather than in the Main Menu, especially if it is buried deep within several sub-menus. To achieve this, open the Main Menu and select the application item listed in it by clicking with the right mouse button. Click on the selection “Add this launcher to panel” and a copy of the application icon will be placed on the Panel.

To make a copy of this on the Desktop, you can drag the icon from the Panel over to the Desktop.

To delete an application icon from the Panel right-click on it and select the “Remove from Panel” option.

To delete an application icon from the Desktop, right click on it and select the “Move to Trash” option.

The Panel: The Panel has many useful utilities called applets. Applets are small applications that run on the Panel. As discussed in Chapter

One, by default, the taskbar and the workspace switcher applets are placed and run on the Panel. Other useful applets that may be placed on the Panel include:

– clock
– sticky notes
– volume control

Placing an Applet on the Panel: To place an applet on the Panel, move the mouse over to an empty space on the Panel and right-click it. Select the item "Add to Panel" and from the sub-menus select the applet to place on the Panel. For example, to place the "sticky notes" applet, select:

Add to Panel → Accessories → Sticky Notes

To prevent accidental removal of an applet, you can lock it on the Panel by right-clicking on its icon and selecting "Lock".

Removing an Applet from the Panel: To remove an applet, right-click on the applet icon and select "Remove from Panel". If the applet is locked, you will have to unlock it first by right-clicking on the icon and selecting "Unlock".

Configuring the Panel: To change the properties of the Panel, right-click on the Panel and select "Properties".



Fig. 3.18: General Properties of the Panel

Setting the Date and Time: It is important that the date and the time are set correctly in your system. This will make it easier to manage the system resources and files and also aid in troubleshooting any problems. To set the date and the time, run the Date & Time application tool from the System Settings sub-menu under the Main Menu.

Main Menu → System Settings → Date & Time

Alternatively, you can right-click on the clock (time) applet display on the Panel and choose “Adjust Date & Time”.

A dialog box asking for the root password will appear if you are not logged in as root. This is because the system date and time are important system parameters and so only the system administrator or root is allowed to do it. After entering the correct root password, you can change the date using the displayed calender and the time in the boxes provided. Select the Time Zone tab and check to see that the timezone selected is correct. If not, correct it by selecting the appropriate timezone region. Usually, the system uses the local time, so do not select the “System clock uses UTC” checkbox.

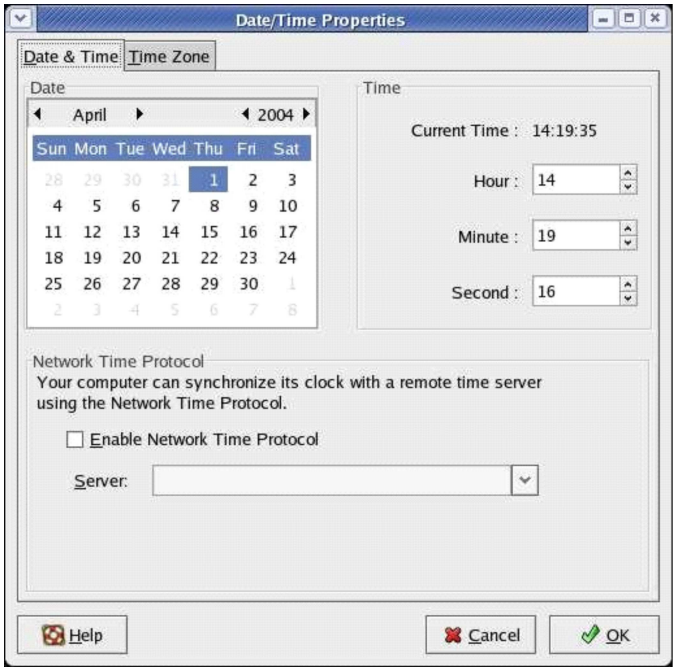


Fig. 3.19: Setting the Date and Time

Click on the OK button after all is done to enable the new date and time. You can check for the new date and time by resting your mouse over the clock (time) applet display on the Panel at the bottom of the screen.

3.3.3 Files and Folders

The File System: One of the most powerful features of a modern computer system is the ability to store data in a form which can be easily retrieved and transported or copied across to other computer systems or media. Data created and accessed by a user in the computer is stored in what is called a *file*. This concept of an electronic file to store electronic data mimics the physical world usage of a file to store data written on paper. This mimicry is carried further by the organisation of these electronic files into electronic folders or directories. Like a physical folder, an electronic folder can contain very many files. A folder may also have sub-folders or subdirectories.

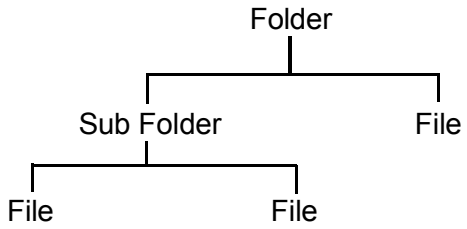


Fig. 3.20: The Hierarchical File System

Files and folders can be created, copied, moved (i.e. transferred) and deleted. The folders themselves are organised in a hierarchical manner starting at the root of the file system. Each user is given a home directory and upon logging in, the user is placed in his home directory.

Ownership and Permission: As the system is set up to handle multiple users concurrently, it needs to have in place the mechanism with which it can control the security and privacy of the file system. It needs to be able to control access to the file system resources for each individual user. In addition it also has to be able to control access at a group level i.e. users who belong to the same group can be given certain privileges with respect to the file system operations.

To be able to perform these features, associated with each file or folder are the categories:

– owner
– group
– others

and the file permissions:

– read (r)
– write (w)
– execute (x)

and the directory permissions:

– read (r)
– write (w)
– access (x)

Associated with a file are the categories like **owner**, **group** and **others** and the permissions which these have on the file. The userid of the user who creates a file by default becomes the owner of the file. When a userid is created on the system it is assigned to one or more groups. (For some Linux systems, when a userid is created on the system, a group which has the same name as the userid is also created and the new userid belongs to this group.) When a user logs in the userid will be assigned to a default group. This default group to which the file owner belongs is associated with the file and this is known as the group owner. All of the other userids on the system which are not the owner of the file or belong to the group associated with the file, are placed in the category “others”.

By default, the owner has to ***read*** and ***write*** permission for a *non-executable file* and ***read***,***write*** and ***execute*** permission for an *executable file*. The group associated with the file has read permission for a non-executable file and read, execute rights for an executable file. The others group has read only permission for non-executable file and read, execute rights for an executable file.

The rights and ownership concepts described above apply to directories too. However, since a directory cannot be executed,

access rights is substituted for execution rights. Access to a directory means that the userid with the appropriate permission can descend into the directory (i.e. change directory to it).

The File Manager application described below may be used to view and modify the ownership and permissions of a file or folder. Only the owner of a file/folder can change its permissions and only the *superuser* or *root* can change the ownership of a file/folder.

Using The File Manager: As it is possible for a user to create and store hundreds and thousands of files and folders, a File Manager is needed to assist the user to manage and manipulate these files and the file system on which it resides.

In this section a brief description of the File Manager and how to use it is given. For more details you should consult the "Help" application on the Main Menu. From the Help application choose:

Desktop → Nautilus File Manager

To start using the File Manager double-click on the home directory icon on the desktop. (This may be named "username's Home" where username is the username of the user currently logged in.)

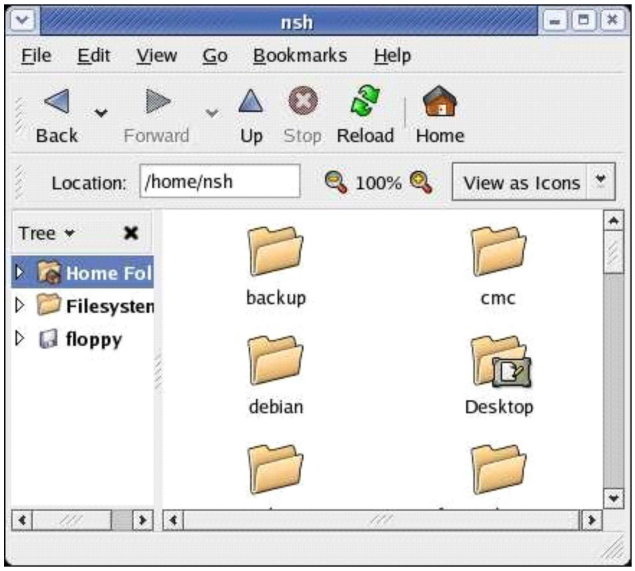


Fig. 3.21: The File Manager

By default, the File Manager window consists of a side pane and a main view pane. At the top of the window just under the window titlebar, are the menu bar and the location bar.

The view pane displays the files and folders contained in the current directory that the user is in. These can be displayed as icons (default) or changed to display them as a listing of filenames.

The side pane contains an icon that represents the current file or current folder. The side pane also contains information about the current file or current folder. A hierarchical (tree) view of the file system on the computer can be obtained from the side pane. By navigating through this, you can access files and folders outside your home directory (provided of course that you have the permission to do so).

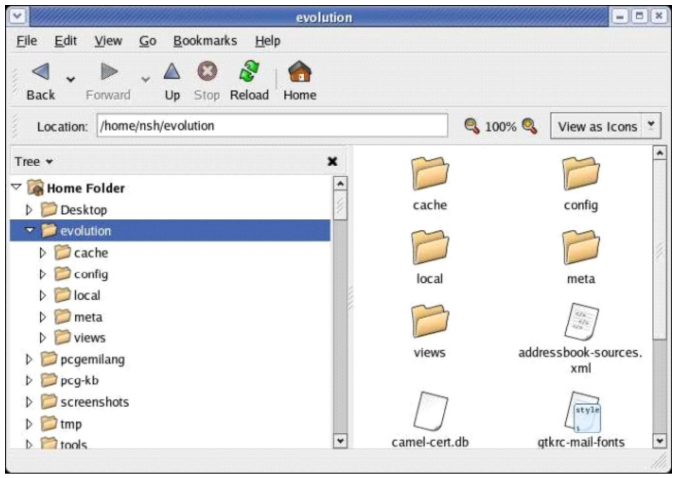


Fig. 3.22: The side Pane with a Tree View of Folders

Accessing Files and Programs: From the view pane of the File Manager, you can open files as well as run (launch) executable programs by double clicking on the icon representing them.

Creating New Folders: To create a new folder under the folder you are currently in, move the mouse to the menu bar at the top (just beneath the titlebar of the window) and click on File and then click on Create Folder. A new folder will appear on the view pane and you will have to type in the name of the new folder. Copy Files and Folders.

To copy a file, click on the file in the view pane to select it. Then select from the menu bar at the top,

Edit → Copy File

Next, open up the folder to which you want to copy the file and then select from the menu bar at the top,

Edit → Paste Files

Another way to copy a file is to right click on the file icon and then select "Copy File". Then navigate to the icon of the folder where you want the copy to be placed in and then right click on the folder icon and select "Paste Files into Folder".

The procedures above can be done with folders too.

To copy more than one file or folder at a time, select multiple files/folders by holding down the CTRL key while clicking on the files or folders.

Moving Files and Folders: Moving a file or folder is different from copying in that a copy of the file/folder is not made, i.e. only one copy of the file/folder exists, and the file/folder is transferred from one folder to another.

To move a file, click on the file in the view pane to select it. Then select from the menu bar at the top,

Edit → Cut File

Next, open up the folder to which you want to move the file and then select from the menu bar at the top,

Edit → Paste Files

Another way of moving a file is to drag and drop the file into the destination folder.

The procedures above can be done with folders too.

To move more than one file or folder at a time, select multiple files/folders by holding down the CTRL key while clicking on the files or folders.

Renaming Files and Folders: To rename a file, click on the file in the view pane to select it. Then select from the menu bar at the top,

Edit → Rename

and then type in the new name.

Alternately, you can also right-click on the file and then select "Rename".

The procedures above can be done with folders too.

Deleting Files and Folders: To delete a file, click on the file in the view pane to select it. Then select from the menu bar at the top,

Edit → Move to Trash

Alternately, you can select the file and then use the DELETE key on the keyboard to delete the file. This has the same effect as above of moving the file to the Trash folder.

It is still possible to salvage a deleted file from the Trash. To do this double-click on the Trash icon on the desktop to open up the Trash folder. Then you can move the file you want to salvage to the desired folder. Note that if you delete the file from the Trash then it cannot be recovered anymore.

The procedures above can be done with folders too.

To delete more than one file or folder at a time, select multiple files/folders by holding down the CTRL key while clicking on the files or folders.

Viewing and Modifying the Permissions of a File or Folder: To view the owner and group of a file/folder and/or to modify its permission settings, select the file/folder and select from the menu bar at the top,

File → Properties

Click on the Permissions tab. The owner and group of the file/folder are displayed as well as the associated permissions.

To change the permissions, modify the check boxes accordingly to the new permissions desired. Click on Close to apply the changes.

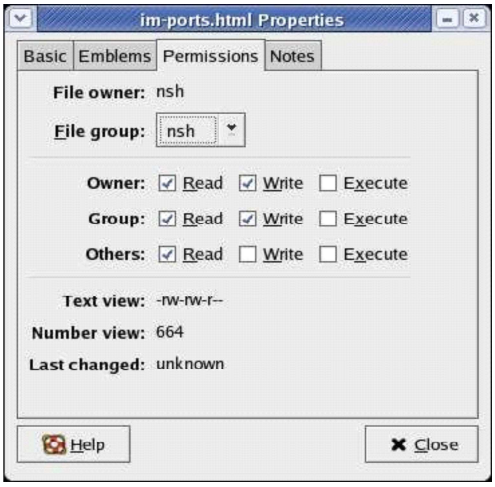


Fig. 3.23: Ownership and Permission Properties of a File

The Trash Can: The trash can icon on the desktop contains files and folders which you have deleted using the File Manager or an application run from the desktop. As noted above, it is a special folder in which it is still possible to move back the deleted items to another folder so as to "undelete" them. However, over time, the trash folder will be filled up as many files and folders get deleted through usage. So, it is a good idea to periodically empty or delete the items found in the trash.

To empty the trash, right-click on the trash icon and select "Empty Trash". Otherwise you can open the trash by double-clicking on it and then select to delete all the items in the trash folder using the File Manager functions described above.



A text file here refers to a file which contains pure text printing characters only. Some types of files e.g. those created by a word processor, while appearing to be text-only actually contain other non-printing characters and hence are not pure text files.



EXERCISE 3.1

1) Use the File Manager to perform the following:
   – Create a new folder called *testdir* under the home directory
   – Copy the following files into the folder *testdir:*
   – /etc/services, /etc/hosts
   – Access the *testdir* folder:
   – Open the two files there to view their contents, then close them
   – Rename the file *hosts* to *hosts.backup* and *services* to *services.copy*
   – Move the file *hosts.backup* to the /usr/tmp directory
   – Delete the file *services.copy*
   – Move up to the home directory
   – Delete the folder *testdir*
   – Launch the application *xeyes* found in the folder */usr/X11 R6/bin*
2) Check the trash can and restore the deleted file(s) there.

Using a Text Editor: Very often it is necessary to use the computer to input some text or to modify the data in a text file. While a full-fledged word processor like OpenOffice.org's Writer may be used,

it can be an over kill since the sophisticated features and formatting available with a word processor are not needed in many cases. A text editor can be used instead. The Linux system comes installed with many text editors. In this section we shall be looking at how to use a text editor which comes with the GNOME Desktop – gedit.

Starting Gedit: Gedit may be started from the Main Menu,

Main Menu → Accessories → Text Editor

After gedit has started, you can immediately type in text into the main edit window. The mouse is also used extensively in gedit, to move around the document as well as to perform editing functions.

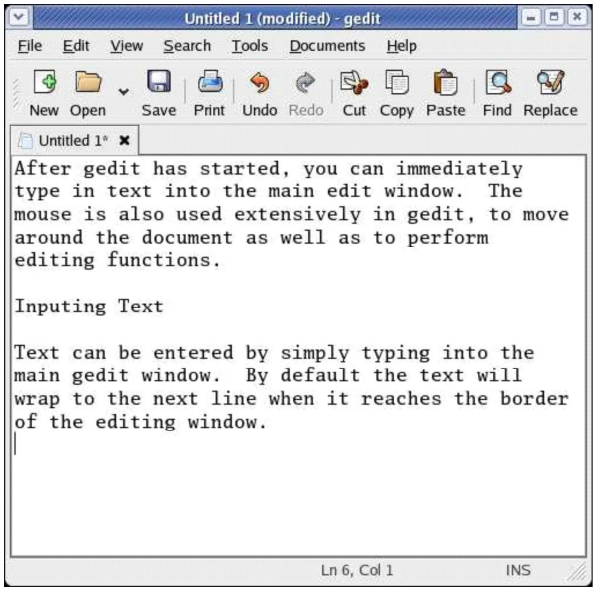


Fig. 3.24: The Gedit Text Editor

Inputting Text: Text can be entered by simply typing into the main gedit window. By default the text will wrap to the next line when it reaches the border of the editing window.

Marking Text: The mouse may be used to mark a block of text by clicking on the beginning of the block and dragging the mouse to the end of the block before releasing it.

Deleting Text: Text can be deleted one character at a time by moving the mouse to the character in question and using the DELETE and BACKSPACE keys. The DELETE key will delete the character on which the cursor is resting on (the current cursor position) while the

BACKSPACE key will delete the character just before the current cursor position.

If there is a lot of text to be deleted, a more efficient way to delete text is to mark it first and then press the DELETE key on the keyboard.

Cut, Copy and Paste: Text can be edited by marking it first and then selecting Cut, or Copy from the buttons at the top. “Cutting” will result in the marked text being copied into a storage buffer and then deleted from the editing window, while “Copying” will result in just a copy of the text being copied into the storage buffer without the deletion of the original text. The text cut/copied in this way can then be pasted into another part of the document by moving the mouse to the desired location and selecting the Paste button at the top.

Another way of invoking these functions is to click on the Edit option on the top menu bar and selecting the desired function, e.g.

Edit → Cut

Edit → Copy

Edit → Past

Undo: After performing an editing function, e.g. deleting a block of text, you can undo the action by clicking on the Undo button at the top. The Undo function can also be invoked using the Edit menu selection at the top.

Saving Text: The text typed in can be saved by clicking on the File option on the main menu bar at the top and selecting,

File → Save As ...

Move to the folder you want to save in the file and enter the name of the file to save.

Find and Replace: The Find button can be used to locate a text string. By default this search is not case sensitive and will also locate the text if it is part of a word. These default options can be changed from the Find dialog.

The Replace button can be used to locate a text string and replacing it with another. Again, the default options of noncase sensitive and matching part of a word can be changed.

The Find and Replace functions can also be performed using the Search menu selection at the top.

Opening a File: You can edit a text document already saved on the system by opening it. From the main menu at the top choose,

File → Open

and select the file to be opened.

Preference: Preferences can be set from the Preferences menu selection,

Edit → Preferences

The Preferences you can set include the font, colour and point size to be used, tab spacing, autowrapping, and autoindenting.

Help: More information on how to use gedit can be obtained by selecting Help from the menu at the top.



EXERCISE 3.2

1) Use the gedit text editor to type in the first page of this guide. Save the document in a file on your home folder.
2) Open the saved file and edit it by:
   – deleting the second paragraph on page one
   – adding the statement “End of Chapter” to the last page
   – save the edited document as a new file.

3.4 CONCEPT OF PATHS

In Linux, all file systems (Hard Disks, CD-ROMs, Floppy Disks, ZIP drives, network mounts, etc. …) are “mounted” onto one logical tree. The top of the hierarchy is traditionally called **root** which is represented by a */* (**slash**). In Linux, everything is treated as a file. A directory is a file. It is a file that contains a list of files and information belonging to those files. This would include things like who “owns” (created) the file, how long it is, and who can use it. Since a directory is simply a list of files, it can contain any file in it, including other directories.

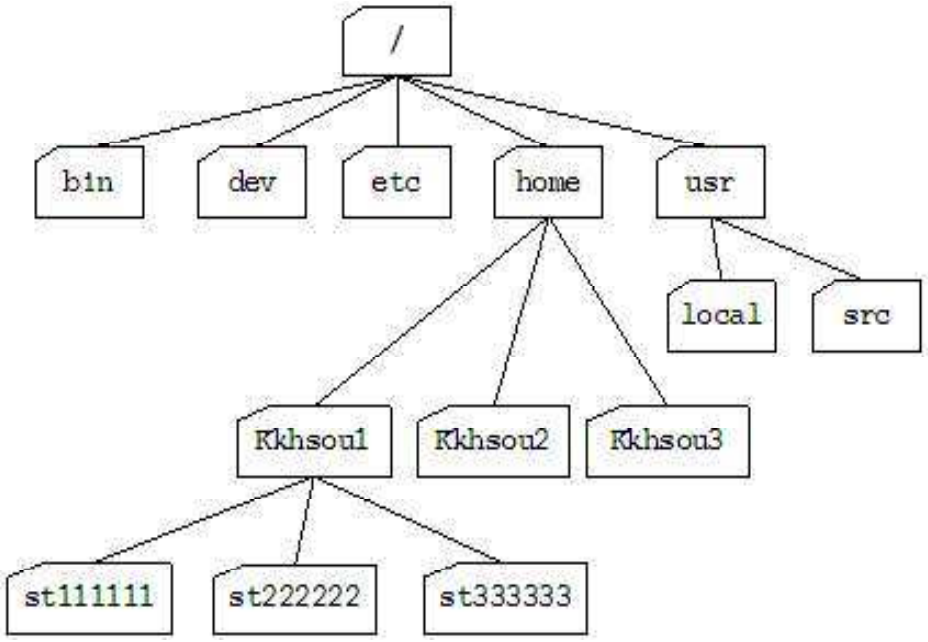


Fig. 3.25: Part of file-system Tree

Absolute and Relative Names: You can specify a file or directory by its path name. There are two ways of expressing the path name: Full (absolute) path name or relative path name. The full path name starts with the root. /, and follows the branches of the file system, each separated by, until you reach the desired file.

However, a relative path name specifies the path relative to your current working directory. Relative path names are more convenient because they are shorter, but they must be used with care. They never begin with / (slash). Now, we have to introduce two special directory entries:

- the current directory
- the parent directory

As an example, (Using the above diagram) assume that your current working directory (your current position in the file system) is **kkhsou1**:

1) A file named "**test.txt**" is placed inside the directory "**st111111**".
   The absolute (full) name is: ***/home/kkhsou1/st111111/test.txt***
   The relative name is: ***st111111/test.txt*** OR ***./st111111/test.txt***
2) A directory named "**DEC**" is placed inside the directory "**kkhsou2**".
   The absolute (full) name is: **/home/kkhsou2/DEC**
   The relative name is: **../kkhsou2/DEC**

3) A file named "**csh**" is placed inside the directory "**bin**".
   The absolute (full) name is: **/bin/csh**
   The relative name is: **../../bin/csh**
4) A directory named "**st123456**" is placed inside the directory "**kkhsou1**".
   The absolute (full) name is: **/home/kkhsou1/st123456**
   The relative name is: **st123456** OR **./st123456**



CHECK YOUR PROGRESS

Q.1: Fill in the blanks:

a) Operating system is a collection of several .....
b) Linux is a ..... operating system.
c) LINUX was conceived by .....
d) Most of the LINUX code is written in the high leve language .....
e) The shell acts as the ..... interpreter in LINUX.
f) When a file is created the default access permission is .....

Q.2: In LINUX, full pathname originates from–

a) your home directory
b) the root directory
c) your working directory
d) None of the above

Q.3: A write permission to a directory allows a user to:

a) edit and create files located in the directory
b) edit but not create files located in the directory
c) search a directory for the presence of a file
d) none of the above



3.5 LET US SUM UP

- LINUX is multiuser and mulltitasking operating system.
- Linus Torvalds was the creator of the LINUC operating system.

- LINUX system is described as *Kernel* and *Shell.*
- *Kernel* is a main program of LINUX system. It controls hardwares i.e. CPU, memory, hard disk, network cards etc.
- *Shell* is the interface between user and kernel. Shell interprets user's input as command and pass them to Kernel.
- In LINUX data, directory, process, hard disk etc (almost everything) are expressed as a file.
- Files are put in a directory. All directories are hierarchical structure (*tree* structure)
- User can put and remove any directories on the tree. Top directory is "/", which is called slash or *root*. Users have their own directoory (*home directory*). *Path* means a position in the directory tree.
- Linux works well in all these environments, and because it is multi-user and multi-tasking, it can perform virtually any job or group of jobs quickly and efficiently.
- There is a special user called the **root** or **superuser** (this user is usually created during the system installation) which has unlimited access and rights to all the system files and resources.
- The Panel contains the Main Menu icon, the application launcher icons, a notification area and applets. The Menus & Toolbars tool is used to customize the appearance of menus, menu bars, and toolbars.
- The graphical desktop can be regarded as a workspace drawing an analogy with the working area on a real physical tabletop. Programs are run, documents displayed and files opened on the workspace.
- To shutdown, at the Main Menu, perform a logout. Logging out will inform the system that you are no longer using the system's resources.
- To be able to control access at a group level i.e. users who belong to the same group can be given certain privileges with respect to the file system operations. To be able to perform these features, associated with each file or folder are the categories:
  – owner
  – group
  – others
  – access (x)

and the file permissions:

– read (r)

– write (w)

– execute (x)



3.6 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: a) System Programs; b) Multiuser; c) Linus Torvalds; d) C; e) Command; f) r w

Ans. to Q. No. 2: (b)

Ans. to Q. No. 3: (a)



3.7 FURTHER READING

1) Craig: *Teach Yourself Linux in 24 Hours*; B. P. B. Publication.

2) Witherspoon, C: *Red Hat Linux 6 Fast & Easy*; B. P. B. Publication.



3.8 MODEL QUESTIONS

Q.1: Suppose you have a computer system having the LINUX OS. Now solve the following problem–

i) Show the procedure how to set the date and time of the system.

ii) Changing the appearance of the desktop clock.

iii) Create a launcher to run the text editor command “gedit” on the Desktop. Use an appropriate icon for the text editor.

iv) The Mozilla web browser is a popular web browser application for Linux systems. It has an easily recognisable icon– a red dinosaur head.

- Locate the Mozilla application in the Main Menu and place it on the Panel as well as the Desktop.
- Launch the Mozilla application from:

*** ***** ***

UNIT 4: LINUX COMMANDS

UNIT STRUCTURE

4.1 Learning Objectives

4.2 Introduction

4.3 Linux Commands

4.3.1 Internal Commands

4.3.2 External Commands

4.4 Directory Handling Commands

4.5 File Handling Commands

4.6 Utility Commands

4.7 File Permissions

4.8 Text Editor *vi*

4.9 Let Us Sum Up

4.10 Answers to Check Your Progress

4.11 Further Reading

4.12 Model Questions

4.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- identify the various types of linux commands
- describe the uses of directory handling commands
- identify and use the file handling commands
- describe the utility commands
- learn about the different modes of *vi* text editor.

4.2 INTRODUCTION

From the previous units we have come to know that Linux is GUT based os and controler computer. It controls both the hardware (things we can touch– such as keyboards, displays, and disk drives) and the software (application programs that we run, such as a word processor). Some computers have a single-user OS, which means only one person can use

the computer at a time.Many older OSes (like DOS) can also do only one job at a time. But almost any computer can do a lot more if it has a multiuser, multitasking operating system like UNIX or LINUX.

It comes with a huge number of powerful application programs. You can get many others for free on the Internet. (The GNU utilities, freely available from the free software foundation, are very popular.) You can thus do much more at a much lower cost. Not only are the applications often free, but some versions of LINUX itself are also free. Like the free applications, these free versions of LINUX are usually of excellent quality. They're maintained by volunteer programmers who want a powerful OS and are frustrated by the slow, bug-ridden development of OSes at big companies. In this unit we will concentrate on some basic LINUX commands and one text editor named ***vi*** editor.

4.3 LINUS COMMANDS

A ***command*** given to UNIX or LINUX is a series of characters that one type. These characters consist of words separated by white spaces. The first word is the command itself and the rest are the command's arguments. The arguments provide information that the command may need for its execution. UNIX or LINUX commands are case -sensitive. That is, *cp* is different from CP and Cp. It should remember that one must type all the commands in lowercase letters.

Commands are issued to the shell at the command line. A command line comprises commands, the line of instructions, options and any command-line arguments that one may provide. For example, the **$ man cp**.

Commands are entered at the shell prompt ($,#). Prompt is merely a symbol that appears at the start of a command line. This lets one know that LINUX is ready and waiting for your commands on the keyboard. For example,

$ mkdir assam. ↵

In UNIX or LINUX, a command line may have more than one command seperated by semicolons (;), pipes (|) or ampersands(&). For example **$ ls - l | more**.

4.3.1 Internal Command

Internal commands are such programs that are built-in into the operating system and reside in the memory along with the kernel.

That is, they get loaded in the memory along with the kernel at the time of booting. These commands will be available as long as the system is kept on. The commands such as *cp, mv, rmdir, ls* etc are all internal commands.

4.3.2 External Command

External commands are not built into the operating system, but loaded from other program files. These program files are available on a pendrive/hard disk and are loaded in the memory of the computer when specially asked for. That is, they must be externally called or introduced into the operating system for execution.

4.4 DIRECTORY HANDLING COMMANDS

The tree structure of the LINUX file system makes it easy to organize your files. After you make and edit some files, you may want to copy or move files from one directory to another, rename files to distinguish different versions of a file, or give several names to the same file. You may want to create new directories each time you start working on a different project.

A directory tree can get cluttered with old files you don't need. If you don't need a file or a directory,delete it to free storage space on the disk.The sections below explain how to make and remove directories and files.

Display Current Directory:

pwd: To display the current directory the command **pwd** (Present Working Directory) is used. Open a command line interface and type **pwd**.

The format is:

$pwd

Creating a Directory: Like other OS, Linux also provided the facility for creating directory. Under a directory we can keep so many diffrent types of files and sub-directory.

mkdir: To create a new directory, use the mkdir command. The format is:

$mkdir dirname ↵

where *dirname* is the name of the new directory. To make several directories, put a space between each directory name. To continue our example, you would enter:

$ mkdir spy boston dine ↵

To create sub directory under some existing directory, we proceed as follows.

$ mkdir /spy/assam ↵

In this case we create a new sub directory assam under an existing directory spy. To create a number of sub directories under a directory with one mkdir command, seperate the directory names with spaces. For example, the following commands:

$ mkdir sales/east sales/west sales/north sales/south ↵

To set the mode of newly created directory we use the following command:

$ mkdir -m dirname ↵

To make any missing parent directories we use the following command:

$ mkdir -p

will create four directories under the sales directory.

We may fail to create a ditectory in the fillowing cases:

a) The directory with the specified name already exists.
b) The user does not have authorization to create a directory.
c) There may exist an ordinary file by that specified name in the current directory.

Moving to a Different Directory (CD): You can change the working directory using the **cd** command followed by the pathname (it may be absolute or relative path) to which you want to move. For example, to move to the sales directory, type the following command:

$cd sales

Note that in the above command, we have made use of the relative pathname. After giving the above command, if you give the **pwd** (present working directory) command, you will see the result as *usr/mini/sates*

The cd command without any subsequent pathname always takes as user back to the user's HOME directory.

UNIX or LINUX uses a single dot (**.**) to represent the current working directory and double dots (**..**) to represent the parent directory of the current working directory. For example, typing **cd..** command will take user "mini" to the directory /user/mini, since that is the parent directory of sales.

Removing a Directory (mhdir): The command **rmdir** followed by the directory name is used for deleting a directory. The directory to be deleted must be empty for this command to work. For example, to remove the empty directory "assam" we type the following command.

$rmdir assam

We can also delete more than one directory by seperating the multiple directory names, with spaces as separators. For example, to remove "assa", ''bina", ''ram" directories we type the following command.

$ rmdir assa bina ram

We can also use the **rm -r** command to remove a directory and everything in it. But it can be dangerous for beginners. For example, to remove non empty directory "assam"

$rm -r assam

To remove the directory and ancestors we use the following command:

$ rmdir -p dirname

The command **rm *** is used to remove all files in that directory. For example, to remove all files from the directory "assam"

$ rm * assam

We may fail to remove a directory in the following cases:

1) The directory is not empty.
2) Our current working directory is not at least one level above the directory we have chosen to remove, hierachically.

3) We cannot remove a directory if we are placed in one of its sub-directories.

Listing the Contents of Directories (ls): To list the names of the files and the sub directories of the current directory, **ls** command is used. Its function is equivalent to the DIR command in MS-DOS. For example, to list the contents of current directory:

$ ls

the user will get the informations that is list of the files and the sub-directories under that.

The order of display would be numerals —> uppercase —> lowercase characters. That is, the list is displayed with numerlas having precedence over alphabet and in alphabet, uppercase letter have precedence over lower case letters.

A directory may contain several files, and the user may be interested in knowing whether a particular file is available or not. In this case, one has to use ls with the filename as follows.

$ls test.c

If file is found it displays; otherwise File Not Found error message will be displayed. The command *ls* can be used along with a number of options. Options are the predetermined arguments used along with a minus sign (-). The different options force the command to work differently. There should be a space between the command name and options. There should not be any white-space between the minus sign (-) and the option name.

For example, the following command will display file and directory names in multiple columns.

$ ls -x

Some other options along with its functions are given in table 4.1.

Table 4.1: Different option for is command

Options	Purpose
ls -x	get multi-column output.
-r	list files and sub-directories in reverse order
-t	list files and sub-directories with time order.

-a	list all files, including the normally hidden files
-c	list files by inode modification time.
-i	list the inode for each file.
-l	display permission, owner, size, modification, time etc. along with file and directory names.
-h	display names of hidden files
-q	display filenames having non-printable character.

pwd Command: To find which directory you're currently in, use the **pwd** (present working directory) command.

The **pwd** command takes no arguments. For example,

$pwd

/users/john

pwd prints the absolute pathname of your working directory.

Binary Directories: Binaries are files that contain compiled source codes or machine codes. **Binaries** can be executed on computer. Sometimes binaries are called **executables**.

The **/bin** directory contains **binaries** for use by all users. For example, to show all the listed files under bin directory following command is used.

$ ls /bin

Simillarly:

/sbin contains **binaries** to configure the operating system.

Also **binaries** found in **/ lib** (libaries) directory.

/sbin is used shared libaries located in **/ lib**

/boot directory contains all files needed to boot the computer.

/etc contains all the machine specific configuration files.

/dev contains all the device files that are not actually located on the hard disk.

4.5 FILE HANDLING COMMANDS

To handle files some important commands like **file, touch, rm, cp**, **mv** and **rename** are used.

Creating Files (using cat): To create a file, type the command **cat** at the shell prompt, followed by a > character and the filename. Its function is very simple like the command in DOS. For example, to create a file called test1 under the current directory, enter the following command.

$ cat > test1 ↵

After you press enter key, you will be prompted to enter the contents of the file. Type the required data and press ctrl+D keys to terminate the command line.

Transfering Contents from One File to Another File: We can also pass the contents of one file to another file. For example, if we wish that the contents of test1 should also be there in another file called test2, enter the following command:

$cat test1 > test2

If the file test2 already exists, it will get overwritten. If you want to append the data to the file and not overwrite it, use double right chevron as shown below:

$ cat test1 >> test2

You can also create a bigger file out of many smaller files using the cat command and the > operator.

The form:

$cat file1 file2 > newfile

Listing the Contents of File (using cat command): To list the contents of a file, we can use the command cat followed by the filename. A simillar command in DOS is the TYPE command.

For example, to list or display the contents of the file test1 we type the following command:

$ cat test1

Absolute path names can be given to show the contents of a file lying in another directory. The cat command can also display the contents of more than one file by separating the different filenames with spaces, as given in the following command:

$ cat test1 test2

The above command will display the contents of the two files in different rows.

Copying Files (cp): The command **cp** is used to copy a file into another. MS-DOS equivalent for cp is ths COPY command. Like in DOS, the cp command requires two arguments - a source filename, the contents of which are to be copied, and a target filename, to which the contents are to be copied.

For example, to copy the file test1 into another file test2, we type the following cp command:

$ cp test1 test2

If we want the system to ask for confirmation before copying the files we use the following command using cp with -i option.

$ cp - i test1 test2

We can copy more than one file at a time to a single directory by listing the pathname of each file we want copied, with the destination directory at the end of the command line. We can use relative or absolute path names as well as simple file names. For example, let's say your working directory was */users/carol* To copy three files called *ch1*, *ch2*, and *ch3* from */users/john* to a subdirectory called *work* (that's */users/carol/work*), by entering following command:

$ cp ../john/ch1 ../john/ch2 ../john/ch3 work

Deleting Files (Using command rm): To delete or remove a file, the *rm* command is used. This is equivalent to the DEL command in DOS. For example, to remove or delete the file test1, we type the following command:

$ rm test1

We can use *rm* with *-i* option, if we want the system to ask for confirmation, like in use of cp command. For example,

$ rm -i test1

To remove a directory, use the command rm with -r option. Here -r stands for recursive deletion. For example, the following command will remove the directory accounts

$ rm -r accounts

Unlike **rmdir, rm -r** will remove directories even if they are not empty. When we use wildcards (*) with **rm**, that is **rm*** be sure we are deleting the right files! If we accidentally remove a file we need, we can't recover it unless we have a copy in another directory or in the system backups.

$ rm *

CAUTION: Do not enter **rm *** carelessly. It deletes all the files in your working directory.

If we want to enter a command like **rm c*** (remove all file names starting with "c") but instead enter **rm c *** (remove the file named **c** and all files!). We may type following commands:

$ rm c*

$ rm c *

Again (**?**) character matches a single character. For example, to remove the files that the name contain five characters in their file name, the first three characters being "uni" we use the following command:

$ rm -x uni??

Moving Files (Using command mv): A file can be shifted or moved from one directory to another by using the **mv** command. Equivalent command in DOS is MOVE. The following command will be displace the directory north from "sales" directory to the directory "user/mini".

$ mv north /user/mini/north

The above command does not create a copy of "north" but just displaces it.

Directories can also be renamed using the mv command. For example, the following command will rename the directory "west" as "east".

$ mv west east

Files can also be renamed in the same way. If the file in the destination directory exists, it will be overwritten and if it does not exist, it will be created.

$ mv test1 test2

mv also has **-i** option like **cp** and **rm** commands and works in the same way.

Viewing Through pg, more, tail & head Commands: If the output of some commands like **cat** or **ls** is too long to fit in a single screen, the output will be scrolling off your screen and you will not be able to view the output. To overcome these problems we can use pg command. For example, the following commands will make us see the contents of file test1, one page at a time:

$ pg test1

more command allows us to view the output one screen at a time. For example, to view the output of **ls** command, operated on the file test1, one screen at a time, type the following command:

$ more test1

After each screen is displayed press enter key to display the next line or press the spacebar key to display the next screen full.

head Command: To display the top of a file, the **head** command is used. For example, the following command will list the first 5 lines of the file test1.

$ head - 5 test1

$head test1

When we use without option, the head command will display the first ten lines.

tail Command: Another command called **tail** displays the bottom of a file. The following command will display the last ten lines of the file test1.

$ tail test1

To display the last 4 lines of the file test1 we can type the following command:

$ tail - 4 test1

lp Command: If we have a long file, it may be best to print it so that we can see it all on paper. The command **lp** or **lpr** prints a file (on to paper as opposed to the screen display). Your system will probably have one or the other but not the both.

The syntax is: **$ lp** *option(s) filename(s)*

$ lpr *option(s) filename(s)*

Printers on UNIX systems are usually shared by a group of users. After you enter the command to print a file, the shell prompt returns to the screen and you can enter another command, however,seeing the prompt doesn't mean that your file has been printed. Your file has been added to the printer queue to be printed in turn.

The system administrator has probably set up a default printer at your site. To print a file named bills on the default printer, use the lp or lpr command, as in this example:

$ lp bills

We may have the following options that are recognized by the **lp** commands.

Options	Purposes
lp --	Marks the end of options
lp - E	Forces encryption when connecting to the server
lp - U	(username), Specifies the username to use when connecting to the server
- d	(Destination) prints files to the named printer
- h	(hostname), Choose an alternate server
- i	(job -id) Specifies an existing job to modify
- n	(copies) Set the number of copies to print 1 to 100.
- m	Sends an email when the job is completed
- s	don't report the resulting job ID (Silent mode)
- q	(priority), set the job priority from 1 to 100. The default priority 50.

4.6 UTILITY COMMANDS

date Command: An example of single-word command line is date. Entering the command date displays the current date and time:

$ date

Tue Nov 4 13:39:24 EST 1997

As you type a command in command line, the system simply collects your input from the keyboard. Pressing the [RETURN] key tells the shell that you have finished entering text and that it can start executing the command.

who Command: Another simple command is **who**. It lists each logged-on user's username, terminal number, and login time. The **who** command can also tell you who is logged in at the terminal you are using.

$ who

shyam tyy01 03:45

The next command is **who am i**. This command line consists of the command (who) and arguments (am i).

$ who am i

cactus! john tty23 Nov 6 08:26 (rose)

The response shown in this example says that:

1) "I am" John.
2) I'm logged on to the computer named "cactus."
3) I'm using terminal 23.
4) I logged in at 8:26 on the morning of November 6.
5) I started my login from another computer named "rose."

Remember all versions of **who am i** give the same information.

cal Command: *cal* is a handy tool that we can invoke any time to see the calendar of any specific month, or a complete year. If we wish to see the calendar for the month of December of the current year, we can provide this as applying following cal command:

$ cal dec

December 2008

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	2	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

With cal command we can produce the calendar for any month or year between the year 1 and 9999.

echo Command: The LINUX system also features a programming facility. We can assign a value to a variable at the prompt:

For example,

$ x=55

and then evaluate the value of this variable with the echo command as follows:

$echo $x

55

Again, we can also display any information in the screen using this echo command as follows.

$ echo "We are all Indian. We are proud to be an Indian**"**

We are all Indian. We are proud to be an Indian.

passwd Command to Change your Password: Security is not much of a concern in the DOS/ Windows environment(until NT), but that was because there was not really much these systems could do to establish a secure environment. However, UNIX or LINUX can, and if we really care about protecting our files, then make sure that no one else is able to use our account without our consent. If our account still does not have a password, or has one that is already known to others, you should change it immediately.

The **passwd** command is used to change passwords, and we use this command as follows:

$ passwd

UX:passwd: INFO: Changing password for local

Old password: ********

New password: *******

Re-enter new password: *******

passwd, when invoked by an ordinary user, asks for the old password, after which it demands the new password twice. When you enter here is not displayed on the screen for obvious reasons; If everything goes smoothly, the new password is registered by the system, and the $ prompt is returned.

4.7 FILE PERMISSIONS

Concept of File Permissions: UNIX or LINUX is a multiuser operating system, that is, more than one user can work on a LINUX system at the same time. Therefore LINUX provides a mechanism known as file permissions, which protect a user's files from tampering by other users. This mechanism lets files and directories be "owned" by a particular user.

For example, if John created the files in his home directory, then John owns those files and will get access to them.

UNIX or LINUX also lets files to be shared among groups of users. For example, if the user John desired, he could alow or deny access to his files by other users. He can also specify whether others can only read or read and write a particular file or directory.

Every file is owned by a particular user. The user who creates the file is by default the owner of the file, however ,files can also be owned by a particular group. This group is by default the group to which the user belongs.

The system administrator can grant file or directory access to more than one group of users.

Groups are usually defined by the type of users who access the machine. For example, in a university, user may be placed into group , such as student, staff, faculty or guest. There are also a few system-defined groups, such as **bin** and **admin**, which are used by the system itself to control access to resources. However, actual users do not belong to any of the system groups.

Types of file/directory Permission: File and directory permissions fall into three main divisions. These are:

a) read permission (r)
b) write permission (w)
c) execute permission (x)

Read permission lets a user can read the contents of the file, or in the case of directories, list out the contents of directory.

Write permission lets the user can modify or write data into a file. In the case of directories, the user can create new files or delete files within that directory.

Execute permission lets the user can run the file as a program or shell script (if the file is a program or shell - script) for directories, execute permission lets the user change the directory in question.

These permission may be granted to the three classes of users namely, the owner of the files, the group to which the files belong and to all other users. Each class may have different permissions. For example, the owner may have read, write and execute permission on a file, the group may have only read permission and others may not have any permission at all.

Interpreting File Permission: The **ls** command with the **-l** option displays a "long" listing of the file attributes.

```
$ ls -l
total 20
drwxrwxr--  2 john student   4096  Nov 31  01:45  asam
drwxrwxr-x  3 satis facultyt 1096  Nov 31  01:50  ghy
-rwxr-xr-x  1 raj student    2096  Nov 31  02:15  test1
drwxr-xr--  3 rasmi faculty  96    Nov 31  02:45  west
-rw-rw-r--  4 jon student    496   Nov 31  03:45  test2
$
```

Here, in the above example, we see that "total 20". This number represents the number of blocks which are occupied by these files.The other attributes are discussed in that order in the following:

File type: The first character in the listing represents the file type, that is whether the file is a regular file or directory. A '-' sign indicates regular file and 'd' represents directory. For example, test1 and test2 are regular files and asam, ghy and west are directories.

File Permission: The second character represent file permissions. The string "rwxrwxr--" lists, in order, the permission granted to the file's owner, group and others. The first three characters "rwx" represent the permissions granted to the file's owner, john. Character

"r" stands for "read", "w" stands for "write"and "x" stands for "execute". Thus, john has read,write and execute permission to the directory asam. The next three character ("rwx") represent the group's permissions on the file. The last three characters (r--), represent the permission granted to every other user on the system. Here other users can only read the directory, since only an "r" appears.

Links: The third field indicates the number of links that are associated with the file. This is nothing but the number of names(aliases) the file has.

Ownership: The fourth field shows the owner of the file. The fifth field shows the group to which the file belongs.

File size: The sixth field specifies the size of the file in bytes.

File- Access Time: The next three fields indicates the time the file or directory was last modified.

Filename: The last field shows the name of the file or directory.

Modification means changing the file contents. Change of attributes are not considered as file modification.

Permission Dependencies Directory Permission: The permission granted to a file also depends on the permission of the directory in which the the file is located. For example, even if a file is set to *rwxrwxrwx*, other users cannot access the file nuless they have read and execute access to the directory in which the file is located. Simillarly to access a file one must also have access-rights to the directory in which the file is located. For example, if john wants to restrict access to all of his files, he should set the permission to his home directory /home/john to *rwx*------. In this way, no other user has access to his directory, and all files and sub-directories within it. He does not hane to worry about the individual permission on each of his files.

In other words, to access a file we must have execute access to all the directories along the file's pathname, and read (or write or execute) access to the file itself.

Changing Permission: Typically, users on a LINUX system are open with files. The usual set of permissions given to a file is rw-r--

r--, which lets other users read the file but not change it in any way. The usual set of permission given to directories is rwxr-xr-x, which lets other users look through your directories, but not create or delete files within them.

However, many users wish to keep other users out of their files. Setting the permissions of a file to rw-------- will prevent any other user from accessing the file. Likewise, setting the permissions of a directory to rwx------ will keep other users out of the directory in question.

In order to change these default permissions and set your own, **chmod** command is used. Only the owner of a file can change the permission on that file. That is, only the user who owns the file can execute the chmod command. The syntax of chmod is:

$ chmod category operation permission filename

Here the argument category specifies the user's cases. The argument operation specifies whether the permission is to be granted or denied. The argument permission specifies the type of permission. The different values for these arguments are listed in table 4.2:

Table. 4.2: Values of Chmod Argument

Category	Operation	Permission
u (owner)	+ (assign)	r (read)
g (group)	– (deny)	w (write)
o (other)	= (absolute assignment)	x (execute)
a (all)		

In the first argument we supply one or more of a (all), u (owner), g (group) and o(other). Then we specify whether we are adding right (+) or taking them away (-). Finally, we specify one or more r (read), w (write) and x (execute) permissions.

For eaxmple, to give all users execute access to the file test2, type the following command:

$ chmod a+x test2

If none of a,u,g, or o is specified, then a is assumed.

For example, the above given command is equivalent to:

$ chmod +x text2

To remove execute permission from users other than the owner, the command will be.

$ chmod og-x test2

To remove read, write and execute permission from users other than the owner and users in the files group, the command will be as follows:

$ chmod o-rwx test2

The assignemnt operator = is used when we want to perform absolute assignment of permissions. Unlike + and - operator that adds or memov < the specified permission, the = operator assigns the specified permission and removes other permissions.

For example, if a file test1 had the permission -rwxr-xr-x, the command:

$ chmod ugo=r test1

will assign read permission to the owner, group and others and revokes the other permissions. That is, now the file will have the permission r--r--r--.

Octal Notation: Instead of the character arguments, one can give numeric arguments to the *chmod* command. The notation takes the form of an octal representation of the permissions, and it is assigned like–

read permission	=	4
write permission	=	2
execute permission	=	1

When more than one permission is associated with a particular user class, the respective numbers are added.

For example, if a directory has the read and execute permissions for the owner, the octal representation for the owner's permissions will be 4+1=5. This exercise is repeated for the other categories (group and others).The result is a three-digit octal number, with each octal digit describing the permission for each category. That is, the first digit represents the permissions for the user, the second digit for the group and third digit for others.

Note: Permission assignement by octal notation is absolute like the = operator.

For example, instead of the command:

$ chmod a+x test1

We can also give the command as follows:

$ chmod 111 test1

'1' indicates execute permission. Similarly, instead of the command:

$ chmod ugo+rw test1

We can give the command:

$ chmod 666 test1

6 indicates read and write permission (4+2)

To assign all permissions to the owner, read and write permission to the group; and only executable permission to others ,we can use either of the following two commands:

$ chmod u=rwx, g=rw, o=x test1

or

$ chmod 761 test1

Note: 777 signifies all permission and 000 indicates absence of all permissions.

Ownership: When a file is created, the creator becomes the owner of the file and the group to which the owner belongs becomes the group-owner. Thus, if user "john" created a file report, he becomes the owner of the file, and john's group ("student") becomes the group-owner.

However, when one copies a file owned by someone else, the ownership of the copy is transferred to that and can manipulate the attributes of the copy at will.

Changing Ownership: There are two commands that can be used to change the ownership of a file or directory. These are:

a) chown

b) chgrp

chown (change ownership): The command chown is used to grant the ownership of a file to another user. For example, to grant thw ownership of a file test1 to john, the user raj has to use chownin the following way:

$ chown john test2

Note: Ownership once surrendered, cannot be reestablished.

Therefore, the command:

$ chown raj test2

will give an error, since the owner of the file is now john and raj cananot run the chown, chgrp or chmod commands from now on.

chgrp Command (change group): The command chgrp changes the group ownership of the file. This command, like chown, can only be used by the owner of the file, irrespective of whether the user belongs to the same group or not.

Note: By default, the group -owner of a file is also the group to which the owner belongs.

4.8 TEXT EDITOR vi

The **vi** editor is a screen-based editor and it is the most powerful text editor available in almost all UNIX and LINUX systems. The **vi** editor has many powerful features to aid programmers. Working with **vi** editor is entirely different from that of any modern simplified editor that you are familiar with. This editor has remarkable editing capabilities. It is rather a very simple editor, if all the elementary **vi** commands are at your finger tips.

LINUX features a number of vi editors, of which **vim** (vi improved) is the most common. Apart from vim, there are **xvi**, **nvi** and **elvis** which have certain exclusive functios not found in the Berkeley version.

Three Modes: The three modes of vi editor are:

a) Command Mode
b) Input Mode
c) Ex Mode

Command Mode: Command mode is the mode where you will be giving commands to the vi editor. When you invoke vi, this is the mode where you will be placed initially.

Input Mode: When you give the commands i, I, a , A, R, s, S, or O then you get shifted to the input mode of vi editor. It is also known as insert mode or editing mode. In this mode you can add text to the

file. For example, when you want to insert text at the current cursor position, give i in the command mode. you will now get placed in the input mode and you can perform the intended mode. For example, if you type 'a' here it will not be interpreted as an 'append' command, but as character 'a' to be inserted in the file.To move back to command mode press Esc key.

Ex Mode: Ex mode of vi editor enables us to give commands at the command line. To move to the ex mode, press ':'(colon) if we are in command mode. If we are in the input mode, first press Esc key and then press ':' to move to the ex mode. Cursor moves to the bottom of the vi screen and we can provide the ex mode commands here. Some impotant and common ex mode commands are given below:

Command	Used to:
:w	save the changes we have made to the file
:q	quit vi without saving changes
:wq	save and quit
:q!	force quit if'q' does nor work

Ex mode is named after the ex line-editor of UNIX or LINUX. The commands we give the ex mode are actually the ex line-editor's commands.

Invoking vi Editor:

$ vi test1: The file 'test1' will opened in vi editor. If the file does not exist, vi will create a new file with that name. In either case we will see a screen full of the tilde'~' which indicates empty lines.

Remember that vi editor opens in the command mode. That means we are now in the command mode.

Let us start by inserting some text in our file "test1". For that we will have to switch to the 'input mode' . We can see that to insert text 'i' command is used. Press 'i' and type the following text.

Saving Text and Quitting *vi* Editor: Having inserted the text, we now need to save these changes. To save the changes, first press Esc key to move to command mode. Now press ':' to move to ex mode. Note that now the cursor has moved to the end of the screen.

Type ‘w’ to save changes. If we want to quit vi, type ‘q’. We can also save and quit vi by giving the command ‘wq’.

Note: We can also save the file in the command mode by presenting Z (uppercase) twice, i.e ZZ.

When we quit vi, a note at the bottom of the screen indicates the name, number of lines and number of character in the file.

Editing File in *vi*: Invoke again the vi editor, by typing vi test1 at the command prompt. We will see the text we have typed followed by multiple empty lines, each starting with a ‘~’ sign. The cursor appears at the beginning of the first line of the file. Let us first learn how we can move this cursor to a particular line or character.

Adding Text: Let us now append some text to the file. For this, first move cursor to the last line of the file. Pressing ‘G’(uppercase) will move the cursor to the beginning of the last line in the file. Additionally, you can precede ‘G’ with the line number where we want the cursor to move. For example, to move to 6th line then we type ‘6G’.

We can move to the beginning of a file by pressing ‘1G’. Now press ‘o’ to open a new line below the last line. By pressing ‘o’ we are in the ‘input mode’.

Inserting Text in The Middle: If we missed any character(s) while typing, then move the cursor to that particular line or word. Press Esc key to move to the ‘command mode’. Now press ‘i’ to insert the character(s) before the current cursor position or press ‘a’ to insert the character(s) after the current cursor position. ‘A’ will append text at the end of current line.

Replacing Text: Text character can be replaced with the ‘r’, ‘R’, ‘s’ and ‘S’ commands. Before we use these commands switch to the command mode, if we are currently in the insert mode.

First, move the cursor to the character which we want to replace. Pressing ‘r’ followed by a character replaces the existing character with the character given after ‘r’. To replace more than one character use ‘R’ followed by the characters. The existing text will get overwritten with these characters. The replacment of text with ‘R’ command is limited to the current line only.

To replace a character with multiple characters without overwriting the subsequent text, use ‘s’ followed by the characters.

‘S’ (uppercase) replaces the entire line irrespective of the cursor position. That menas, it deletes the current line completely and makes it ready for a new line of text.

Pressing ‘R’, ‘s’ and ‘S’ shifts you to the ‘input mode’. Press Esc key to move back to the command mode.

Deleting Text: To delete a character, first move the cursor to that particular character position. Press ‘x’ to delete a single character. We can also precede ‘x’ with the number of characters we want to delete. For example, ‘4x’ will delete four characters. Note that the rest of the line gets adjusted automatically.

If we go on deleting all the characters in the current line, the characters to the left of the current line start getting deleted, instead of that in the next line.

Note: To delete the current line press ‘dd’ . To delete more than one line use ‘#dd’, where # represents the number of lines we want to delete.

‘X’ (uppercase) is used to delete a character to the left of current cursor position. To understand it better, it can be said that ‘x’ functions like the delete key and ‘X’ funtions like the Backspace key.

Undoing and Repeating Commands: The repeat command (a dot(.)) can be used to repeat the most recent command. For example, if we have recently given the ‘a’ command and appended one character, we can repeat the whole process again by pressing a . (dot). We will have the character inserted two times.

The undo (u) command can be used to recover from the most recently committed mistake. For example , press u and we will see that the second character we have inserted using (.) disappears.

Let us now save our file and quit vi editor, Rememeber to press Esc key to switch to command mode if you are not in it already and then use :wq to save the file and quit vi.



CHECK YOUR PROGRESS

Q.1: Hidden files in LINUX can be listed using–

a) ls -t b) ls -h c) ls -a d) none of the above.

Q.2: In LINUX, the command to delete all files started with sg and ending with hj is–

a) rm sg*hj b) rm sg?hj

c) rm -prefix sg -suffix hj c) none of the above.

Q.3: The LINUX command **rm -r project** will–

a) delete the file project from the current directory.

b) delete all the files from the directory project and all its subdirectories.

c) recursively delete the directory project and all its subdirectories.

d) None of the above.

Q.4: UNIX can have–

a) one file system only.

b) n file systems, if it had n hard disks.

c) any number of file systems depending upon the number of partitions.

d) none of the above.

Q.5: The UNIX command “ls | more” displays a list of–

a) files in the current directory one line at a time.

b) files in the current directory one screen time

c) all the files in the current directory and then wait for the next command

d) none of the above.

Q.6: The UNIX, full pathname originates from–

a) your home directory b) the root directory

c) your working directory d) none of the above.

Q.7: The chmod command in UNIX–

a) change the current execution status from user mode to kernel mode.

b) makes the file hidden so that it cannot be seen using the ls command.

c) changes the access permissions of a file or directory.

d) none of the above.

Q.8: Which out of the following commands would you use to change your password in UNIX system–

a) pass b) passwd c) chpass d) setpass

Q.9: The execute permission to a file allows a user to

a) execute the file b) edit the file

c) list the contents of the file d) none of the above.

Q.10: The chown command in UNIX changes the–

a) home directory of a user

b) access permissions of a file or directory

c) priority of a process.

d) none of the above.

Q.11: Find out the TRUE and FALSE statements:

i) The command pwd shows the name of the current directory.

ii) UNIX makes a distinction between the text files and the binary files.

iii) The command mv file1 file2 renames file1 as file2.

iv) The more command appends new lines to a file.

v) A binary file is also known as an ASCII file.

vi) The command rm * removes all files from the current directory.

vii) A file having protection 466 cannot be used by anybody other than the owner.

viii) Only the super user can view the password of a user in the file /etc/passwd file.

ix) chmod u-x gives the user permission for execution.

x) ls command is used to list the files and directories.



4.9 LET US SUM UP

- **Linux Command:** A command given to UNIX or LINUX is a series of characters that one type. These characters consist of words separated by white spaces. The first word is the command itself and the rest are the command's arguments. The arguments provide information that the command may need for its execution.
- **Internal Commands:** Internal commands are such programs that are built-in into the operating system and reside in the memory along with the kernel.
- **External Commands:** External commands are not built in into the operating system, but loaded from other program files.
- To create a new directory, we use the mkdir command.
- You can change the working directory using the **cd** command
- The command **rmdir** followed by the directory name which is to be removed is used for deleting a directory.
- To list the names of the files and the sub -directories of the current directory, **ls** command is used. its function is equivalent to the **DIR** command in MS-DOS.
- To find which directory you're currently in, use the **pwd** (print work ing directory) command.
- To create a file, type the command **cat** at the shell prompt, followed by a > character and the filename.
- A file can be shifted or moved from one directory to another by using the **mv** command.
- The command **cp** is used to copy a file into another. MS-DOS equivalent for cp is ths COPY command.
- **more** command allows us to view the output one screen at a time.
- The **who** command can also tell you who is logged in at the terminl you're using.
- The **passwd** command is used to change passwords

- File and directory permissions fall into three main divisions. These are read permission (r), write permission (w), execute permission (x).
- In order to change these default permissions and set your own, **chmod** command is used.
- There are two commands that can be used to change the ownership of a file or directory. These are:
  a) chown
  b) chgrp



4.10 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: (c)
Ans. to Q. No. 2: (a)
Ans. to Q. No. 3: (c)
Ans. to Q. No. 4: (c)
Ans. to Q. No. 5: (b)
Ans. to Q. No. 6: (b)
Ans. to Q. No. 7: (c)
Ans. to Q. No. 8: (b)
Ans. to Q. No. 9: (a)
Ans. to Q. No. 10: (d)
Ans. to Q. No. 11: i) (T), ii) (F), iii) (T), iv) (F), v) (F), vi) (T), vii) (F), viii) (F), ix) (F), x) (T).



4.11 FURTHER READING

1) Sumitabha, D. (1998): *Unix: Concept and Applications*; Tata McGraw-Hill.
2) Tanenbaum, A. S. & Woodhull, A. S. (1987): O*perating Systems: Design and Implementation*; Prentice-Hall.



4.12 MODEL QUESTIONS

Q.1: What is the difference between **cp** and **mv** commands? Explain with an example.

Q.2: Distinguish between:

a) **cat** and **more** commands

b) **tail** and **head** command

Q.3: List out the different options and significance of **ls**.

Q.4: A file has got a protection 756 (octal). What protections does it really have?

Q.5: Explain how you can implement the cut, copy and paste operations in vi.

Q.6: Give the syntax for opening a file in vi editor. Also explain the different options.

Q.7: How can you access ex commands in vi?

*** ***** ***

UNIT 5: MS WINDOWS OPERATING SYSTEM

UNIT STRUCTURE

5.1 Learning Objectives

5.2 Introduction

5.3 Windows Operating System

5.4 Components of Windows

- 5.4.1 The Desktop
- 5.4.2 Icons
- 5.4.3 Taskbar
- 5.4.4 My Computer

5.5 Working with Files and Folders

- 5.5.1 Creating Files and Folders
- 5.5.2 Naming Files and Folders
- 5.5.3 Renaming Files and Folders
- 5.5.4 Deleting Files and Folders

5.6 Cut, Copy, Paste

5.7 Let Us Sum Up

5.8 Answers To Check Your Progress

5.9 Further Reading

5.10 Model Questions

5.1 LEARNING OBJECTIVES

After going through this unit, you will be able to

- learn about Windows Operating System
- learn about Desktop and its various parts
- manage My Computer and its various parts
- learn about how to work with files and folders by creating, deleting, moving, renaming, and copying them.

5.2 INTRODUCTION

This unit provides an outlook on Microsoft Windows operating system. In this unit, the learner will learn how to work with Windows operating system

using its various in-built features. The discussion begins with the basic components of Windows operating system. Similar to the way we keep files or documents in file folders for easy reference at office or home, computers also contain folders which hold different documents or files. The concept of computerised file and folders are also included in this unit.

5.3 WINDOWS OPERATING SYSTEM

Microsoft Windows is a series of operating systems and Graphical User Interfaces (GUI) produced by Microsoft. Microsoft first introduced an operating environment named *Windows* in November 1985 as an add-on to MS-DOS in response to the growing interest in graphical user interfaces. Microsoft Windows came to dominate the world's personal computer market. Windows XP is an operating system introduced in 2001 by Microsoft and is a part of the Windows family of operating systems. This new operating system provides a new interface that uses a more stable and reliable environment than the previous versions of Windows like MS Windows 98, MS Windows 2000 etc.

One of the biggest features of *Microsoft Windows XP* is its multi-user environment. Everyone who uses the computer must *login*, and each unique user has their own "user" folder where application preferences and documents are stored.

5.4 COMPONENTS OF WINDOWS

5.4.1 The Desktop

When Microsoft Windows is opened, the Desktop appears as shown in (Figure 5.1). It is the first screen visible after logging in Windows Operating system. It is a part of a **Graphical User Interface** (GUI) where icons, menus and dialogue boxes on the screen represent programs, files, and options on computer. We can move file and folders to the desktop for easy access.

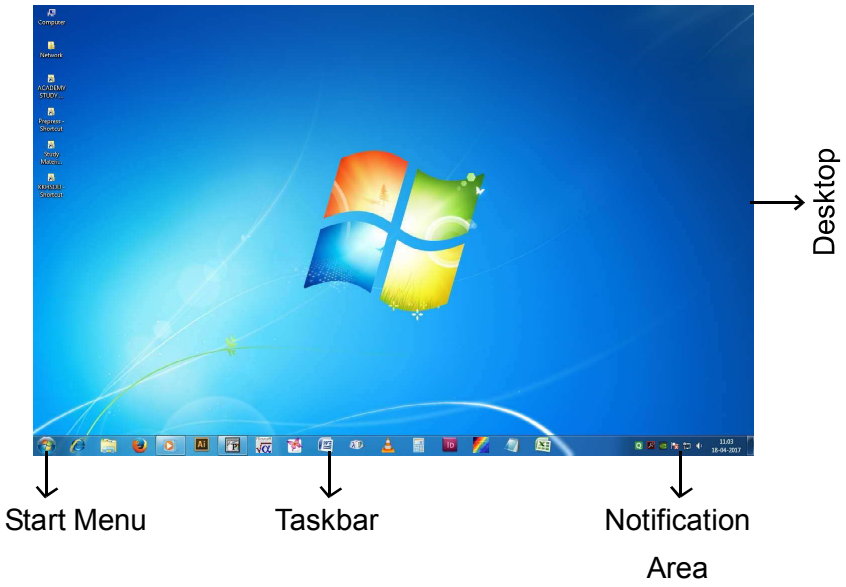


Fig. 5.1: The Desktop

- **Creating desktop shortcuts:** Desktop shortcuts are great for quick, easy access to our frequently used programs. Suppose we are using *MS word* very frequently. We can create a MS Word shortcut clicking the following:

Start > ***Programs*** > Right click on ***Microsoft Word***. Then a menu will appear. Click on ***Send To*** > ***Desktop(Create shortcut)***

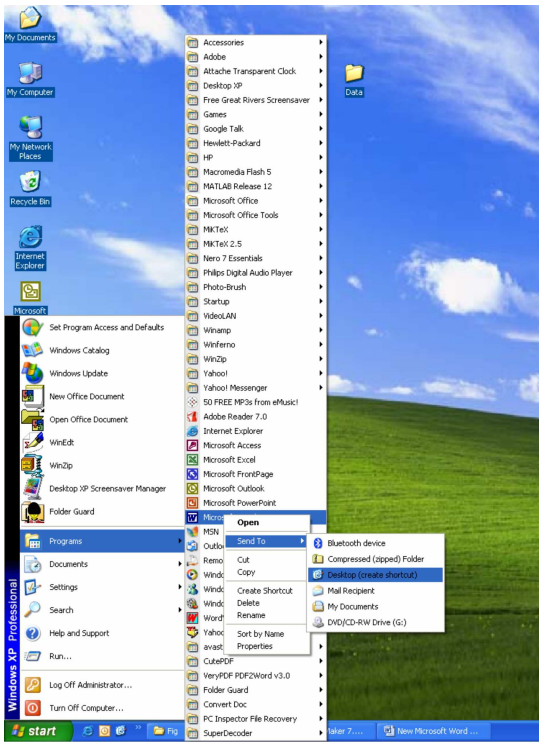


Fig. 5.2: Creation of Desktop Shortcut

- **Customizing the Desktop:** We can customise Windows desktop according to our preference. To customise desktop, we have to right-click anywhere on the desktop and select ***Properties*** from the shortcut menu. This displays ***Display Properties*** dialog box as shown in Figure 5.3. This dialogue box contains five tabs at the top of the window. These are: *Themes, Desktop, Screen Saver, Appearance, and Settings.*
- **Themes:** These are collections of sounds, icons, fonts, colours, and screen savers, which often come with their own background graphfics, based on a specific theme. We can select from a range of in-built themes from the **Theme** drop-down box.



Fig. 5.3: The *Display* Properties Dialog Box

- **Appearance:** This tab allows us to change the windows style, colour scheme and font size. We can choose either *Windows XP style* or *Windows Classic style* from the *Windows* and *Buttons* drop-down box. Each of these style has predefined colour schemes, which can be selected from the *Color* scheme drop-down box. These schemes are used to provide colour to all the working objects like icons, title bar etc. The *Font size* drop-down box is used to select the size of the font in the folders, icons, title bar and so on.

- **Settings:** This tab allows us to change the number of colours or display resolution from the *Color quality* and *screen resolution* section respectively. We can increase the screen resolution from 800 x 600 to 1024 x 768 pixels.

5.4.2 Icons

Icons are small graphical images that can represent a computer's programs, files, folders and printers amongst other things as shone in (Figure 5.4). To activate the program/file/folder that an icon represents we simply double click on it with the left mouse button. This will activate the icon and either start a program or open a file/folder. The icons on our computer desktop can be renamed by right clicking on them and selecting ***rename***. They can be deleted by right clicking and selecting ***delete***. It is possible to create our own Windows desktop icons for programs, files, folders, etc.

To organise desktop icons, we can right click anywhere on the desktop and then select ***Arrange Icons By*** option to arrange the icons on the desktop according to name, size, or type. The main icons that appear by default on the Desktop are listed below:

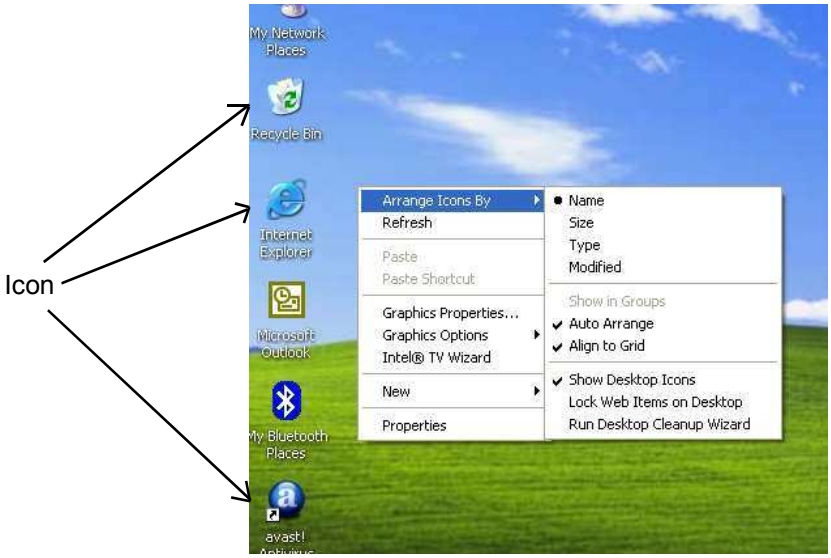


Fig. 5.4: Icons on the Desktop

- **My Computer:** When we open this icon, it displays documents and files which are already stored there.

- **My Network Places:** It enables us to communicate with network resources on the network for file shearing purpose.
- **My Documents:** It is used to store documents. It also contains data files, which can be easily organised.
- **Recycle Bin:** When we delete a file, Windows XP will place the file into the recycle bin (instead of deleting it altogether). This allows us to **restore** the file in case we delete it by mistake. For example, if we delete a file by accident we could double click on the recycle bin icon to see its contents. Inside we will see the file/folder that we deleted. To restore the file back to where we deleted it from, we have to right click on it and then have to select **restore** from the menu that appears.
- **Internat Explorer** : It is a ***web browser*** developed by Microsoft and included as part of the Microsoft Windows operating systems. It provides access to ***Internet***.

Web Browser: A program used to view HTML documents.

Internet: A computer network consisting of a worldwide network of computer networks to facilitate data transmission and exchange.

Default: An option that is selected automatically unless an alternative is specified.

5.4.3 Taskbar

The taskbar as in Figure 5.5 is another important part of the Windows XP operating system. By ***default***, the taskbar is placed at the bottom of the desktop screeen. The taskbar can be moved to any of the four sides of the desktop and can be resized to display more buttons. One of its main uses is to switch between any open programs or documents. Taskbar includes *Start Button*, *Quick Launch Toolbar*, and *System Tray*.



Fig. 5.5: Taskbar

- **Start Button:** The Start button is used to access the Windows XP start menu.The start button (Figure 3.6) is a very important part of Windows XP. Clicking on the start button opens up what is called the start menu, the start menu is used to access our programs, settings, printers and more.



Fig. 5.6: Start Button

- **Quick Launch Toolbar:** The Taskbar also displays its own set of toolbars; the most frequently used is the Quick Launch toolbar. It is a customisable toolbar that displays single click links to programs and commands. By default, the Quick Launch toolbar displays links to Internet Explorer, Media Player, and Show Desktop.
- **System Tray:** The system tray displays icons for programs that are loaded into memory, although not all programs place an icon in this tray.
- **Window Tab:** The Windows tabs are used to switch between any open windows (programs, folders, documents, etc). We can switch between the windows by simpy clicking on the corresponding window tab, clicking a second time on the same tab will minimise the window.
- **The Win XP Clock:** The clock sits on the taskbar and displays the system time. Hovering the mouse pointer over the clock will reveal the date, to learn how to change the time and date.
- **Turn off Computer:** If we click on the ***Turn Off Computer*** option on the start menu, we will see a dialogue box. It has three options:
  - **Standby:** This will put our computer into standby mode. Windows XP will power down components like the monitor and hard drive to preserve energy.
  - **Turn Off:** Shuts down the PC. This option to turn off our PC.
  - **Restart:** This will restart the computer.

5.4.4 My Computer

My Computer provides access to hard drives and removable drives of the computer.



Fig. 5.7: My Computer Icon

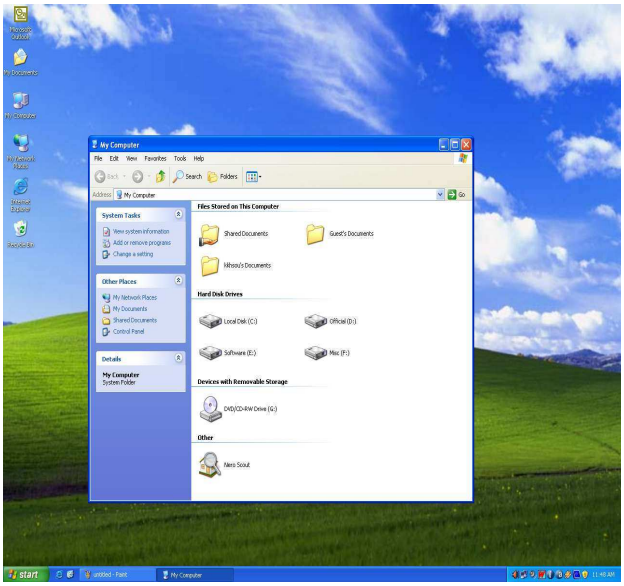


Fig. 5.8: Icon inside My Computer

- **Title bar:** The title bar contains the name of the program we are working with and in some cases the name of the opened document also appears. In the top right corner of the title bar we find the minimize, maximize/restore, and close buttons.



Fig. 5.9: The Title bar of My Computer

We can use the **minimize** button to temporarily decrease the size of a window or remove a window from view. While a window is minimized, its title appears on the taskbar.

The **maximize** amplifies the size of the window to the whole screen.

After we minimize a window, if we click the **restore** button, the window will return to its former size.

The **close** button closes the window. If we have modified the document, we are asked if we want to save the changes before closing.

Maxi

Minimize  Close

Fig. 5.10: The Buttons in the title bar

- **Menu bar:** The menu bar has the program functions organized in menus. For example, when we click on *File*, the different

operations that can be performed with files are displayed. All the operations that the program allows us to do are located within the different menus of this bar.



Fig. 5.11: The Menu bar

To open a menu with the keyboard, we need to press the *Alt* key plus the underlined letter of the menu we want to open. For example Alt+F opens the menu *File*.

If on a menu, an option appears with a *small arrow pointing to the right* it means that option contains other options. If we want to open the menu related to that option, we only need to maintain the option highlighted for a few seconds and the menu will appear.

If we find an option that is *grey* or *lighter colored* it means that it is not possible to select it at that moment.

- **Toolbar:** The toolbar contains icons to immediately execute some of the most often used commands. These commands can also be executed from the menu bar.

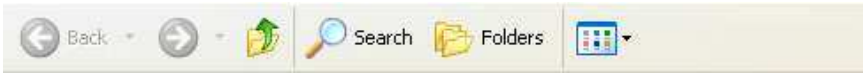


Fig. 5.12: The Toolbar

- **Scroll bar:** The scroll bars allow us to move throughout the page in a quick and simple manner. We can simply drag the bar with the mouse or click on the arrows. There are bars to move along the width of the page and bars to move along the height of the page.



Fig. 5.13: The Scroll bar

- **Status bar:** The status bar is used to give the user certain information.This information varies depending on the window that is opened at the time. In this case, it tells us that we have 11.3 GB free space on the disk, and that the total size of the disk is 19.5 GB.This bar is not shown by default in Windows XP. We can activate or disactivate it from the menu *View* > *Status Bar.*



Fig. 5.14: The Status bar



CHECK YOUR PROGRESS

Q.1: Fill in the blanks:

i) Screen saver is used to prevent damage to ..... element in the monitor.

ii) By default, the taskbar in the Windows XP is located on the ..... of the desktop.

iii) ..... button returns the window to its original size as it was before the maximized state.

iv) Full form of GUI is .....

Q.2: State whether the following statements are true (T) or false (F).

i) The settings of Windows XP once set cannot be changed.

ii) Status bar contains the Maximize, Restore and Closed buttons.

iii) When we delete a file, Windows XP will place the file into the recycle bin.

iv) Files and folders can be restored from the recycle bin.

5.5 WORKING WITH FILES AND FOLDERS

A ***file*** is defined as an organised and structured collection of related information stored in the computer's memory. The information can be of different types such as text, data or images.

Folders provide a very convenient and useful method of storing files and subfolders in an organized way on a computer's hard drive or on a removable drive. We can also create folders inside a folder. Figure 5.15 shows three folders(namely Software,Official Data and Records) and two files(flower.jpg and biodata.doc) inside the D:drive.

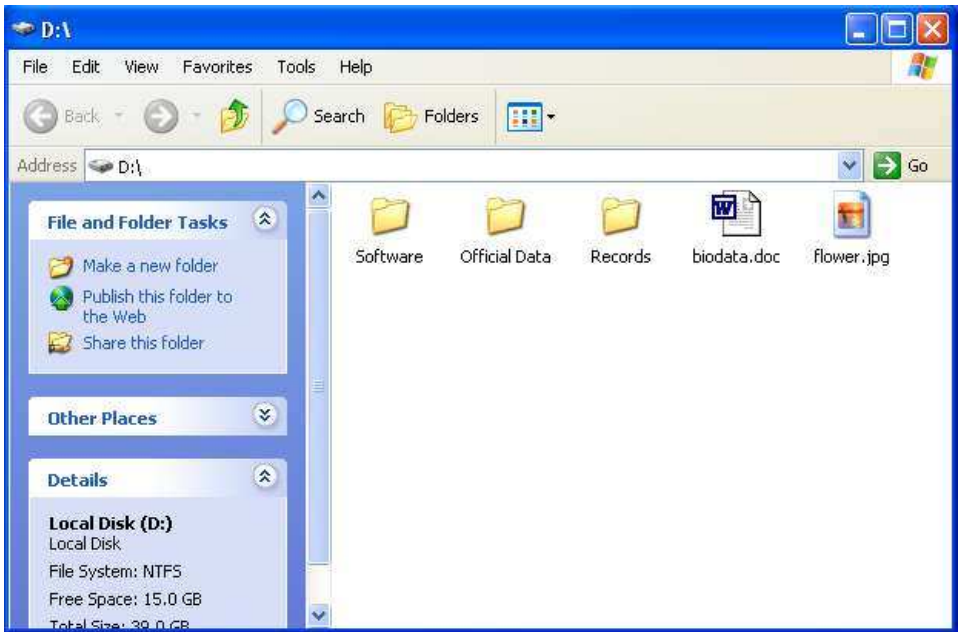


Fig. 5.15: Folders and Files inside D:drive

5.5.1 Creating Files and Folders

Folders can be created on desktop, in My documents or at any other specified location. To create a new folder, say, on desktop we can *right click* on the desktop and select ***New*** from the shortcut menu.This displays another pop-up menu, from where we have to select the ***Folder*** option and type the name for the new folder (Figure 5.16). Files can also be created in the similar manner.

Another way to create a New Folder is as follows:

1) Open ***My Computer***.
2) Click on the drive or folder in which you want to create a new folder.
3) On the ***File*** menu, point to ***New***, and then click ***Folder***
4) Type the name for the new folder, and then press ***Enter*** key.



Fig. 5.16: Creation of New Folder on Desktop

5.5.2 Naming Files and Folders

Names for *folders* can be up to 255 characters in length. A descriptive short name is usually a better choice. We may use spaces and underscores in folder names, but we cannot use the following characters: * : < > | ? " \ or /.

File names can contain upto 255 characters and can contain numbers, letters, spaces and some symbols. Most files have an extension, in addition to the name. The extension is assigned by the program that creates the file. For example, Microsoft Word creates a file wth ***.doc*** extension. In general, image file have the following extentions: ***.jpg***, ***.gif***, ***.bmp*** etc.

5.5.3 Renaming Files and Folders

Sometimes it may be required that a file or folder should be renamed. To rename a file or a folder, we have to first select the file or folder and then have to follow any of the following steps :

- Right click on the file or folder name and the select ***Rename*** option from the shortcut menu. Enter the new name and press ***Enter*** key.

- After selecting, click the file name or folder name to highlight it, then enter the new name and press ***Enter*** key.

5.5.4 Deleting File and Folders

If we want to delete a particular file or folder, we have to first select the file or folder and then have to follow any of the following steps:

- Right ckick on the file or folder and select the ***Delete*** option from the shortcut menu.
- Press ***Delete*** key from the keyboard.
- Drag the file or folder into the *recycle bin*.

When we perform any of the above steps, Windows XP displays a message on the screen asking us to confirm deletion of file or folder. To delete the file we have select **Yes**, or **No** to keep the file.

If we want to permanently delete a file or folder, first we have to select the file or folder and then press ***Shift + Delete*** key from the keyboard.



Clipboard: The Clipboard is the storage area for items that have been cut or copied. Each time we execute Cut or Copy, we replace the old information on the Clipboard with whatever we just cut or copied. We can paste Clipboard information as often as we like, until we replace it with something else.

5.6 CUT, COPY, PASTE

The Cut, Copy, and Paste commands are nearly universal. These three functions are used by almost every Windows program and perform more or less the same function in each of them. We can cut, copy, and paste files,folders, programs, text etc.

- **Cut:** When we cut something, it is deleted from its current location and saved to the ***clipboard***. Information saved to the clipboard stays there until new information is either cut or copied. Each time we execute cut or copy, we replace the old information on the clipboard with whatever we have just cut or copied. We can paste the same clipboard information as often as we like.
- **Copy:** *Copy* is similar to *cut* except that the original item is not deleted. When we copy something, a copy of the item is saved to

the clipboard. Information stored on the clipboard stays there until new information is either cut or copied. Each time we execute cut or copy, we replace the old information on the clipboard with whatever we just cut or copied. We can paste clipboard information as often as we like, until we replace it with something else.

- **Paste:** We can place information on the clipboard wherever we like. Execute the paste command and the information we have cut or copied is placed wherever cursor is located.

There are three major methods of *cutting*, *copying*, and *pasting*. The three methods are: *using the menu*, *using keyboard shortcuts* and *using icons*. We will review all of them. In most programs, they will work exactly as described below:

- **Using the Menu:**

  Cut:

  - Select what you want to cut.
  - Click *Edit*, which is located on the *menu bar*. A drop-down menu will appear.
  - Click *cut*.

  Paste:

  - Place the cursor at the point where you want to place the information that is currently on the clipboard.
  - Click *Edit*. A drop-down menu will appear.
  - Click *Paste*.

  Copy:

  - Select what you want to copy.
  - Click *Edit*, which is located on the *menu bar*. A drop-down menu will appear.
  - Click *Copy.*

- **Using Keyboard Shortcuts:**

  Cut

  - Select what you want to cut.
  - Press *Ctrl* and *x* button at the same time from the keyboard i.e., *Ctrl+x*.

Paste:

- Place the cursor at the point where you want to place the information that is currently on the Clipboard.
- Press Ctrl+v.

Copy:

- Select what you want to copy.
- Press *Ctrl+c*.

- **Using Icons:**

Cut

- Select what you want to cut.
- Click the *Cut* icon.

Paste:

- Place the cursor at the point where you want to place the information that is currently on the Clipboard.
- Click the *Paste* icon.

Copy:

- Select what you want to copy.
- Click the Copy icon.



CHECK YOUR PROGRESS

Q.3: State whether the following statements are true (T) or false (F).

i) If a file is already opened we cannot rename it.

ii) We can give a new name to a folder while it is already opened.

iii) To delete a file permanently we have to press *Alt+Delete* key from the keyboard.

iv) We cannot use * < > ? " / \ : symbols in folder name.

v) *.gif* is an extension name of text file.

vi) *Crtl + p* command is used to paste a folder.

vii) The shortcut key for copying a seleted item is *Ctrl+c*.

viii) *Toolbars* contain the *Cut*, *Copy* and *Paste* icon.



5.7 LET US SUM UP

- The new interface of *Windows XP* provides more stable and reliable environment than the earlier versions of Windows.
- The *desktop* contains several icons, menus, dialogue boxes and shortcuts to various applications, files etc.
- An *icon* is a graphic image. Icons help us to execute commands quickly. To execute a command by using an icon, we have to click the icon.
- *My computer* icon is placed on the desktop to allow users to access disk drives, view and manage files and folders and also to control panel and installed printers.
- *Windows explorer* is a useful feature of Windows XP that allows the users to organise files and folders stored on the computer in an efficient manner.
- The *Title bar* displays the name of the current file and the name of the current program.
- The *Menu bar* displays the program menu.
- *Toolbars* generally display right below the menu, but we can drag them and display them along any of the window borders. We can use the icons on the toolbars to send commands to the program.
- The *Status bar* provides us with information about the status of our program.
- A *file* is a organised and structured collection of related information stored in the computer's memory.
- A *folder* is a collection of files and subfolders. Any file or folder can be renamed, moved or copied to any other location on the computer.



5.8 ANSWERS TO CHECK YOUR PROGRESS

Ans. to Q. No. 1: i) phosphor; ii) bottom; iii) Restore;
iv) Graphical User Interface

Ans. to Q. No. 2: i) False; ii) False; iii) True; iv) True

Ans. to Q. No. 3: i) True; ii) True; iii) False; iv) True; v) False; vi) False; vii) True; viii) True



5.9 FURTHER READING

1) Bott, E. (2002): *Faster Smarter Microsoft Windows XP*; Microsoft Press.
2) Stanek, W. R. (2006): *Introducing Microsoft Windows Vista*; Microsoft Press.



5.10 MODEL QUESTIONS

Q.1: Write down the steps to create a new folder inside F: Drive.

Q.2: What is Menu bar and Toolbar.

Q.3: Write down the steps to rename a folder.

Q.4: How to change the desktop background?

Q.5: What is icon? Describe any two desktop Icon.

*** ***** ***