Srinivas Institute of Management Studies

Pandeswar, Mangalore – 575 001

BACKGROUND STUDY MATERIAL

LINUX ENVIRONMENT

BCA V Semester

Compiled by

Mrs. Lathika K

SIMS, Mangalore

2015

LINUX Environment	
UNIT-1	
Chapter 1	**General Overview of the system**
1.1	History
1.2	System Structure
1.3	User Perspective
1.3.1	The File System
1.3.2	Building Block primitives
1.4	Operating System Services
1.5	Assumptions about Hardware
1.6	Assignment 1
Chapter 2	**Introduction to the Kernel**
2.1	The Unix file system
2.2	Architecture of the Unix Operating System
2.2	Introduction to the System Concepts
2.3.1	An overview of the file subsystem
2.4	Inodes
2.4.1	What is an inode in LINUX?

2.5	System Administration
2.6	Assignment 2
Chapter 3	**The Buffer Cache**
3.1	Buffer headers
3.2	Structure of the buffer pool
3.3	Reading and writing disk blocks
3.4	Advantages and disadvantages of the Buffer cache
3.5	Assignment 3
Chapter 4	**Internal representation of files**
4.1	Inodes
4.2	Structure of the regular file
4.3	Directories
4.4	Conversion of a pathname to an inode
4.5	Super block
4.6	Allocation of Disk blocks
4.7	Other file types
4.8	Assignment 4
UNIT -II	
Chapter 5	**System calls for the file system**
5.1	open system call

UNIT-I
Chapter 1
General overview of the system

1.1 History

In 1965, Bell telephone Labs joined an effort with general Electronic Company and Project MAC of Massachusetts Institute of technology to develop a new operating system called **Multics**. Many people who later took part in the early development of the UNIX system participated in the **Multics** work at Bells Labs.

With the end of their work on the Multics project, members of computing Science Research centre at bell Labs were left without a "convenient interactive computing service". in attempt to improve their programming environment, Ken Thompson, Dennis Ritchie and others sketched a paper design of s file system that later evolved into an early version of the UNIX file system.

To create a better development environment, Thompson and Ritchie implemented their new system design on the PDP-7, including an early version of the UNIX file system, the process subsystem and a small set of the utility programs. The new system was given the name UNIX.

Although this early version of the UNIX system held much promise, it could not realise its potential until it was used in a real project. Thus while providing a text processing system for the patent department at Bell Labs, the UNIX system was moved to a PDP-11 in 1971.the system was characterised by its small size: 16Kbytes for the system, 8KByets for the user programs, a disk of 512 Kbytes and a limit of 64KByets per file.

In 1973, the Operating system was rewritten in C. In 1974, Thompson and Ritchie published a paper describing the UNIX system in the communications of CAM. By 1977, the number of UNIX system sites had grown to about 500, of which 125 were in universities. UNIX systems became popular in the operating telephone companies, providing a good environment for the program development, network

transaction operations services and real time services. Licences of UNIX systems were provided to commercial institutions as well as Universities.

In 1977, Interactive System Corporation became the First Value Added Reseller of a UNIX system. 1977 also marked the year that the UNIX system was first ported to a non-PDP machine, the Interdata 8/32.

With the growing popularity of microprocessors, other companies ported the UNIX system to new machines. In the period from 1977 to 1982, Bell Labs combined several AT & T variants into a single system, known commercially as UNIX system III. Bell Labs later added several features to UNIX System III, calling the new product UNIX system V and AT &T announced official support for system V in January 1983. However, people at the University of California at Berkeley had developed a variant to the UNIX system, the most recent version of which is called 4.3 BSD for VAX machines, providing some new, interesting features.

Several reasons have been suggested for the popularity and success of UNIX system is listed below.

The system is written in high level language, making it easy to read, understand, change and move to other machines.

1. It has a simpler user interface that has the power to provide the services that users want.
2. It provides primitives that permit complex programs to be built form simpler programs.
3. It uses a hierarchical file system that allows easy maintenance and efficient implementation
4. It uses a consistent format for files, the byte stream, making application programs easier to write.
5. It provides a simple, consistent interface to peripheral devices.
6. It is a multi user, multi process system; each user can execute several processes simultaneously.

7. It hides the machine architecture form the user, making it easier to write programs that run on different hardware implementations.

1.2 System structure

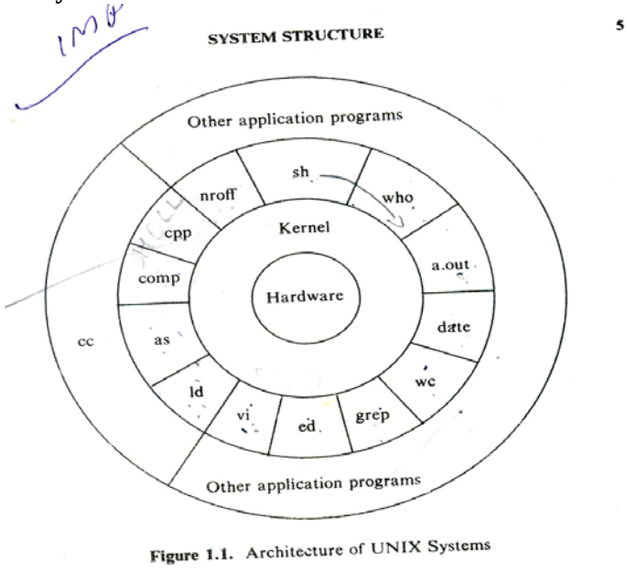


Figure 1.1. Architecture of UNIX Systems

The above figure 1.1 depicts the architecture of the **Unix** system. The hardware at the centre of the diagram provides the OS with basic services. The OS interacts directly with the system. Viewing the system as a set of layers, the OS is commonly called the system kernel or kernel.

Programs such as the shell and editors(ed and vi) shown in the outer layers interacts with the kernel by invoking a well defined set of system calls. The system calls instructs the kernel to do various operations for the calling program and exchange data between the kernel to do various operations for the calling program and exchange data between the kernel and the program. Several programs shown in the figures are commands(grep, wc, date etc). Private user programs may also exist in this layer as indicated by the program whose name is a.out (the standard name for executable files produced by the c compiler). Other

application programs can build on the top of lower level programs, hence the existence of the outer most layer in the figure. For example, the standard C compiler, cc is in the outermost layer of the figure.

1.3 User perspective

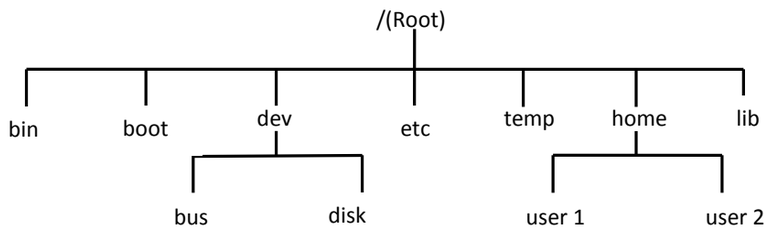
This section briefly reviews high-level features of the UNIX system such as the file system, the processing environment, and building block primitives.

1.3.1 File system

The UNIX file system is categorized by

- A hierarchical structure
- Consistent treatment of file data
- The ability to create and delete files
- Dynamic growth of files
- The protection of file data
- The treatment pf peripheral devices

Unix System File Structure



A file system is the methods and the data structures that an operating system uses to keep track of a file on a disk or partition; i.e the way the files are organized on the disk.

The file system is organized as a tree with a single root node called root denoted by /, every non leaf node of the file system is a directory of files and file at leaf nodes of the tree are either directories, regular files or special device files.

Different types of files can be seen in Unix operating system. Files in Unix operating system are classified as *ordinary files, regular files, directory files* and *special files*.

Ordinary files include a variety of file types like application files, data files, document files etc.

Application files are usually executable files and using these files different applications are executed. There are applications for performing tasks like word processing, doing calculations, browsing the Internet, managing mails, doing image manipulations and so on.

Data files store different types of data and these files can be image files, audio files, video files or simply text files.

Document files store multiple types of information such as text, drawings, images and so on.

Directory files are related to different directories of the system and these store information related to different directories.

Unix special files are device driver files, link files or similar other types of files.

Different physical entities are also represented as files in Unix. Thus, printers, modems and other devices are represented as files. Using special files, the operating system can control different devices connected to the system. Such special files used to control devices are known as device driver files. **Device driver files** are essential for proper communication between the operating system and the connected devices.

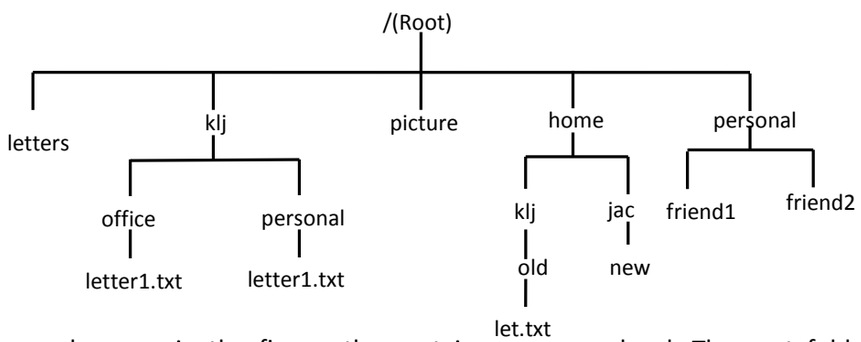
Different files used by Unix operating system are also arranged in a hierarchical manner. **Root** folder is at the top in the hierarchy. All the files and folders of Unix system are included in this folder. **Root** directory has several sub folders named as **bin, boot, dev, etc, temp, home** and so on. The folder named **bin** stores all binary files. For storing boot files, **boot** folder is used. **Device files** necessary for

the proper working of attached peripherals are stored in the folder named **dev.** The folder named **etc** stores different files for configuring the system and devices. The folder named **temp** is a temporary folder and this folder stores temporary files. Files stored in this folder are deleted after use.

Different users of the system are included in the **home** folder. Depending on the number of users, the home folder has several sub folders. Library files required for running the system are stored in the **lib** folder

Paths and Path Names

In a computer system it is possible to find several folders. Each folder may contain several sub folders as well as many files. To locate any file in the system it is necessary to specify the exact path where the file resides in the memory. Simply providing the file name means that the file is located in the directory where the user is located at that time. The concept of paths and path names will be clearer by considering the entire storage space of the system as a single large storage unit. This large storage unit is usually known as root and this is denoted as a single slash character (/). Several folders can be created in the root folder and these are the storage compartments of the root storage space. Each of the partitions can be partitioned further to get several sub folders. The organization described below



As can be seen in the figure, the root is at the top level. The root folder is partitioned into a number of folders namely letters, klj, picture, home and personal. These folders lie just below the root partition. These folders are

represented *as / letters, /klj, / picture, / home and / personal* respectively. The representation indicates the exact paths of the folders and is known as **path names**. Folders named klj and personal are having two sub folders beneath each one. Folders named **office** and **personal** lie below the folder klj. The exact path names of these two folders are written as **/klj/office**, and **/klj/personal** respectively. Both the sub folders store one file each and the names of files are written below the folder name. Sub folders to home folder are **klj** and **Jac** and their exact path names are given as **/home/klj,** and **/home/jac** respectively. The names of files stored in the sub folders are written below the name of the folder. Folders named **friend1** and **friend2** lie as sub folders to the parent folder named personal and their exact path names are given as **/personal/friend1** and **/personal/ friend2.** Similarly the exact path names to folders old and new can be given as **/home/klj/ old**, and **/home/klj/new**.

Names given to files and folders located within a folder must be different. Also different sub folders of a folder must be named differently. But sub folders of different parent folders can use the same name. Files in different folders can use the same file name. For example, as can be seen in the figure, the same file **letter1.txt** is stored in two different folders. This is possible since different folders represent different storage locations. The exact path names of these two files are **/klj/office/letter1.txt** and **/ klj/ personal / letter1 .txt**.

When a user logs into the system, the user is automatically placed in the home folder of the user. Home folder of any user is located as the sub folder to the /home folder. For example, when the user **klj** logs into the system, the user is placed in the folder **/home/ klj.** During this time **/home/klj** is the current directory of the user **klj.**

Exact path names denote the exact locations where files are stored. Such path names are also known as **absolute path** names. In path names, the first slash indicates the **root** folder. Following slashes separate the names of different sub folders. File name appears as the last item in path names. To shorten path names, it is usual to specify the path names relative to the current working directory of the user. Such path names are called **relative path names**. Consider the file tree shown already in Figure 5.2. Suppose the user working in the directory **/home/klj** requires specifying the file **let.txt** stored in the sub folder named old. The path

can be specified in two ways as follows: **/ home/ klj/old / let. txt** and **. / old / let. txt**

Both the path names identify the same file. The first path name is the **absolute path** name whereas the second path name is the **relative path** name. In the relative path name a dot character is used to indicate the current position of the user and the path name begins with a dot. In a similar manner, to refer to a parent directory above the current directory, a user uses two dot characters in relative path names.

Permissions to access a file is controlled by access permissions associated with the file. Access permissions can be set to read, write and execute permissions for 3 users; the file owner, file group and everyone.

UNIX system treats device files as if they were files. Devices are special device files, occupy node positions in the file system directory structure. Programs access devices with the same syntax they use when accessing regular files.

Processing environment

A program is an executable file and a process is an instance of the program in execution. Many processes can execute simultaneously on Unix systems. This feature is called multiprogramming or multitasking. Various system calls allow processes to create new processes, terminate processes, synchronous stages of process execution and control reaction to various events.

Generally, the system calls allow users to write programs that do sophisticated operations. As a result, the kernel of the Unix system does not contain many functions that are part of the “kernel” in other system. The most important example of such a program is the **shell.** Shell is the command interpreter program that is executed by the users after logging into the system.

The shell allows 3 types of commands. First, a command can be an executable file that contain the object code produced by the compilation of the source code(for example, a.out). Second, a command can be an executable file containing a

sequence of shell command lines. Finally, a command can be an internal shell command instead of an executable file.

The internal commands makes the shell, a programming language in addition to a command interpreter and include commands for looping(for –in-do-done and while-do-done), commands for conditional execution(if-then-else-fi), a case statement command, a command to change the current directory of a process etc.

For example, typing the command **who** causes the system to execute the program stored in the file **\bin\who** which prints a list of people who are currently logged into the system.

1.3.2 Building block primitives

Unix provides OS primitives that enable users to write small, modular programs that can be used as building blocks to build more complex programs. One such primitive visible to the shell users is the capability to **redirect I/O**.

Process can access files they read from their standard input file, write to their standard output file and write error messages to their standard error files.

Pipes

The pipe ('|') operator is used to create concurrently executing processes that pass data directly to one another. It is useful for combining system utilities to perform more complex functions. For example:

```
$ cat hello.txt | sort | uniq ↵
```

creates three processes (corresponding to cat, sort and uniq) which execute concurrently. As they execute, the output of the **hello.txt** is passed on to the **sort** process which is in turn passed on to the **uniq** process. **uniq** displays its output on the screen (a sorted list of users with duplicate lines removed). Similarly:

```
$ cat hello.txt | grep "dog" | grep -v "cat" ↵
```

finds all lines in hello.txt that contain the string "dog" but do not contain the string "cat".

Redirecting input and output

The output from programs is usually written to the screen, while their input usually comes from the keyboard (if no file arguments are given). In technical terms, we say that processes usually write to **standard output** (the screen) and take their input from **standard input** (the keyboard). There is in fact another output channel called **standard error**, where processes write their error messages; by default error messages are also sent to the screen.

To redirect standard output to a file instead of the screen, we use the **>** operator:

$ echo temp ↵

The output of the file temp is

hello

$ echo temp > output ↵
$ cat output ↵

The output of the file output is

hello

If we want to append the output of the echo command to the file, we can use the >> operator:

$ echo bye >> output ↵
$ cat output ↵

The output is

hello
bye

Standard input can also be redirected using the < operator, so that input is read from a file instead of the keyboard:

$ cat < output ↵

The output is

hello
bye

1.4 Operating system services

The services provided by the kernel are

1. Controlling the execution of processes by allowing their creation, termination, suspension and communication
2. Scheduling processes for execution on the CPU. Processes share CPU in a time shared manner. The CPU executes a process, the kernel suspends it when its time quantum terminates and kernel schedules another process to execute. The kernel reschedules the suspended process later.
3. Allocating main memory for an executing process. If the system runs low on free memory, the kernel frees memory by writing a process temporarily to secondary memory called a swap device.
4. Allocating secondary memory for efficient storage and retrieval of user data.
5. Allowing processes controlled access to peripheral devices such as terminals, tape drives, network devices.

1.5 Assumptions about hardware

The execution of user processes on UNIX systems is divided into 2 levels; user level and kernel level. When 2 processes execute a system call, the execution mode of the process changes from user mode to kernel mode. The differences between the 2 modes are:

1. Processes in user mode can access their own instructions and data but not kernel instructions and data. Processes in kernel mode can access kernel and user addresses.

2. Some machine instructions are privileged and result in an error when executed in user mode. Although the system executes in one of two modes, the kernel runs on behalf of a user process. The kernel is not a separates set of processes that run in parallel to user processes, but it is part of each user process.

1.6 Assignment 1

2 marks questions

1) Write any two reasons for the success of UNIX operating system
2) Define a program and a process
3) What do you mean by a path? Differentiate full path name and relative path name.
4) What are pipes? Name any 2 kinds of pipes.

5-6 mark questions

1) Discuss about the evaluation of the UNIX operating system
2) List and explain the operating system services
3) Write a note on the file system of Unix Operating system

Chapter 2
Introduction to the Kernel

2.1 The Unix file system

Every item stored in a UNIX file system belongs to one of four types:

1. **Ordinary files**

   Ordinary files can contain text, data, or program information. Files cannot contain other files or directories. Unlike other operating systems, UNIX filenames are not broken into a name part and an extension part (although extensions are still frequently used as a means to classify files). Instead they can contain any keyboard character except for '/' and be up to 256 characters long (note however that characters such as *,?,# and & have special meaning in most shells and should not therefore be used in filenames). Putting spaces in filenames also makes them difficult to manipulate - rather use the underscore '_'.

2. **Directories**

   Directories are containers or folders that hold files, and other directories.

3. **Devices**

   There are two types of devices in UNIX - **block-oriented** devices which transfer data in blocks (e.g. hard disks) and **character-oriented** devices that transfer data on a byte-by-byte basis (e.g. modems and dumb terminals).

4. **Links**

   A link is a pointer to another file. There are two types of links - a **hard link** to a file is indistinguishable from the file itself. A **soft link** (or symbolic link) provides an indirect pointer or shortcut to a file. A soft link is implemented as a directory file entry containing a pathname.

Kernel

In computing, the **kernel** is a computer program that manages input/output requests from software and translates them into data processing instructions for the central processing unit and other electronic components of a computer. The kernel is a fundamental part of a modern computer's operating system.

System call

When a computer program (in this case called a *process*) makes requests of the kernel, the request is called a system call. A system call is a mechanism that is used by the application program to request a service from the operating system.

2.2 Architecture of the UNIX Operating system

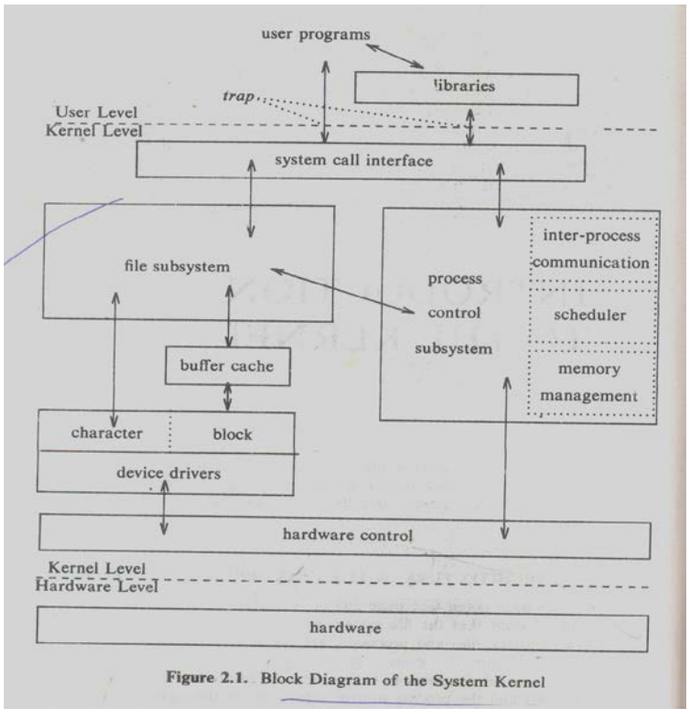


Figure 2.1. Block Diagram of the System Kernel

The above figure gives a block diagram of the Kernel, showing various modules and their relationships to each other. It shows the file subsystem on the left and the process control subsystem on the right. These are the 2 major components of the kernel.

The figure shows 3 levels: user, kernel and hardware. The system call and library interface represent the border between the user programs and the kernel(i.e user programs talks to the kernel with the help of the system calls and the library interfaces)

The figure partitions the set of system calls into those that interact with the file subsystem and those that interact with the process control subsystem. The file subsystem manages files, allocating file space, controlling access to files and retrieving data to users. Processes interact with the file subsystem via a set of system calls such as open, close read, write etc.

The file subsystem accesses file data using a buffering mechanism. This buffering mechanism controls the data flow between the kernel and secondary storage devices.

Device drivers are the kernel modules that control the operation of peripheral devices. Block I/O devices are random access storage devices. A tape driver(example for an I/O device) may allow the kernel to treat a tape unit as a random access storage device. The file subsystem also interacts directly with "raw" I/O device drivers without the interventions of a buffering mechanism. Raw devices, sometimes called character devices, include all devices that are not block devices.

The process control subsystem is responsible for process synchronization, IPC, memory management and process scheduling.

The memory management module controls the allocation of memory. If at any time the system does not have enough physical memory for all processes, the

Formatted: Font: (Default) Calibri, 14 pt

kernel moves them between main memory and secondary memory so that all processes get a chance to execute.

The scheduler module allocates the CPU to processes. It schedules them to run until they voluntarily leave the CPU or until the kernel pre-empts them when their runtime exceeds a time quantum. The scheduler then chooses the highest priority process to run. The original process will run again when it is the highest priority process available.

The hardware control is responsible for handling interrupts and for communicating with the machine. Devices, such as disks or terminals may interrupt the CPU while a process is executing. If so, the kernel may resume execution of the interrupted process after servicing the interrupt.

2.3 Introduction to system concepts

Formatted: Font: (Default) Calibri, 14 pt, English (India)

This section gives an overview of some major kernel data structures and describes the function of modules.

2.3.1 An overview of the file subsystem

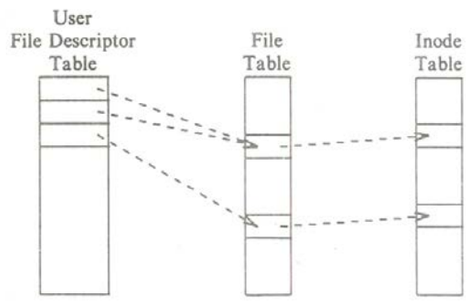
Formatted: Font: (Default) Calibri, 14 pt, English (India)

The internal representation of a file is given by an inode, which contains a description of the disk layout of the file data and other information such as the file owner, access permissions and access times.

The kernel contains two other data structures- the file table and the user file descriptor table. There will be only one file table and it contains the information of all the files in the disk. the user file descriptor table is allocated per process. When a file is created, it is given a number called **file descriptor.**

The kernel returns a file descriptor for the **open** and **creat** system calls, which is an index into the user file descriptor table. When executing read and write system calls, the kernel uses the file descriptor to access the user file descriptor table, follows pointers to the file table and inode table entries, and, from the inode, finds the data in the file.

Formatted: Font: (Default) Calibri, 14 pt



File system layout

In the original Unix file system, Unix divided physical disks into logical disks called *partitions*. Each partition is a standalone file system. We will use the term ``file system'' when referring to a single partition.

A file system has the following structure

Formatted: Font: (Default) Calibri, 14 pt

1. The boot block occupies the beginning of a file system and may contain the bootstrap code that is used to boot the machine. Although only one boot block is needed to boot the system, every file system has a boot block.
2. The superblock describes the state of the file system- how large it is, how many files it can store, where to find free space on the file system and other information.
3. The inode list is a list of inodes that follows the superblock in the file system. Administrators specify the size of the inode list when configuring a file system. The kernel references inodes by index into the inode list. One inode is the **root** inode of the file system.
4. The data blocks start at the end of the inode list and contain file data and administrative data.

Boot block	Super block	Inode list	Data blocks

2.4 Inodes

Each and everything in Linux is accessed by a file. Even block devices like Hard disks, and CD/DVD's are nothing other than a file. The operating system(Linux) files inside a file system(be it ext2,ext3 or ext4) are not really accessed by their name. The file system recognizes a file not by its name but by a number. That number through which the operating system reaches the location and other attributes of that file is called as an **inode number**.

2.4.1 What is an INODE in Linux?

Its a data structure that keeps track of all the information about a file. You store your information in a file, and the operating system stores the information about a file in an inode(sometimes called as an inode number). Information about files(data) are sometimes called metadata. So you can even say it in another way, "An inode is metadata of the data."

Whenever a user or a program needs access to a file, the operating system first searches for the exact and unique inode (inode number), in a table called as an inode table. In fact the program or the user who needs access to a file, reaches the file with the help of the inode number found from the inode table.

2.5 System administration

There is no difference between administrative processes and user processes. They use the same set of system calls available. They are distinguishable from general user processes only in the rights and privileges they are allowed. Internally, the kernel distinguishes a special user called the super user with special privileges. A user may become a super user by going through a login password sequence or by executing special programs. Administrative processes do functions like disk formatting creation of new file system, repair of damaged file system, kernel debugging etc

2.6Assignment 2

2 mark questions

1. What is a boot block?
2. What is a super block?

5-6 mark questions

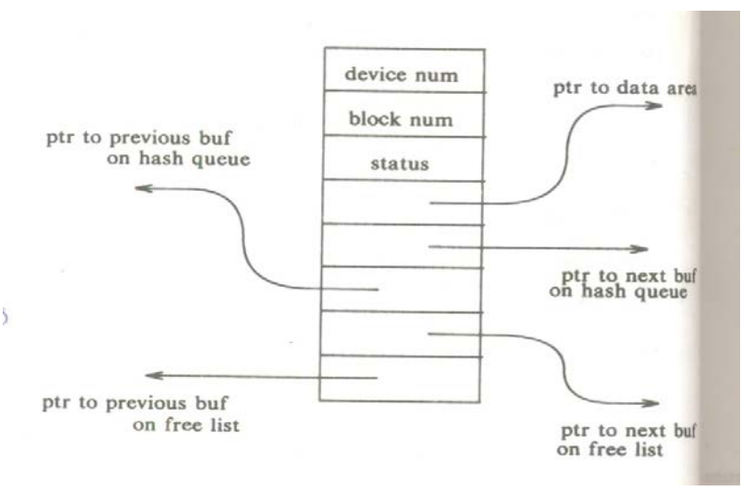
1) With a neat diagram explain system kernel
2) Write a note on system administration.
3) Write a note on file system layout.

Chapter 3
The buffer cache

Buffer cache

Reading from a disk is slow compared to reading from disk. It is common to read the same part of a disk several times during short period of time. By reading the information from the disk only once and then keeping it in memory until no longer needed, one can speed up all, but the first read. This is called disk buffering and the memory used for the purpose is called buffer cache.

3.1 Buffer headers



A buffer consists of 2 parts: memory array and a buffer header

Memory array contains data from the disk and buffer header identifies the buffer. The buffer header contains a **device number** and **block number** field. The **device number** is used to uniquely identify the partition in which the block resides and **block number** is used to identify the block.

The status of a buffer is a combination of the following conditions

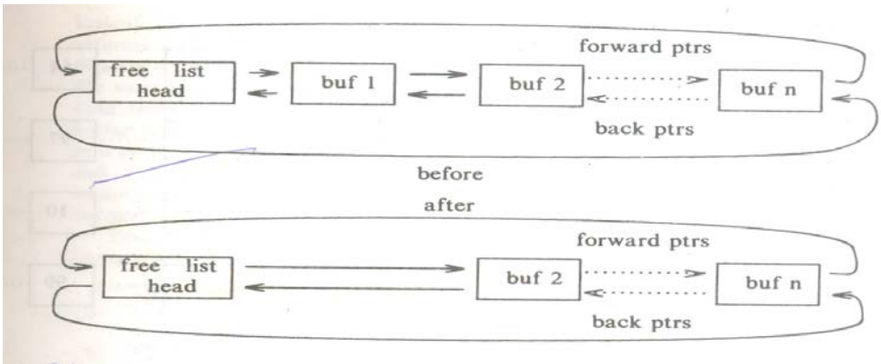
1. The buffer is currently busy.
2. The buffer contains valid data
3. The kernel must write the buffer contents to disk before reassigning the buffer. This condition is known as delayed write.
4. The kernel is currently reading or writing the contents of the buffer to disk.

The process is currently waiting for the buffer to become free.

3.2 Structure of a buffer pool

The kernel caches data in the buffer pool according to a least recently used algorithm. After it allocates a buffer to a disk block, it can not use the buffer for another block until all other buffers have been used more recently.

The kernel maintains a free list of buffers that preserves the least recently used order. The free list is a doubly circular linked list of buffers with a dummy buffer header that marks the beginning and end. Every buffer is put on the free list when the system is booted. The kernel takes a buffer from the head of the free list when it wants any free buffer. When the kernel returns a buffer to the buffer pool, it usually attaches the buffer to the tail of the free list, occasionally to the head of the free list, but never to the middle. As the kernel removes buffers form the free list, a buffer with valid data moves closer to the head of the free list.



3.3 Reading and writing disk blocks

When a process requests some data, it asks for the kernel for it. Kernel finds the data in the buffer, reads the data from it and delivers the data to the process.

If the required data is not present in the buffer, then the kernel asks the hard disk driver to get the data from the hard disk and puts the process to sleep, the hard disk driver asks the hard disk controller to get the data from the hard disk. After getting the data from hard disk, the controller dumps it in a buffer called DMA buffer. But hard disk driver does not know that work is done. To inform the work is done, the controller interrupts CPU. At this situation the processor calls the interrupt service routine (a function related to the hard disk). When interrupt service routine is executed, hard disk driver will come to know that the work is done. Hard disk driver informs this to the kernel and the kernel provides the required data to the process.

The process for writing the contents of a buffer to a disk block is described below. The kernel informs the disk driver that it has a buffer whose contents should be written to a disk block. The disk driver schedules the block of data for I/O. If the write is synchronous, the calling process goes to sleep awaiting I/O completion.

If the write is asynchronous, the process tells the kernel that write is asynchronous. So the kernel issues a write request to the hard disk driver and does not care about its completion now and process will continue its work.

Algorithm for reading a disk block

Algorithm bread /*block read*/

Input: file system block number
Output: buffer containing data
{
    Get buffer for block
    If(buffer valid data)
        Return buffer;

```
        Initiate disk read;
Sleep(event disk read complete);
Return (buffer)
}
```

Algorithm for Block Read Ahead

```
Algorithm breada            /*block read and read ahead */

Input 1) file system block number for immediate read
2) file system block number for asynchronous read
Output: buffer containing data for immediate read
{
        If (first block not in cache)
        {
                Get buffer for the first block;
                If(buffer data not valid)
                        Initiate disk read;
        }
        If (second block not in cache)
        {
                Get buffer for second block;
                If (buffer data valid)
                        Release buffer;
                Else
                        Initiate disk read;
        }
        If(first block was originally in cache)
        {
                Read first block
                Return buffer;
        }
        Sleep (event first buffer contains valid data);
        Return buffer;
}
```

Algorithm for writing a disk block

```
Algorihm bwrite          /*block write*/

Input: buffer
Output: none
{
        Initiate disk write;
        If (I/O synchronous)
        {
                Sleep(event I/O complete);
                Release buffer;
        }
        else if(buffer marked for delayed write)
                Mark buffer to put at head of free list;
}
```

3.4 Advantages and disadvantages of buffer cache

1. By using a buffer cache, we can have a common interface for input/output. this makes the system design simpler.
2. For writing data directly to the hard disk requires that the data is hardware aligned.
3. It can reduce the amount of disk traffic because it is not necessary that every time the kernel should search data in the hard disk. The data is stored in the buffer cache after the first read. This increases overall system throughput and decreasing response time.

Since there is single image of disk block in the cache, file integrity problem will not occur.

Disadvantages

1. Since the kernel does not write data immediately to the hard disk for a delayed write, the system may crash. That causes the disk data in an incorrect state.

2. When system calls involving file system is invoked, the data from the user space is copied to the kernel. Thus there exists 2 copies of same data and there is an overhead involved in creating that extra copy. When large amount of data is being read or written, this can cause high inefficiency and reduced performance.

3.5 Assignment 3

1. What is a buffer cache? Explain the structure of a buffer pool.
2. Write a note on reading and writing disk blocks
3. What are the advantages and disadvantages of a buffer cache?
4. Write an algorithm for reading a disk block
5. Write an algorithm for writing a disk block
6. Write an algorithm for read write ahead.

Chapter 4
Internal representation of file

This chapter describes the internal structure of files in the UNIX system. The inode contains the information necessary for a process to access a file such as file owner ship, access rights, file size and location o file data in the file system

4.1 Inodes

When a file system is created, data structures that contain information about files are created. Each file has an inode and is identified an inode number in the file system where it resides.

The inode contains the information necessary for a process to access a file such as file ownership, access rights, file size etc.

Disk inode consists of the following fields

File owner identifier

Ownership is divided between an individual owner and a group owner and defines a set of users who have access rights to a file. The super user has access rights to all files in the system.

File type

Files may be type regular, directory, character or block special files.

File access permissions

The system protects files according to 3 classes: owner, group owner and other users. Each class has access rights to read, write and execute files.

File access time

This gives the time the file was last modified, when it was last accessed and when the inode was last modified.

Number of links

Number of links to the file, represents the number of names the file has in the directory hierarchy.

The kernel saves the data in distinguishable disk blocks. The inode identifies the disk block that contains the file's data.

File size

Data in a file is addressable by the number of bytes from the beginning of the file starting from the byte offset 0.

Incore Inode

In memory, a copy of disk inode is present and this copy contains more information than the disk copy inode. This is called **incore inode.**

The incore copy of the inode contains the following fields in addition to the fields of the disk inode

1. **The status of the incore inode** indicates
   a. Whether the inode is locked
   b. Whether the process is waiting for the inode to become unlocked
   c. Whether the incore representation of the inode differs from the disk copy, as a result of a change to the data in the inode.
   d. Whether the incore representation of the file differs from the disk copy inode as a result of a change to the file data
2. The logical device number of the file system that contains the file
3. **The inode number**: the inode number of the disk copy is stored in the incore inode

4. **Pointers to another incore inodes**: all incore inodes will be in the form of a linked list. If one incore inode is found, the other can be searched easily.
5. **A reference count**: this indicates the number of instances of the file that are active.

4.2 Structure of a regular file

Disk inode is a one that represents a file and file contains data. The file will be in terms of disk blocks and each disk block on a disk is addressable by a number called disk block number.

Inode contains a table which consists of a set of disk block numbers. If file data are stored in terms of disk blocks sequentially, there is a risk of fragmentation. But we can obtain the required data block of a file by just knowing the start block address and the file size in the inode.

Suppose a user creates 3 files A,B and C each consisting of 10 disk blocks of storage qnd these 3 files are stored continuously. If the user then wishes to add 5 blocks of data to the middle file B, the kernel would have to copy file B to a place in the file system that had room for 15 blocks of storage. The disk blocks previously occupied by the file B's data would be unusable except for files smaller than 10 blocks.

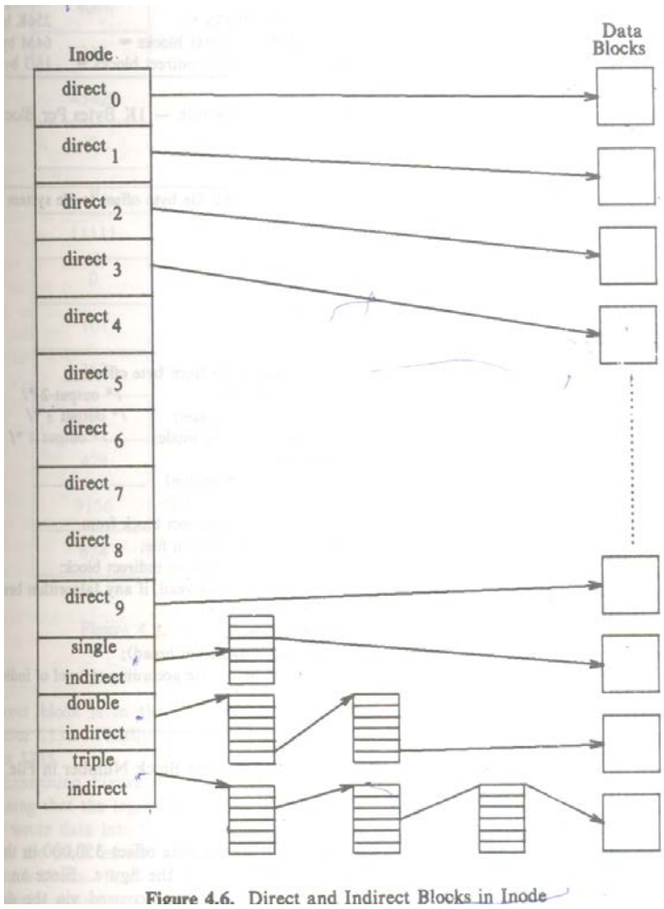
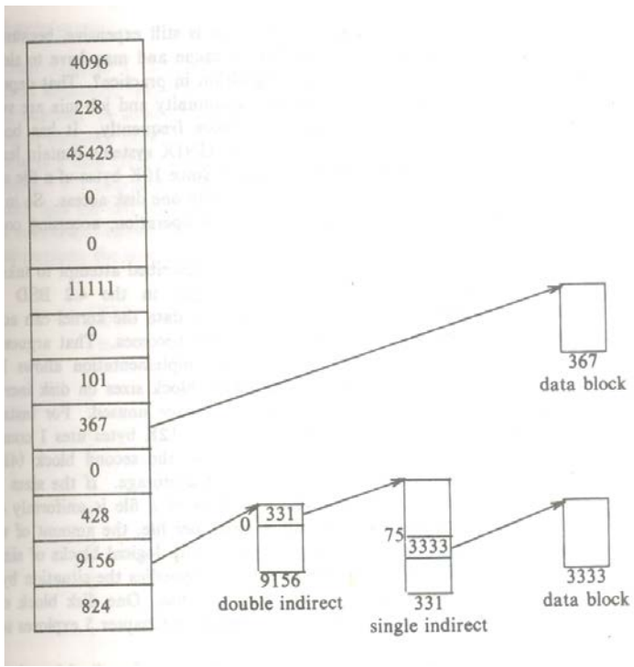


Figure 4.6. Direct and Indirect Blocks in Inode

Kernel allocates one block at a time for a file and the data in the file is spread throughout the file system. In this case the table consisting of block numbers in the inode is difficult to manage.

To keep the inode structure small and allow large files, the table of contents of disk blocks are designed as shown in the figure. The system V unix system runs with 13 entries in the inode table of contents. The block marked “direct” in the figure contains disk block number that contains actual data. The block marked

"single indirect" refers to a block that contains a list of direct block number. The block marked "double indirect" contains a list of single indirect block numbers and block marked "triple indirect" contains a list of double indirect block numbers



4.3 Directories

A directory is a file whose data is a sequence of entries , each consisting of an inode number and the name of a file contained in the directory. A pathname is divided into separate components by the slash(/) character. The kernel stores data for a directory just as it stores data for an ordinary file using inode structure and levels of direct and indirect blocks. Processes may read directories in the same way they read regular files but the kernel reserves exclusive right ti write a directory.

The access permissions of a directory have the following meaning. Read permission on a directory allows a process to read a directory, write permission allows a process to create a new directory or remove old one, execute permission allows a process to search the directory for a file name.

4.4 Conversion of a pathname to an inode

The algorithm **namei** parses the pathname one component at a time, converting each component into an inode based on its name and the directory being searched and eventually returns the inode of the input pathname. **Namei** uses intermediate inodes as it parses a pathname, call them working inodes. The inode where the search starts is the first working inode. During each iteration of the **namei** loop, the kernel makes sure that the working inode is indeed that of a directory.

The kernel does a linear search of the directory file associated with the working inode. For example, suppose a process wants to open a file /lathika/temp. When the kernel starts parsing the file name, it encounters / and gets the inode of the root directory /. Kernel makes root as the current working inode, after checking that the current inode is a directory, the kernel searches **root** for a file whose name is **lathika.** It accesses data in the root directory block by block and searches each block , one entry at a time until it locates an entry for “lathika”. On finding the entry, kernel releases the inode for the “root” and allocates inode for “lathika”. After finding that **lathika** is a directory, the kernel searches ”lathika” block by block for a file named “temp”. On finding this, the kernel releases the inode for “temp” and allocates inode for “temp”. Since the pathname is ended kernel returns that inode.

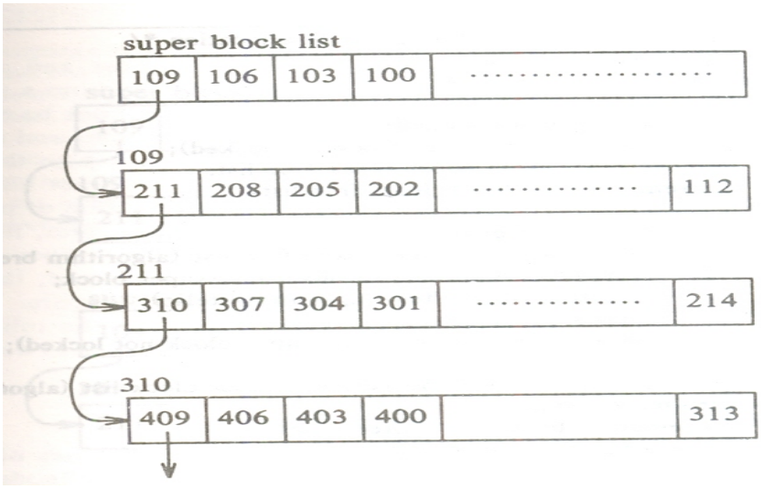
4.5 Superblock

The superblock consists of the following fields

1. Size of the file system
2. The number of free blocks in the file system

3. A list of free blocks available on the file system
4. The index of next free block in the free block list
5. The size of the inode list
6. The number of free inodes in the file system
7. A list of free inodes in the file system
8. The index of the next free inode in the free inode list
9. Lock fields for the free block and free inode lists
10. A flag indicating that the super block has been modified.

4.6 Allocation of disk blocks



When a process writes data to a file, the kernel must allocate disk blocks from the file system. The superblock contains a list of free disk blocks in the file system. The utility program **mkfs** organizes the data blocks of a file system in a linked list such that each link of a list is a disk block that contains an array of free disk block numbers. The first entry in each block is the pointer to the next block.

In the above diagram, the first block is the super block list. When kernel wants to allocate a block from a file system, it allocates a block from the file system; it allocates the next available block in the superblock list i.e starting with 100). Once allocated blocks cannot be reallocated until it becomes free.

Assume that the kernel started assigning of blocks from the superblock list to a process , starting from 100th block, next it allocates 103, next 106 then 109. But 109 is the last block. Kernel cannot allocate 109 because it already has data. Still it allocates block number 109 to a process by copying all the contents of block number 109 to superblock list, since superblock list is empty. After doing this, kernel allocates 109 to the process. The kernel does not directly write the data to the disk block. It writes the data through buffer. i..e kernel creates a buffer for each data block. If the file system contains no free block, the calling process receives an error.

If a process writes a lot of data to a file, it repeatedly asks the kernel for blocks to store the data, but the kernel assigns only one block at a time. The program **mkfs** tries to organise the original linked list of free block numbers so that block numbers related to a file are near to each other. This helps performance, because it reduces disk seek time and latency when a process reads a file sequentially.

The algorithm free for freeing a block is the reverse of one for allocating a block. If the superblock list is not full, the block number of the newly freed block is placed on the superblock list. If the superblock list is full, the newly freed block becomes a link block. The kernel writes the superblock list into the newly freed block and writes this block to the disk. It then places the block number of the newly freed block in the superblock list.

4.7 other file types

The UNIX system supports two other file types: pipes and special files. A pipe sometimes called a FIFO, differs from a regular file. Once data is read from a pipe, it cannot be read again. Also, data is read in the order that it was written to the pipe.

The other file type in the UNIX system are special files, including block device special file s and character device special files. Both type specify devices and therefore the file inodes do not reference any data.

4.8 Assignment 4

1) Explain the different fields of incore inode and disk copy inode
2) With a neat diagram, explain the structure of a regular file
3) Write a note on super blocks
4) With a neat diagram explain the allocation of disk blocks
5) Explain the conversion of a path name to an inode.

UNIT-II
Chapter 5
System calls for file system

This chapter deals with system calls for the file system. It starts with system calls for accessing existing files such as open, read, write, lseek and close, then presents system calls to create new files namely, creat and mknod and then examines the system calls that manipulate the inode: chdir, chroot, chown, chmod, stat and fstat. It investigates more system calls: pipe and dup. Linl and unlink change the structure of the file system hierarchy, mount and unmount extend the file system tree visible to users.

5.1 Open

The open system call is the first step a process must take to access the data in a file. The syntax of the open system call is

fd=open(pathname, flags, modes);

Where pathname is the filename, flags indicate the type of open (such as for reading or writing) and modes gives the file permissions if the file is being created. The open system call returns an integer called the **user file descriptor.**

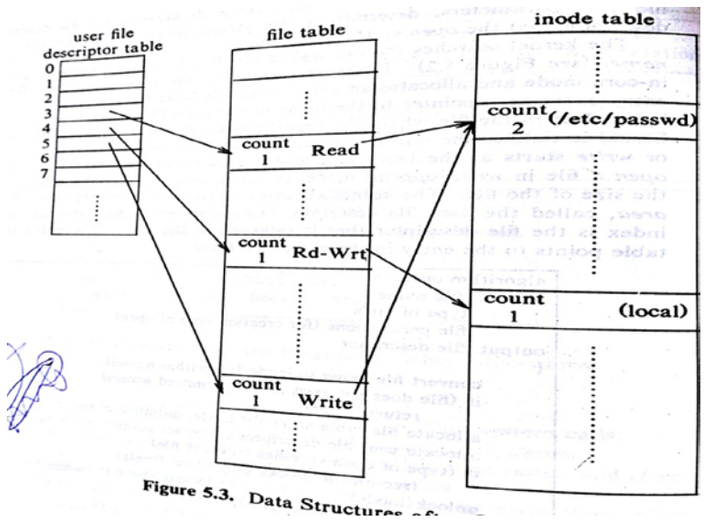


Figure 5.3. Data Structures

The kernel searches the file system for the filename parameter using the algorithm **namei**. To access a file, kernel needs to know the inode of a file. Every file has an inode. The copy of the disk inode will be there in the memory and it is called incore inode. Kernel checks permissions for opening the file after it finds the incore inode and allocates an entry in the file table for the open file. The file table entry contains a pointer to the inode of the open file and a filed that indicate the byte offset in the file, where the kernel expects the next read or write to begin. The kernel initializes the offset to 0 during the open call, meaning that the initial read or write start at the beginning of a file by default. A process can open a file in write append mode, in which case the kernel initializes the offset to the size of the file. The kernel allocates an entry in a private table called the user file descriptor table and notes the index of this entry. This index is the file descriptor that is returned to the user.

Suppose a process executes the following code, opening the file "/etc/passwd" twice, once for reading and once for write only and the file "local" once for reading and writing.

Fd1=open("/etc/passwd", o_rdonly);

Fd2=open(“local”, o_rdwr);

Fd3=open(“/etc/passwd”,o_wronly);

The above figure shows the relationship between the inode table, file table and the user file descriptor tables. Each open call returns a file descriptor to the process and the corresponding entry in the user file descriptor table points to a unique entry in the file table, even though the file “/etc/passwd” is opened twice. The file table entries of all instances of an open file point to one entry in the incore inode table. The process can read or write the file “/etc/passwd”, but only through the file descriptors 3 and 5 in the figure.

5.2 Read

The syntax is

number=read(fd, buffer, count);

Where fd is the file descriptor returned by the open system call. Buffer contains the address of the data structure that will contain the read data on successful completion of the call. Count is the number of bytes the user wants to read and number is the number of bytes actually read. The kernel gets the address of the file table entry from the user file dcriptor table. It now sets several I/O parameters in the u area. After setting the I/O parameters in the u area, it follows the pointers from the file table entry to the inode, locking the inode before it reads the file.

The kernel converts the byte offset into a block number and it notes the byte offset in the disk block where I/O should begin and how many bytes in the block it should read. Then it copies the data from the block into the buffer and then from the buffer to the target address in the user process. It updates the I/O parameters in the U area according to the number of bytes it read, incrementing the file byte offset and the address in the user process where the next data should be delivered and decrementing the count of bytes it needs to read to satisfy the user

read request. It the user request is not satisfied, the kernel repeats the entire cycle, converting the byte offset to a block number, reading the block from disk to a system buffer, copying data from the buffer to the user process, releasing the buffer and updating the I/O parameters in the U area.

The cycle completes either when the kernel completely satisfies the user request, when the file contains no more data or if the kernel encounters an error in reading the data from disk or copying the data to user space.

5.3 Write

The syntax is

Number=write(fd, buffer,count);

Where meaning of fd, buffer, count and number are the same as they are for read system call.

If a file does not contain enough blocks to store the data which was written by the process, the kernel allocates a new block using algorithm **alloc**. The inode is locked for the duration of write, because kernel may change the inode when allocating new blocks. When the write is complete, the kernel updates the file size entry in the inode if the file has grown larger.

5.4 Adjusting the position of the file I/O- Lseek

The ordinary use of read and write system calls provide sequential access to a file, but processes can use the lseek system call to position the I/O and allow random access to a file. The syntax is

Position=lseek(fd, offset,reference);

Where **fd** is the file descriptor identifying the file, **offset** is the byte offset and **reference i**ndicates whether offset should be considered from the beginning of the file, from the current position of the read/write offset or from the end of the

file. The return value **position** is the byte offset where the next read or write will start.

5.5 File creation- Creat

The creat system call creates a new file in the system. The syntax is

Fd=creat(pathname, modes);

Where fd, pathname and modes mean the same as they do in the open system call. If no such file previously existed, the kernel creates a new file with the specified name and permission modes. If the file already existed, the kernel truncates the file, i.e releases all existing data blocks and sets the file size to zero.

The kernel parses the pathname using the algorithm **namei**. When it arrives at the last component of the pathname, namely filename that it will create, **namei** notes the position of the first empty directory slot in the directory and saves the offset in the u area.

If the directory has no empty slot, the kernel remembers the offset of the end of the directory and creates a new slot there.

Assuming that, no file by the given name previously existed, the kernel assigns an inode for the new file using the algorithm **ialloc**. It then writes the new file component and the inode number of the newly allocated inode in the parent directory at the byte offset saved in the u area. Afterwards it releases the inode of the parent directory. The parent directory now contains the name of the new file and its inode number.

If the given file already existed, before the **creat** system call, it already has an inode number. The kernel finds its inode while searching for the filename. The old file must allow write permission for a process to create a new file by the same name, because the kernel changes the file contents during the **creat** system call. It truncates the file, freeing all the data blocks using the algorithm free, so that the file looks like a newly created file.

5.6 Pipes

Pipes allow transfer of data between processes in a FIFO manner. Conceptually, a pipe is a connection between 2 processes, such that standard output form one process becomes the standard input of other process. There are 2 kinds of pipes: named pipes and unnamed pipes. Processes use the open system call for named pipes but the pipe system call to create an unnamed pipe.

Pipe system call

Pipe() is a system call that facilitates inter process communication. It opens a file, which is an area of main memory that is treated as a “virtual file”. One process can write to ths “virtual file” or pipe and another related process can read from it. If a process tries to read before something is written to the pipe, the process is suspended until something is written. The syntax for creation of a pipe is **pipe(fdptr);**

Where fdptr is the pointer to an integer array that will contain the 2 file descriptors for reading and writing the pipe. Because a pipe does not exists before its user, the kernel must assign an inode for it on creation. It also allocates a pair of user file descriptor and corresponding file table entries for the pipe: one file descriptor for reading from the file and another for writing the pipe.

The kernel assigns an inode for a pipe forma pipe device. A pipe device is just a file system from which kernel can assign inodes and data blocks for pipes. The kernel then allocates 2 file table entries for the read and writes descriptors respectively and updates the information in the incore inode. Each file table entry records how many instances of the pipe are open for reading and writing. Initially 1 for each table entry and the inode reference count indicates how many time the pipe was opened. Initially 2, one for each file table entry. Finally, the inode records byte offsets in the pipe where the next read or write of the pipe will start.

5.7 Mounting and unmounting file systems

A physical disk consists of several logical sections called partition. Each partition is a device in UNIX. Each device has a name, called device file name. A section of a disk may contain a logical file system (NTFS, FAT etc). The mount system call connects the file system in a specified partition of a disk to the existing file system hierarchy. The syntax of mount system call is **mount (special pathname, directory pathname, options);**

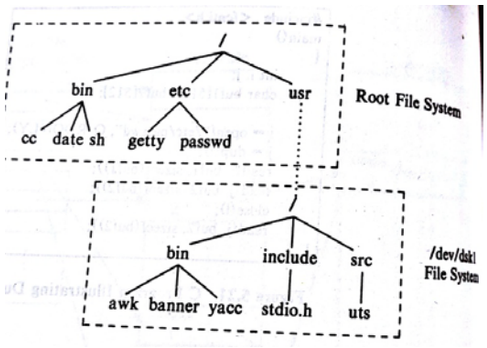
Where special pathname is the name of the device special file of a disk section containing the file system to be mounted, directory pathname is the directory in the existing hierarchy where the file system will be mounted and options indicate whether the file system should be mounted "read only".

If a process issues a system call mount("/dev/dsk1", "/usr", o);

The kernel attaches the file system contained in the portion of the disk called /dev/dsk1 to the directory **/usr** in the existing file system tree. The file /dev/dsk1 is a block special file meaning that it is the name of a block device. After completion of the mount system call, the root of the mounted file system is accessed by the name **/usr.**

The kernel has a mount table with entries for every mounted file system. Each mount table entry contains

1. A device number that identifies the mounted file system (in our example, dsk1 is the device and it has a number called device number. /dev/dsk1 is the file path where dsk1 is the file name.)
2. A pointer to the buffer containing the file system super block.
3. A pointer to the root inode of the mounted file system(mounted file system also contains a root and this root has an inode i.e / of the /dev/dsk1)
4. A pointer to the inode of the directory that is the mount point.(usr is the root file system)



5.8 Unmounting a file system

The syntax is unmount(special filename);

Where special filename indicates the file system to be unmounted. When unmounting a file system

1. Kernel accesses the inode of the device to be unmounted(i.e kernel accesses the inode of dsk1)
2. Retrieves the device number of the special file(it retrieves the device number of dsk1)
3. Releases the inode(releases the inode of dsk1)
4. Finds the mount table entry whose device number equals that of the special file.

Before the kernel actually unmounts a file system, it makes sure that no files on that file system are still in use by searching the inode table for all files whose device number equals that of the file system being unmounted.

5.9 Link system call

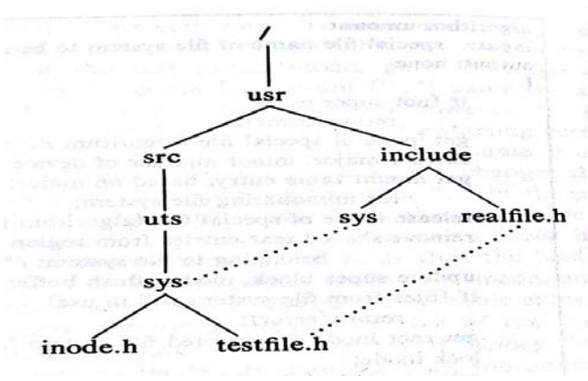
The link system call links(assigns) a file to a new name in the file system directory structure. The syntax is link(source filename, target filename);

Where source file name is the name of the existing file and the target file name is the another name the file will have after the completion of the link call. For example, after executing the system calls,

Link("/usr/src/uts/sys", "/usr/include/sys");

Link("/usr/include/realfile.h", "/usr/src/uts/sys/testfile.h");

The following 3 pathnames refer to the same file /usr/src/uts/sys/testfile.h, /usr/include/sys/testfile.h and /usr/include/realfile



The kernel allows only a super user to link directories. Link system call gives another file name to the previous one. The kernel first locates the inode for the source file using algorithm **namei**, increments its link count, updates the disk copy of the inode and unlocks the inode, it then searches for the target file. If the file is present, the link call fails and the kernel decrements the link count incremented earlier. Otherwise it notes the location of an empty slot in the parent directory of the target file, writes the target file name and the source file inode number into that slot and releases the inode of the target file's parent directory. The link count of the source file inode is greater that it was at the beginning of the call. For example, when executing

Link("temp", "dir/temp1");

The kernel locates the inode for the file **temp**, increments its link count, remembers its inode number say 74. It locates the inode of **dir**, the parent directory of **temp1**, finds the empty directory slot in **dir** and writes the file name **temp1** and the inode number 74 into the empty directory slot. Finally it releases the inode for **temp.** If the link count of **temp** had been 1, it is now 2.

5.10 Unlink

The unlink system call removes a directory entry for a file. The syntax is **unlink(pathname);** where pathname identifies the name of the file to be unlinked from the directory hierarchy. If a process unlinks a given file, no file is accessible by that name until another directory entry with that name is created. For example

Unlink(“myfile”);

Fd=open(“myfile”, “o_rdonly);

The open system call should fail, because the current directory no longer contains a file called **myfile.**

The kernel first uses a variation of algorithm **namei** to find the file that is to be unlinked. It accesses the incore inode of the file to be unlinked using algorithm **iget.** After checking error conditions and other parameters, the kernel clears the filename from the parent directory. This is done by writing a 0 for the value of the inode number in the directory entry.

5.11 Assignment 5

1) Write a note on read() system call
2) Write a note on write system call
3) What are pipes? Name 2 types of pipes.
4) What is a mount system call? Explain
5) What is a link system call? Explain.
6) Explain creat system call

Chapter 6
The I/O subsystem

The I/O subsystem is a component of the OS and all the device drivers are managed by the I/O subsystem. The I/O subsystem allows a process to communicate with peripheral devices such as disks, tape drives, terminals, printers and networks and kernel modules that control devices known as device drivers. Device driver is the software that talk to the devices. Systems may contain one disk driver to control all drives, one terminal driver to control all terminals and one tape driver to control all tape drives. Two brands of tape drives may treat the devices as 2 different device types and have 2 separate drivers.

6.1 Driver interfaces

The Unix system contains 2 types of devices, block devices and character devices. Block devices such as disks and tapes look like random access storage devices to the system. Character devices include a;; other devices such as terminals and network media.

The user and device communication is through filesystem. Every device is considered as a file. Every devie has a name that looks like a file name and is accesses like a file. The device special file has an inode and occupies a node in the directory hierarchy of the file system. The device file is distinguished from other files by the file type stored in its inode either **“block” or “character special”.** If a device has both **block and character** interface, it is represented by 2 device files, its block device special file and its character device special file.

6.1.1 System configuration

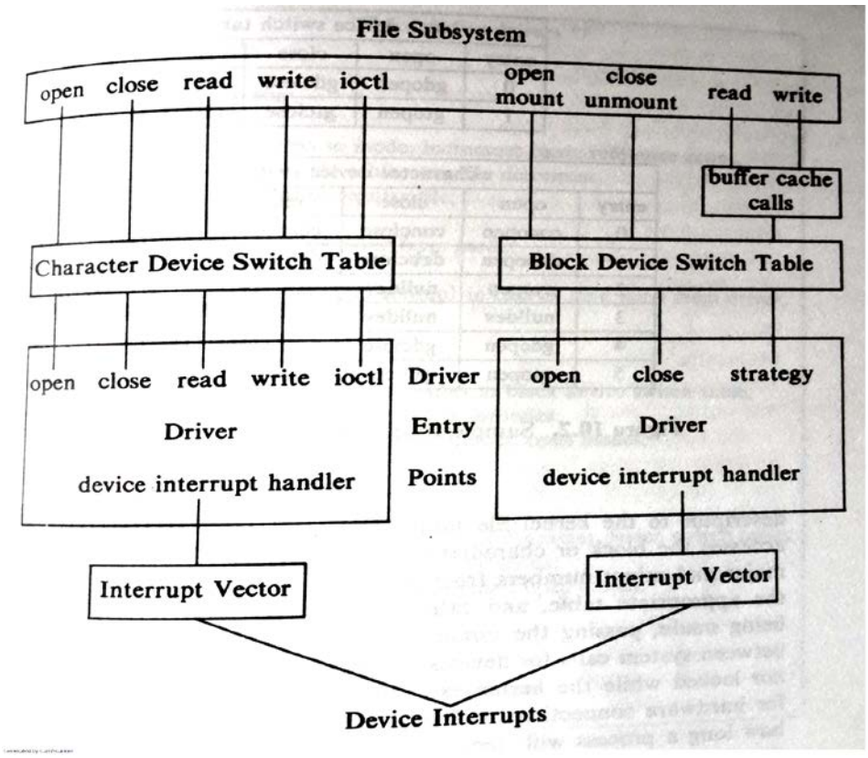
System configuration is the procedure by which administrators specify parameters that are installation dependent. Some parameters specify the sizes of kernel tables such are process tables, inode tables, file table and the number of buffers to be allocated for the buffer pool.

There are 3 stages at which device configuration can be specified.

1. Administrators can hardcode configuration data into files that are compiled and linked when building the kernel code.
2. Administrators can supply configuration information after the system is already running.
3. Finally, self identifying devices permit the kernel to recognize which devices are installed.

The kernel to driver interface is described by the block device switch table and the character device switch table. Each device type has entries in the table that direct the kernel to the appropriate driver interface for the system calls. The open and close system calls of a device file funnel through the two device switch tables , according to the file type. The mount and unmount system calls also invoke the device open and close procedures for block devices. Read, write and ioctl system calls of character special file pass through the respective procedures in the character device switch table. Read and write system calls of block devices and files on mounted file systems invoke the algorithms of the buffer cache.

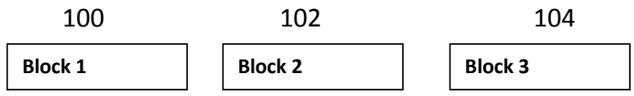
The hardware to driver interface consist of machine dependent control registers or I/O instructions for manipulating devices and interrupt vectors. When a device interrupt occurs, the system identifies the interrupting device and calls the appropriate interrupt handler.



6.2 Disk drivers

Disk units in Unix systems are configured into sections (partitions) that contain individual file systems. For example, a disk contains 4 file system; an administrator may leave one unmounted, mount another “read only” and mount the last two “read write”. Even though all file system exists on one physical unit users can not access files in the unmounted file system, nor can any users write files in the read only file system.

The disk driver translates a file system address consisting of a logical device number and block number to a particular sector on disk.



Assume 1000. This is the device number for D:\

The file system treats a partition as a set of blocks. It assigns an address to each block which is relative to the partition itself. To read and write from the hard disk, we need to specify an address on the hard disk. So we should translate a given block number on a particular partition to the absolute address in the hard disk. This can be done by using the device no. of the partition and the given file system block number.

The sizes and lengths of disk sections have been fixed according to the disk type. For instance the DEC RP07 disk is partitioned into sections as shown below.

Section	**Start block**	**Length in blocks**
0	0	64000
1	64000	944000
2	168000	840000
3	36000	672000
4	5040000	504000
5	672000	336000
6	840000	168000
7	0	1008000

Suppose files /dev/dsk0, /dev/dsk1, /dev/dsk2 and /dev/dsk3 corresponds to sections 0 through 3 of an RP07 disk and have minor numbers 0 through 3.

Assume that the size of the logical file system block is same as that of a disk block. If the kernel attempts to access block 940 in the file system contained in /dev/dsk3, the disk drive converts the request to access block 336940

(Section 3 starts at block 336000, 336000+940= 336940) on the disk.

The sizes of disk sections vary and administrators configure the file systems in sections of appropriate size. Large file systems go into large sections and so on. Sections may overlap on disk. For example, sections 0 and 1 in the RP07 disk are disjoint but together they cover the entire disk. Section 7 also covers the entire disk. The overlap of sections does not matter, provided that the file system contained in the sections are configured such that they do not overlap.

The use of fixed section restricts the flexibility of disk configuration. The information of disk partitions should not be put into disk drivers. It should be places in a particular table because, if we place those information in the driver, we ca not change the partition size etc.

6.3 Terminal drivers

These drivers are used to control transmission of data to and from terminals. Terminals are user interfaces to the system. To accommodate interactive use of Unix system, terminal drivers contain an internal interface to the line discipline modules, which interpret input and output. In canonical mode, the line discipline converts new data sequences typed at the keyboard to a canonical form (what the user really meant) before sending the data to a receiving process. The line discipline also converts raw output sequences written by a process to a format that the user expects.

In raw mode, the line discipline passes data between processes and the terminal without such conversion.

For example, if programmers typed some incorrect spellings and these can be erased using the erase key. i.e the user can erase the mistakes and enter

corrections. The terminal sends the entire sequence to the machine including erased characters. In canonical mode, the line discipline buffers the data into lines and processes erase characters internally before sending to the reading process.

The functions of a line discipline are

1. To parse input strings into lines
2. To process erases characters
3. To process a “kill” characters that invalidates all characters typed so far on the current line.
4. To write received characters to the terminal
5. To expand output such as tab characters to a sequence of blank spaces
6. To generate signals to processes for terminal hang-ups, line breaks or in response to a user hitting the delete key
7. To allow raw mode that does not interpret special characters such as erase, kill or carriage return

6.4 Streams

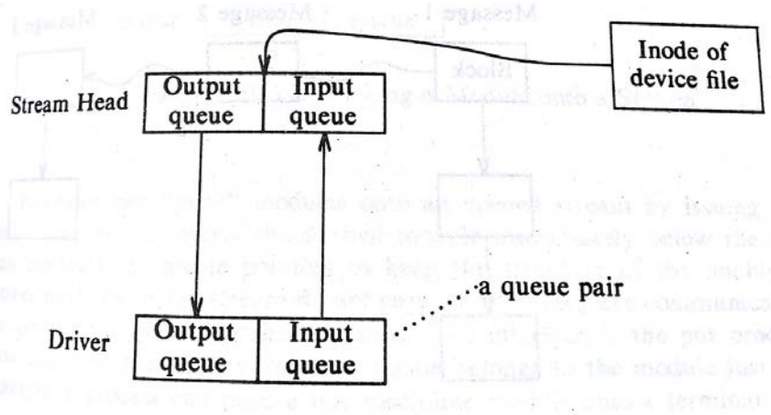
There existed no standard mechanism of communication between processes and device drivers. A new scheme was devised to enable standardized communication between device drivers and processes.

A stream is a full duplex connection between a process and a device driver. It consists of a set of linearly linked queue pairs, one member of each pair for input and other for output. Eventually after several manipulations by queue pairs, the information reaches the required device driver or the process. i.e when a process writes data to a stream, the kernel sends the data down to the output queue. When a device driver receives input data, kernel sends the data up to the input queue to a reading process.

Character device drivers have the above mentioned mechanism in its operation. It is not necessary for all character device drivers to implement this mechanism. Even the kernel uses this scheme, if a character device uses it. If not, the kernel communicates with the character devices using previously mentioned

mechanisms involving the switch table with variable function pointers(such as open, close etc)

For the first open of a stream driver, the kernel allocates 2 pars of queue, one for the stream head and another for the driver.



When a character device special file is opened, the kernel allocates 2 pairs of queues, one for the stream head (means head of the queue) and another for the driver. The inode of the character device special file contains a pointer to the stream head. When data is written to the device file, kernel sends it to the stream head, which passes the data the lower level layers and ultimately data reaches the driver.

6.5 Assignment 6

1) Write a note on driver interfaces?
2) Write a note on disk drivers
3) Write a note on terminal drivers
4) Write a note on streams?

Chapter 7
Interprocess communication

Interprocess communication mechanism allows arbitrary processes to exchange data and synchronize execution. Several forms of Interprocess communication are pipes, named pipes and signals. The other forms of Interprocess communication are

7.1 Process tracing

The UNIX system provides a form of IPC fro tracing processes useful for debugging. A debugger process forks a child to be traced and controls its execution with the **ptrace()** system call, setting and clearing breaks and reading and writing data in its virtual address space.

The debugger forks a child process which invokes the **ptrace()** system call and now execs the program to be traced. When a debugger forks, there are 2 debuggers- parent debugger and a child debugger. Now child debugger calls the **ptrace()** system call and informs the kernel that it has taken birth for tracing a program. Now it execs the program to be traced (i.e it exec(a.out), suppose **a.out** is a program to be traced). At this point the memory space of the child debugger will be replaced by **a.out**

Ptrace()

The ptrace() system call is used to debug a program. The syntax of the **ptrace()** system call is

ptrace(cmd, pid, addr, data);

ptrace() is used to issue a command **cmd** to the process whose process id id **pid**. The command can be for reading or writing data, resuming execution and so on. **Pid** is the process id of the traced process. **Addr** is the address of the memory location to read or write data. Data is an integer value to be written.

7.2 System V IPC

The UNIX system V IPC package consists of 3 mechanisms. They are shared memory, message queues and semaphores. These 3 mechanisms share common properties. They are

1. Each mechanism consists of a table whose entries describe all instances of the mechanism.
2. Each entry contains a numeric key.
3. Each mechanism contains a get() system call. Using this system call, a process can create a new instance of a mechanism or retrieve an existing one.
4. Each IPC entry has a permission structure that includes the Used ID & group ID of the process that created the entry, a user and group ID set by the control system call and a set of read write permissions for user, group and others.
5. Each entry contains other status information such as the process ID of the last process to update the entry and the time of last access or update.
6. Each mechanism contains a "control" system call to query status of an entry, to set status information or to remove entry from the system.

7.2.1 Messages: there are 4 system calls to send a message from one process to another. They are msgget(), msgsnd(), msgctl(), msgrcv().

1. Msgget() this returns the already existed message queue or creates a new one and returns the id of the newly created message queue.

The systax is msgqid= msgget(key, flag); where msgqid is the id of newly created message queue.

2. Msgsnd() the process uses the msgsnd() to send a message.

The syntax is **msgsnd** (msgqid, msg, count, flag);

where msgqid is the id of a message queue returned by the msgget() system call. Msg is a pointer to a character array, count gives the size of the character array

and flag specifies the action the kernel should take if it runs out of internal buffer space.

3. **Msgrcv()** a process receives message from another process using the system call **msgrcv()**.

   The syntax is **count = msgrcv(id, msg, maxcount, type, flag);**

where id is the message descriptor, **msg** is the address of the structure to contain received message, **maxcount** indicates that process is ready to accept maximum of **maxcount** data. **Type** is the message type the user wants to read and flag specifies what kernel does if no messages are on the queue.

4. **Msgctl()** this is used to control the message queue. Using this system call, the kernel issues command to the message queue.

   The syntax is **msgctl(id, cmd, mstatbuf);**

   where **id** is identifies the message descriptor, **cmd** specifies the type of command and **mstatbuf** is the address of the data structure that will contain the result of a query.

7.2.2 Shared memory

Using this mechanism 2 or more processes exchange messages using same memory. The important system calls in this mechanism are

1. **shmget();** creates a new region of shared memory or returns the shared memory id. The syntax is **shmid=shmget(key, size, flag);** where size is the number of bytes in the region.

2. **shmat():** it attaches a particular shared memory segment to a process so that it can be used by the process. The syntax is **vitraddr=shmat(id, addr, flag);**

3. **shmdt():** this detaches a shared memory region from the process. The syntax is **shmdt(addr);**

4. **shmctl():** this is used to control shared memory regions. The syntax is **shmctl(id, cmd, shmstatbuf);**

7.2.3 Semaphores

This mechanism is slightly different from another 2 mechanisms. This system is not directly used to pass messages. This allows processes to synchronize execution by doing a set of operations. The important system calls come under this are

1. **Semget():** creates an array of semaphores. The syntax is **id=semget(key,count,flag);**

2. **Semop():** using this, processes s manipulate semaphores. The syntax is **oldval=semop(id, oplist, count);**

3. **Semctl():** this is used to control semaphores. The syntax is **semctl(id, number, cmd, arg);**

7.3 Sockets

To provide common methods for Interprocess communication and to allow use of sophisticated network protocols, the BSD system provides a mechanism known as sockets. The kernel structure consists of 3 parts: the socket layer, the protocol layer and the device layer. The socket layer provides the interface between the system calls and the lower layers. The protocol layer contains the protocol modules for communication (TCP/IP) and the device layer contains the device drivers that control the network devices.

Processes communicate using client-server model; a server process listens to a socket, one end point of a two way communication path, and the client process

communicate to the server process over another socket, the other end point of the communication path, which may be on another machine.

The 4.2 BSD system supports the UNIX system domain for communicating on one machine and the Internet domain for process communicating across a network.

Each socket has a type- a stream socket or datagram. The stream socket allows sequeneced reliable delivery of data . Datagram do not guarantee sequenced, reliable delivery. But they are less expensive than stream sockets. The TCP provides stream socket service and user datagram protocol provides datagram service in the Internet domain. The socket mechanism consists of several system calls.

Socket()

The socket system call establishes the end point of a communication link. The syntax is

Sd=socket(format, type, protocol);

Format specifies the communication domain (the UNIX system domain or the Internet domain), type specifies the type of communication over socket and protocol indicates particular protocol to control the communication. Processes use the socket descriptor sd in other system calls.

The close() system call closes sockets.

Bind()

The bind system call associates a name with the socket descriptor. The syntax is

Bind(sd, address, length);

Sd is the socket descriptor and the address points to a structure that specifies an identifier specific to the communications domain and protocol indicates a

particular protocol to control the communication. Length is the length of the address structure.

Connect

The connect system call requests that the kernel make a connection to an existing socket. The syntax is

Connect(sd, address, length);

Sd is the socket descriptor, address is the address of target socket that will form the other end of communications line. Both sockets must use the same communications domain and protocol. If the type of the socket is a datagram, the connect call inform the kernel of the address to be used on subsequent send calls over the socket.

Listen()

When a server process arranges to accept connections over a virtual circuit, the kernel must queue incoming requests until it can service them. The listen system call specifies the maximum queue length

Listen(sd, qlength);

Where sd is the socket descriptor and qlength is the maximum number of outstanding request.

Accept()

The accept system call receives incoming requests for a connection to a server process. Syntax is

Nsd= accept(sd, address, addrlen);

Where sd is the socket dercriptor, address points to a user data array that the kernel fills with the return address of the connecting client, and addrlen indicates the size of the user array. Accept returns a new socket descriptor nsd, different from the socket descriptor sd.

The send and recv system calls transmit data over a connected socket
Count()

Count=send(sd, msg, length, flags);

Where sd is the socket descriptor, mwsg is a pointer to the data being sent, length is its length, and count is the number of bytes actually sent. The flags parameter may be set to the value SOF_OOB to send data “out-of-band” meaning that data being sent is not considered part of the regular sequence of data exchange between the communicating process.

Shutdown()

The shutdown() system call closes a socket connection.
Shutdown(sd, mode);

Where mode indicates whether the sending side, the receiving side or both sides no longer allow transmission.

7.4 Assignment 7

1) What is process tracing?
2) Explain system V IPC.
3) Write a note on messages. What are the system calls for messages?
4) With syntax give any 4 system calls for shared memory.
5) What is a socket? What are the system calls for sockets?

UNIT III
Chapter 8
An Introduction to LINUX

8.1 What is Linux?

Linux has features that are comparable or better than other proprietary operating systems. With the availability of several advanced applications based on Linux, the use of this operating system has increased. More and more applications are being deployed in Linux platform in this era of burgeoning IT growth. Stability, support, increased availability of skills and resource pools, and the application maturity, along with lower cost of ownership make this operating system acceptable in an increased level in different sectors.

8.2 About Operating Systems

Computer is an electronic device capable of performing several computing and mathematical operations. Computers are also used for playing games, viewing pictures and photos, watching movies and for hearing audios. Browsing the Internet, sending and receiving mails, chatting, exchanging ideas and socializing are also now possible with the use of computers. And, computers need some basic software or program to do all these activities. These types of basic software that are needed to control the different activities are called system software or operating systems. System software makes the working of computers easy. System software also control the operations of the CD drive, enable the use of modems and other peripherals connected to the computer. They help computer users to create files and folders, copy files and run applications. In other words, operating systems perform different management operations such as file management, memory management, device management and process management.

Several types of operating systems have evolved through the past several years. Some of the classifications of operating systems are real time operating systems, server operating systems, single user operating systems, multi user operating systems and so on. Operating systems are now used also in mobile phones, PDAs

(Personal Digital Assistants), music players, game consoles, digital cameras and several other gadgets. In these devices operating system software are embedded in the hardware. Unix, Windows, Solaris, Mac are some of the operating systems currently available for desktop systems. Linux is an operating system used in a variety of hardware platforms.

8.3 Free and Open Source Software

Open source, freeware and free software are now available for anyone and for any use. The demands of consumers as well as enterprises are ever increasing with the increase of IT usage, it is quite natural that a single solution provider cannot produce all the needed solution, due to this, IT environment of today have different solutions from different vendors.

Freeware and free software are different from each other. **Freeware** software is a type of software that is available free of cost and these can be easily distributed without any restrictions. There are no license agreements for freeware software. The definition of free software is given by the Free Software Foundation (FSF). But in free software, the meaning of the word free is different and it does not mean that the software is available at zero cost.

- Free software user is provided with the complete freedom to run the software for any purpose,
- freedom to study the working of the software,
- freedom for adapting the software to different needs,
- freedom to redistribute copies,
- freedom to improve the software and for releasing improved versions of the software.

To improve the software and to adapt it to different needs, the source code of the free software must be easily available. Any software can be treated as free software, if the users are provided with all the above freedoms.

In the initial period all the software was available as free software. Earlier computer users were scientists and developers, and they did not consider the software as a business asset. So there were no restrictions in copying the software or modifying it. But the main difficulty was the lack of standardization

between different software. As a result, software worked in a system based on a particular architecture could not run in another machine having a different architecture. In order to overcome the difficulties associated with the restrictions in using different software, in 1983, Richard Matthew Stallman started the **GNU** or **Gnu Not Unix** Project. This project was aimed to spread the message of using free software and to develop a free operating system similar to but different from the Unix operating system. In continuation of this project, later in 1985, Free Software Foundation (FSF) was formed. The face of the African animal gnu was chosen as the logo (see the figure) of this project.

With the activities of the Free Software Foundation and the GNU project, several types of free software were made available. The GNU text editor Emacs and the GNU C compiler GCC were the very successful initial software. Two other notable developments were the, Berkley Software Distribution (BSD), which is a reimplementation of the Unix operating system and the X Window system, a free, network transparent graphical computing environment. The work progressed and by the year 1990 many of the components of a free operating system named GNU operating system, except the kernel, were developed. Development of the kernel—called as Hurd——was found to be very difficult. In 1992 Linux kernel was made available under the GNU General Public License (GPL). GNU operating system used the Linux kernel and developed a complete operating system and in this way the free operating system GNU-Linux was created.

Open source software means that the source code of the software is openly available. This is contrary to proprietary software, where the source code is kept secret. Open source software license provides certain license to end users. This

license provides the right to access the source code of the software, make copies of the software and distribute it and make improvement to the software. So anyone can distribute or sell the software. Also anybody can modify the software to suit his needs, and can provide support to users for the software. Open source software is developed collectively and hence these are transparent due to which bug fixing is easy. These types of software are continuously modified and improved by adding new features and facilities. As a result, these types of software also have a better quality

Parallel software development and debugging are other notable features of open source software development process. Linux happens to be the first name that comes to people when speaking about free and open source software. But Linux is not the only software that is available in this category. **Gedit** is a lightweight free text editor and can be found in many Linux distributions, BSD and Unix systems. This editor software supports most standard editing features and has a powerful plug-in system for adding new features. For graphic based applications, one of the most commonly used applications is **Gimp**. Several image manipulations can be done using this software. Gimp is free and is available in different versions for different operating systems.

Internet applications include applications for browsing the Internet, mail applications and messaging services. Mozilla Firefox, Mozilla Thunderbird, Iceweasel, Gaim, Evolution are some of the commonly used applications for different Internet based activities. MYSQL is an open source database application available for different platforms. Besides these, there are several other open source software that are used as firewalls, IDS and so on. These are free software and are available in different versions suitable for different operating systems. An absolutely free, open, feature-rich multi-platform office productivity suite is **OpenOffice.org suite**. The user interface and the different functionality of this suite are very similar to other commercial products like Microsoft Office. Python is an easy to learn and a powerful programming language. It has a high-level data structure and a simple but effective approach for object-oriented programming.

This is an ideal language for scripting and rapid application development in many areas on most platforms. **Apache** web server is an open source software that has emerged as the most popular web server.

8.4 Origin of Linux

DOS was the once dominated computer operating system. But the source code of DOS was not made available by the developers of the operating system. Along with DOS, another operating system available was Unix. Unix was developed in 1969 and was developed using C programming language. Unix was different from other operating systems available at that time and Unix offered several advantages. Unix made it possible for different software to work on any hardware. Unix could work on a wide range of machines irrespective of the architecture or configuration. Another feature of Unix operating system was the ability for networking. This ability helped computers to interconnect themselves as well as to connect to the Internet. Due to these added features several derivatives of this operating system such as Sun OS Solaris, BSD (Berkely Software Distribution) and so on were soon developed. BSD was released as free and open source software and the license allowed the use of the operating system for any commercial purpose.

The increased price of Unix and the difficult user interfaces kept Unix away from users. Efforts to develop a low cost operating system progressed and this led to the development of **Minix**. **Andrew S Tanenbaum**, a Dutch Professor, wrote this operating system to teach the students about the inner working of operating systems and the source code was freely distributed. **Minix** was developed for educational use and was released under BSD license. This operating system was written in C programming language and assembly language. Thus, for the first time, the code for an operating system was made available. Along with Minix, another operating system released was **GNU** (Gnu's Not Linux). Both these operating systems were similar to **Unix** but used no code from Unix. This was the time when developers were actively involved in discussions related to the

development of open source software. In **1991 Linus Trovalds**, a student of Helsinki University developed an initial operating system and posted to newsgroups for feedbacks and discussions.

Actually this was the initial version of the kernel used for the development of Linux operating system. With some modification, based on the discussions at newsgroups, an initial version of **Linux 0.01** was released in mid September and was put on the Net. This was the beginning of the evolution of Linux operating system. Depending on user requirements new versions of Linux operating system were released from time to time. The main difficulty with the different versions is that these versions provided support for only a limited number of hardware components, such as hard disks, keyboards or monitor types. Soon, the different versions were made available through different ftp sites. The efforts made by several developers led to the release of **Linux** for actual use in computers.

Linus Trovalds later came to be known as the father of Linux operating system. He adopted the penguin nicknamed as **Tux** as the Linux mascot (see Figure 1.2). Later, Linux community adopted Tux as the mascot of Linux kernel and majority of people associate Linux with Tux. Tux can be seen in Linux related websites and along with Linux products and applications. Soon Linux was included under GNU General Public License (GPL) making the source code freely available for copying, studying as well as for editing purposes. GNU operating system used this Linux kernel to develop a complete operating system and thus the free operating system GNU/ Linux was formed.

After this, several commercial vendors entered the scene for developing advanced operating systems. With the introduction of graphical interfaces such as KDE and X-Window, the popularity of Linux operating system also increased. **Red Hat** and Debian **were** the first two vendors who entered the scene for distributing Linux.

8.5 Linux Kernel

- The central nervous system of Linux operating system is the kernel.
- The kernel is responsible for running the entire system.
- It is responsible for all the basic functions performed by the operating system.
- Kernel manages memory, controls different devices, execute commands, handles different errors and so on.
- The kernel is under constant revision and development by the Linux community. With every revision, new and new features are incorporated to the Linux kernel.
- Thousands of lines of codes are modified and tested before a new stable version of Linux is released for use.
- A full featured Linux kernel that provided the abilities for the management of different resources of the system was Linux 2.4. Later versions added several new features to this initial Linux version.
- Linux kernel is developed in a modular architecture. **Modular architecture** means that the kernel is built using several modules or subsystems.
- Basically there will be a core that is made up of the minimum code for running the system.

- This kernel performs different functions of the operating system such as device management, memory management, process management and file management with the support of the core.
- Basically Linux kernel is made up of several device drivers.
- Device drivers are file systems through which users and programs interact with the hardware and other connected devices.
- Basically a tree like architecture is followed in Linux operating system for different device drivers.
- This type of architecture separates drivers for different platforms and for different devices.
- Interrupt handling functions performed by the kernel helps to control different hardware systems. This is done by setting priorities for interrupts, enabling and disabling interrupts and scheduling interrupts.
- Bus architecture of computers is responsible for the efficient working of internal as well as external devices connected to them.
- Commonly used bus architectures are known by names such as ISA, PCI, MCA, USB and so on. Several devices such as keyboards, mouse, speakers, printers, scanners and so on can be connected to the system using USB (Universal Serial Bus) and Linux supports this architecture efficiently.
- Also a number of I/O devices are supported by Linux kernel. Different devices that are supported include IDE, SCSI and SATA disk drives.
- Newer versions provide support for increased numbers of controllers.
- Different floppy disks, tapes, DVDs and CDs are also supported by Linux kernel.
- Linux kernel also supports serial ports and parallel ports, modems, sound cards, TV and radio tuners and a variety of video cards.
- Support for different file systems suitable for different platforms is another feature provided by Linux kernels.
- Different systems use different types of file systems namely FAT, FAT32, N TFS, VFAT, HPPS and so on.
- Support for different partition tables is another feature offered by Linux.

- Several processes and threads can be created cheaply and easily using the Linux kernel.
- Also the kernel helps to share different files, memory and processes. Communication between processes and applications are also done efficiently by the kernel.
- A complete implementation of networking protocols such as TCP, IP, UDP, NetBEUI and so on are included with the kernel.
- Linux kernel supports several new networking devices and offers several networking functions.
- Firewalls, packet filters and the like have become common components of Linux kernels. These features help the easy integration of new as well as old components in networks.
- Protocols for messaging, mail operations as well as other common Internet services were included from earlier versions of Linux.
- Availability of documentation is another feature of Linux kernel. Detailed documentation files provide necessary support for running the operating system.

8.6 Linux Features

Linux operating system is based on Unix and its several features are adapted from Unix. Linux is compatible with the new popular industry trends such as virtualization and cloud computing. Several popular applications including BI, CRM, ERP, database and middleware are now deployed in Linux platform. Linux has emerged as a low cost alternative for different uses. Different features of Linux operating system can be explained under different groups as follows:

8.6.1 Free and Inexpensive

Linux versions are available as free distributions and can be downloaded free from the Internet. Paid distributions of Linux are available at reduced prices. Linux is available under General Public License (GPL), which requires free distribution of the source code. All patches and updates are also available under this license. This license provides freedom to copy and modify the software without any restrictions. Also the modified versions are distributed under the same license as the original one. Besides these advantages, several applications working in Linux

operating system are available either as free or under GPL. Such Linux applications can be downloaded from different Internet sites. Updates of several applications can also be downloaded and can be used free.

8.6.2 Stable and Flexible

Linux is flexible, as it is an open source operating system. Different companies test Linux to ensure hardware compatibility and Linux is found to be stable in several hardware components. Also, the security of Linux is better when compared to other operating systems. Virus attacks in Linux are less when compared to proprietary operating systems. Virus propagation mainly takes place through pirated copies. For Windows operating system, innumerable numbers of security software are available. But in the case of Linux it is not so. This difference is because several user groups are constantly updating Linux software by adding new features and facilities. Security holes are fixed easily during the updating processes. Moreover, the kernel and file systems of Linux are derived from the secure Unix operating system. Hence the security offered by Linux is strong. Besides these, Linux distributions are now made available with a default antivirus software for scanning the system and for detection and removal of viruses from the system.

8.6.3 Fast and Reliable

Linux operating system is fast and so it can manage different resources efficiently and effectively. One reason for the increased speed is the low memory requirement due to its modular architecture. Linux has the ability to add new modules as and when required and to release the resources after the use. The backup option provided by several Linux distributions offers facilities for simple backing up in an easy manner. By setting different options, the system backup can be set to take place automatically in different styles such as daily backup, weekly backup, incremental backup, full backup and so on.

8.6.4 Easier to Use

Graphical interfaces in Linux offers several facilities for ease of use. A number of graphical interfaces, each having a different layout, can be chosen by the user for

installation. To make the interface more attractive, options are available for customization. For quick access, applications can be placed as shortcuts in the desktop or can be added to the panel. Drag and drop functionality, point and click facility are some of the features of graphic user interfaces. Forward and backward buttons help to navigate through forward and backward windows at ease. Using the single click option, files and folders can be opened by single clicking of the mouse. Different operations including the complex tasks of system administration can be done using a number of mouse clicks, when using graphic interfaces. Linux installs most of the necessary drivers during the time of its initialization and hence new devices can be added easily. So restarting of the computer after installing device drivers is not necessary.

8.6.5 Robust Programming and Application Environments

Highly robust and capable programming environments are available in Linux. Different software are available as free and are covered under the GNU Public License. Compilers for C, C++ languages, assembler, Perl, Python are available in Linux operating system. Scripts in Linux operating system are similar to batch files in Windows systems, but they perform better. Office productivity tools, e-mail solutions, Web applications, database systems and middleware applications are some of the common applications bundled along with Linux operating system. Typically, Web servers, FTP servers, mail servers, disk and print servers are available as bundled applications along with Linux operating system.

8.6.6 Built-in Networking Ability and Easy Access to the Internet

Another important feature of Linux operating system is the built-in networking ability. Networking allows the sharing of files and devices connected in networks. Also, a single Internet connection can be shared among different networked computers. The Internet is based on Unix and the different protocols used in the Internet are developed from Unix operating system. Naturally Internet connectivity and networking of systems can be done easily when using Linux. It is to be noted that majority of web servers are running under Linux operating system. Most Linux distributions are provided with a web browser for surfing the Net.

8.6.7 Increased Returns

With Linux, getting maximum ROI (Return on Investments) is possible in an easy manner, as Linux has proved to be very cost-effective and easily maintainable. Linux runs speedily and efficiently on several architecture in which Windows operating systems cannot run. Installation of newer versions of Windows operating system requires new machines with increased power and large memory. RAM, CPU, disk, audio, video and the like are also obligatory for installing new Windows versions. But this is not the case with Linux. Requirements for Linux operating system are less and hence this system can work effectively in older machines. Moreover several low cost solutions are now available for Linux operating systems. Hence Linux is found to be more affordable.

8.6.8 Multi User and Multi Tasking Ability

Linux is a true multi user and multi tasking operating system. Unix operating system from which Linux was developed is a multi tasking and multi user operating system. This ability helps several users to work simultaneously as well as to perform several tasks at the same time, so also, when using Linux operating system.

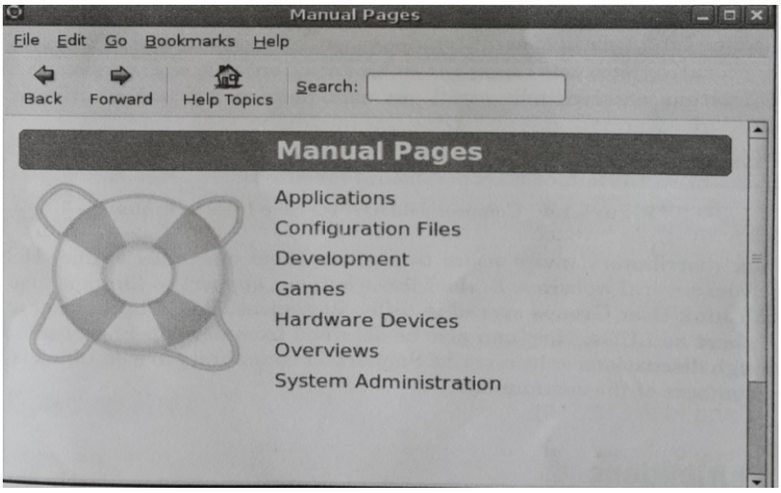
8.6.9 Entertaining and Educative

Several applications available for Linux operating system help in the running of audio, video as well as other multimedia files. Support for CD, DVD and other media help to play high quality digital audio and video files in Linux. Support for TV and radio tuner cards help to watch television shows as well as to hear radio broadcasts. Now several games are installed by default, while installing Linux operating system. These games make Linux more entertaining. Availability of several educational applications suited to students of different groups as well as different standards is another feature of Linux.

8.6.10 Readily Available Help

Help option can be seen included with several Linux versions. **Exhaustive manual pages** are provided with several new releases. These manual pages are divided

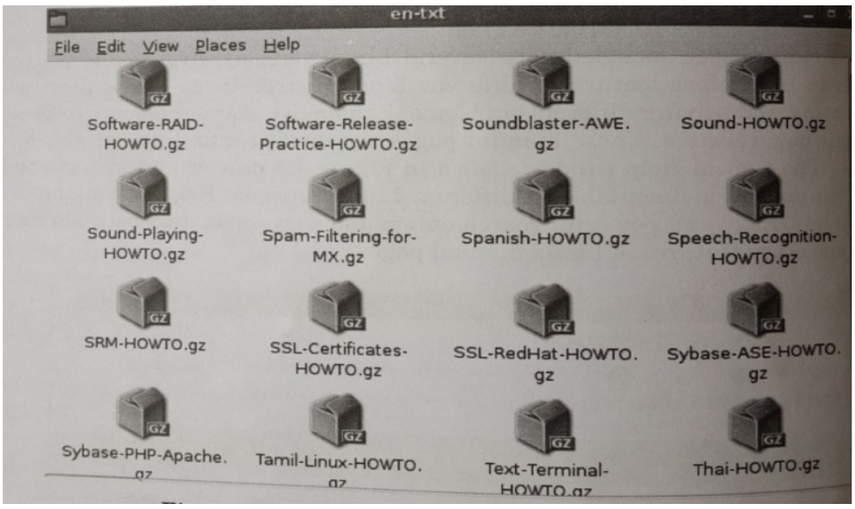
into different sections for ease of use. The typical Help window shown in Figure 1.3 provides an idea of the help facility in manual forms available in different Linux versions.



Help for different applications are also available by clicking the Help menu available in the menu bar of different Linux applications. Clicking the menu displays a window showing the list of headings available. Clicking the heading opens the linked file having the necessary explanations for the selected item. Also, majority of Linux applications are provided with an instant pop up help tip window when the cursor is placed above buttons or icons. Another form of help available in Linux operating system is the presence of several HOWTO files. These files are available for almost all topics and these files are stored in compressed forms. Extracting the compressed files to a folder makes the files readable.

The figure below shows a set of such help files available with a Linux distribution. Apart from the built-in help option, commercial support is also offered by several vendors as well as support enterprises. Different Linux distributors provide commercial support to Linux in different ways such as network support (software and updates), basic support (24x7 global bases) and premier support (life time). To provide speedy support, some Linux distributors invite users to register their

machines online. Help is also available from several websites. Besides these options, support to Linux is also provided by several Linux User Groups spread in different regions. These Linux user groups are known in short as LUGs. Help can also be obtained from developers spread around the world through discussions or by e-mails. Support for community driven Linux is available from the members of the community.



8.7 Linux Distributions

Linux was first made available under the GNU General Public License and this operating system was distributed along with the source code. Several communities modified the source code for different needs. This led to the release of different Linux versions. Different versions of Linux are known in general as Linux distributions or Linux distros, in short. **Distribution** is a specific edition of Linux. Distribution is made up of the Linux kernel, windows management system, desktop environment and a set of software.

Basically Linux distributions are available in two types. One is the paid type and the other is the free distribution type. **Paid type** distribution is also known as

commercial distribution while the free type is known as community version. For mission critical applications paid Linux distributions are preferred. Some vendors release the commercial version along with the community version. Red Hat Linux and Novell's SUSE Linux are examples of commercial distributions. Debian and Fedora are examples of free Linux distributions. CentOS is an example of a community compiled enterprise Linux distribution. Red Hat Linux and Fedora are released by the same vendor with different terms and conditions. Open SUSE and commercial SUSE are the two different types of Linux released by the same vendor. Releasing the same software with different sets of terms and conditions is called dual licensing. Depending on the nature of application, -Linux distributions are mainly divided into two versions namely **desktop** version and **server** version.

8.8 Linux Opportunities

This operating system is mainly used for operations like web hosting, software development, device embedding and so on. With the addition of increased technical features such as virtualization, real time computing and the like more and more applications are now deployed in Linux. For developing mobile applications also, Linux is widely used. To satisfy the variety of requirements, Linux is available in different platforms such as desktops, servers, mainframes, mobile systems and so on. It provides support for new devices and new hardware easily. Software development in Linux platform is done using languages such as C++, Java, Perl, Python, and so on. Now websites are developed using LAMP (Linux, Apache, MYSQL and Python) in Linux platform and hence the requirement of LAMP professionals is high. Developing device drivers for devices and new peripheral devices is another area of opportunity for Linux professionals.

8.9 Assignment 8

1) What are free and open source software?
2) Write a note on Linux Kernel
3) What are the Linux features? Explain in brief.
4) Write a note on Linux Distributions

Chapter 9
Linux Distribution and installation

9.1 About Linux Distributions

Different Linux distributions are evolved one after the other and are derived from the same kernel. Depending on the type of installation files of the distribution, Linux distributions are divided into two groups namely **rpm based distributions** and **deb (Debian) based distributions**. Red Hat Linux and its derivatives are classified as rpm based distributions. Some of the distributions that can be included under this group are Fedora, Mandriva, CentOS, SUSE and the like. Debian distributions and their derivatives namely Ubuntu, Knoppix, Damn Small Linux, Linux Mint belong to the second group. The working of the system as well as the arrangement of files for both types of distributions is identical.

9.1.1 RPM Based Distributions

Red Hat Linux operating system is the Linux version distributed by Red Hat. This is available as desktop and server versions. This operating system can work in different types of computers such as desktops, servers as well as in computers having different architectures and processors. Fedora Linux is an operating system developed under Fedora project sponsored by Red Hat. This is a robust and matured operating system. The installer of Fedora Linux is known as **Anaconda**. New versions are released with added features. New release of this distribution comes with different desktop environments namely GNOME, KDE, LXDE and so on.

Another distribution based on rpm is **Mandriva Linux**. Earlier, this distribution was known as **Mandrake Linux**. This is also based on Red Hat and is commercially backed. It is one of the most common and popular Linux distributions.

SUSE Linux is a commercial Linux version and is very secure. Free version of this distribution is known as **OpenSUSE** Linux. This is a common and popular operating system and is rpm based.

9.1.2 Deb Based Distributions

Debian-GNU Linux is a comprehensive and non-commercial Linux distribution. Debian is a volunteer based Linux distribution. This Linux version supports systems of different architecture. It has two desktop environments—Gnu Network Object Model Environment (**GNOME)** and K Desktop Environment **(KDE).** Several distributions are derived from Debian. This distribution has regular new releases. Ubuntu Linux is an easy to use Linux version.

Ubuntu operating system has two versions—the desktop version and the server version. This operating system is derived from Debian Linux. Ubuntu is an African word and it means humanity to others. Two types of graphic user interfaces namely GNOME and KDE are available for Ubuntu. The version in which K desktop environment is used is known as Kubuntu.

Kubuntu is also an African word and it means towards humanity. Knoppix Linux is one of the first distributions that came with a live CD distribution. Live CD can boot from itself and operate through RAM. This is one of the most popular Linux distributions. This is a deb based operating system. **Damn Small Linux** is a small Linux distribution having a small size of just 50 MB. The distribution is freely available and the latest version can be downloaded from the web. This distribution can also run from a CD and from within the Windows operating system. Firefox web browser is available in this distribution. Other applications available in DS Linux are Sylpheed—the e-mail program similar to Outlook Express in the Windows platform, XMMS (X Multimedia System)—cross platform multimedia player, text editors such as Beaver and Vim, graphics application Xpaint. Office productivity tools such as PDF viewer, calculator, calendar, and dictionary and so on are also available.

9.2 Assignment 9

1) Write a note on RPM based distributions

2) Write a note on Deb based distributions

Chapter 10
Linux Desktop Environment

10.1 X Window System

The X Window System is a simple and complete system used for building graphical user interfaces on most operating systems. X window system provides the basic framework for building the GUI environments such as drawing and moving windows, interacting with the mouse and keyboard and so on. Several user interfaces are created based on this environment for different client programs. Now several different graphical interfaces such as GNOME, KDE and so on for Linux environment have become popular. **Moblin** environment designed for mobile desktop platforms is also common now. **X window system** was originated in 1984 and it was designed for working in network connections. **Unix** based systems use the X window system for graphic purposes. The X window system is based on client—server architecture and this architecture has become popular.

10.2 Graphical Interfaces

The earlier graphical interfaces used for **Unix** systems were neither powerful nor user friendly. When Linux was released with open source code, a number of graphical interfaces were developed. X window and Motif were the two initial graphical interfaces used for Linux systems. Major abilities of the currently used graphical interfaces were absent in these earlier systems, which lacked features such as drag and drop support, networking facility, content wise help etc. Efforts were taken by developers to create a content rich and user friendly graphical interface for Linux. This necessity led to the formation of K desktop environment foundation or KDE foundation. Efforts by this foundation led to the development of a free software. This software system came to be known as KDE.

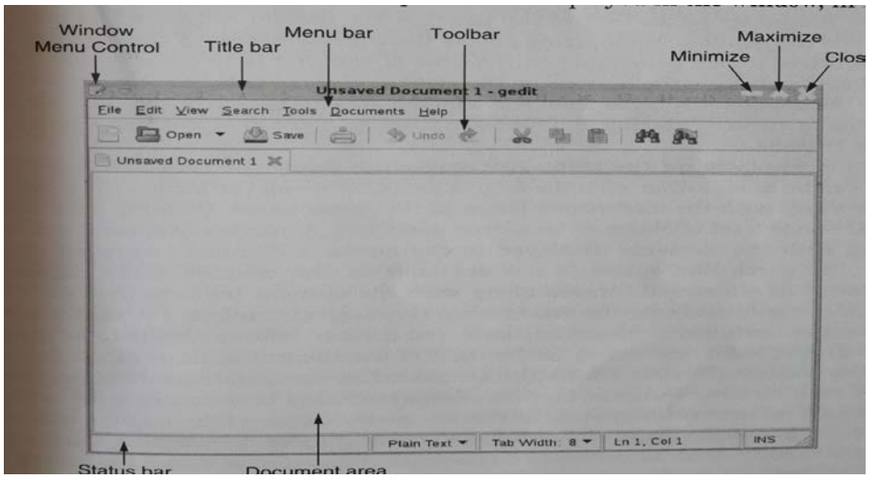
Meanwhile another set of developers actively developed another graphical interface to the Linux system, which came to be known as GNOME (Gnu Network Object Model Environment). KDE is based on Qt graphical library while GNOME is based on GTK library. Both GNOME and KDE systems have the same components.

Applications designed for GNOME run on KDE and vice-versa. The difference lies in the layout and arrangement of the different components. KDE and GNOME can be seen not only in Linux but in almost all Unix based operating systems such as OpenBSD, Tru64, AIX, Solaris and so on. The initial versions of KDE and GNOME were modified several times and new versions were released. Each version came out with new features and added applications. In course of time some other graphic interface systems for Linux such as XFCE (X Forms Common Environment), XPDE (XP Desktop Environment) were also developed. At present, GNOME and KDE are the two widely used graphic interfaces for Linux desktops.

The basic graphical Linux environment is made up of the desktop, icons, menus and windows. Desktop is the name given to the overall working area and is an active component of using computers. in the Linux working environment, different applications, tools, files and other items are arranged systematically on the desktop. Desktop helps to move items and organize different activities. It is possible for users to place frequently used files and folders on the desktop for easy access. Desktop can also be used for opening programs, copying files, connecting to the Internet, using the Internet service, playing games and so on.

Icons are the small figures or symbols displayed on the desktop. Usually icons have their names displayed below them. Icons can represent anything such as a document, application or folder. Different shortcuts can be placed as icons on the desktop for quick access. By default several icons are displayed on the desktop.

Menus are the set of elements displayed, from which the user can select the required element. The elements in the menu list are displayed when the menu element at the top of the list is clicked. Basically, menus are of different types and are known in different names such as pop-up menu or drop-down menu. Pop-up menu is the menu that pops up on the screen when some events take place. The menu that is displayed when the mouse button is right clicked anywhere on the desktop is the pop-up menu. Some pop-up menus appear anywhere on the screen, whereas some others are displayed only in specific places on the screen. Drop-down menus appear as arranged in a hierarchical order.



Another essential part of the Linux graphical desktop environment is the window. Window is a rectangular area having borders on all its four sides. Windows are the activity areas of the application or documents. Different Linux applications run in windows. Windows in which applications run are the application windows and those windows in which documents are opened are the document windows. Another type of window, called dialog window appears on the request of application window. The job of dialog windows is to give alert to problems, ask for confirmation of action, or request input. Some dialogs do not allow interacting with the main application window until the dialogs are closed. These windows are called modal dialogs. Windows that can be left open while working with the main application window are called transient dialog windows. Windows have a title bar which displays the name of the document that is opened or the application that is working. In the case of application window, the name of the application is displayed along with the name of the file in the title bar. Usually in almost all windows, four buttons can be seen in the title bar. The button on the left extreme is the Window Menu Control button. Clicking this buttons opens up a number of options. On the right extreme three buttons are visible. These buttons are used to minimize, maximize and close the window. Minimize button is indicated by a thick

line. Maximize button is indicated by an upward arrow symbol. Close button is indicated by a cross symbol. Clicking these buttons minimizes, maximizes or closes the window. When the window is minimized it appears as an icon at the panel at the bottom. Clicking the maximize button expands the window. Maximizing makes the window to occupy the entire desktop or the entire active region depending on the area of working of the application.

A **menu** is displayed in the menu bar appearing below the title bar. Selection of menu items can be made either with the help of mouse or using the keyboard by pressing the ALT key along with the underlined letter in the menu name. Clicking each item in the menu bar opens a set of elements as a drop down list. A dimmed or greyed element means that the particular option is not available at the current time. Three dots, commonly called as ellipses, displayed along with the element indicate that additional information is needed to execute the action when the element is selected. Usually a dialog box appears when such elements are selected and further information is to be given to the dialog box. A triangle appearing to the right of the element indicates that there are additional sub choices associated with the element. Key combinations displayed on the right side of each element indicate the list of shortcut keys that can be used with the keyboard instead of using the mouse. In certain menu elements, by using check mark indicators to the left of the menu element helps in selecting or de-selecting the element. Help option is commonly available in the menu bar in almost all windows.

Toolbar is the bar appearing under the menubar. It contains buttons for the most commonly used commands. Clicking the button executes the command.

Status bar lies at the bottom of the window and this bar provides information about the current state of viewing.

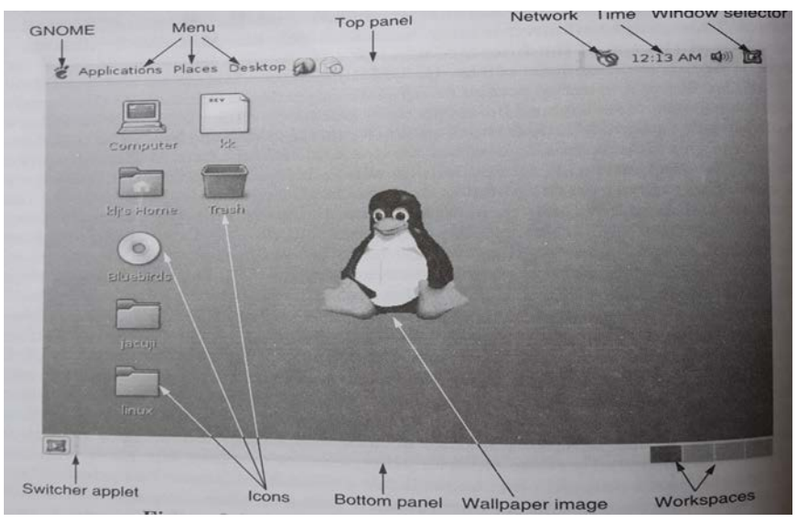
Scroll bars and scroll buttons appear on the right and bottom borders of the window, above the status bar, when there is too much of information to be displayed within the opened window. Scroll bars are the small bars which enable

the user to view the entire contents in the window by dragging it. Vertical as well as horizontal scroll bars help to move the contents either in the vertical or horizontal directions. Linux allows the opening of more than one window at a time. Opening several windows at the same time makes each window to appear one above the other. But only one window can be active at a time. The active window is the window lying at the top of all other windows. Active window is indicated by a colored title bar, distinct from the title bar of other windows.

Now we will discuss the details of the two common Linux desktop environments GNOME and KDE.

10.3 GNOME Desktop

GNOME is the most popular and widely used Linux graphical desktop environment. The different components of the GNOME desktop startup screen are shown in the figure below.



The default start-up screen of GNOME is made up of panels and several icons. GNOME desktop area has top and bottom panels as its boundaries and by default, these two panels appear in the GNOME desktop. Panel is a rectangular area in the

GNOME desktop and the panels provide access to certain actions and information. Different applets are visible in panels. Applet is a small application whose user interface resides within a panel. Another object appearing on the panel is the launcher. Launchers perform specific actions when activated. Launchers are used to start applications, execute commands or open folders. The top panel holds the main menu. By default, date and time indicators, system sound indicator, volume indicator, network enable indicator appear on the top panel and launcher will be seen at the bottom panel.

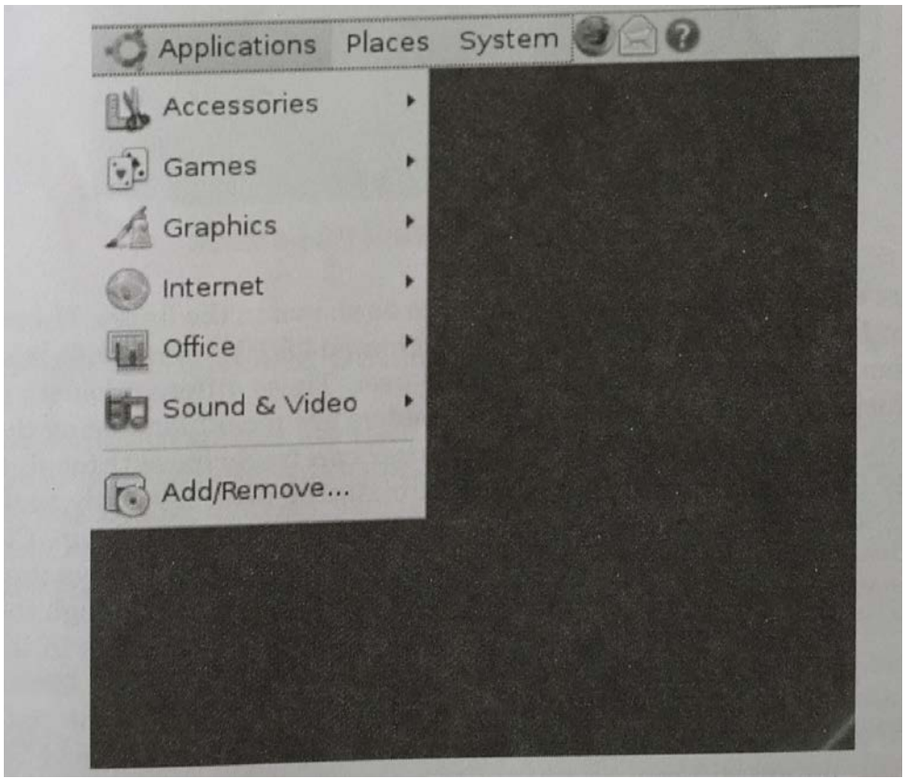
In Linux, every user has a Home folder created while adding the user. The Home folder is placed on the desktop and this Home folder is intended to store all the files of the user. Home folder is displayed on the desktop using a special icon. Special icons for some other purposes are also available in the Linux desktop. The icon Computer available on the desktop gives access to the entire file system, different drives and other removable media. Trash is a special folder in which deleted objects like files and folders are stored. When a file is deleted, it is not permanently removed from the computer memory, but it is moved to Trash folder. If required, such deleted objects available in Trash folder can be restored. Deleting from Trash folder makes the objects permanently removed from computer memory. When a removable media such as CD, flash drive or devices containing files like music player is inserted in the drive, an icon representing the device will appear on the desktop. A file manager application always runs behind in GNOME. The file manager used by several Linux distributions is **Nautilus File Manager**. To open a new Nautilus window, double-click on an appropriate icon on the desktop, or choose an item from Places menu on the top panel.

Another part of the GNOME desktop is the workspace. Workspaces can be considered as virtual screens, which can be switched between, at any time. It is possible to subdivide the desktop into separate workspaces. Each workspace can contain several windows and this helps in the grouping of related tasks together. Every workspace contains the same desktop, the same panels, and the same menus. However, it is possible to run different applications and open different

windows in each workspace. The applications in each workspace will remain there even when switched to other workspaces. By default, four workspaces are available. More workspaces can be created by right clicking on the centre of the workspace and selecting the Preferences option from the pop-up window. This step opens the workspace switcher preferences window, and proceeding further through different options, additional workspaces can be created. Switching between workspaces is done by clicking the switcher applets available at the bottom panel.

The main point of access to GNOME is through the panel menu bar. The basic menu items appearing in the menu bar are Applications, Places and Desktop and these appear as shown in the above figure.

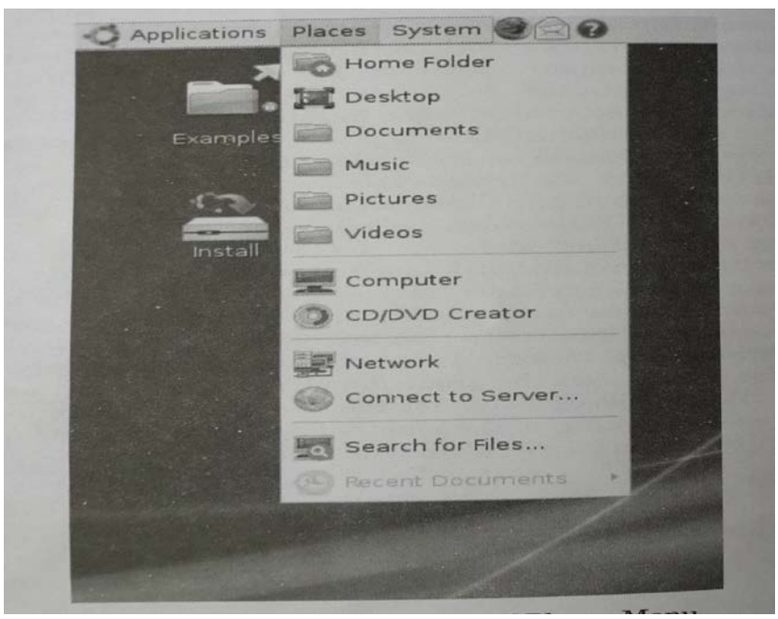
In certain new Linux versions, the menu bar has the items Applications, Places and System. Clicking each of the items opens up a number of elements in the form of a drop-down list. Select Applications in the menu bar to launch applications, Places to open locations in the computer or in the network, and System for customizing the system environment. Clicking the menu Applications opens a list of elements as shown in the figure below.



The elements are actually group names and each group holds several installed applications. An application is a type of computer program that allows performing a particular task. Applications are used to create text documents such as letters or reports, to work with spreadsheets, to listen to music, to browse the Internet, create, edit, or view images, audio and video files and so on. For each of these tasks different applications are available. The different applications are grouped based on their use. The groups commonly available in several Linux distributions are Accessories, Games, Graphics, Internet, Office, Sound & Video, System Tools and so on. Applications for audio and video operations are stored under Sound & **Video group.** Office group has applications for doing office operations such as word processing, spreadsheet operation, database maintenance, presentation

work etc. Applications for Internet related tasks such as web browsing, mail checking, chatting and social networking can be seen under **Internet group**. Photo, image and picture editing and manipulating applications are available under **Graphics group**. Games group has several game applications. Applications that are not included in any of these groups are installed in **Accessories group**. In some distributions additional groups can be seen in the list. When the mouse passes over the group name, a window is opened on the right side and the list of installed applications under the selected group are displayed. User can launch the application by clicking on the required application. Launching the application opens the selected application in a new window.

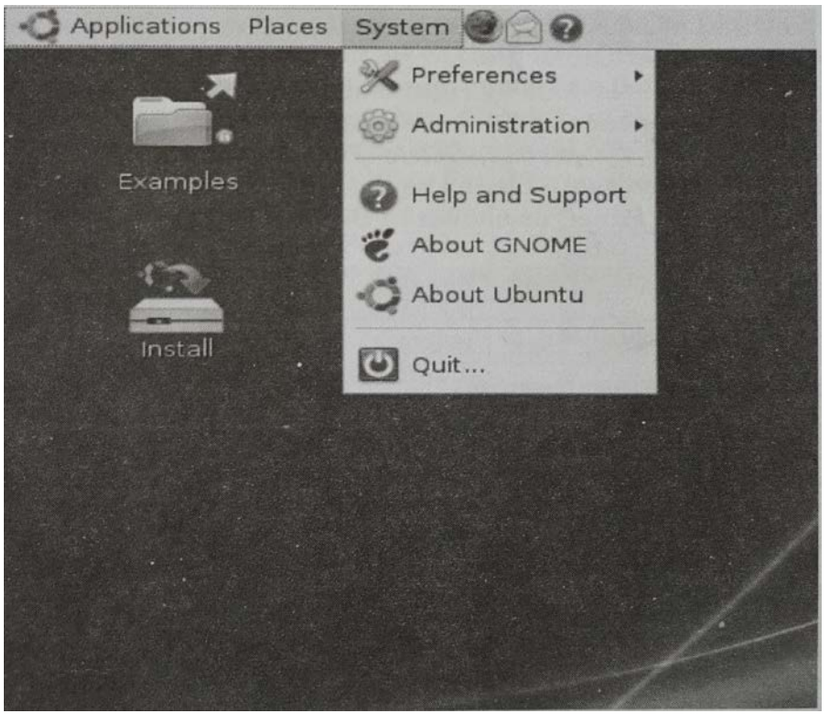
Places are classified into a number of groups as shown in the figure below.



Users can move to Home folder, Desktop folder and other folders created for the user. While adding a new user, the system creates different folders for the user. These different folders are named documents, music, pictures and videos and the

folders are meant for storing the different files of the user, classified under their types. A user can freely move through the folders created for the user. Users other than the administrator user have only limited power and a free navigation is not allowed through all the folders in the system. Clicking the element Computer in the list opens the file manager window and displays the drive and file arrangement in the computer. Administrator user can navigate through the different folders and files displayed. In a similar way moving through computers in the network is also possible. Writing files to CD or DVD is possible by launching the application CD-DVD Creator from the list. Searching for files and displaying the list of recent documents are also done by selecting the options from this list. If the recent documents list is empty, this option will remain unavailable for selection.

Opening the menu System displays a drop-down list of elements as shown in Figure below.



Using the first option in the list it is possible to set different preferences. **Administration** option has applications for performing several system administration jobs. The other options in the list include getting GNOME help and Support and a short description about the distribution. Proper shutting down of the system is essential after a session. Quit option of this list provides the facility for proper shutting down of the system.

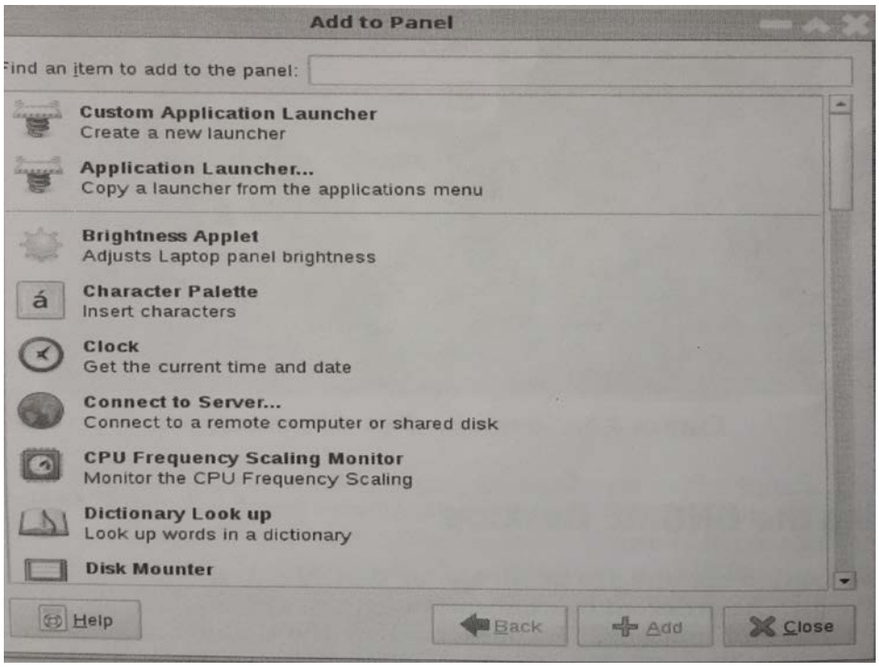
10.4 Customizing the GNOME Desktop

Several customization options are available for GNOME desktops and the options differ with distributions. The different customization options are available for themes, usage of colors, icon sets and animations.

10.4.1 Customization of Panels

Customization of panels helps to add different tools, menus or other applications to panels, change the behavior and appearance of panels, add or remove objects from panels, create multiple panels, hide and show panels and so on. It is possible to select and set different properties, objects, and backgrounds for panels. By default, the top panel contains the menu made up of elements like Applications, Places and System. Besides the menu elements the panel also contains launchers and applets. Web Browser launcher, **Notification Area applet**—which displays icons from other applications**, Clock applet**—showing the current time, Volume **Control applet**—for controlling the volume of the speaker on the system, **Window Selector icon**—which lists all the open windows are some other objects that can be seen in the top panel. By default, the bottom panel contains the Show Desktop button, Window List applet and Workspace Switcher applet. The presence of icons differs with different distributions.

Customization of panels is done by right clicking anywhere on the panel. When this is done, a pop-up window appears and a list is displayed in this window. Different options displayed are Add to Panel, Properties, Delete This Panel, New Panel. To provide more details, a Help element is also available in the list. Clicking the option Add to Panel opens a new window titled Add to Panel, as shown in Figure below.

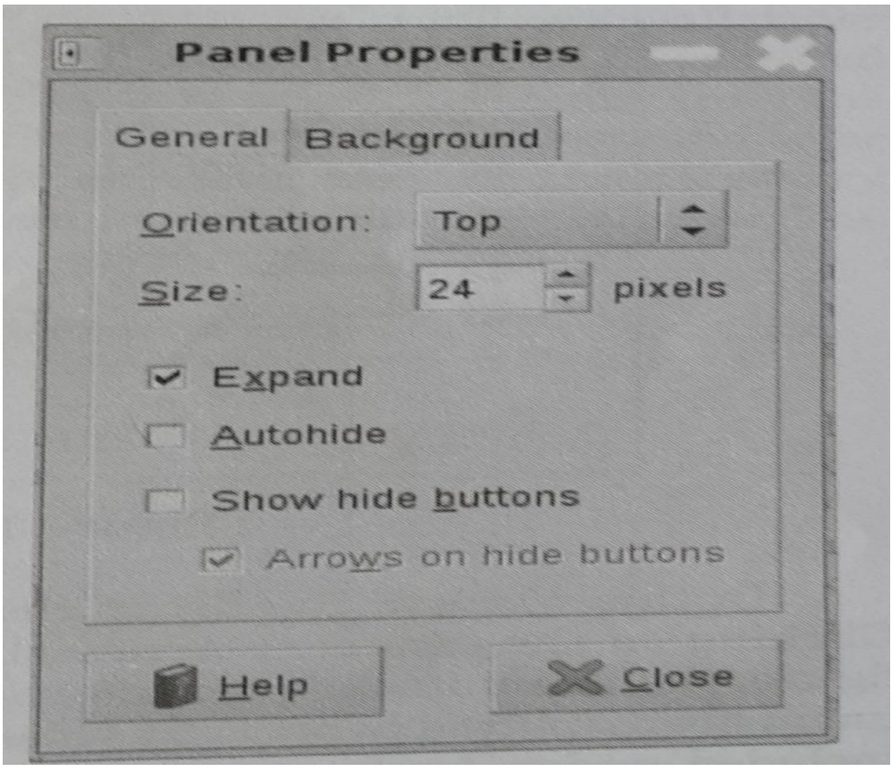


Any item from the list can be selected by clicking on the item. When the item is selected, the Add button gets activated. When clicking the Add button, the selected application is added to the top panel and the icon appears on the top panel. Items not available in the list can also be added to the panel. For this, select the item Custom Application Launcher from the list and proceed to add new items to the list and then to the panel. After adding items to the panel, click the Close button to close the window. Working in a similar way, application launchers, buttons, menus and so on can be added to the panel

To remove items from the panel, right click on the item. From the options displayed on the pop-up window, select the option Remove From Panel. Besides adding items, **drawers** can also be added to panels. **Drawers** are extensions to panels and these enable the storage of different objects in a drawer, as in the case of drawers in tables. It is possible to add, move, and remove objects from

drawers in the same way as is done to add, move, and remove objects from panels. Selecting **New Panel** from the list in the pop-up window adds a new panel. Thus more panels can be added to all the four sides of the desktop. To delete any panel, select the option Delete This Panel from the list and proceed.

Selecting Properties option opens another window on the screen as shown in Figure below.



This window has two tabs named General and Background. Using the options in the General tab, general properties of the panel can be changed. The size of the panel and its orientation can be changed by setting different values in this tab.

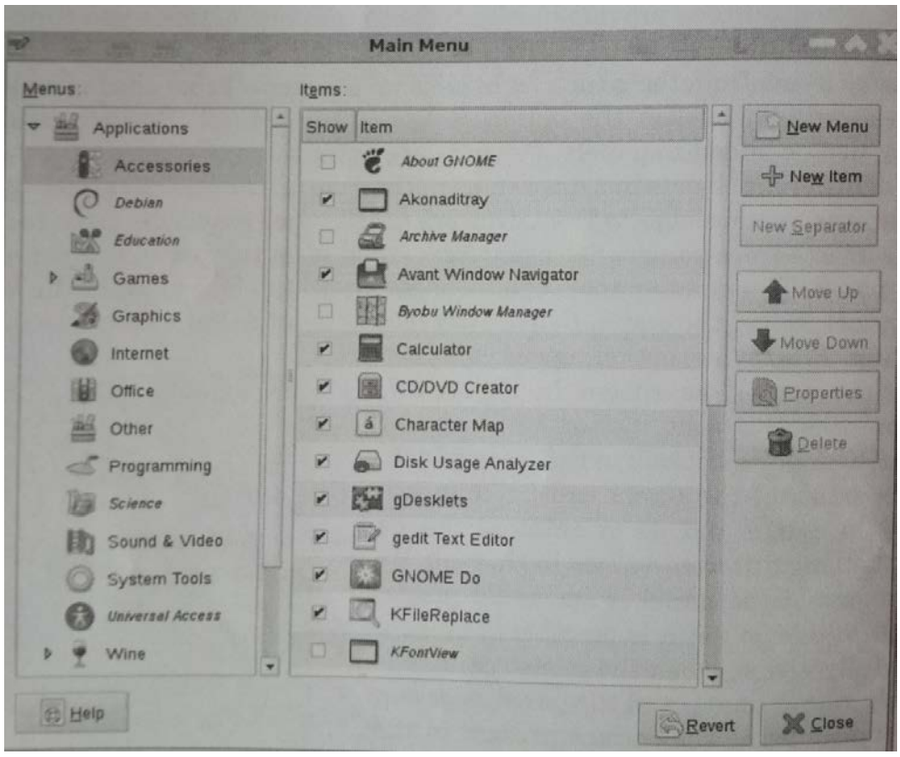
Background image and the panel color can be set differently using the options present in the other tab.

A panel can also be moved to another place. To move a panel to another side of the screen, press and hold the ALT key and drag the panel , to the new location. Click on any vacant space on the panel to begin the drag. A panel that is not set to expand to the full width of the screen can be dragged away from the edge of the screen and can be placed anywhere.

10.4.2 Customization of Menus

Clicking each element of the panel menu bar opens a list of elements as a drop-down list. Different elements are grouped under different sections in the list and the sections are separated using separator elements. Some of the displayed elements in the list are having an arrow mark on the right side indicating that there is another set of elements beneath the item. Customization of menu includes adding a new menu element to the panel, removing an element from the panel, adding a new item to the list or removing an item from the list, changing the order of elements in the list by moving up or down the elements, adding new separator elements to the list and changing display properties.

Elements displayed under each menu item can be removed from the list or new ones can be added to the list in the following way. First right click on the menu item. This displays a new pop-up window. Choose the option Edit Menus from the displayed window. This step opens a menu layout window titled Main Menu, as shown in Figure below.

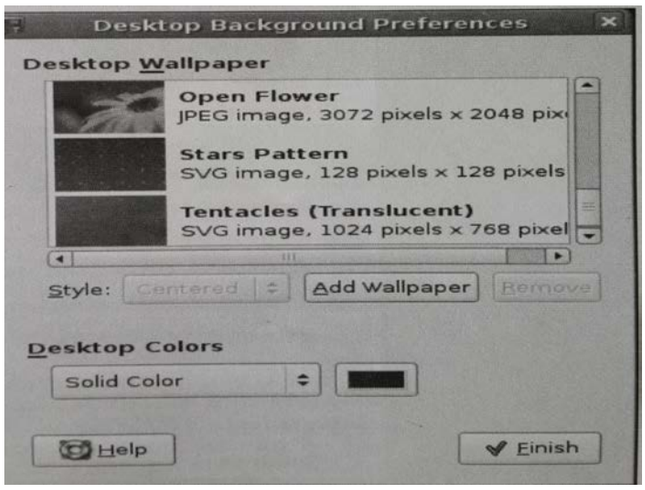


In this window, the list of elements in the menu bar is displayed on the left pane. Clicking on the expander arrow shows or hides sub elements under each menu. In the figure, the menu item **Applications** is in the expanded form and the elements under this menu are displayed beneath the element. **Accessories, Debian, Education** are the elements included under Applications menu. Clicking an element on the left pane displays the list of installed applications, on the right pane. In the figure, the element **Accessories** is selected and the installed applications belonging to this group are displayed on the right pane. All the applications installed in the group need not be visible when the menu is opened. Applications can be made visible by checking on the Show checkbox. By removing the check mark, application can be made invisible in the displayed list. The hidden

application can be added back to the list by checking on the checkbox. As can be seen in the figure, on the right side of the window, a number of buttons with labels are available. The purpose of each button is clear from the label attached to the button. Creating a new menu, adding an item to the menu and adding a new separator are done by clicking the respective buttons. An element in a list can be moved up or down by selecting the element and then clicking the **Move Up** or **Move Down** arrow buttons, to move the selected element up or down in the list. Properties can be changed by clicking Properties button and the selected element can be deleted using Delete button. Clicking Revert button helps to undo the changes made. After finishing the work, click Close button to close the window.

10.4.3 Customization of Desktop

Changing color, style and wall paper of the desktop are the major desktop customizations possible. These are done by selecting the required options displayed in the pop-up window appearing when the mouse is right clicked anywhere on the blank desktop surface. Selecting the option **Change Desktop Background** displays a new window titled **Desktop Background Preferences**, as shown in Figure below.

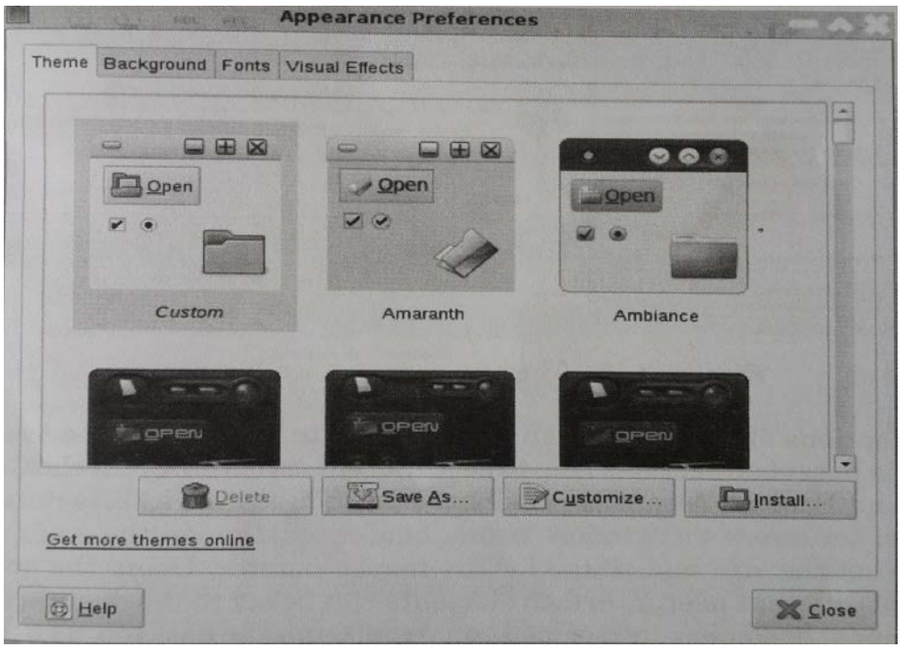


This window displays sample images of a number of wall papers. Depending on the distribution used, the display can be different. Select the desired image from the list by clicking on it. The desktop background wall paper changes to the selected image instantly. The **Style** button allows setting the image in different styles such as centralized or stretched. It is also possible to add new wall paper images to the list from other sources. To remove any image, first select the image from the list using mouse click and then click the **Remove** button. The Desktop background color can also be set with the help of the Desktop Colors option. After making the necessary changes, click the **Finish** button to save the changes made and to close the window. The **Close** button on the title bar can also be used for the same purpose.

Apart from the above discussed customizations, some others can also be done using different tools installed by default, during the installation time of Linux operating system. The different installed tools are displayed when the Preferences element available in the System menu is opened. An overview of the different tools available in Ubuntu Linux can be obtained by studying Figure below.

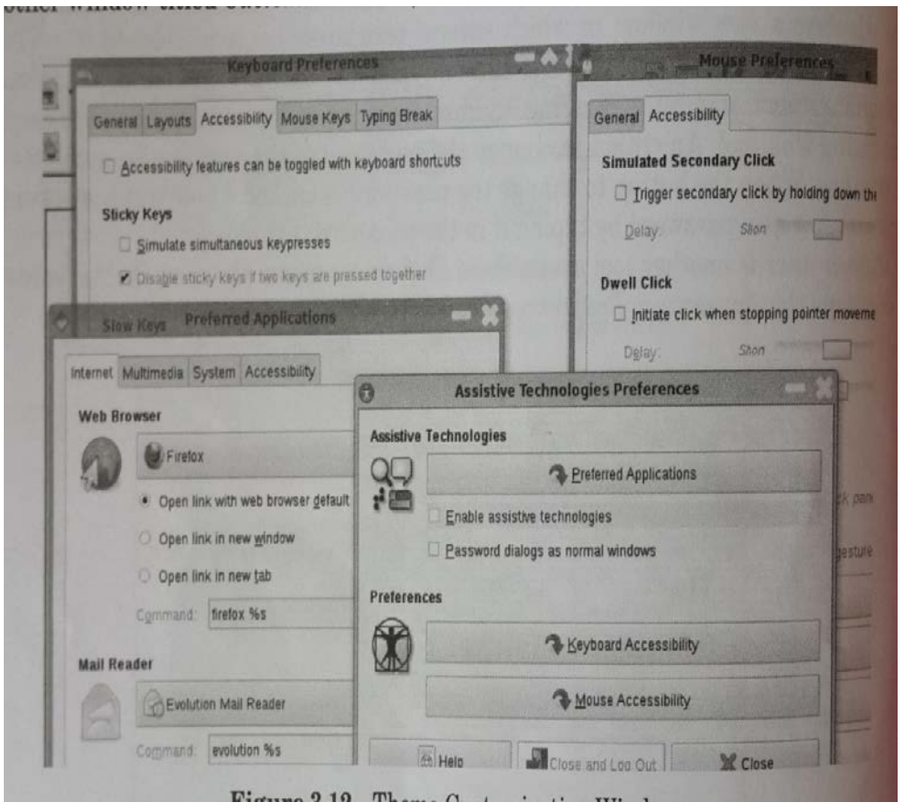
Preference tools are available for several purposes such as hardware customizations related to Bluetooth preference, printer settings, keyboard shortcuts, mouse pointer settings, power management, remote desktop management, audio settings and so on. Also there are tools for setting and managing computer networks, for selecting preferred applications and the like. **About Me** tool helps to set different personal information of the user. Opening this menu displays a new window, in which several text boxes are available. In these text boxes personal information of the user can be typed and saved. E-mail address, contact telephone number, instant messaging address are some of the information that can be saved using this tool. An option to change the password of the user is also available in this window. When the button to change the

password is clicked a new window appears. User can set a new password by typing it in the password text box**. Appearance** is another tool available in Preferences menu. Opening the Appearance tool displays the Appearance Preferences window as shown in Figure 3.11.



As seen in the figure, this window is having four tabs named as Theme, Background, Fonts and Visual Effects. A theme is a group of coordinated settings that specifies the visual appearance of the GNOME desktop. Selecting the Theme tab displays several desktop styles in a preview style, as can be seen in the figure. Any of the themes can be selected and this step changes the overall appearance of the GNOME desktop to the selected theme. After selecting the theme, click the OK button to set the desktop to the selected theme. Using the other buttons available, the selected themes can be deleted, new themes can be added or modified and can be saved as a new theme.

Themes settings affect different parts of the GNOME desktop, such as controls, colors, window frame, icons and pointers. The Controls setting determines the visual appearance of windows, panels, and applets. This setting also determines the visual appearance of other GNOME-—compliant interface items like menus, icons, and buttons. The selected theme can be customized further and can be saved as a new theme. For this, click the **Customize** button appearing at the bottom of the window. This step opens another window titled Customize Theme, as shown in the Figure below.



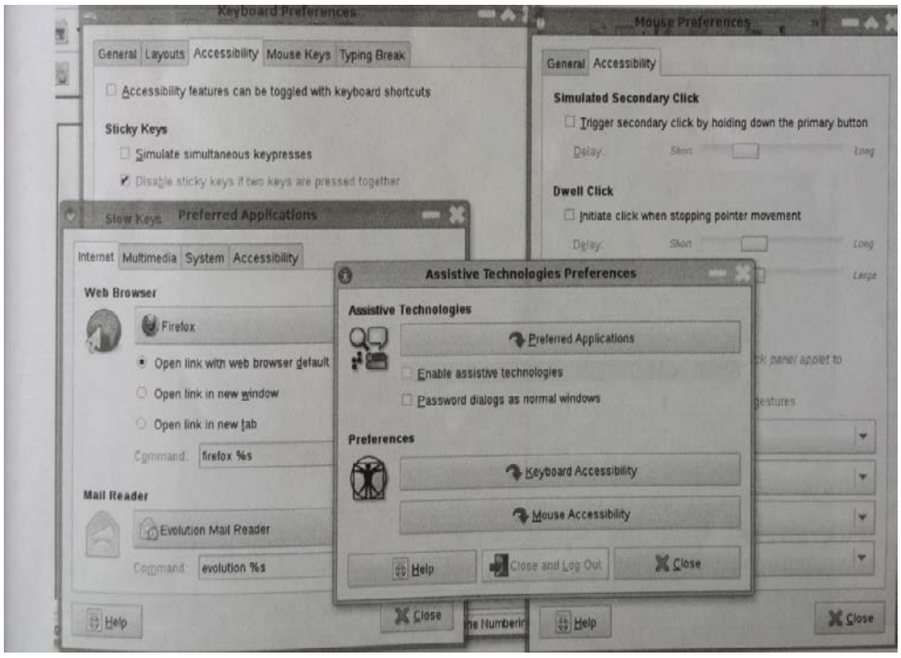
Different options for customization are provided in different tabbed sections, in this window. Options in tabbed sections help to set different styles for Controls, Colors, Window Border, Icons and Pointer. Also it is possible to do such things like

varying background and **text colors, set colors** for window frame border, customize the icons appearing on the screen and set the size and shape for the mouse pointer. Using the options for color setting, colors of different user interface elements can be set to the desired color. The icon setting determines the appearance of icons on panels and the desktop. The pointer setting determines the appearance and size of the mouse pointer. After making modifications to the existing theme, save the modified theme with a new name by clicking the Save As button and proceeding further.

Opening the **Background** tabbed section in the **Appearance Preferences** window displays a set of options. It is possible to customize the desktop background by selecting the image from the displayed list of images. Selecting an image for the desktop background superimposes the image on the desktop background. The desktop background color is visible if a transparent image is selected or if the image does not cover the entire desktop. A solid color or a gradient effect with two colors can also be selected for the background. Selecting the Fonts tab opens a new window which shows the name of the available fonts and point sizes. To change the font, click the font selector button. The font picker dialog opens. Select the font family, style, and point size from the lists. The preview area shows the current choice. Click OK to accept the change and to update the desktop. Fonts used in menus, toolbars, dialog boxes of applications, documents, icon labels and title bars can be modified as per requirement.

Another tab available in this window is the **Visual Effects** tab. Opening the tab helps to set different options to change the animation effects on the desktop. Different animation effects can be set for opening, closing, focusing and minimizing of windows, by setting different options. In Ubuntu distribution, another customization tool available is **Assistive Technologies**. This tool can be used to set applications to start automatically when logging to the system, configuring keyboard accessibility features such as sticky keys, slow keys or bounce keys and configuring mouse accessibility features such as dwell clicking and so on. All the customizations are made by setting values to various options

displayed when opening the different tabs appearing on the window. The typical display of the different options can be seen in Figure below



File Management is another tool available in several distributions. Launching this tool opens a new window titled **File Management Preferences** window. Different configuration options are arranged in different tabs. The options help to modify the view styles of files and folders such as viewing as icon view, list view or compact tview. Options can be set in such a manner that files are opened when the mouse is single clicked or double clicked. Asking for confirmation, when files are deleted is another option that can be set, using this tool.

Another general and commonly done desktop customization is the setting and configuring of screen savers. Screen savers work by displaying moving images on the desktop screen when the computer remains idle for a considerably long period. To stop the screen saver and to return to the working condition, move the

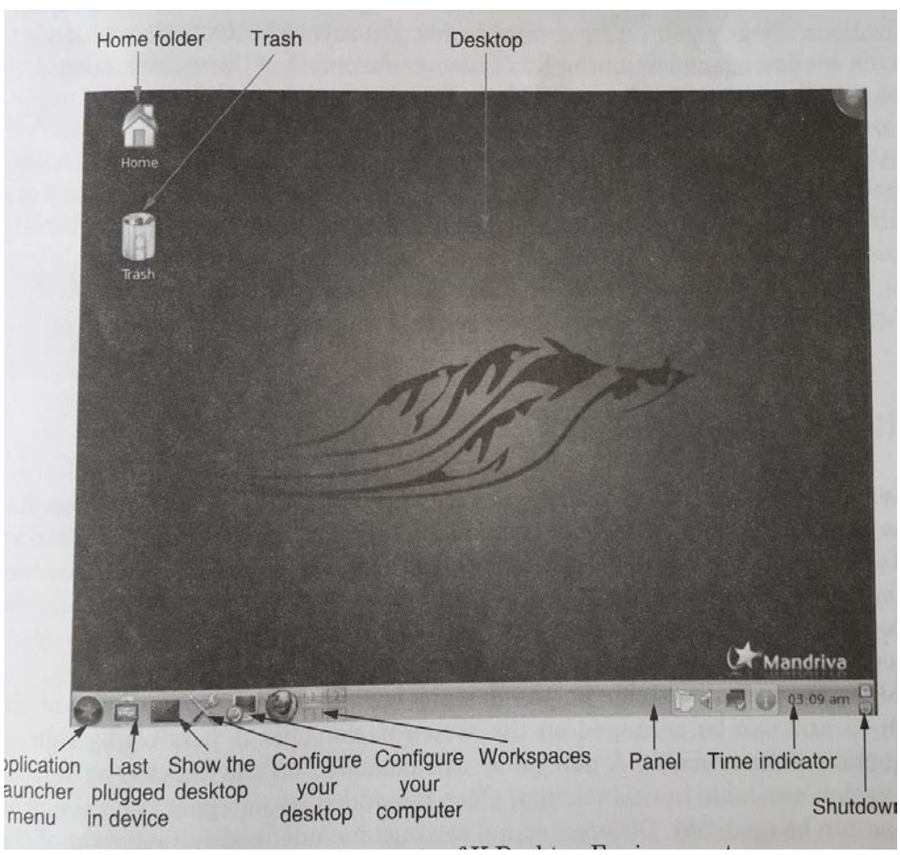
mouse or press any key on the keyboard. Use the Screensaver tool to set the type of screen saver to be displayed and the time after which the screen saver has to start its action. Launching this tool opens the **Screensaver Preferences** window as shown in Figure below.

In the figure, the theme floating feet is selected and its display appears on the right pane. Incidentally it is to be stated that foot is the logo of GNOME and this logo can be seen in the top panel in the GNOME desktop. On clicking the Preview option button in this window the selected theme is displayed on the whole screen. The Blank screen theme displays no image and selecting this option displays only a black screen. Random theme selects a screensaver at random from the list to function as the screen saver. In the window, move the slider to set the time to treat the computer as idle. When the set time has passed with no input

from the mouse or the keyboard, the computer is treated as idle and the screensaver becomes active and displays the image on the screen.

10.5 K Desktop Environment

Another popular desktop Linux environment available in almost all Linux distribution is KDE or K Desktop Environment. This environment was released before GNOME. This desktop environment is used both by Unix and Linux. The display of the typical K Desktop Environment is shown in Figure below.



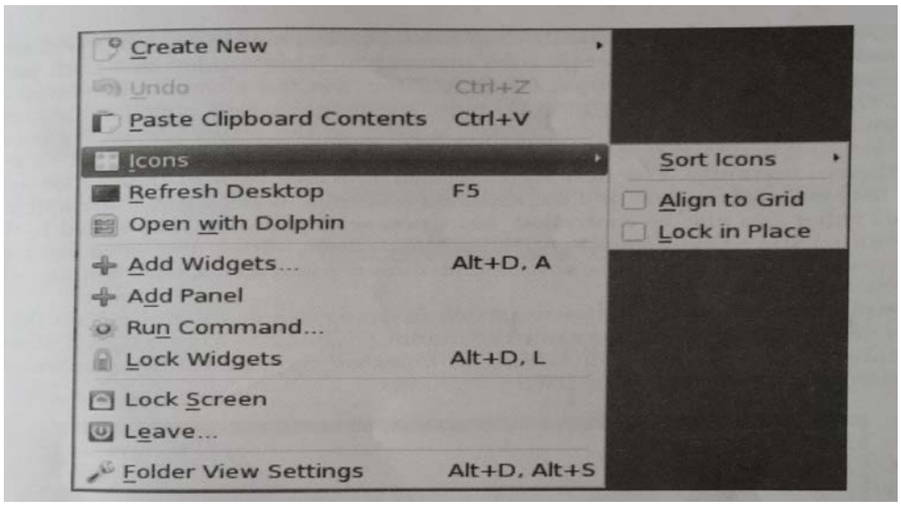
The appearance of different windows, menus etc. are identical to that of the GNOME desktop. This desktop has a number of icons displayed on the desktop. Similar to GNOME, other parts of this desktop environment are panels, windows, menus and buttons. By default, several distributions have a single panel at the bottom of the screen in KDE. Different applets as well as launchers can also be seen in this panel. Date and time indicators, system sound indicator, volume indicator, network enable indicator appear on this panel. **Four** workspaces are available by default and these are also visible in the bottom panel.

Home Folder of the user is created on the desktop during the installation of Linux. Home folder stores all the files of the user. This folder is displayed on the desktop and is indicated by a special icon. Special icons are also used to represent Trash folder and different removable media. The main point of access when working in KDE is the panel buttons. The different buttons in the bottom panel are known by names like Application Launcher menu, Last plugged in device, Show the desktop, Configure your desktop and Configure your computer. Clicking each of the buttons displays a menu in which several elements related to the subject of the clicked button are displayed. The elements in the list provide several choices for the user to make selection. As is clear from the name, the **Application Launcher** button is used to launch applications. The arrangement of applications is similar to the arrangement of applications used by GNOME desktop. Applications are grouped under several heads. Executing Linux text commands is done from the window opened when the **Run Command** element of the menu is opened. Facility for leaving the system is also available in this menu.

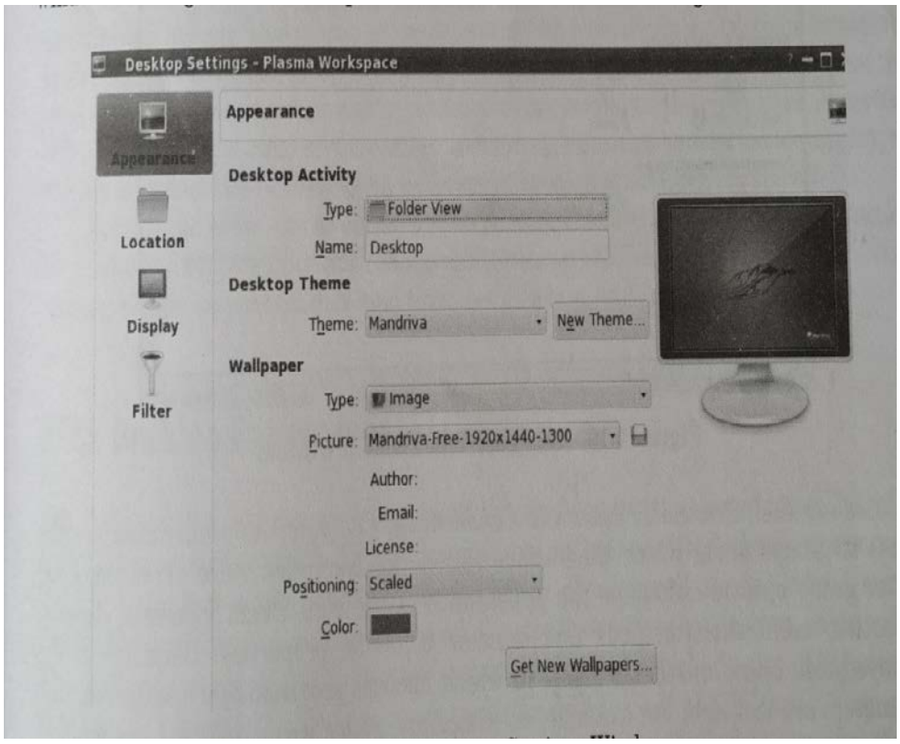
Clicking the Last plugged in device icon displays the list of devices that are recently plugged into the system. Usually CD, DVD or flash drives are plugged into the system for different purposes and clicking the icon provides an easy way to get the details of the devices. **Show the desktop** icon actually works as a toggle switch. Clicking this icon closes all the opened windows and shows the desktop. The other two icons namely **Configure your desktop** and **Configure your computer** are meant for different customization purposes.

10.6 Customizing the K Desktop

Similar to the different customizing options available for the GNOME desktop, K desktop is also provided with a number of customizing options. Customizations of desktop elements are done by setting the different property parameters that are displayed as a pop-up window, when the mouse is right clicked anywhere on the empty space on the desktop. The pop-up window displayed is as shown in Figure 3.16.

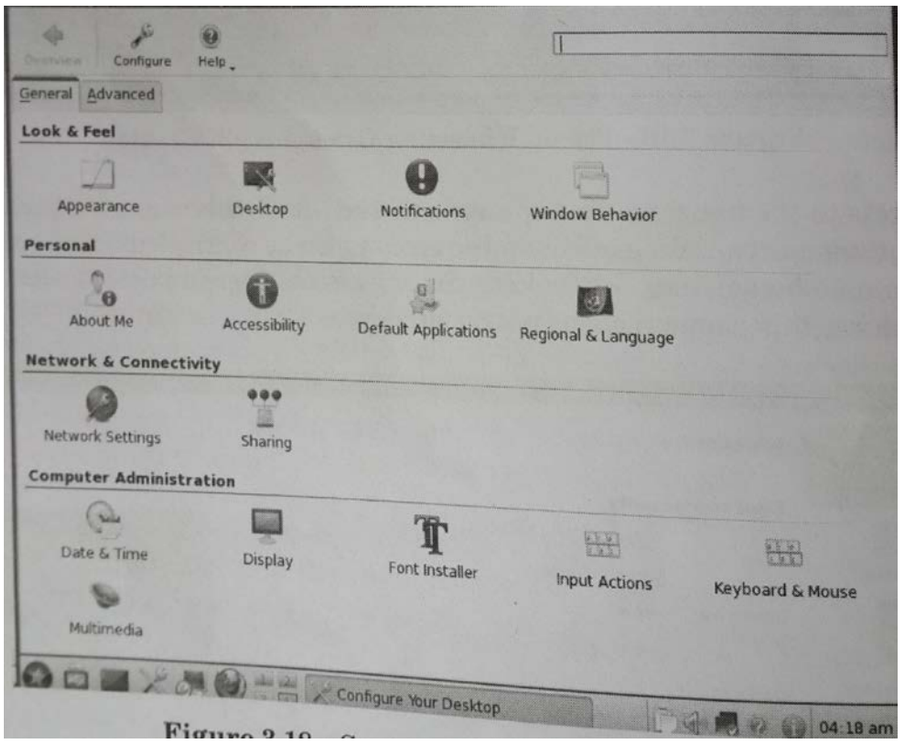


Different icons available on the desktop can be sorted on the basis of time, size and so on and can be arranged on the sorted basis. This is possible by selecting **Sort Icons** option of **Icons** menu. A new panel can be added on the desktop by selecting **Add Panel** option available in this window. Once the added panel appears on the desktop, its settings can be modified. Different panel settings include modifying height of the panel, adding widgets to the panel, adding spacers and the like. Other panel properties such as its alignment, visibility, hiding effect and so on can also be controlled. Panels that are not required can also be deleted. Folder View Settings is another important element of this window. Clicking this element opens a new window as shown in Figure below.

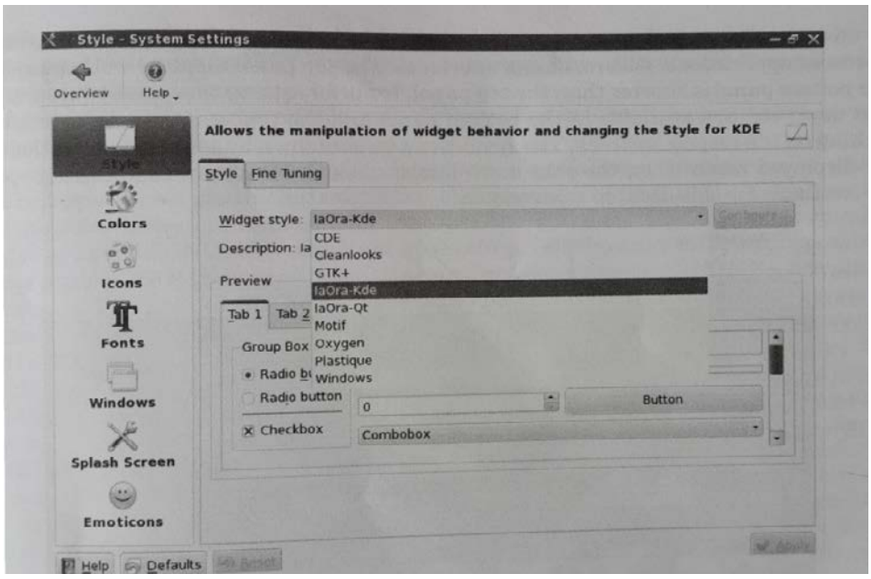


The new window opened is the **Desktop Settings** window. Using the options available in this window several parameters can be customized. This window has a left pane and a right pane. *Appearance, Location, Display, Filter* are the elements displayed on the left pane. Options for setting desktop themes, wall papers, and wall paper images, location of the Home folder and display parameters appear on the screen, when **Appearance** is clicked. Selecting a new theme from the given samples displays a reduced view of the selected theme, instantly. New wall papers can be added to the list. Several wall papers namely Weather, Globe, Mandelbrot, Virus, and so on are available for making selection. After setting values for different parameters, click the Apply button to take effect the new settings.

Opening the configuring option available on the desktop—**Configure Your Desktop**---displays a new window having the same title name, as shown in Figure 3.18. This window has two tabs named **General** and **Advanced.** Opening the general tab displays several configuration options arranged in different groups.



Look & Feel, Personal, Network & Connectivity, Computer Administration are the names of groups under which the various options for configuring are arranged. **Look & Feel group** options customize the appearance of the constituent elements, desktop properties, Notification methods and window behavior properties. Clicking on the displayed icon opens another window in which enough provisions for configuring the parameters are included. For example, clicking **Appearance** icon displays a new window as shown in Figure below.

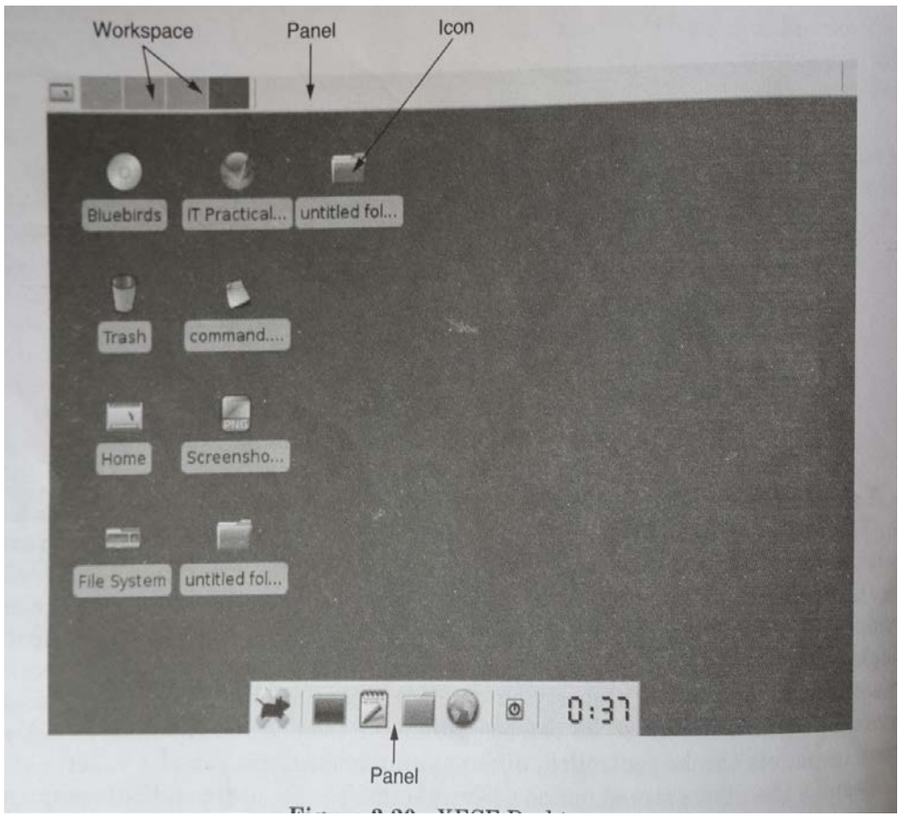


As can be seen from the figure, the displayed window has a left pane and a right pane. The left pane has buttons named style, colors, icons, fonts, windows, splash screen, screensaver and so on. The different appearance settings are controlled by setting different values or parameters for style, colors, icons, fonts, windows, splash screen and emoticons. When any icon on the left pane is selected, a set of options for the selected item is displayed on the right pane. To set new values, select the button on the left pane and then set new values on the property parameters appearing on the right pane. By setting different options, different styles can be customized, icons can be selected for display, use of icons in panels can be controlled, different animation effects can also be set.

10.7 XFCE Desktop Environment

XFCE is a small desktop environment and this desktop is not as popular as GNOME or KDE. This desktop is not widely spread or matured as the other two desktop environments. The major advantage of XFCE desktop is that it is light. A typical display of this desktop is shown in Figure 3.20. The components of this desktop

are panels, icons, menus and windows. During installation, some icons such as Trash and Home folder are created on the desktop. User can also create new icons on the desktop. Icons displayed on the desktop represent applications, folders, files or shortcuts. The top panel displays workspace icons. The bottom panel is shorter than the top panel. Icons for setting time, audio volume, help, shut down etc. are available in the bottom panel and the processes can be activated by clicking on the respective icons. The system can be shutdown by selecting the option from the displayed menu when the shut down icon is clicked. Minimum customizing options are available for this desktop environment.



10.8 Desktop Applications

Several applications get installed by default, while installing Linux operating system. it is possible to find Linux applications for creating text documents, spreadsheets, browsing the Internet, sending mails, creating and editing images, audio and video operations, gaming and so on in almost all Linux distributions. Applications such as web browser, FTP tool, Instant messenger application, chat client and mail client help to make use of the various services available in the Internet. There are also applications for writing to CDs, running multimedia files and educational applications. Different security tools such as firewalls, tools for encryption and decryption are also get installed by default. Debuggers, compilers, website developing and scripting tools such as PHP, Perl, Python are some other common resources found in different Linux distributions.

Some of the common applications that are found in several distributions are the following: **Gedit** is the text editor application used for creating and editing simple text files. No formatting option is supported by this text editor. **Dictionary** application helps in finding spellings and definitions of words. **Calculato**r application helps to do different basic, financial, and scientific calculations. Letters and symbols from the Unicode character set are selected with the help of Character Map. **Nautilus File Manager** is the file management application and this helps for the display of folders and their contents. File manager also helps to copy, move and classify files, access CDs, USB flash drives and other removable media. **Terminal** provides access to the system command line.

OpenOffice.org suite is installed in several Linux distributions. This suite is made up of a number of applications. OpenOffice.org Writer (Word Processor), OpenOffice. org Calc (Spreadsheet), OpenOffice.org Impress (Presentation), OpenOffice.org Base (Database) and OpenOffice.org Draw (Drawing) are the individual applications included in this suite. **OpenOffice.org Writer** is the default word processing application of different Linux distributions. Different operations are controlled with the help of menus and submenus. Options are available to select any character, word, line, sentence, paragraph, page or the complete

document. Deletion of character, word, line, sentence, paragraph, page or document can be easily done in this application. Formatting such as changing font size and its type, line spacing, indentation, search-replace, finding word and replacing are some other options. Insert page numbers, page breaks, headers, footers, add graphics can also be done.

OpenOffice. org Calc helps to create spreadsheet files that can be used for manipulating data and preparation of different types of graphs. The application used for creating presentation files is OpenOffice.org Impress. Database files are created using Base application. A simple drawing application of this suite is **OpenOffice.org Draw**.

A freely distributed software for image editing, photo retouching, image composition and image authoring found in several Linux distributions is **Gimp.** This application works on many operating systems. **Xpaint, F-Spot Image Editor** are the other graphics applications distributed through different Linux distributions. For simple viewing of different image files, there is the **Image Viewer** application.

Amarok is an audio player which supports different audio file formats such as MP3, ogg and wav. **SMPlayer** is the video player which can play files in formats such as MPEG, MP4, DivX, ogv, wmv and so on.

A fast and full-featured web browser available in several Linux distributions is **Firefox.** Firefox is a lightweight browser based on Mozilla. **Konqueror** is another web browser available in several Linux distributions. **Gaim** is an instant messenger application that can be seen in Linux distributions. It supports multiple protocols via modules including AIM, ICQ, Yahoo!, MSN, Jabber, IRC, Napster, Gadu-Gadu and Zephyr.

Evolution is the GNOME mailer, calendar, contact manager, and communications tool. The tools which make up Evolution are tightly integrated with one another and act as a seamless personal information management tool.

Download managers, chat clients are the other common Internet applications available in several Linux distributions.

10.9 Assignment 10

1) Explain the X Window system
2) Write a note on GNOME desktop
3) Write a note on K Desktop Environment
4) Write a note on XFCE Desktop Environment

Chapter 11
Managing Linux Files and Folders

11.1 Linux Files and Folders

Files store information in an electronic form. Files hold different types of information. Files can be seen anywhere in the computer system and these appear in different forms and in different types. Files in computer systems are identified by their names. Different operating systems have different rules for naming computer files. These rules are related to the maximum number of characters that can be used in the file name, use of special characters in the file name and the like. New versions of Linux operating system allows file names with several characters with more than one word separated by space character. It is important that the same file name must not be given to more than one file. Linux system treats lower case as well as upper case letters differently. So files with the same name, but using different letter cases are considered as different files. However, certain special characters are not allowed in file names.

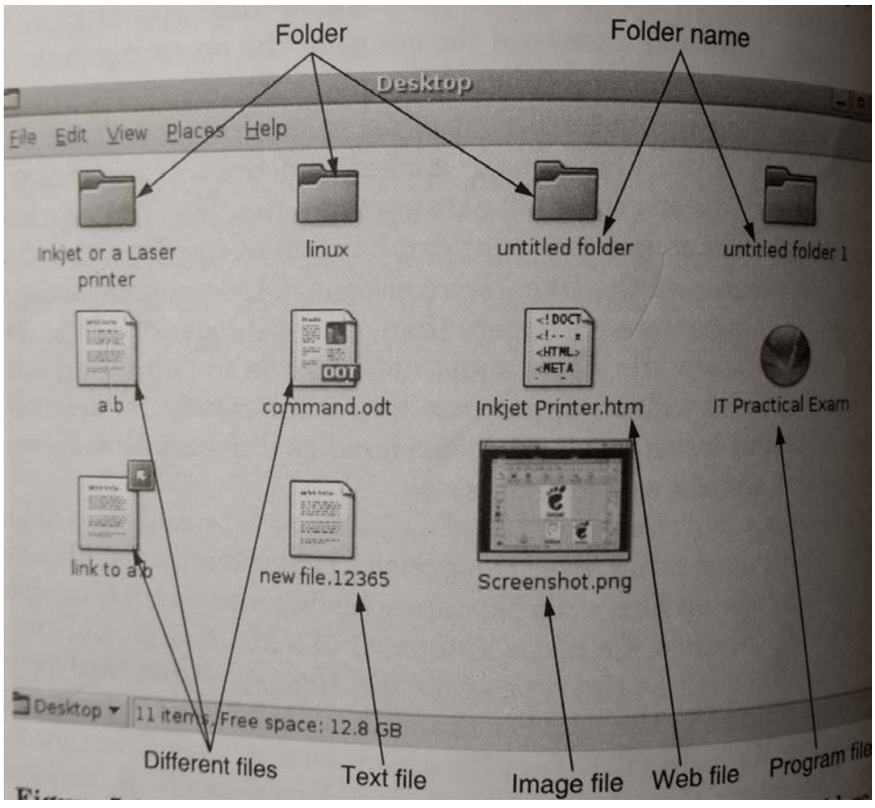
File names have a set of characters used as file name extension. File name extension is the set of characters appearing after the period (.) character in the file name. From the standard file name extension of a file, it is possible to understand some features of the file such as the type of the file, the application used for creating the file and the data contained in the file. For example, a file having the file name extension odt indicates that the file is in PDF Text Document format. OpenOffice.org Writer is the application used to create the file and the file is a document file. A file saved with the file name extension jpg indicates that the file contains image and it is saved in J PEG format and an image editor application can open the file.

Storing all the files in one place in the computer system without any planning or arrangement makes it difficult to locate the required file from the storage. Under such conditions, it is better to store the files in different locations. Files are

grouped based on their type or use and are stored in different locations. This type of storage helps in locating the required file easily. The need for storing files in a structured and neat manner led to the emergence of the concept of folders. **Folders** can be treated as containers for files. Usually different users create their own folders and store the files created by them in their own folders. For example, files connected with correspondences are stored in a folder named '**correspondence**'. Picture files are stored in another folder named '**picture'**, audio files are stored separately and so on. The terms directory and folder mean the same thing and are used interchangeably. Similar to files, folders are also identified by their names.

Rules for naming files are applied for naming folders also. A Folder can contain several files or no files at all. Folders can also contain several sub folders or no sub folders. Folders created within a folder are known as **sub folders or subdirectories**. The folder in which the user is working is known as the **current directory**. Any directory below the current directory is called **sub folder** or **subdirector**y. The directory above the current directory is called as **parent directory**.

Linux operating system treats folders as files. So files and folders within the same folder must have different names. In graphical system used by Linux, files and folders are represented differently using different icons. Also Linux uses different icons to represent different types of files. Thus text files, image files, web files are all represented using different types of icons. The different types of icons used for representation for files and folders are clear from the screen display shown in Figure below.

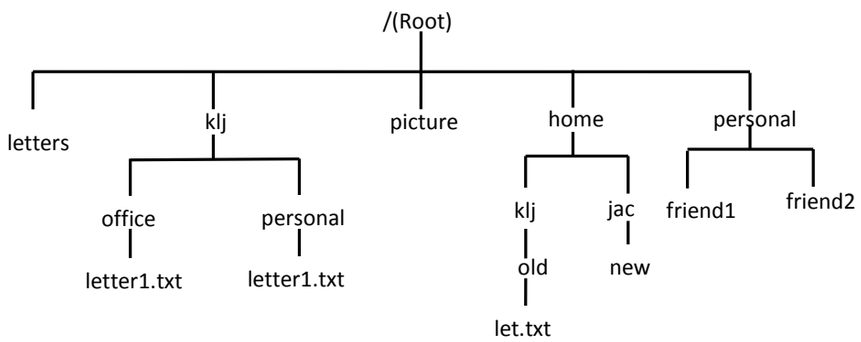


11.2 Paths and Path Names

In a computer system it is possible to find several folders. Each folder may contain several sub folders as well as many files. To locate any file in the system it is necessary to specify the exact path where the file resides in the memory. Simply providing the file name means that the file is located in the directory where the user is located at that time. The concept of paths and path names will be clearer by considering the entire storage space of the system as a single large storage unit. This large storage unit is usually known as root and this is denoted as a single slash character (/). Several folders can be created in the root folder and these are the storage compartments of the root storage space. Each of the partitions

can be partitioned further to get several sub folders. The organization described below

.



As can be seen in the figure, the root is at the top level. The root folder is partitioned into a number of folders namely letters, klj, picture, home and personal. These folders lie just below the root partition. These folders are represented *as / letters, /klj, / picture, / home and / personal* respectively. The representation indicates the exact paths of the folders and is known as **path names**. Folders named klj and personal are having two sub folders beneath each one. Folders named **office** and **personal** lie below the folder klj. The exact path names of these two folders are written as **/klj/office**, and **/klj/personal** respectively. Both the sub folders store one file each and the names of files are written below the folder name. Sub folders to home folder are **klj** and **Jac** and their exact path names are given as **/home/klj,** and **/home/jac** respectively. The names of files stored in the sub folders are written below the name of the folder. Folders named **friend1** and **friend2** lie as sub folders to the parent folder named personal and their exact path names are given as **/personal/friend1** and **/personal/ friend2.** Similarly the exact path names to folders old and new can be given as **/home/klj/ old**, and **/home/klj/new**.

Names given to files and folders located within a folder must be different. Also different sub folders of a folder must be named differently. But sub folders of different parent folders can use the same name. Files in different folders can use the same file name. For example, as can be seen in the figure, the same file **letter1.txt** is stored in two different folders. This is possible since different folders

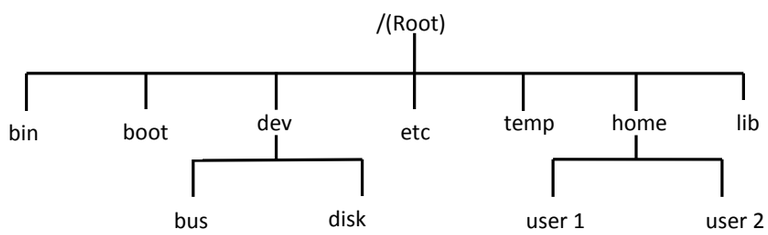
represent different storage locations. The exact path names of these two files are **/klj/office/letter1.txt** and **/ klj/ personal / letter1 .txt**.

When a user logs into the system, the user is automatically placed in the home folder of the user. Home folder of any user is located as the sub folder to the /home folder. For example, when the user **klj** logs into the system, the user is placed in the folder **/home/ klj.** During this time **/home/klj** is the current directory of the user **klj.**

Exact path names denote the exact locations where files are stored. Such path names are also known as **absolute path** names. In path names, the first slash indicates the **root** folder. Following slashes separate the names of different sub folders. File name appears as the last item in path names. To shorten path names, it is usual to specify the path names relative to the current working directory of the user. Such path names are called **relative path names**. Consider the file tree shown already in Figure 5.2. Suppose the user working in the directory **/home/klj** requires specifying the file **let.txt** stored in the sub folder named old. The path can be specified in two ways as follows: **/ home/ klj/old / let. txt** and **. / old / let. txt**

Both the path names identify the same file. The first path name is the **absolute path** name whereas the second path name is the **relative path** name. In the relative path name a dot character is used to indicate the current position of the user and the path name begins with a dot. In a similar manner, to refer to a parent directory above the current directory, a user uses two dot characters in relative path names.

11.3 Linux System File Structure



A file system is the methods and the data structures that an operating system uses to keep track of a file on a disk or partition; i.e the way the files are organized on the disk.

The file system is organized as a tree with a single root node called root denoted by /, every non leaf node of the file system is a directory of files and file at leaf nodes of the tree are either directories, regular files or special device files.

Different types of files can be seen in Unix operating system. Files in Unix operating system are classified as *ordinary files, regular files, directory files* and *special files*.
Ordinary files include a variety of file types like application files, data files, document files etc.

Application files are usually executable files and using these files different applications are executed. There are applications for performing tasks like word processing, doing calculations, browsing the Internet, managing mails, doing image manipulations and so on.

Data files store different types of data and these files can be image files, audio files, video files or simply text files.

Document files store multiple types of information such as text, drawings, images and so on.

Directory files are related to different directories of the system and these store information related to different directories.

Unix special files are device driver files, link files or similar other types of files.

Different physical entities are also represented as files in Unix. Thus, printers, modems and other devices are represented as files. Using special files, the operating system can control different devices connected to the system. Such special files used to control devices are known as device driver files. **Device driver**

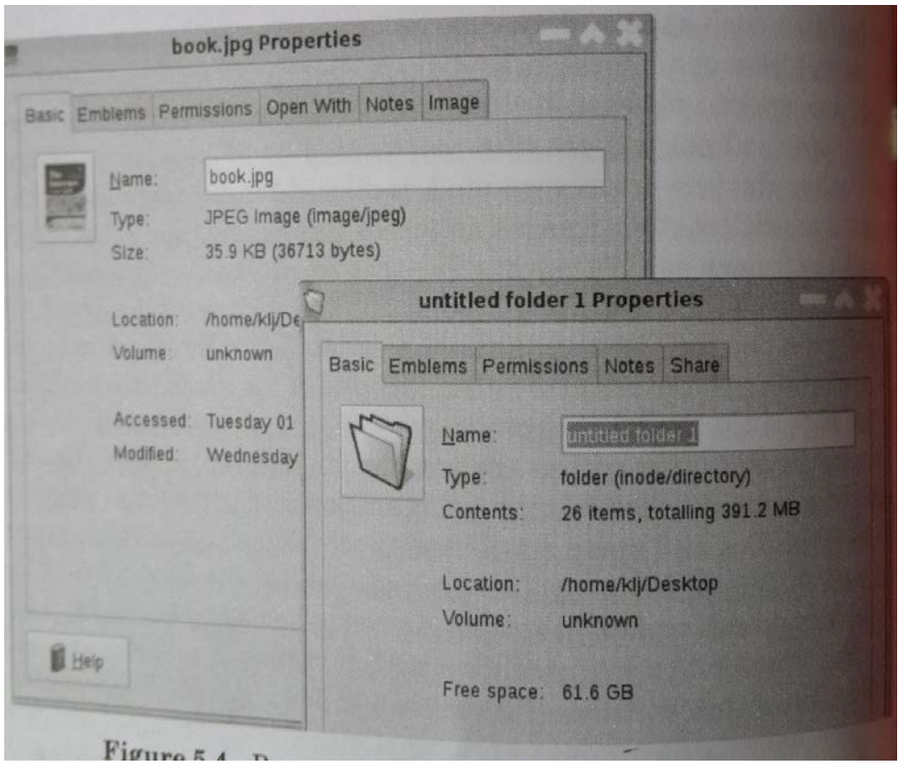
files are essential for proper communication between the operating system and the connected devices.

Different files used by Unix operating system are also arranged in a hierarchical manner. **Root** folder is at the top in the hierarchy. All the files and folders of Unix system are included in this folder. **Root** directory has several sub folders named as **bin, boot, dev, etc, temp, home** and so on. The folder named **bin** stores all binary files. For storing boot files, **boot** folder is used. **Device files** necessary for the proper working of attached peripherals are stored in the folder named **dev.** The folder named **etc** stores different files for configuring the system and devices. The folder named **temp** is a temporary folder and this folder stores temporary files. Files stored in this folder are deleted after use.

Different users of the system are included in the **home** folder. Depending on the number of users, the home folder has several sub folders. Library files required for running the system are stored in the **lib** folder

11.4 Files and Folders Properties

A general idea about files or folders in Linux system can be obtained from the icon and studying the file name extension. More information regarding the file or folder is obtained by right clicking on the **file** or the **folder** icon and selecting the option **Properties** from the pop-up menu displayed on the screen. Selecting this option displays the **Properties window** as shown in Figure below.

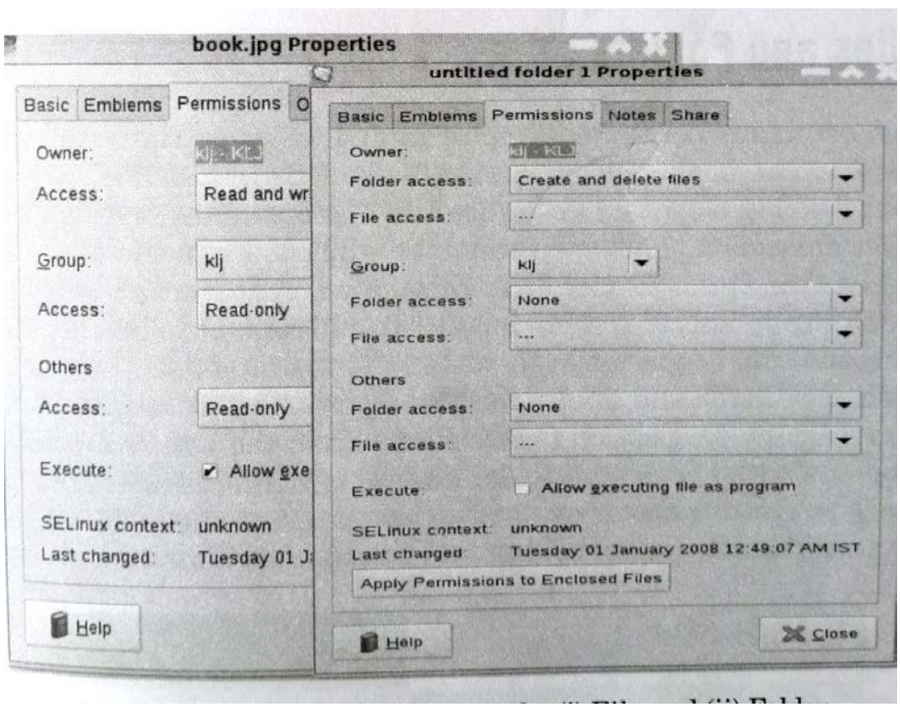


Properties window has a title bar containing the name of the file. This window has six tabs named *Basic, Emblems, Permissions, Open With, Notes and Image*. For a folder, the properties window displays different tabs named as *Basic, Emblems, Permissions, Notes and Share*. Opening the first tab named as Basic displays a general description of the properties of the selected file or the folder. **Name** of the file or folder, t**ype** of the file, its **content, size, exact location** in the system where the file is stored, and the like can be obtained by opening this tab.

Selecting the **Emblems** tab displays a number of emblems and the user can select an appropriate emblem from the list to represent the file or folder.

Permissions to use the file or the folder are the important properties associated with them. There are three kinds of permissions that a user can have for files or folders. These are the permissions to **read** the file, **write** or **edit** the file and to **execute** the file. For directories, the different permissions are for listing the contents of the directory, creating and removing files in the directory and for accessing files from the directory.

Opening the tab named **Permissions** displays the details of permissions available to the file or folder. The display is as shown in Figure below.



Users of Linux operating system are divided into three categories namely **owner** of the file, other users including the **group** of the owner and **all other** users. Each Linux file is assigned to an owner and a group when the file is created. Usually, by

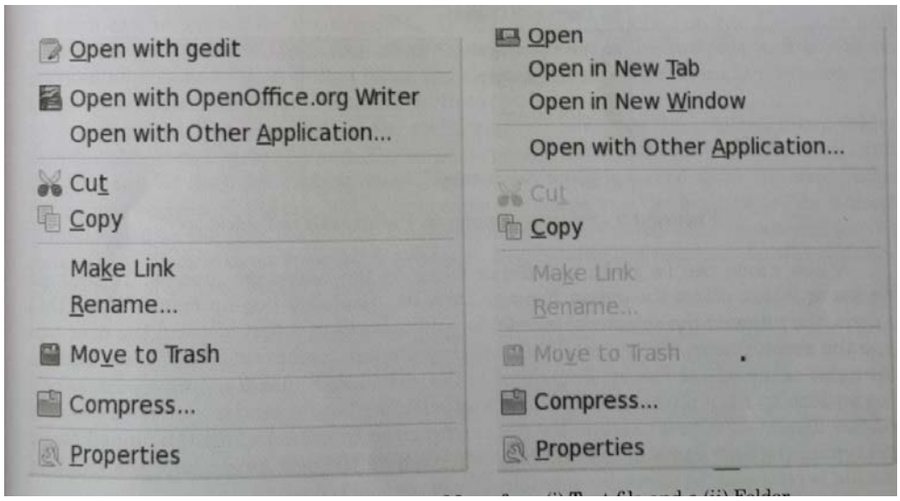
default, the **owne**r is the creator or the **current user** and the **group** is the group of the directory in which the file is created.

Linux makes use of two ways for viewing the contents of files and folders. One way is to double click the file or folder icon. When the folder icon is double clicked, the folder is opened in a file browser window and the content of the folder is displayed in the opened window.

As stated earlier, folders can remain empty or may contain one or more files or sub folders. Different icons are used to represent different types of files and folders and this help to distinguish between files and folders in the folder. A second way of viewing the folder content is to right click on the folder icon and then choosing the open option from the displayed pop-up menu.

In a similar way, to view the contents of a file, its icon is double clicked. It is possible to set the default application for opening any file in Linux system. For example, consider a text file. Linux provides option to configure the system such that all text files can be opened using a text editor like **gedit** or using a word processor application like **OpenOffice.org Writer** application. The **Open with** option available in the file properties window helps to configure the system, as discussed above. Opening this tab displays the list of installed applications using which the file can be opened. Select the ideal application from this list. This step makes all files of the same type to use the selected application for opening them.

Instead of double clicking the icon, right clicking the file icon also provides an option to select an application for opening the file. The list of options available when the file or folder icon is right clicked, is displayed in Figure below.

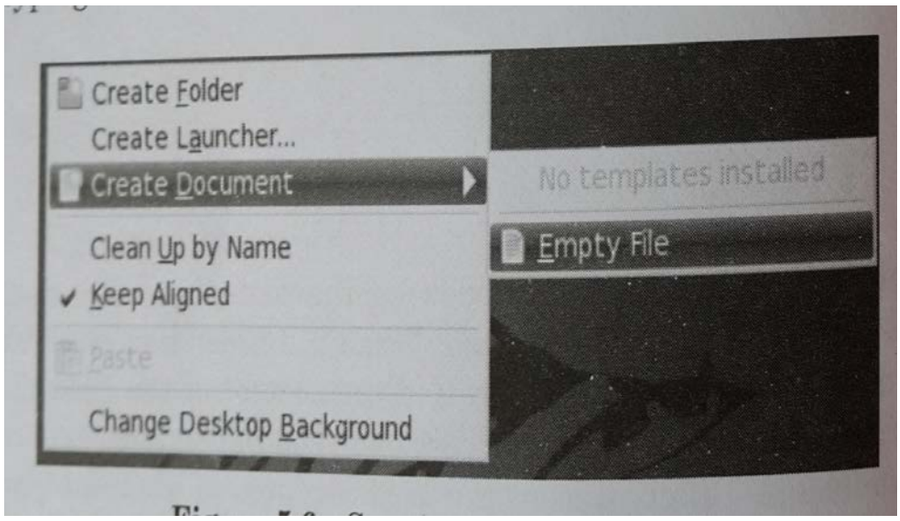


Here two menus can be seen. The first pop-up menu is the menu displayed when a text file is right clicked. In this case, text files are configured to open with **gedit** application and this is the first option displayed on the pop-up menu. Selecting the option to **Open with Other Application** displays the list of installed applications and the user can select the required one from the list. Similarly, for a folder, the first option is the default one and this simply opens the folder. If another application is to be selected for opening the folder the same is done by selecting the option from the menu.

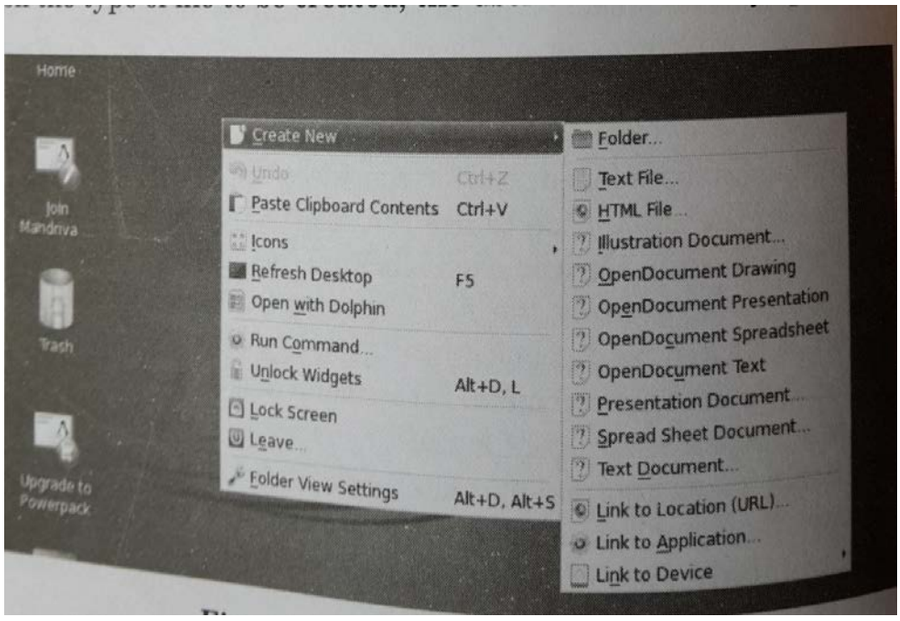
11.5 Creating Files and Folders

Empty files can be created in the desktop or in any folder by right clicking on the free space in the desktop or in the folder. When this is done in GNOME, a pop-up menu is displayed on the desktop as shown in Figure 5.6. The menu has several elements. Select the option **Create Document**. This opens another window. From the newly displayed sub menu option, select the option **Empty File**. This option creates a new text file and the file icon appears with the default file name **new file**. To add content to the file, double click the file icon. The created file is opened in the default text editor. Additions or modification to the contents can be made and finally the file can be saved.

To create a folder below the current folder, select the first option Create Folder from the list. A new folder is created and an icon appears with the default folder named as untitled folder. The folder name is written in a text box and the user can give a new name to the folder at this instant, if required, by typing the new name. Folders can be renamed later also at any time.

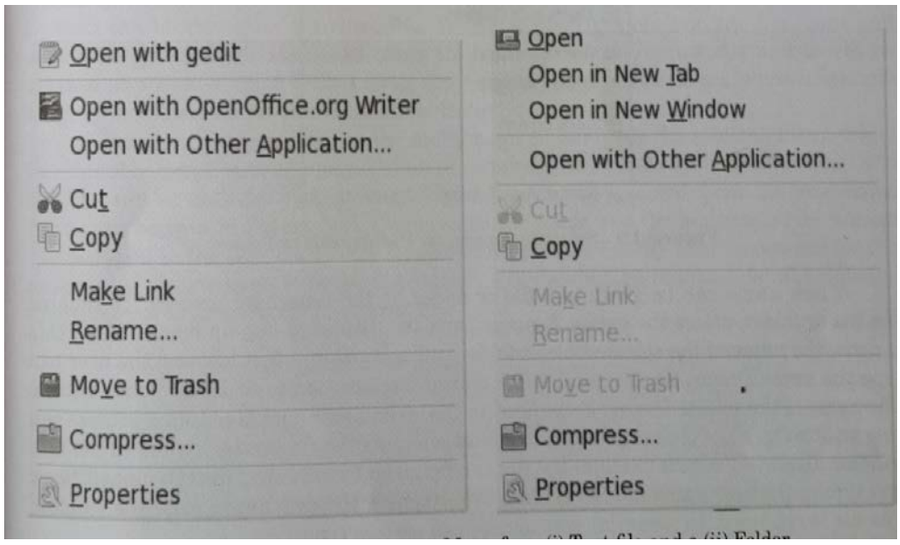


When working in KDE, right clicking on the empty space displays a pop-up window as shown in Figure below, Select the option **Create New** from the list. This displays another window. Here more options are available. A folder can be created by selecting the first option **Folder** from the list. A list of different file types are listed in the menu and depending on the type of file to be created, the user can choose any option from the list.



11.6 Managing Files and Folders

There are several management operations associated with files and folders in Linux operating system. These are *opening, moving, copying, deleting, and renaming* files. All the operations on files and folders are done by choosing the required option from the pop-up menu, when the file or folder icon is right clicked. The pop-up menu displayed when the file icon and the folder icon are right clicked is slightly different and these two pop-up menus are as shown in Figure 5.8. The first options of this menu are related to the opening of the selected files or folders. Files can be opened using the default application or a using another application. To open with the default application, select the first option. To use another application from the displayed list of applications, select the option **Open with Other Application**. A list of applications is displayed. Select the application from the list and proceed to finish the process.

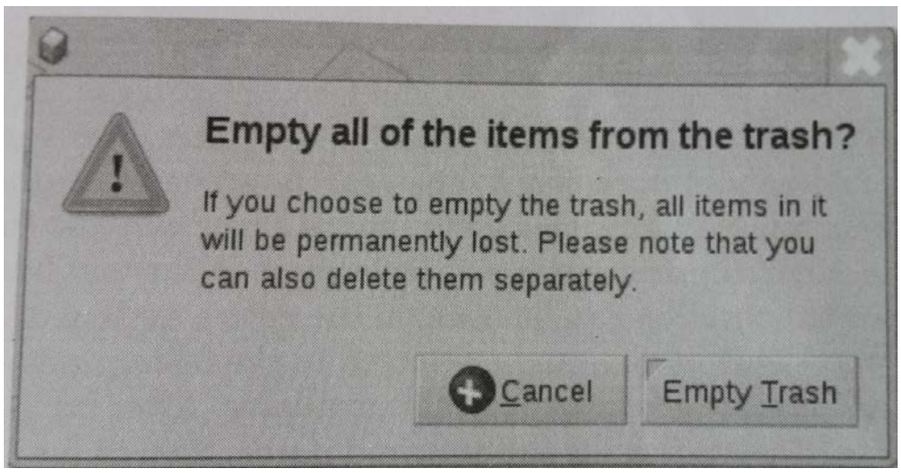


Using the copying process, a file or folder can be copied from one location to another location. To copy a file or folder, first right click the file or folder icon and select the option **Copy** from the pop-up menu. When this is done, the selected file is copied by the mouse. After copying the file, move to the target location and open the target folder. Right click on the empty space in the target folder and select the option Paste from the pop-up menu to paste the copied file in the target location. Some Linux versions show the option as Paste URL, as shown in Figure 5.9. The shortcut keys ctrl+v can be used for pasting the object. Working in a similar way, any file or folder can be copied from one location to another location. To move a file from one location to the target location, select the option Cut instead of Copy in the first step. Copy operation creates a copy of the file in the new location whereas cut operation moves the file from the original location to the new location. If a file or folder having the same name as the copied one is remaining in the target location, a window appears on the screen asking for overwriting the existing files or merging folders. Copying or moving takes place if a confirmation is given. Pasting several objects to the folder makes the folder surface cluttered and unordered. Clicking the option icons from the window

provides option to arrange the objects in an ordered manner sorted based on the different parameters.

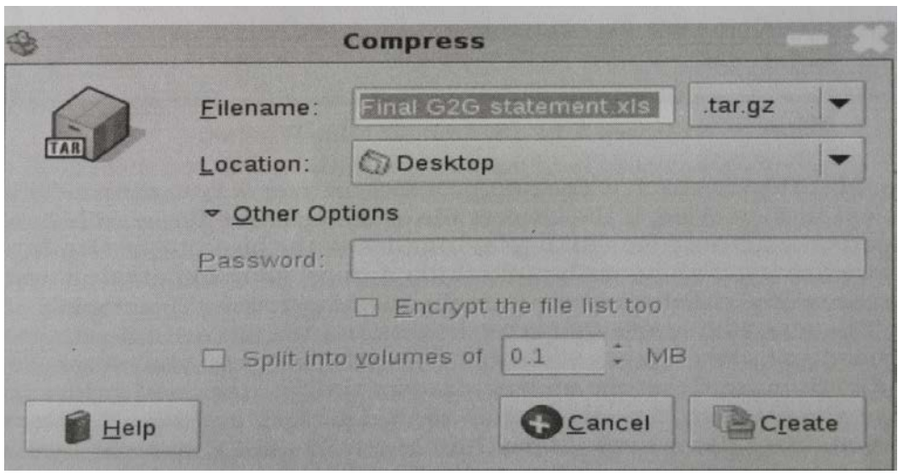
A new name can be given to a file or folder by the renaming process. To rename the file or folder, select the option **Rename** from the displayed pop-up menu. When this is done, the name of the selected file or folder is displayed in a text box and the user can type the new name in the text box. After typing the new name, click the mouse button. The name of the selected object is changed to the given new name. Renaming process can also be done by right clicking the file icon and selecting the **Properties** option. In the new window displayed, a new name to the file can be given by selecting the tab named **Basic** and typing the new name in the text box. After typing the new name, close the window. The file is renamed.

To delete a file or folder, select the file or folder and press the Delete key of the keyboard. The same thing can be achieved by right clicking the icon and choosing the option Move to Trash from the options. The deleted files are moved to the trash folder. The contents of this folder can be viewed by opening the trash folder icon. In several Linux distributions trash folder can be seen on the desktop. Moving the file to the trash folder does not delete the object permanently. Files available in trash folder can be restored to their previous state and to the previous location by the restore option. Restore option can be selected from the list displayed when the deleted object icon is right clicked. For permanently deleting the deleted files, select the option to empty trash or delete permanently. Selecting this option displays a conformation window as shown in Figure below.. Clicking the Empty Trash button removes the deleted files permanently from the computer storage.



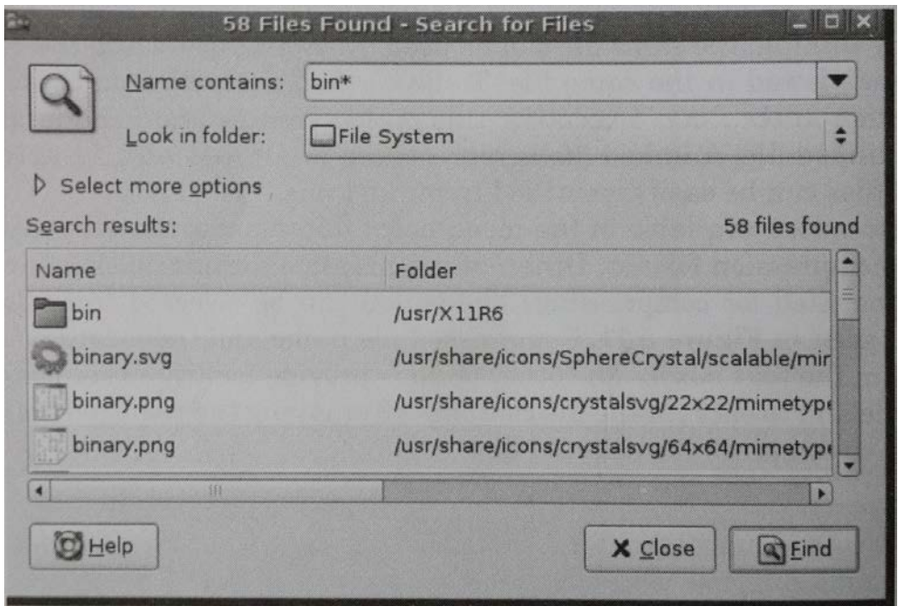
Compress option available in the menu helps to compress the selected file or folder in a suitable compression format. Different compression formats such as ar, jar, zip and the like can be used for compressing. The option can be selected from the drop-down list as can be seen in Figure 5.11. Compressed file name and the location for its storage can be typed in the text boxes. The other options provide facility to set password for the compressed file and to encrypt the file

content. After giving the required details, click the Create button to compress the selected file.



11.7 Searching for Files

Another common file and folder operation available in this operating system is the searching for files, folders and documents. The search option is available in the **Places** menu at the Linux desktop. Selecting this option opens a new window. In the new window, type the name of the folder, file or document to be located, in the text box provided for this purpose. The searching operation can be done on the entire system or in specific folders or selected devices only. This option can be selected in the folder name box. After typing the details, click the **Find** button to start searching for the required items. Search for files is made in the specified directory as well as in the subdirectories of the directory. The details of located files are displayed in the Search results text box appearing at the bottom of the window, as displayed in Figure below.



11.8 Linux File System

The file system used by the operating system plays a major role in the performance of the operating system. Different operating systems use different types of file system formats. Extended file system, reiser file system, high performance file system, virtual file system, NTFS, XFS, FAT, LVM and so on are some of the common file systems used by different operating systems. Extended family of file systems is the most commonly used Linux file system. Current versions of Linux operating system uses a file system known as **ext4** or the fourth extended file system. This is the improved version of the previous file system known as third extended file system or **ext3** file system. The major characteristics of this file system are bigger file system size, larger file size, faster file system checking and the like.

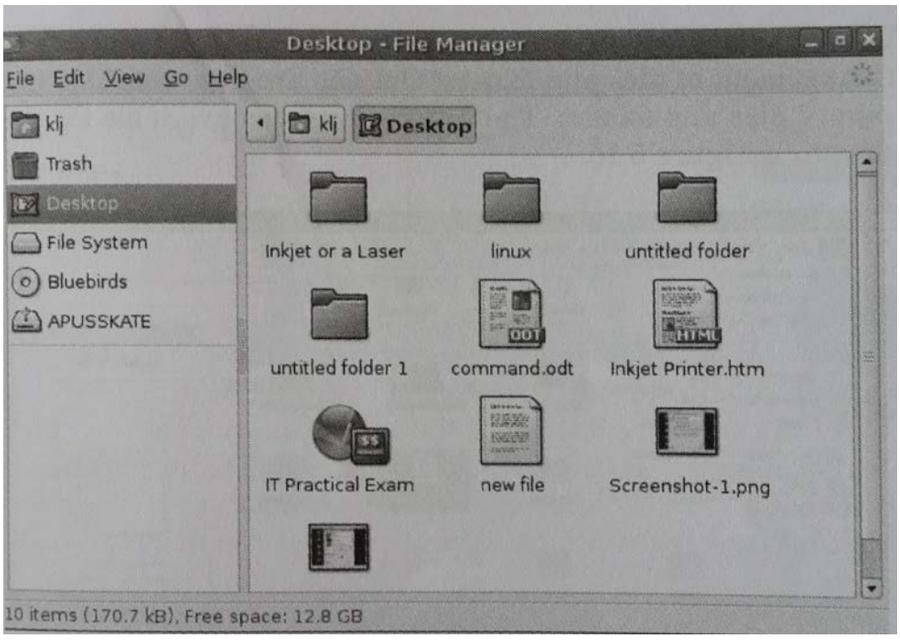
Ext4 file system makes use of 48 bit block addressing scheme. So this file system can address 248 (2 raised to 48) blocks. So this file system can have a maximum

file system size of 1 EB (Exa Byte) and a maximum file size of 16 TB (Tera Byte). Ext3 file system allowed a maximum of 32,0000 sub folders in a single folder. But in ext4 file system there is no restriction on the number of sub directories.

11.9 Linux File Managers

File managers can be used to perform a variety of activities such as creating folders and documents, organizing files into folders, displaying files and folders, searching and managing files, running scripts and launching applications, customizing the appearance of files and folders, opening locations in the computer, writing data to CD or DVD, installing and removing fonts, managing the desktop etc.

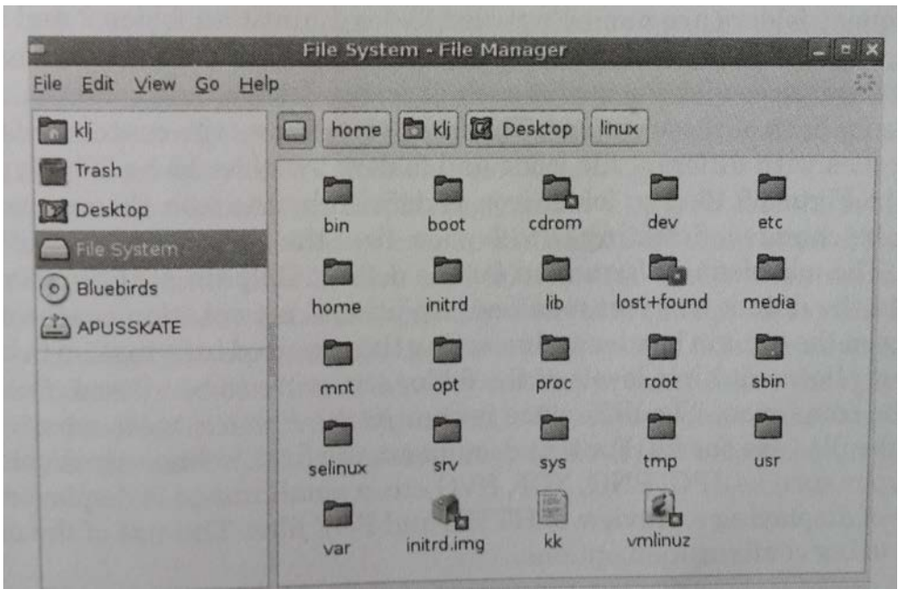
File managers control different file activities, control hardware, switch between users and keep flow of data between users. File managers always run when working the system. If required, file manager applications can be activated either from the menu appearing at the top panel in GNOME desktop or by double clicking any folder icon on the Linux desktop. Different Linux distributions have different file manager programs. In GNOME, file manager application is available in the list of tools displayed when **System Tools** is selected from the list appearing on opening **Applications** menu. File manager available in majority GNOME versions is known as **Nautilus** and for KDE it is either **Konquero**r or **Dolphin**. The display of Nautilus File Manager window is shown in Figure below.



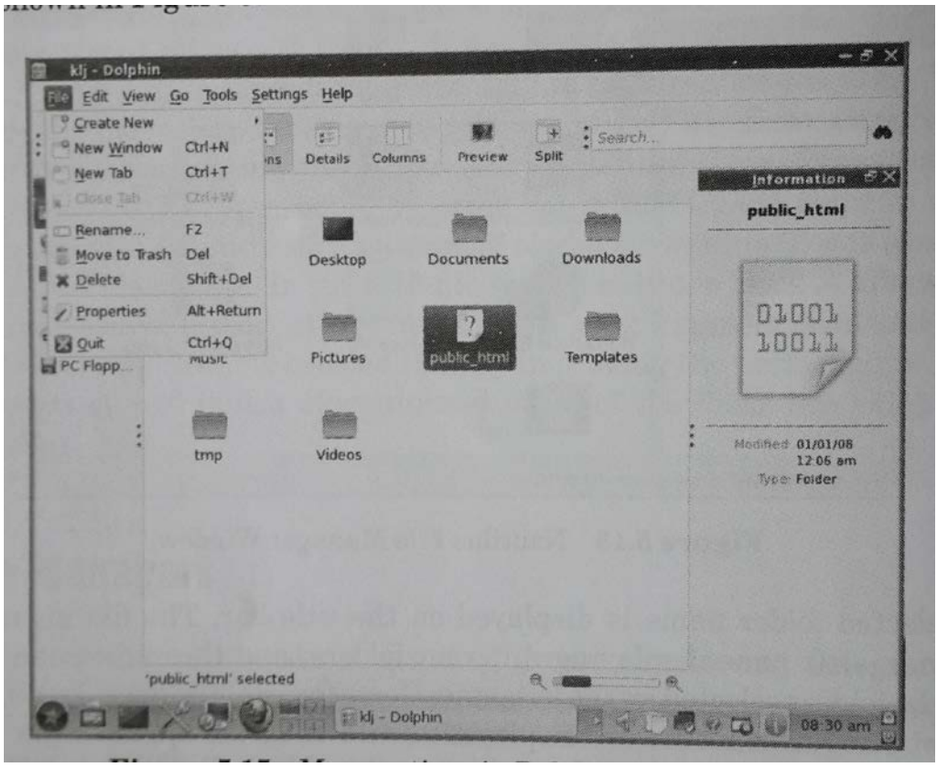
File manager window is similar to the file browser window. The default interface of the file manager window is made up of title bar, menu bar, tool bar, location bar, workspace, information panel, status bar etc. This layout varies with the distributions and the versions installed. The location bar displays the path to the current folder. Workspace shows the contents of the current folder. Status bar shows the data related to the status of the selected file or the selected folder. When no file is selected, the status bar shows the number of files and folders in the current folder.

The selected folder name is displayed on the title bar. The file manager window has two panes——left pane displaying different folders, and the right pane showing the contents of the folders selected on the left pane. Home folder of the user, Trash, File System and different removable media are displayed as folders in the file manager window. The desktop is also displayed as a folder. Selecting this folder displays the contents of the desktop on the right pane. In the file manager window, the root folder is indicated as a square button. When File System is

selected on the left pane of the file manager window, details of the hierarchical arrangement of files in the file system is displayed on the right pane. The display is as shown in Figure below.



Several options are available on clicking the menu in the menu bar and the options help in the management of files and folders. Options are also available in the File menu for creating empty files and folders. The different options available for the Dolphin File Manager is shown in Figure below.

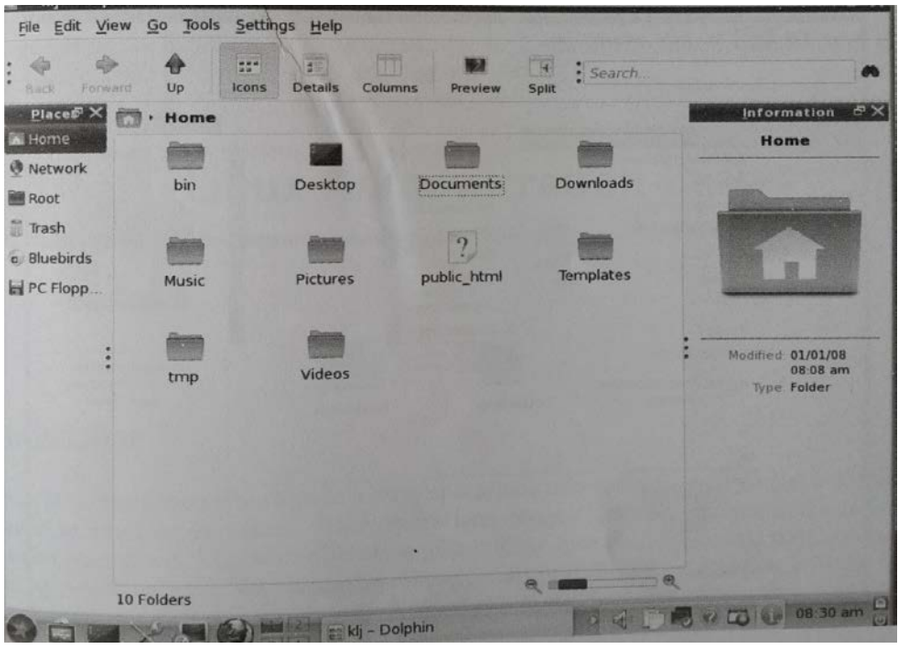


New folders and files can be created, files and folders can be renamed and can be deleted by selecting the appropriate menu options. Instead of selecting the menu options, keyboard shortcut can also be used. Creating folder option creates a folder below the current folder. By default, Linux assigns the name untitled folder to the newly created folder. Subsequent folders are named untitled folder 1, untitled folder 2 and so on.

Several view options can be set using the options available when opening View menu. Usually, there are three viewing modes such as *icons, details,* and *columns*. The **Icons** view shows the contents of the folder by representing files with different file icons and folders by folder icons. The typical display is as shown in the above figure. The **folde**r icon is defined by the icon theme currently used, and

this can be changed. Selecting **Details** view lists the contents along with additional information. The additional information for the default Dolphin configuration is the size of the file and the date it was last modified. Additional information can be displayed by right clicking on the column header and selecting the required information to be displayed. Columns view allows multiple levels of the folder structure to be viewed. Preview view is similar to the Icons view. The difference is that in the Preview view, wherever possible, previews of the files are shown. For text documents the first 14 lines are displayed and for image file types such as J PG, PNG, XCF, SVG etc, a small image is displayed. Dolphin is also capable of displaying a preview of HTML and PDF files.

File managers are capable of displaying hidden files and folders available in the selected folder. Hidden files and folders can be distinguished from other files and folders by the colour used. Usually hidden files and folders are fainter than normal files and folders. Hidden files and folders are displayed by selecting the option **to view hidden** files. File managers can sort the contents of the current folder in a number of ways such as by name, date, size and in descending order. For sorting process select the option from the View menu. The process is clear from Figure below.. The Information panel shows extended information about the selected file. The details are listed just below the large preview of the file.



11.10 Assignment 11

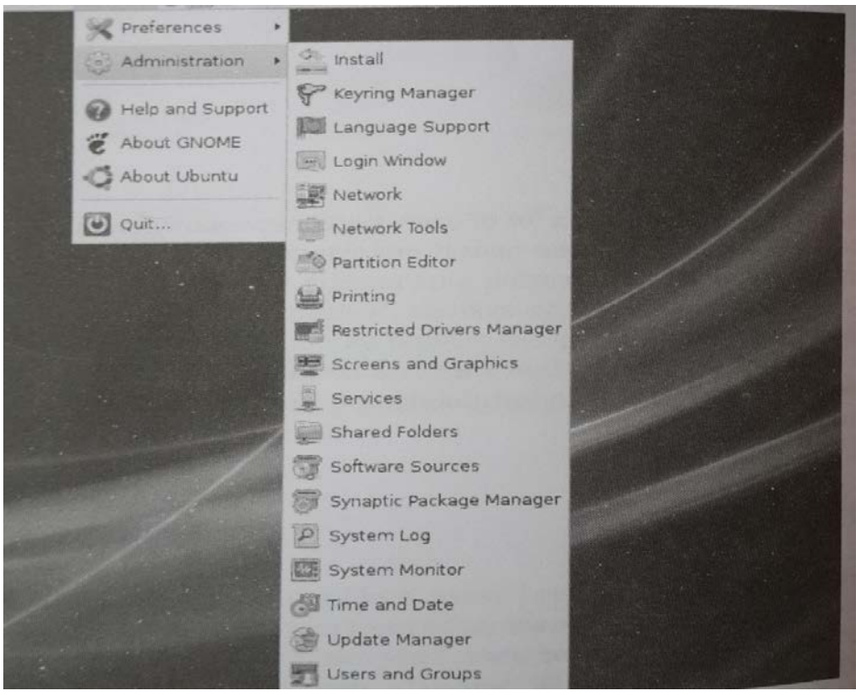
1) Write a note on Linux Files and Folders
2) Write a note on path and path names.
3) With a neat diagram , explain Linux System File structure
4) Explain the steps in creating Files and folders in Linux

UNIT IV
Chapter 12
Linux Administration Basics

12.1 Administrator Powers

A Linux system will be having several users, and Linux treats different users differently. Out of the different users, the superuser is the most powerful user. Superuser is also known as the root user or the administrator user. This user is created at the time of installation of the operating system and has a separate password known as the administrator password. Administrator user has unlimited powers.

1. Administrator user has access to all parts of the system including different file systems.
2. All the commands can be used by the administrator user, whereas other users can execute only certain commands, depending on the rights granted.
3. Local as well as remote systems can be mounted by the root user.
4. The superuser can control the system usage and set different configurations.
5. The superuser can modify the essential parts of the system including the setting of different languages for menus and windows.
6. Different log files can be viewed by the administrator user.
7. Superuser can create new users by assigning user names and passwords as well as can change the names and passwords of existing users.
8. Superuser can view as well as edit files created by other users of the system.
9. In multi-boot systems, the superuser can set the boot operating system.

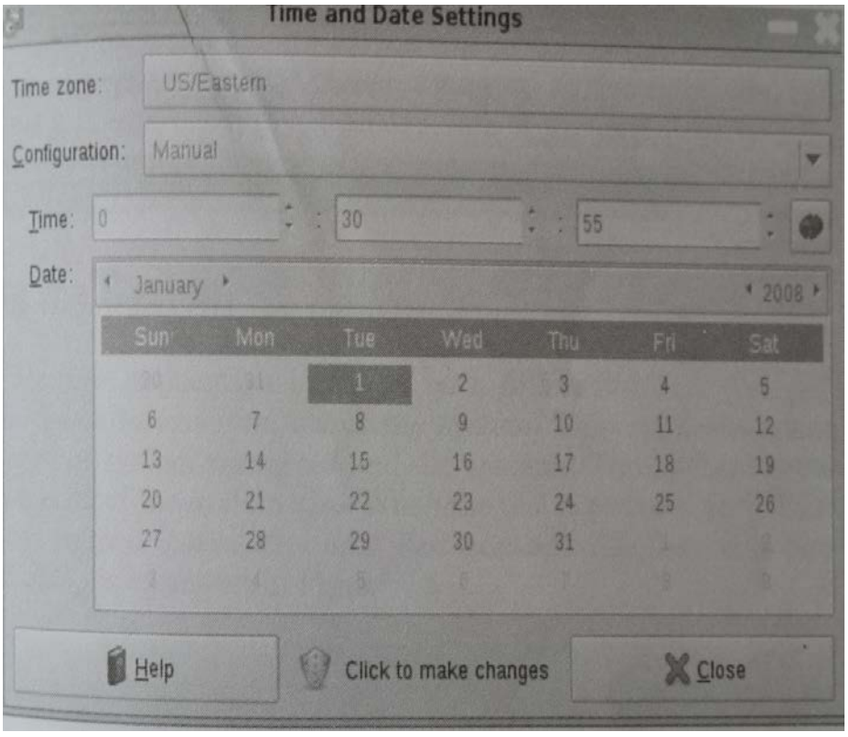


Different administration operations are done from System menu in GNOME desktop. When opened, this menu displays two options—**Preferences** and **Administration**. Different administration options are available by selecting Administration option. Different options available for the administrator for GNOME desktop is displayed in the above figure. When the **administrator** option is activated, the system prompts for the administrator password. Proceeding further stages for administration operations is possible only after giving the correct administrator password.

12.2 Administering Time and Date

The simplest administration operation when using Linux operating system is the administration of time and date. The time administration tool available with Linux distribution allows setting the time, date and time zone of the system. When the Time and Date tool is launched, it displays the Time and Date Settings window as

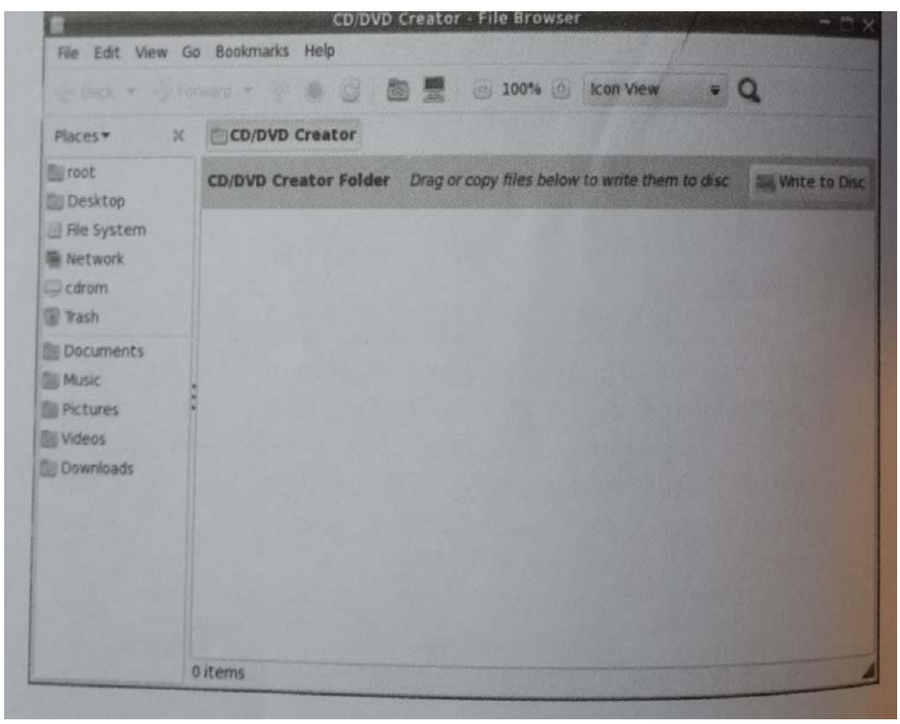
shown in Figure 6.2. Clicking the button-Click to make changes—seen at the bottom of the window helps in making a selection of the date in the calendar and the user can make necessary changes. Using the different options available, the administrator user can set different changes.



To save the changes made by the administrator, click the save button. The system prompts for the administrator password and the changes are saved when the correct password is given.

12.3 Writing to CD/DVD

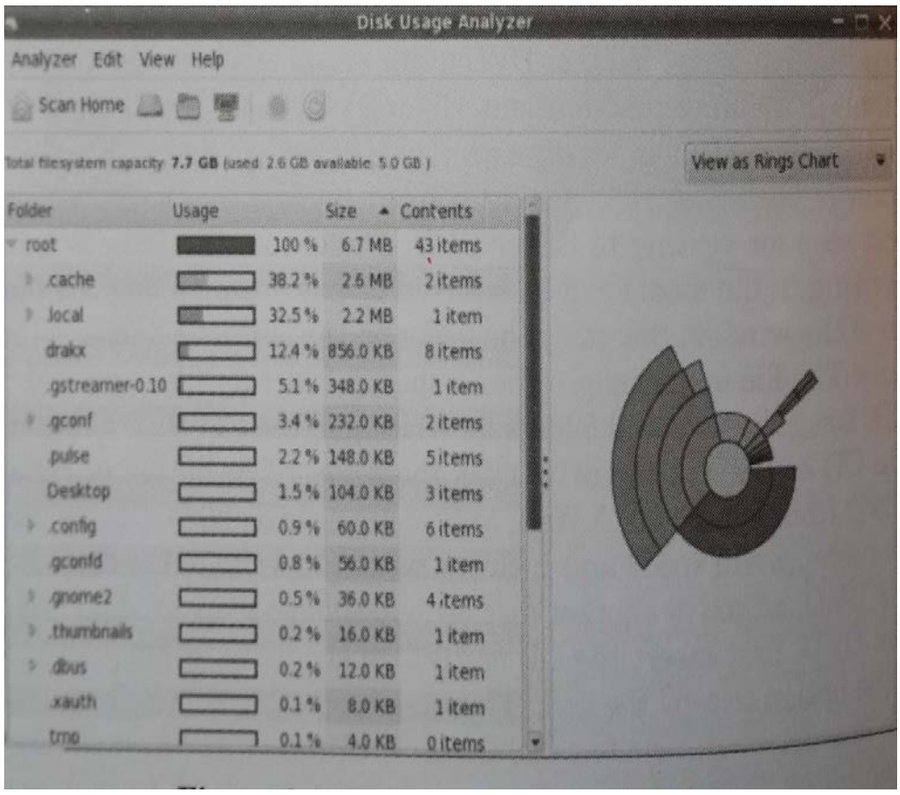
Writing to a CD or DVD is very useful for backing up important documents and files. For this, the computer must have a CD or DVD writer. When the application is launched, the opening window is displayed as shown in Figure below.



The first step for writing to CD or DVD is to choose the files. Required files and folders for writing to the media can be selected by navigating along the places listed on the left pane of the window. Selecting the folder on the left pane displays the contents on the right pane. Drag the files and folders for writing to the CD-DVD Creator folder. After inserting the CD or DVD in the drive click the Write to Disc button or choose the option Write to CD/DVD from File menu. A **Write to Disc** dialog is displayed. Set values for options such as disk, name, write speed and click the button to write to the disk. It is possible to copy a CD or DVD, either to another disc or to an image file stored on the computer. To copy a CD or DVD, first insert the CD or DVD in the drive. Right click on the CD icon and choose the option to copy the disk. This step opens the **Write to Disc** dialog window. It will then eject the original disk, and ask to change it for a blank disk. Writing takes place after inserting the disk.

12.4 Disk Usage Analyzer

Tools for analyzing disk usage are available in several operating systems. A commonly available tool in several Linux distributions is **Disk Usage Analyzer**. When the application is launched, the opening window is displayed as shown in Figure below.

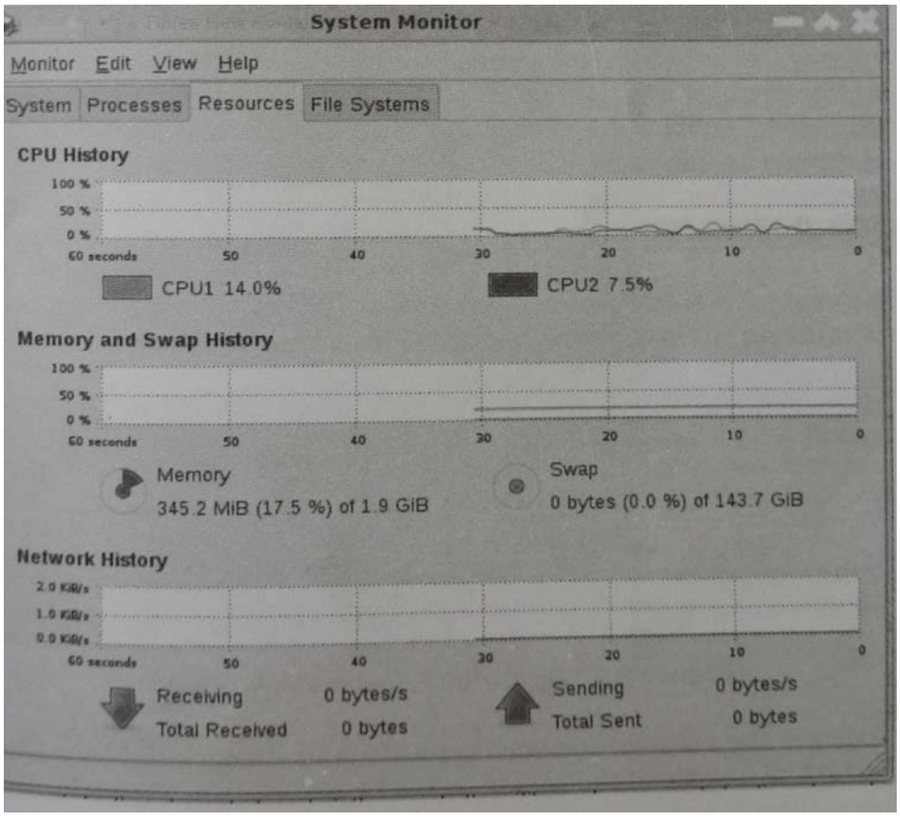


Folder and file wise memory usage can be obtained from the list. The details are arranged in a tabular format along with the size occupied by each item. A representation of the usage in a graphical format is also displayed on the right side. By studying the occupied memory, it is possible to get an estimate of

memory size that can be reclaimed by deleting some unused applications or outdated files.

12.5 Monitoring the System

The **System Monitor** application available with different Linux distributions enables to display the basic information about the system. This application monitors system processes, usage of system resources and file systems. When the tool is activated, a window having four tabbed sections named *System, Processes, Resources* and *File Systems* is displayed on the screen. The display is as shown in Figure below.



When the **System** tab is opened, various basic information about the computer's hardware, software and the system status are displayed.

Active processes and how various processes are related to one another can be obtained by opening the **Processes** tab.

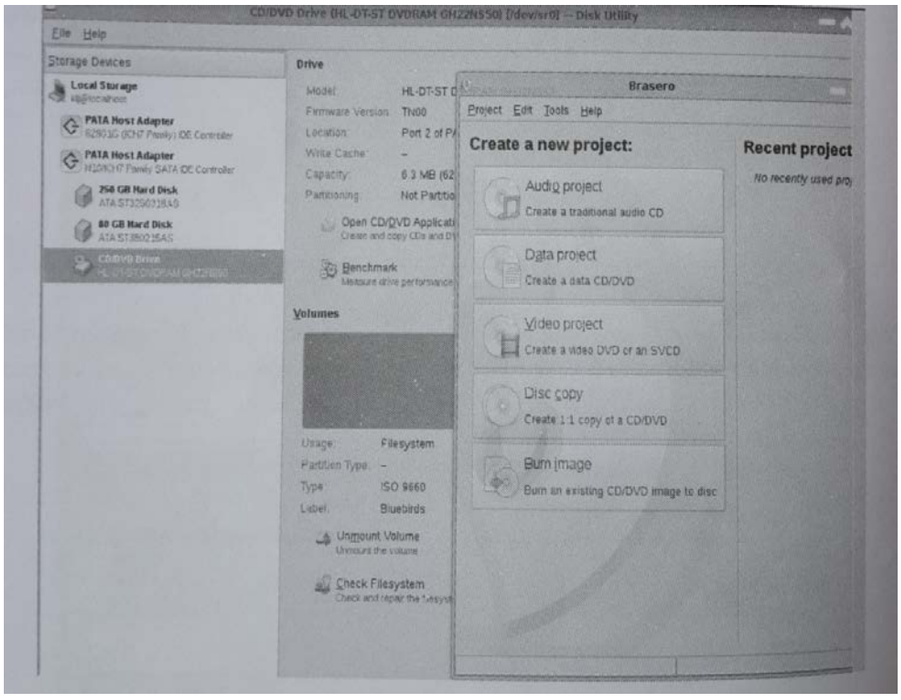
The current usage of different system resources such as CPU, memory, network usage can be obtained from the **Resources** tab.

File Systems tab lists all mounted file systems along with basic information about each one.

12.6 Managing Drives and Media

For managing drives and media, a disk utility application is also installed in Linux operating system. This application provides information about the drive as well as the different storage media connected to the system and provides options for mounting or dismounting the volume, checking file system, editing and deleting partitions as well as for formatting volume. A facility for measuring the performance of the drive is also available in some applications.

Disk Utility is the name of one such type of application available in Ubuntu Linux versions. The typical display of the screen, when launched, is as shown in Figure below.

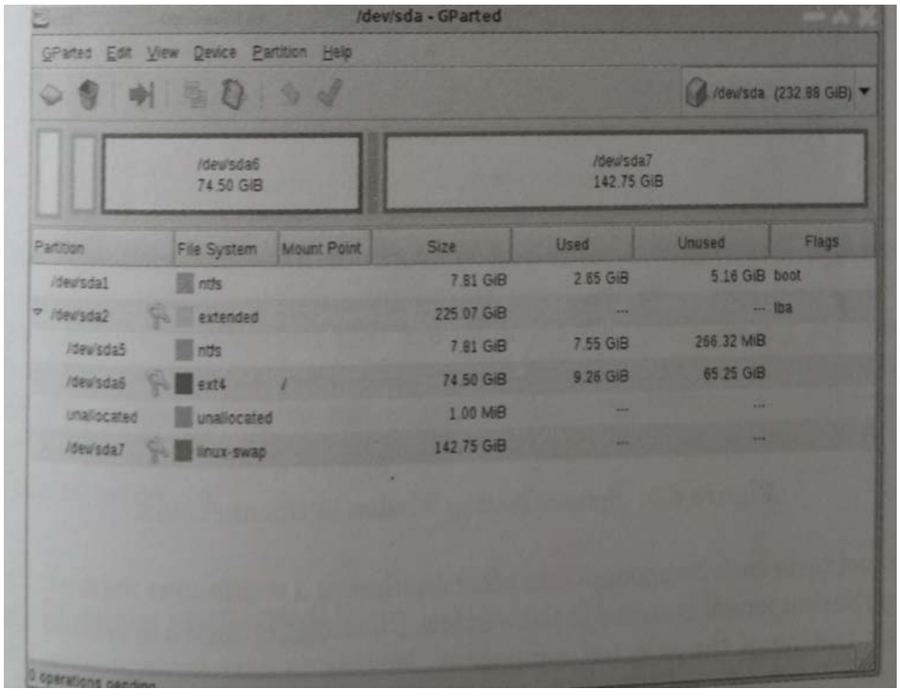


As we can see here, the window is divided into two portions. Left portion provides the details of the different storage media available. On selecting any of the media, more details are displayed on the right pane. A number of buttons are provided in the right pane and clicking the button performs the operation stated below the button. The common operations possible include formatting the drive, checking the file system, benchmarking, editing and deleting partitions and so on. In the figure, the option CD / DVD Drive is selected on the left pane and the various corresponding options appear on the right pane. Clicking the option to open the CD / DVD application activates Brasero application. This is the application for creating and copying CD and DVD. The application allows burning data to CD or DVD, burn audio CD, copy CD and DVD, create video DVD, create image files, erase CD and DVD and check the integrity of discs and disc images. Suitable option can be selected by clicking the required button from the window. To burn

data to a CD, click the button for copying data. This step opens another window and the user can select the required files for writing to CD. After selecting the files click the button to write to CD and proceed to finish the process. A cover editor for CD and DVD is also available with this application.

12.7 Creating and Editing Disk Partitions

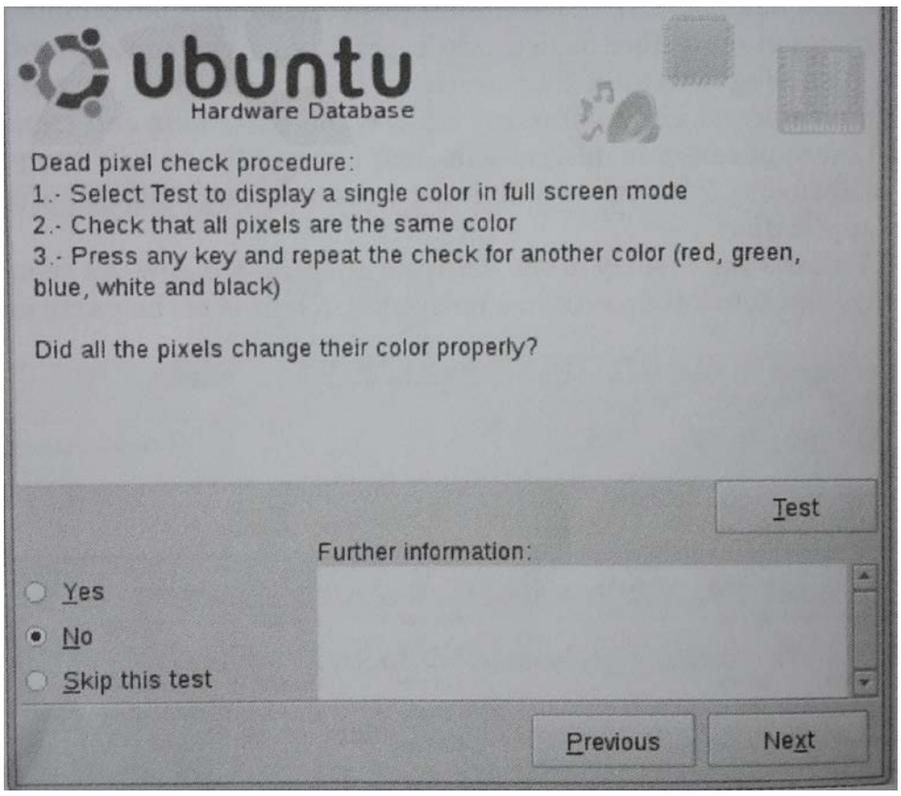
Different applications are available in different Linux distributions for creating and editing disk partitions. The application available in the GNOME version of Ubuntu distribution is **GParted.** The name stands for **GNOME Partition editor**. This partition editor enables the creation, reorganization, and deletion of disk partitions. Enabling and disabling partition flags such as **boot** and **hidden** is also done by this application.



The above figure shows the display of the **GParted** opening window. As seen in the figure, this tool displays the details of partitions in the disk. Details of the partitions of the hard disks displayed include partition name, file system used, mount point, size, used and empty memory sizes and flags.

12.8 System Testing

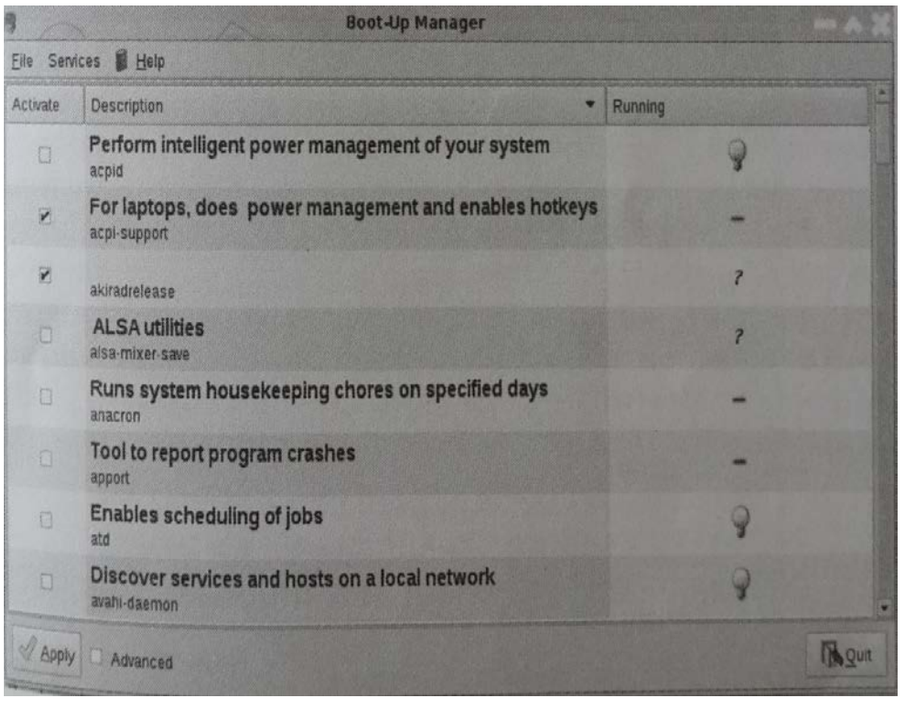
Launching the tool for testing the system opens a new window displaying all the hardware components attached to the system. In Ubuntu distribution the screen display is as shown in Figure below.



The tool tests each component one after another in a continuous manner. Procedure for testing the component is stated in the window. Clicking the button to test the component performs a testing of the selected component. Moving to next screens provides facility to test other hardware components also. Working in a similar manner all hardware components can be tested for their working.

12.9 Boot-Up Manager

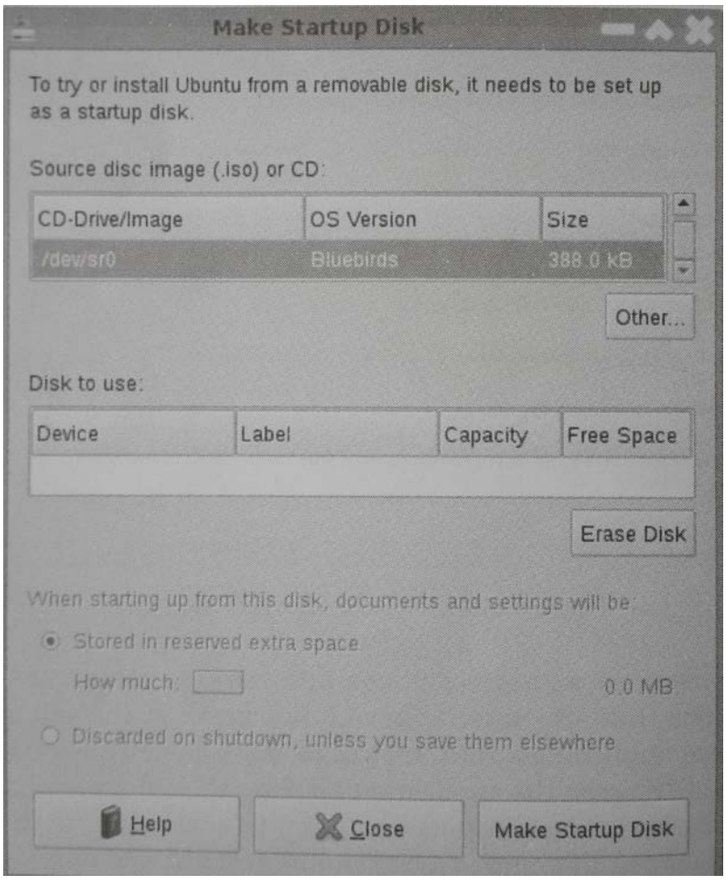
Boot-Up Manager is a graphical tool that provides facilities for starting or stopping services as well as for configuring applications to start at boot time. When this tool is activated, it displays the graphical window as shown in Figure below.



A list of applications with descriptions and working status are displayed in this window. Applications that are running have a tick mark in the square box seen on the left side while those applications that are yet to start do not have the tick mark in the square box. Right clicking on the application displays the option to activate or deactivate the selected application, depending on whether the application is in deactivated or in activated state. Thus running applications can be stopped and applications that are yet to start can be activated.

12.10 Making Startup USB Disk

USB Startup Disk Creator is an utility available in several Linux versions. This utility creates an image of the operating system in an USB medium. This image is known as a **Live USB**. Live USB can be used to install Linux on the computer for running Linux without affecting the system hardware. To make a live USB disk, plug the device to the USB drive of the system and make sure that the system recognizes the disk. Launch the application from the menu. This opens a new window as shown in Figure below.



Under the source disc image option, specify the image. The image from a CD can also be used for creating the live USB disk. After setting different options, click **Make Startup Disk** button. This step creates the required startup disk. The USB can be safely removed and can be used as a live USB for booting the system. To boot from the live USB disk, set the boot option to the disk and power on the computer. The system boots from the USB disk.

12.11 Viewing Log Files

Log viewer applications are available in all Linux distributions. One of the log viewer applications available in several Linux versions is **System Log Viewer**. System Log Viewer is useful because it provides an easier, more user-friendly display of different event logs. Also System Log Viewer comes with functions that can be used to manage the logs, including a log monitor and log statistics display.

When the application is launched for the first time, the application displays several log files by default. Opened logs are listed on the left side of the application window. Details of the log selected in this list are displayed on the right side window. By default, **System Log Viewer** monitors every opened log. Log information is displayed in the status bar and this includes the number of lines in the log, size of the log in bytes, date the log was last modified. The log can be copied if required and pasted to another editor.

12.12 Installing New Languages

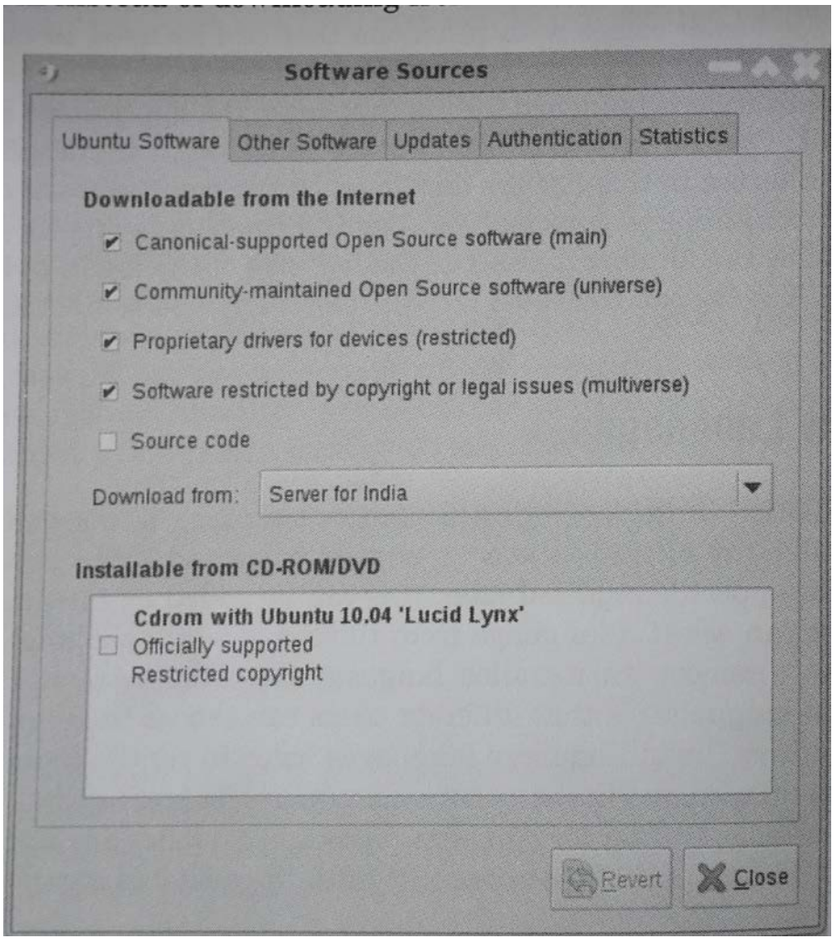
During the installation of the operating system, a language is selected for installation purpose. The administrator can select a language from the list of available languages in the list. It is also possible to remove an installed language from the system. The advantage of installing different languages is that different users can choose the required language in their language settings. Display for numbers, dates and currency amounts can be set using the new language. The changes take effect only after the next logging in to the system.

12.13 Downloading Applications and Installing New Releases

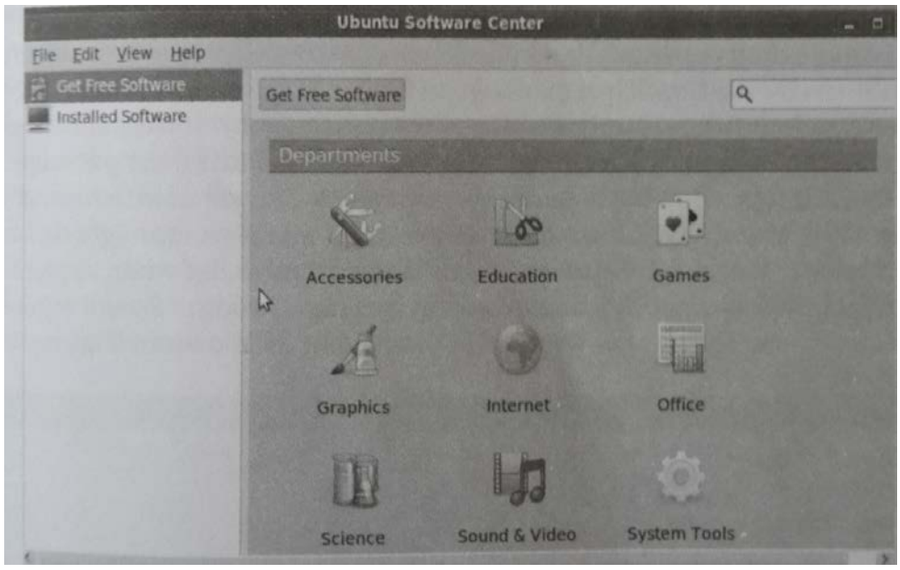
Several Linux application software are available for downloading from the Internet. The software available is of different types. These are available as community supported or canonical supported open source, in the form of proprietary software or in the form of licensed software. These different software are stored in different servers located at different places. Also several http and ftp servers archive Linux resources, which can be downloaded and can be used freely.

Ubuntu Linux distribution provides a tool for software downloading from websites and for checking whether the installed software has new releases available. **Software Sources** is the tool used for this purpose. When this tool is launched, it displays a window as shown in Figure below. Different options can be set for software downloading by opening different tabs appearing in this window.

New and updated Linux versions are released regularly. System updating involves the downloading and installation of the latest updates of the applications and packages used in the Linux version. Different updates are available from different sources. Using the above tool it is possible to configure the system such that updating takes place automatically at the set time intervals such as daily, on alternate days, weekly updates and so on. If the system is working in online mode, the updating takes place at the set time intervals automatically. If new Linux releases are available in a CD or DVD medium, the updating process can be also done using the medium instead of downloading from website.



A virtual catalog of several free applications suited for Ubuntu distribution is available in **Ubuntu Software Center**. When launching this tool, a window as shown in Figure is displayed.



Clicking the link Installed Software appearing on the left pane of the window displays the list of installed software in the system. Selecting **Get Free Software** on the left pane displays a list of departments on the right pane. Here different software are grouped under different departments or categories. Double clicking the icons of departments displays a list of available applications under the selected category. Suitable application can be selected from the list. The selected application can be installed with the click of a mouse button. An Internet connection is necessary to install an application available in this list.

To install a particular application available in the list, first locate the application using the **search** option. Click the application and then click the arrow button to go to the screen of the program. Click the button for installing the software. The time taken for installation depends on the size of the software and the speed of Internet connection. Once installed, its name appears in the **Applications** menu in the same category in which the application is listed in the **Software Center**. Working in this way, applications under different categories can be installed. To

remove an installed application, first locate the application in the list. Select the application and then click the **Remove** button appearing in the window.

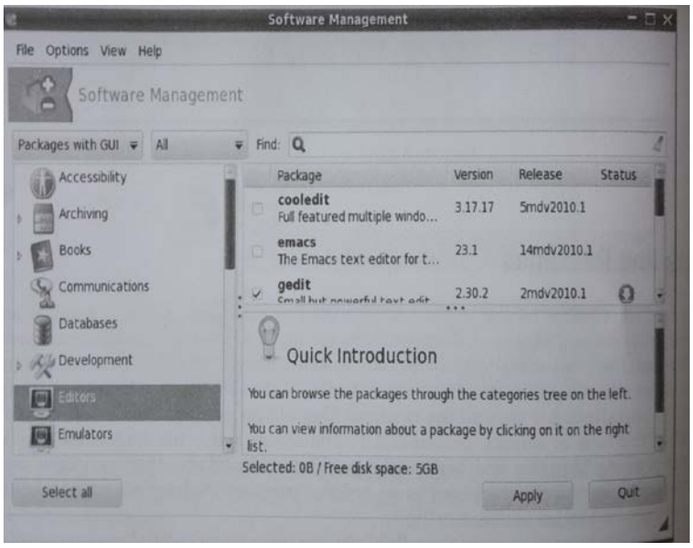
12.14 Installing Packages

Several Linux software applications are bundled as packages. Linux packages are mainly available in two types namely RPM packages and DEB packages. RPM stands for Red Hat Package Manager and DEB stands for Debian. These packages can be identified by their file names extensions. These files are executable files for Linux operating system. Several of these packages can be downloaded from the Internet.

To install a packaged file, the first step is to copy the files in the local folder. After copying the files, double click the file for installation of the package. Necessary permission to execute the file must be available for installing the package. The installation takes place only if all the dependencies are satisfied. Otherwise an error window will appear.

Another method for installing Linux packages is the use of **Linux Package Manage**r applications for installing packages. Package Managers install, remove, configure, upgrade and downgrade single and multiple packages. RPM based and Debian based distributions use two different package managers for package installation. But both package managers appear identical and work in a similar manner.

In Mandriva Linux distribution, the package manager is launched by selecting the **Install & Remove Software** option displayed in the list when **Applications** menu is opened. The option is also available by navigating through **System > Administration**. When the option is selected, a new window is displayed, as shown in the figure.



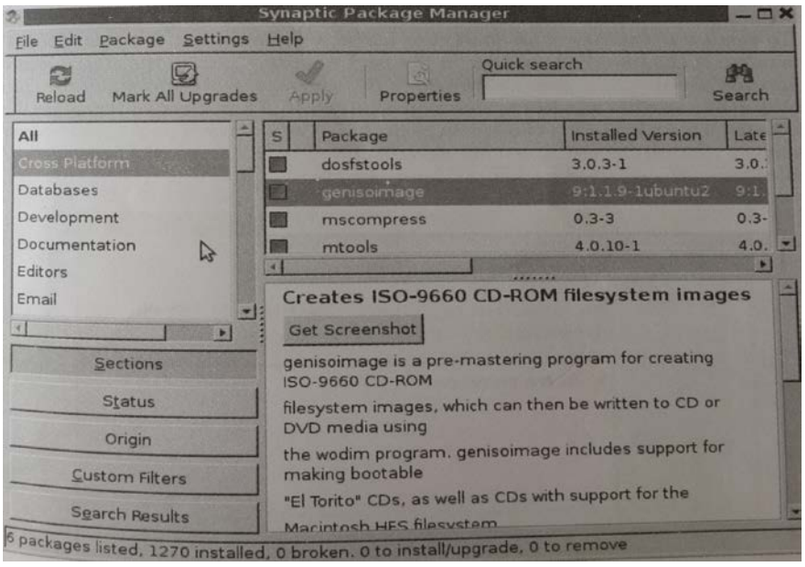
This new window is the **Software Management** window and this window lists all the installed applications, categorized under different groups. Browse the group to get the list of applications available under the group. Installed applications are indicated with a tick mark on the square box at the beginning of the package name. Right clicking on the application displays more details about the selected package in the description box.

To install a new package, click on the square box. This step opens a new window which displays a summary of additional memory needed. Click the OK button for confirmation of installation. Installation process starts and asks for inserting the medium on the drive. Insert the medium and click OK button. The package is installed and the name of the application appears under a group in the **Applications** menu, in which the package was displayed.

To remove the installed software, first locate the software in the package list in the **Software Management** window. Remove the tick mark in the square box appearing on the left of the name of the application by clicking on the square box. Click the **Apply** button appearing on the bottom of the window. This step opens a

new window in which a summary is displayed. Proceed by clicking the OK button to remove the software.

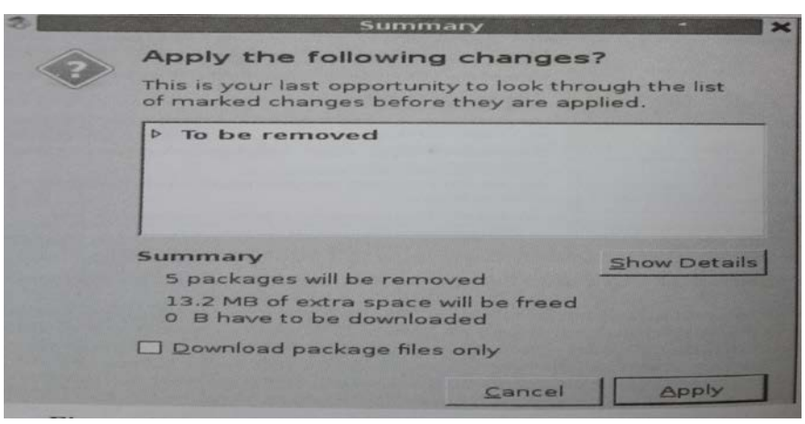
Ubuntu Linux distribution makes use of **Synaptic Package Manager**, as the package manager for installing packages. To install packages using this package manager, activate the application by selecting **System** from the menu bar and choosing **Administration** option and then **Synaptic Package Manager**.



Similar to any window, this window has a title bar, menu bar, tool bar and so on. Opening the menu bar displays a set of options. Toolbar displays a number of icons for providing main actions. The window is mainly divided into a left pane and a right pane. Depending on the selection of category made on the left pane (category selector window), the corresponding details of available and known packages will be displayed on the right pane. The installed packages are indicated by a dark square at the beginning of their names where as packages that are yet to be installed are indicated by a white square at the beginning of their names. Right click on the name of the package. A short description of the selected package will be displayed in the bottom right pane (description field).

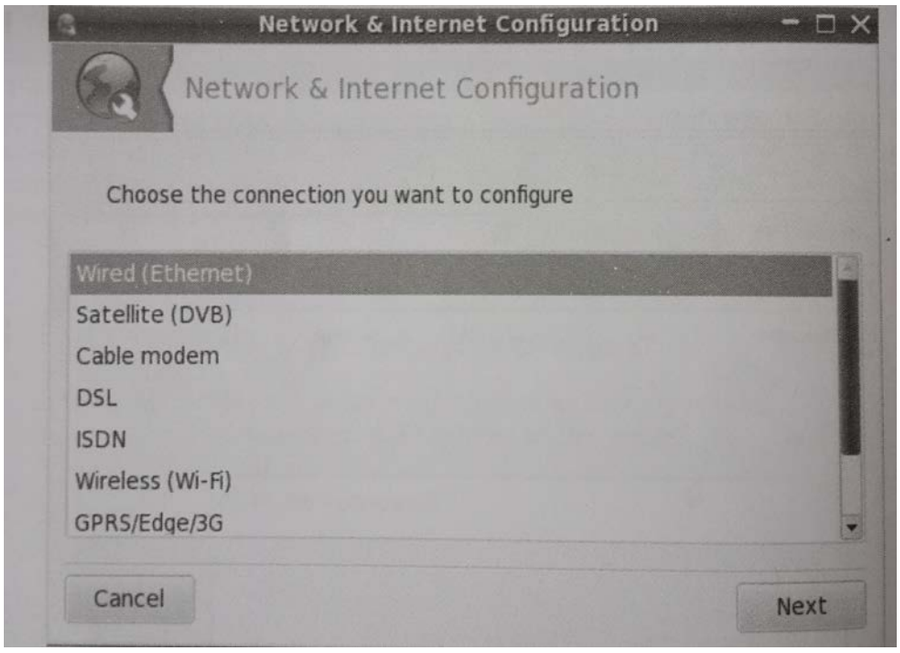
Installed applications can be marked for removal or upgrading, while those are yet to be installed can be marked for installation, from the options displayed on the pop-up window. Options can also be set by opening Edit menu of the application. Different menu options are for reloading the packages as well as for marking upgrades of packages. After making changes, save the changes by clicking the Apply button on the tool bar. Clicking the Reload button on the tool bar reloads the packages from Internet sites, if the computer is working online. Otherwise a downloading error will appear on the screen and the process will come to a halt. Packages are made available through so called repositories. Repositories can be located on different media such as CD/ DVD, local hard disk, the Web (http, ftp) or in remote file systems. A repository contains the packages and along with them, an index that provides basic information about the packages, such as the required dependencies and short description.

To remove a package, first mark the package for removal. For this, double click on the name of the installed package in the package list. Click on the status icon of the package and choose the option for removing the package. Save and apply the changes. This step opens another window as shown in the following figure. This is the confirmation window and asks for the confirmation of selected packages. A summary of packages removed and the memory that can be reclaimed are also displayed in this window. Clicking the Apply button removes the selected packages from the system.

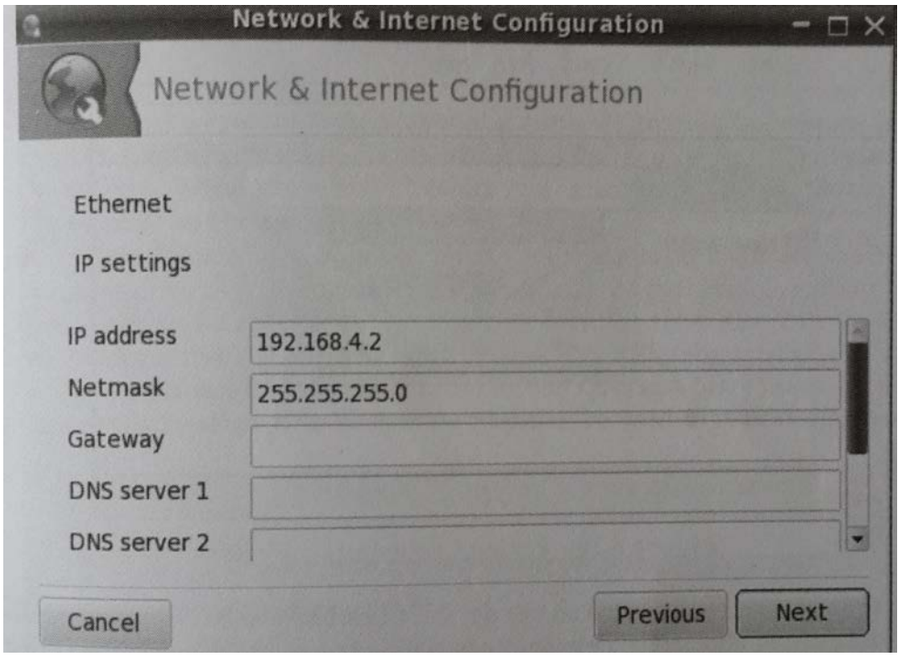


12.15 Setting Up and Managing Computer Networks

Several tools for **configuring and managing** computer networks are installed by default during the installation of Linux operating system. In GNOME desktops a network icon is displayed on the top panel. Right click on this icon and select the option **Configure Network** from the displayed list. This operation starts the **Network & Internet Configuration** wizard. The first screen displays the list of different connections available for configuring .



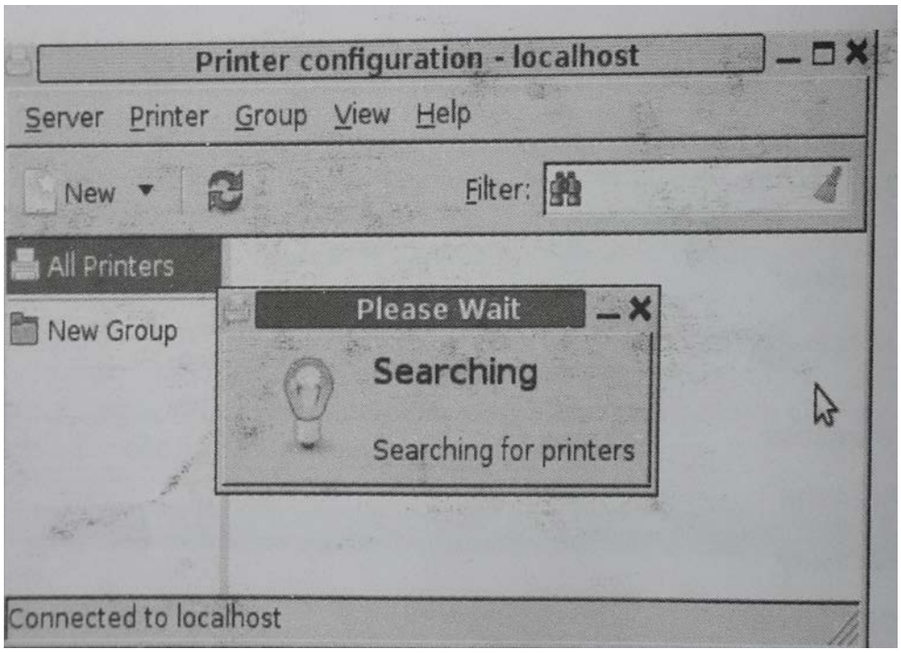
The different connections can be wired network, wireless network, using cable modem etc. For setting up a wired network, choose the option Wired (Ethernet) from the list and click the Next button. The computer detects the Ethernet interface available and its name is displayed in the next window. Configuring the protocol for connection can be made either automatically or manually. Select the manual configuration option and move to the next window. The new window displayed as shown in the figure below.



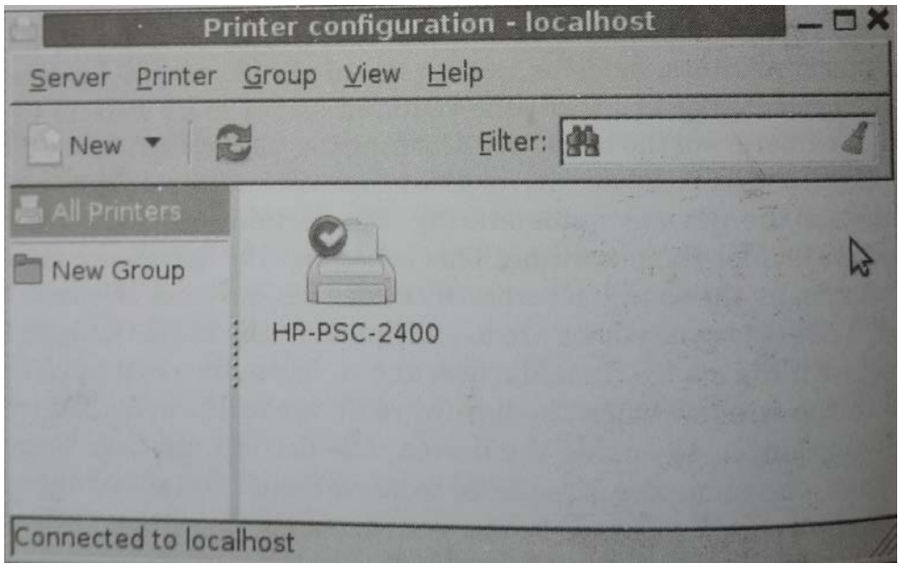
The IP settings are done in this new window. Set the IP address and netmask for the computer. In this window the IP address is given as 192.168.4.2 and netmask as 255.255.255.0. Click the Next button and proceed further and finish the processes. The same step is to be done in all the computers connected in the network. The IP addresses used in different computers must be like 192.168.4.3, 192.168.4.4, and so on. The netmask used must be 255.255.255.0 in all the member computers in the network. Once the configuration is finished a connection is established between the networked computers by clicking the button for establishing connection. Using suitable tools or text commands it is possible to check whether a connection is established between the computers.

12.16 Adding Printers

Linux operating system allows adding new hardware to the system. Using this facility, it is possible to add printers, modems, scanners and the like. Linux operating system configures most of the devices automatically. But certain devices require proprietary drivers to be installed for their working. This is because the drivers for those devices are not freely available, as the manufacturers of the devices have not released the hardware details. Device drivers for any device are usually provided by the manufacturer and this is available along with the device. To add a new driver, locate the device driver file from the list displayed in the window when the hardware driver tool is launched from the menu. Press the activate button to enable the driver. The driver may have to be downloaded from the Internet. The computer is required to be restarted for the changes to take effect. To install the printer software in the system, select the option **Printing**. This opens the printer configuration window as shown in the following figure.

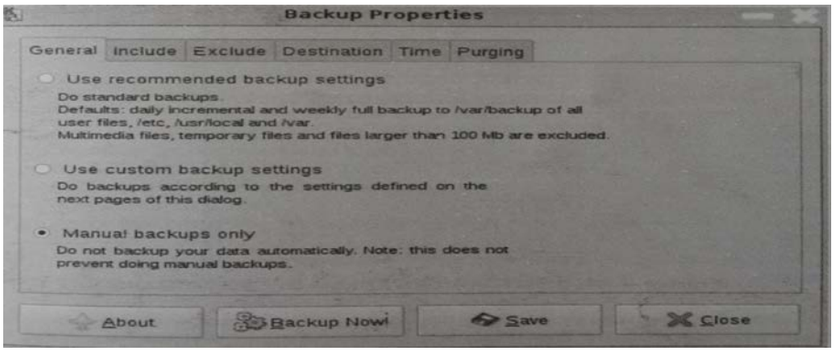


To add a printer, connect the printer to the computer and switch on the device. Click the **New** button in the configuration window. The system checks for new printers connected to the computer and installs the necessary printer software. Different configuration steps for a printer involve assigning port, setting time limits and the like. After selecting the parameters, proceed to install the printer by clicking the **OK** button. If the printer is successfully installed, its icon appears in the window as shown in the following figure. In this way more printers can be added to the system. Right clicking on the printer icon displays a pop-up window and the user can select different options for setting different printer configurations. Double clicking the icon displays the printer properties window and it is possible to set different configuration options for the printer, in this window.



12.17 Backing Up Files

The backed up files are used for restoring the system to the previous state whenever the system fails. Linux installs tools for simple backups of the system. Tools for restoration of the system are also installed by default. When the system backup tool is launched, the window as shown in the following figure appears on the screen.

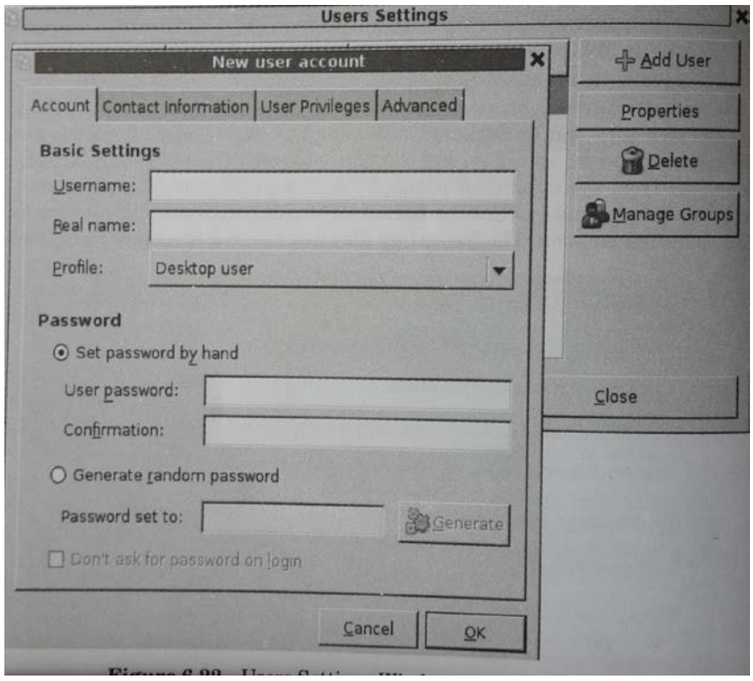


As seen from the above figure, it is possible to set different options for availing system backups. To backup the system manually, click the button for backing up files. This step backs up the files in the selected location. The backed up files can be used to restore the system.

12.18 Managing Users and Groups

A major function of Linux administrator is the management of users and groups of the system. User and group management jobs involve adding and deleting users and groups. During the installation of Linux, the administrator user is created. Other users are created and the passwords are assigned by the administrator user. When the tool for managing users and groups is launched, it opens a window titled **Users Settings** and this window displays the details of current users and groups.

To add a new user, click the button **Add User** displayed on the right side. This step opens a new window titled **New user** account, above the previous window, as shown in Figure 6.22.



The window has a number of tabs and opening each tab displays a set of options for user configuring. Opening the first displays a number of text boxes for typing the details of the new user. After typing the details, click the OK button to add the new user. A user can be deleted by selecting the user from the list and then clicking the **Delete** button. Several other user configuration options can also be set using this tool. Users are added to different groups. Users belonging to different groups have different powers and abilities. Managing user groups involve adding new groups, adding users to groups, deleting groups and so on.

12.19 Assignment 12

1) Write a note on Administrator powers
2) Write a note on Writing to a CD/DVD
3) How do you monitor the system in Linux?
4) Write a note on managing Drives and Media

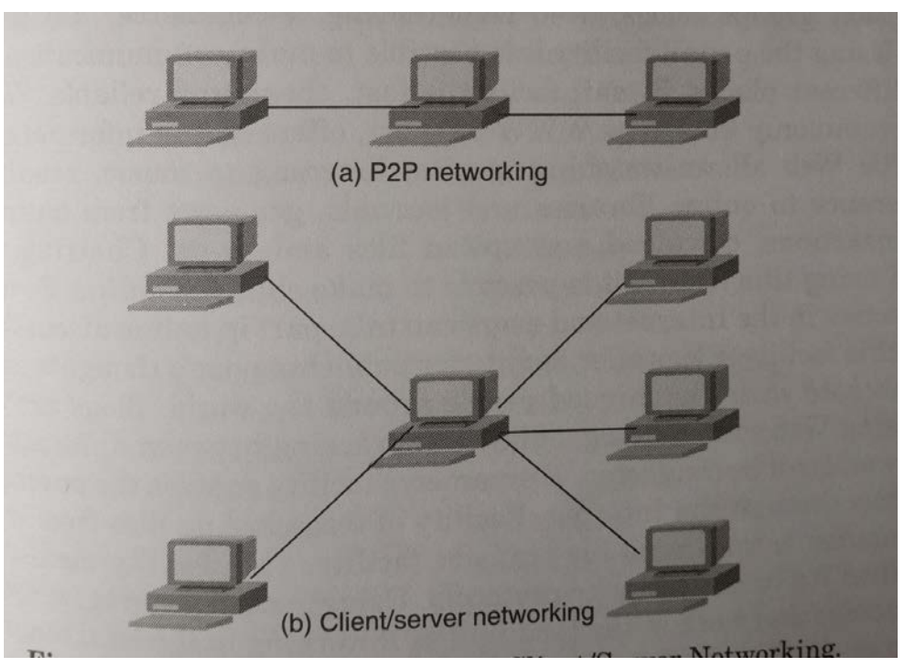
Chapter 13
Networking and Using the Internet

13.1 Networking Fundamentals

A computer network consists of a group of interconnected computers such that any computer in the network can share files as well as other resources available in the network. Depending on the medium used for linking computers, the networks can be either a wired network or a wireless network. OSI networking model, introduced by Open System Interconnect, is the standard model used for networking computers. This model divides the functions of networking protocols into a series of layers. Each layer is called a stack and can be implemented either in hardware or using software. The different layers from the bottom level to the top level are named as physical layer, data link layer, network layer, transport layer, session layer, presentation layer and application layer. Physical layer is the lower most layer and this layer defines the different hardware components of the system. Application layer is at the top in the hierarchy and this layer acts as the interface of the user to interact with the system.

For networking computers or devices, certain protocols are necessary for the effective working of networked systems. In computer networks, data is transmitted between the networked computers by dividing the data into a number of packets. At the receiving end, the different data packets are assembled in the correct order to create the file. TCP/IP has become the universal standard suite of protocols used to provide connectivity between networked devices. This protocol ensures that the data received at the receiving end is exactly the same as the data sent. This protocol also ensures that the data is transmitted without any errors. The IP address is divided into four sets of 8 bits numbers separated by decimal points. Each number in the address can vary from 0 to 255. Since there are four numbers in the address, the system is known as IPv4 (version 4). The emerging standard is the IPv6 (version 6), which makes use of six numbers in the address. New Linux distributions are capable of using both types of standards.

Basically, computer networks are of two types. In one type the member computers are connected in a serial fashion. The first computer is connected to the second computer, the second one is connected to the third one and so on. This type of networking computers is known as peer to peer networking or P2P networking. In the second type of computer network, a number of computers are connected to a central computer. The central computer is known as the server computer and the computers connected to it are known as client computers. This type of network architecture is known as client-server architecture. The arrangement of computers in these two types of networking is clear from the following Figure.



The clients make requests to the server for different operations and the server satisfies the requests of client computers.

Computer network formed with the member computers spread around a small area is called a Local Area Network or LAN, in short. Such LAN computers are interconnected to form a bigger computer network spread in a wide area and this type of network is known as a **Wide Area Network** or WAN, in short. **Metropolitan Area Network** or MAN is the name given to the computer network spread around a metropolitan area.

A computer network can also be formed by linking all the computers of an organization located at different places. This is a private computer network for the exclusive use of the organization. Such type of private network is known as an intranet.

Depending on the area of operation computers are linked using different media. A wired network is formed using cables while wireless networks make use of satellite links or radio signals for linking.

13.2 Basics of the Internet

The Internet has its origin in 1968, when the Department of defense, USA built a network of computers called **ARPANET**. The Internet is formed from a large number of networked computers. This large network of computers is made up of several interconnected LANs and WANs. The different computers forming the Internet are working on different operating systems and are based on different protocols. Working of the Internet is based on client/server architecture. To connect computers working on different systems, TCP/IP is used as the standard protocol of the Internet.

The Internet provides several services. The most widely used Internet services are e-mail or electronic mail and World Wide Web (WWW). The other services offered include chat, discussion groups, blogs, Web conferencing, e-commerce, file transfer, remote logging etc. Using the e-mail facility it is possible to make communication between users located at different places. E-mail facility is fast, cheap and reliable. World Wide Web or the Web, commonly known as WWW in short, offers online information on any topic instantly. The Web allows watching

movies, listening to music, reading documents, making reference to online libraries and journals, get news from online newspapers, conduct transactions, download and upload files and so on. Chatting facility is also popular and using this facility it is possible to make chatting online. Several discussion groups are active in the Internet and users can take part in online discussion on different topics with this facility. Blogs offer facility for publishing one's thoughts and ideas online and to collaborate with like minded people around the world. Blogs are also known as web logs. Using Web conferencing, online conferencing between different people located around the world can be conducted. E-commerce facility enables the purchase and selling of commodities through the Internet. Facility of transferring files from one computer to another computer is known as file transfer facility. This facility makes use of special protocols called ftp or file transfer protocols. Using remote logging, a user can log into another computer and work at the terminal as if working in the local computer. All these services can be obtained by running the applications installed in Linux operating system for the required purpose. To make use of the different facilities available in the Internet, it is necessary to connect the system to the Internet. Connectivity to the Internet is provided by service providers who are known as Internet Service Providers or ISPs. Service providers have different methods for connecting a computer to the Internet. The different methods of getting connectivity include use of telephone lines, use of mobile phones or use of satellite inks. In majority of cases the connectivity is made using a special device called modem. Once getting connected to the Internet, different application software can be launched to get the required service from the Internet.

13.3 Internet Applications in Linux

While installing Linux operating system, several applications for using different Internet services will be installed by default. Also several free and open source applications can be downloaded from different web sources and these can be used for getting the necessary services from the Internet. The following table provides a list of some of the common applications available in several Linux

distributions and the use of the applications. Most of these applications are installed by default when installing Linux operating system.

Application name	**Use**
aMSN	MSN messenger
Blogtk	Blog
Bluefish	Web development
CheckGMail	Gmail notifier
Chromium	Web Browser
Ekiga	VoIP
Elinks	Web Browser
Empathy	Chatting, VoIP
Epiphany	Web Browser
Evolution	Mail management
Filezilla	File Transfer
Firefox	Web browser
Firestarter	Desktop firewall
gFTP	File transfer
gTwitter	Read and post to Twitter
Gwibber	Microblogging
Kasablanca	File transfer

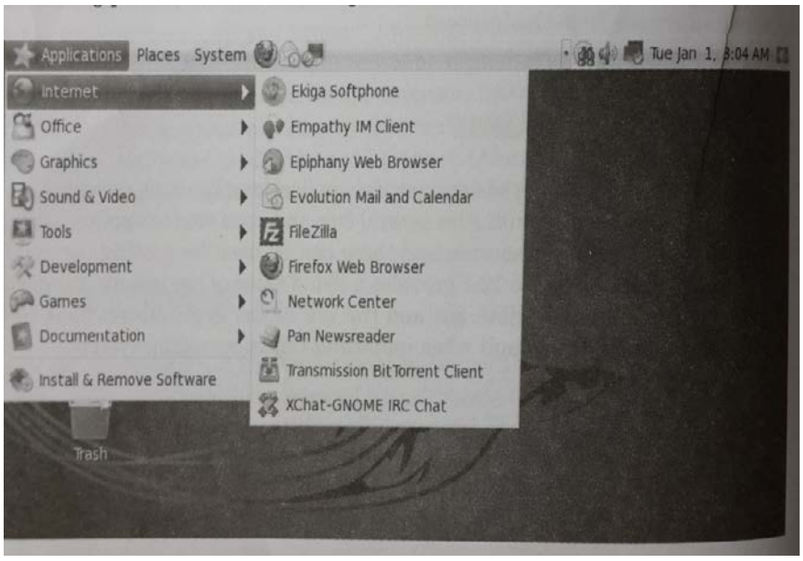
KFlickr	Filickr uploading
Konqueror	Web browser
Kopete	Instant messaging
KVIrc	Internet chat
Pan	News reader
Pidgin	Messaging
Pan	News reader
Thunderbird	Mail management, RSS
TS client	Remote desktop client
Tucan manager	File transfer
Vinagre	Remote desktop viewer
Xchat	Chatting
Yarssr	RSS reader

13.4 About Web and Websites

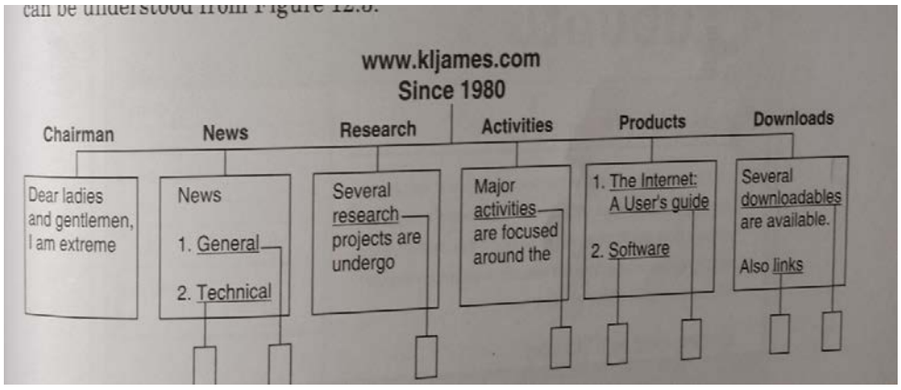
World Wide Web is the multimedia portion of the Internet. This is the most active and the fastest growing part of the Internet. This is also the information store of the Internet. Terms such as **Cyberspace** and **Information super highway** are also used to describe the web. Different types of files are stored in the web. The different files available in the web include audio files, video files, movies, data or simply text documents. Files in the web are known as **web pages**. Similar to the files in a local computer, files in the web are identified by file names. For creating web pages, languages such as HTML (Hyper Text Markup Language), XML (Extensible Markup Language) etc. are used. The web is made up of several web

pages. Every web page available in the Internet has a unique name. It is the Internet address of the web page. The group of web pages collectively forms a website. Websites are identified by their names known as website addresses or domain addresses or domain names. The web address is unique and is also known as URL (Uniform Resource Locator). The first page of any website is known as the home page of the site. To view web pages, it is necessary to visit the website. It is possible to locate any page in the Internet by specifying the URL. URL actually points to a specific file located in a computer connected to the Internet. When a user requests for a web page using a web browser, the requested file is copied to the user's computer for its display. The copied file is then viewed in the local computer.

Different types of websites can be found in the web. There are official websites loaded with official information of offices, government agencies, educational institutions and the like. Commercial websites are intended for commercial uses and many of them offer facilities for online trading. Details of products offered for sale will be displayed in these websites. Personal websites are mainly created for personal use and these contain the person's family details, achievements and so on. There are certain websites that are designed for use by special groups of people or special activities such as for sports, games, events, social networking etc. Depending on the type of the websites and the nature of activities, the content and the layout of the web pages included in the website varies. A web page may contain one or more links to other web pages or to other websites. Actually websites are made up of several linked web pages. The structure of a website can be understood from Figure below .



Links in web pages are given to text as well as to images or to any other object in web pages and such links can be easily identified. Usually links appear as underlined texts. When the mouse pointer passes above a link, the cursor arrow changes to a hand symbol and this is an indication of the link to that object. When a linked object is clicked, it opens another web page. The newly opened page can be in the same website or in some other websites, depending on the address given to the link. By opening links provided in web pages, a visitor to a website can move from one web page to another without any limit. This process of visiting websites one after another is known as **web browsing** or simply, browsing. Separate software is needed to visit websites and to view web files. Such software for visiting websites and viewing web pages is known as **web browser** or simply browser.



13.5 Assignment 13

1) Write a note on networking fundamentals

2) Write a note on Internet applications in Linux

3) Write a note about web and websites

V semester Linux Environment Question Bank
UNIT-I

Questions carrying 2 marks.

1. Name any two features of UNIX system.
2. Give the differences between .and . . .
3. What is a path name? Define full path name and relative path name.
4. Differentiate a process and a program.
5. Differentiate the user mode and the kernel mode.
6. What is the function of process control subsystem of the kernel architecture?
7. What is the importance of buffer cache in kernel architecture?
8. Give the difference between boot block and super block.
9. Give a brief note on system administration.
10. Define a buffer. What are the parts of a buffer?
11. How will be the data placed in buffer pool?
12. What is the difference between the synchronous write and asynchronous write?
13. Name any two advantages of the buffer cache.
14. Define an i-node. Name the fields of the i-node.
15. Define in core i-node. Name the fields of the in core i-node.
16. What is reference count?
17. What is a directory? Give the difference between a file and a directory.
18. What is the function of namei?
19. What is the difference between block device special file and character device special file?
20. Give the differences between bread and breada.

Questions carrying 5 marks and above.

1. Name any five services of operating system.
2. Give the structure of a file system with a neat diagram.
3. Give the structure of the buffer header with a neat diagram.
4. Write a note on the structure of the buffer pool.
5. Write bread algorithm
6. Explain breada algorithm.

7. Write bwrite algorithm.
8. What are the advantages and disadvantages of buffer cache?
9. Give the structure of the disk i-node.
10. What is the difference between disk i-node and incore i-node? Explain the structure of in-core inode.
11. Give the structure of a regular file with direct and indirect block.
12. How can we convert a path name to an inode?
13. List out the contents of the super block.
14. Write a note on allocation of disk blocks. Explain with example.
15. Write a note on other files in UNIX operating system.

UNIT-II

Questions carrying 2 marks.

1. Name any two system calls which returns file descriptor values and give the syntax.
2. Define U Area. What are the different I/O parameters saved in U Area.
3. What do you mean by file locking and record locking?
4. Define the pipe. Name the types.
5. What is the difference between the named pipes and unnamed pipes?
6. Name any two system calls which defines file system structure.
7. List out the entries of the mount table.
8. Give the syntax of umount and unlink system call.
9. Write a note on file system abstractions.
10. What is the function of the disk driver?
11. Define a socket.
12. What are the three mechanisms of System V IPC.
13. Define a stream.
14. Name the elements of the queue of the stream.
15. Define the term ‘line discipline’ in terminal driver.

Questions carrying 5 marks and above

1. Define and explain the function of open system call.
2. Define and explain the following system calls with syntax. i. Read ii. Write iii. Close iv. Create

3. What is the importance of lseek system call? Explain with syntax.
4. What is the difference between creat and mknod system call? Explain with syntax.
5. Write the function of pipe system call.
6. What is the difference between mount and link system call? Explain with syntax and example.
7. Write a note on disk driver.
8. Name the functions of the line discipline.
9. Write a note on streams.
10. Write process tracing algorithm.
11. Write a note on network communications.
12. Give the structure of system V IPC.

UNIT-III

Questions carrying 2 marks.

1. Differentiate the terms ‘free software’ and ‘freeware software’.
2. Define the types and versions of LINUX distributions.
3. Give examples for RPM and Deb distributions.
4. Expand the term KDE and GNOME.
5. What is Gedit?
6. Name the different applications of OpenOffice.org.
7. Define and explain the different wild card characters supported by LINX OS.
8. What is the importance of X window system?
9. Name the initial graphical interfaces used for LINUX.
10. Define the terms ‘application window’ and ‘document window’.
11. Define and expand XFCE desktop environment.
12. Define a file name and file name extension.
13. What do you mean by a path name? Give example.
14. Give the classification of files in LINUX OS.
15. Name the different wildcard characters used in LINUX.
16. What does the search string *.[ch] indicates?
17. Name the different file system types used in LINUX.
18. What is the difference between the ‘hidden files and folders’ and ‘other files and folders’?
19. Define Dolphin and Nautilus.

20. Name any two web browsers in LINUX.

Questions carrying 5 marks and above.

1. Define and explain the features of LINUX.
2. Write a note on LINUX distributions.
3. Give the difference between RPM based distributions and Deb based distributions with example. 4. Write a note on Xwindow system.
5. Write a note on components of LINUX window.
6. Write a note on GNOME desktop.
7. Explain in brief K desktop environment.
8. Write a note on XFCE desktop environment.
9. Define and explain different types of files in LINUX OS.
10. Define and explain the components of file manager window.
11. Write a note on files and folders properties.
12. Give the structure of LINUX file system. Explain with example.
13. Write a note on managing files and folders in LINUX.
14. How can we search for files in LINUX? Explain in brief.
15. How can we create files and folders in LINUX?

UNIT-IV

Questions carrying 2 marks.

1. Write the use of Disk usage analyzer.
2. What is GParted?
3. What are the advantages of using live versions?
4. What is Live USB?
5. Define the system log viewer.
6. What are the various types of Linux packages?
7. Write the functions of LINUX package manager.
8. What is the use of synaptic package manager?
9. What does .ar and .gz indicates?
10. What is the use of back up files?
11. What is IP address? Give the difference between IPv4 and IPv6.
12. Define P2P networking.
13. Define client/server architecture.

14. Define LAN,WAN and MAN.
15. What is the difference between internet and intranet?
16. Name the services provided by the internet.
17. Define the term ISP. Write the use of ISP.
18. What is the importance of Cyberspace?
19. Define and expand the term URL.
20. What are the importance of backed up files?

Questions carrying 5 marks and above.

1. Write a note on powers of an administrator in LINUX OS.
2. How can we write the contents to CD/DVD in LINUX?
3. Write a note on Ubuntu software center tool.
4. How can we install LINUX package?
5. Explain the services provided by the internet.
6. Write a note on the following.
    i. Managing drives and media in LINUX OS.
    ii. Creating and editing disk partitions LINUX OS.
    iii. Monitoring the System LINUX OS.
    iv. Managing users and groups LINUX OS.
7. Write a note on Synaptic Package Manager.
8. How can we configure computer network in LINUX? Explain with example.

9. Differentiate P2P networking and client/server networking with a diagram.
10. How can we differentiate a web and website? Explain with example.
11. Write the uses of the following internet applications in LINUX.
    i. aMSN
    ii. Blogtk
    iii. Bluefish
    iv. CheckGMail
    v. Chromium
    vi. Ekiga
    vii. Elinks
    viii. Filezilla
    ix. Firefox
    x. gFTP Credit Based Semest

BCACAC 312

Credit Based Fifth Semester B.C.A. Degree Examination, Oct/Nov 2014

(New Syllabus 2014-15 Batch)

LINUX ENVIRONMENT

Time: 3 hours **Max. Marks: 100**

Note: Answer any ten questions from Part A and answer any one full question from each Unit in Part B

1

a) What are the companies involved in developing Multics
2*10=20
b) What is the use of the following system calls:
i. Brk()
ii. Signal()
c) What is super block? List any 4 components of super block?
d) Write the purpose of the following algorithm
i. Bmap
ii. Alloc
e) List the conditions of buffer status field
f) What are the three stages of device configuration?
g) Explain lseek system call.
h) Expand the term KDE and GNOME.
i) Define kernel. List the activities of LINUX Kernel.
j) Name any 2 web browsers in LINUX.
k) Write the functions of LINUX package manager.
l) Define the term ISP. Write the use of ISP.

PART-B
UNIT-I

2

a) Explain block read and ahead algorithm(breada)
b) Discuss UNIX file system with diagram.

c) List the features of Unix file system.
d) Expalin with figure the table of contents filed of an inode.
(8+4+4+4)

3
a) Expalin the various services of UNIX Operating System.
b) Explain the Kernel architecture of the UNIX system with neat digarm.
c) Explain the structure of Buffer Pool. (6+8+6)

UNIT-II

4
a) With the syntax explain the following system call
i) Write
ii) Mknod
iii) Mount
iv) Creat
v) Pipe
b) List the functions of line discipline
c) Explain the properties of system V IPC package.
(10+4+6)

5
a) Write a note on disk driver.
b) Explain sockets model with diagram.
c) Explain the read and link system call
d) Explain the system configuration with diagram.
(4+6+4+6)

UNIT-III

6
a) List and explain the features of GNOME desktop.
b) Give the difference between RPM based distributions and Deb based distributions with example.
c) Explain in brief the K desktop environment. (8+6+6)

7

a) Write a note on components of LINUX window.
b) How can we create files and folders in LINIX?
c) Explain the features of LINUX. (8+6+6)

UNIT-IV

8

a) How can we write the contents to CD/DVD in LINUX?
b) Explain the services provided by the Internet.
c) Write a note on Ubuntu software center tool.
(8+6+6)

9

a) How can we differentiate a web and websites? Explain with example.
b) Differentiate P2P networking and client/Server networking with a diagram.
c) Write the uses of the following internet applications in LINUX.
   i. Chromium
   ii. Ekiga
   iii. Elinks
   iv. Filezilla (8+6+6)