Case Studies

Questions raised by Balaram:

1)Security in windows 2000
2)Explain with example the role of autoexc.bat and config.sis files in ms-dos.
3)compaction and coalescing
4)mode switching
5)Explain files and disk management in windows 2000.
6)What are different methods to keep track of which disk block go with which file?
7)Explain different layers of UNIX Operation System with figure.
8)multimedia files.
9)File System in Linux Operating System.

Q1.Security in windows 2000:
Every windows 2000 user (and group) is identified by a SID(Security ID). SIDs are binary numbers with a short header followed by a long random component. Each SID is intended to be unique worldwide. When a users starts up a process, the process and its thread run under the user's SID. Most of the security system is designed to make sure that each object cab be accessed only by threads with authorized SIDs.

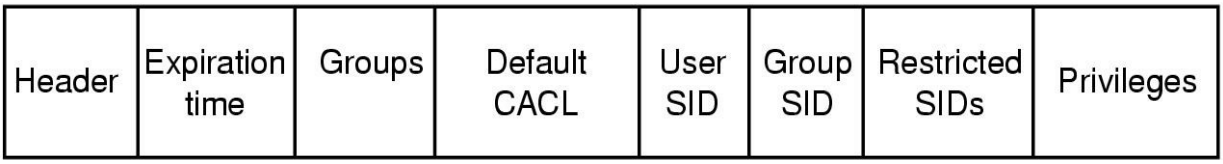

Fig: Structure of an access token

Each process has an access token that specifies its SID and other properties. It is normally assigned at login time by winlogon and is shown in fig above.
Another basic concept is the security descriptor. Every object has a security descriptor associated with it that tells who can perform which operation on it.

A security descriptor consists of a header followed by a DACL(Discretionary ACL) with one or more ACEs(Access Control Elements). The two main kinds of elements are Allow and Deny . An allow elements specifies a SID and a bitmap that specifies which operations processes with that SID may perform on the object. A deny element works the same way, except a match means the caller may not perform the operations. For example, Ida has a file whose security descriptor specifies that everyone has read access, Elvis has no access. Calthy has read/Write access and Ida herself has full access. This simple example is illustrated as shown in fig below.

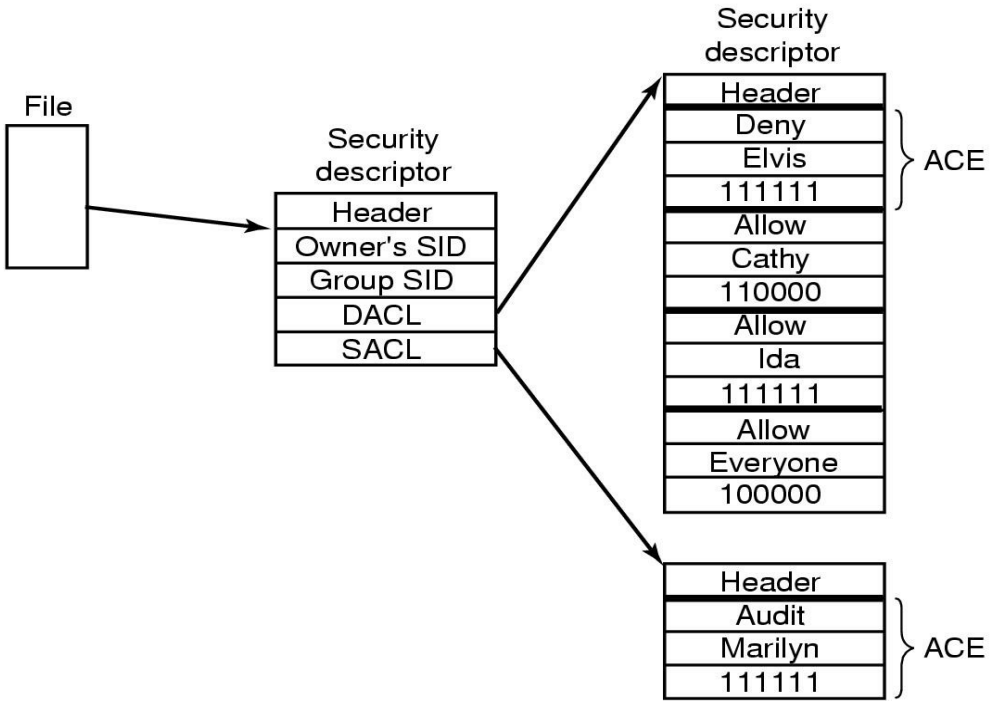

Fig: an example security descriptor for a file.

Q. 2. Explain with example the role of autoexc.bat and config.sis files in ms-dos:

The autoexec.bat and the config.sys were files created for MS-DOS and Windows 3.x as an easy solution of loading the files required for various devices as well as the operating system to properly run. These files are required for later revisions of MS-DOS and Windows 3.x to load. Because Microsoft is trying to steer away from MS-DOS, these files are not required for Windows 95, Windows 98, Windows NT, Windows ME, Windows 2000, Windows XP, or later operating systems. However, in some cases it may still be necessary for users to edit or configure these files.

The auotexec.bat and the config.sys are most commonly edited by the MS-DOS command file Edit. To edit these files, type **edit c:\autoexec.bat** to edit the autoexec.bat file, or **edit c:\config.sys** to edit the config.sys file. If the mouse drivers are not loaded properly you will not have the capability of navigating the mouse.

If you have Windows 95, Windows 98, or later versions of Windows it is recommended that you use the sysedit command; to run this program, click Start, Run, and type sysedit.

Autoexec.bat layout

Below is an example of what an autoexec may look like.

```
@echo off
SET SOUND=C:\PROGRA~1\CREATIVE\CTSND
SET BLASTER=A220 I5 D1 H5 P330 E620 T6
SET PATH=C:\Windows;C:\
LH C:\Windows\COMMAND\MSCDEX.EXE /D:123
```

@echo off Tells DOS to just read the lines but don't echo them back to the screen.
SET SOUND=C:\PROGRA~1\CREATIVE\CTSND This example is for the particular sound card that we have in one of the machines that we have. The set sound is telling the computer to send all sound events that the computer may run to that directory.
SET PATH=C:\Windows;C: Sets the computer to look in the C:\windows directory or the root if a command used at the prompt is not found.

Config.sys layout
Below is an example of what the config.sys may look like:
DEVICE=C:\Windows\HIMEM.SYS
DOS=HIGH,UMB
DEVICE=C:\Windows\EMM386.EXE NOEMS
FILES=30
STACKS=0,0
BUFFERS=20
DEVICEHIGH=C:\Windows\COMMAND\ANSI.SYS
DEVICEHIGH=C:\MTMCDAI.SYS /D:123

DEVICE=C:\Windows\HIMEM.SYS The Himem.sys line is a very important line; this line will allow you to load drivers into high memory. If this line is not present, Windows 3.x will not load.
BUFFERS=20 Buffers line is to load buffers into memory allowing Windows to load memory.

Q. 3)compaction and coalescing:
These are the terms used for preventing fragmentation arises in memory management (variable partitioning) .
In computer science, **coalescing** is the act of merging two adjacent free blocks of memory. When an application frees memory, gaps can fall in the memory segment that the application uses. Among other techniques, coalescing is used to reduce external fragmentation, but is not totally effective. Coalescing can be done as soon as blocks are freed, or it can be deferred until some time later (known as deferred coalescing), or it might not be done at all.

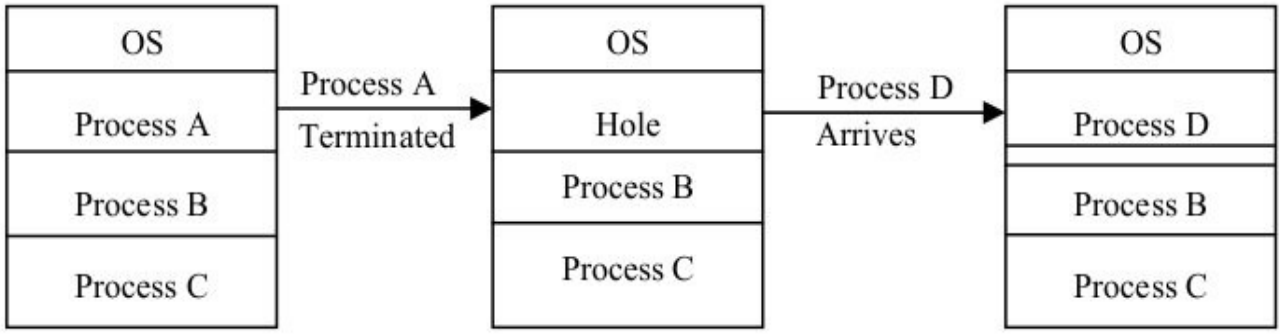

Fig: Variable size partition

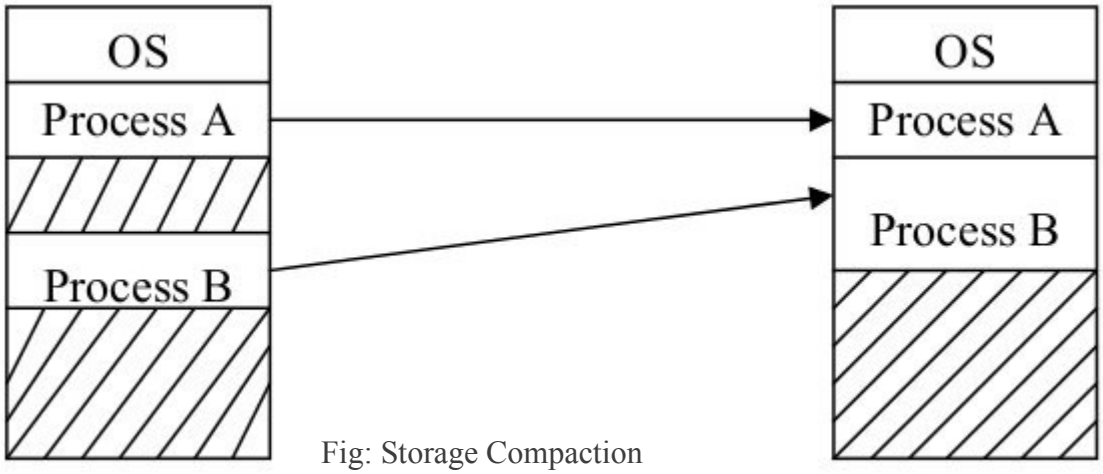

Fig: Storage Compaction

External fragmentation problem can be resolved by coalescing holes and storage compaction. Coalescing holes is process of merging existing hole adjacent to a process that will terminate and free its allocated space. Thus, new adjacent holes and existing holes can be viewed as a single large hole and can be efficiently utilized. There is another possibility that holes are distributed throughout the memory. For utilizing such scattered holes, shuffle all occupied areas of memory to one end and leave all free memory space as a single large block, which can further be utilized. This mechanism is known as Storage Compaction, as shown in Figure above.

Q. 4. mode switching:
Go through the Context Switching. User mode and Kernel. mode.

Q. 5)Explain files and disk management in windows 2000.

Create a Folder
With hard drives getting larger, and more applications being installed, it is very important to organize your hard disk using folders. It is important for you to understand how your hard disk is organized. A folder on your hard disk is like a folder in your filing cabinet. You use it to save documents for an application, project or tasks of a similar type (e.g. budget information). To create a new folder, make the window active where you want the folder to appear. Choose File --> New --> Folder.

The default name of the folder is “New Folder”. Notice that the words “New Folder” are highlighted in blue. This means that the words are selected. Simply type in a new name for the folder (it is common for people to want to click their mouse button on the highlighted text. Doing this will mean having to delete the characters “New Folder”. More on this later). Hit <Enter> or click on a blank area in the current window to complete the folder creation and naming process.
Move a Document to a Folder Using Drag-and-Drop Method
Copy a Document to a Folder Using Drag-and-Drop Method
Delete a Document
Recovery of Deleted Document from Recycle Bin

Disk Management:
Here you have to explain about creating new partition and formating and deleting partitions.

Q. 6)What are different methods to keep track of which disk block go with which file?
Here you have t explain about the file system allocation methods.

Probably the most important issue in implementing file storage is keeping track of which disk blocks go with which file. Various methods are used in different operating systems.
File Allocation Method:

- Contiguous Allocation
- Linked List Allocation
- Linked List Allocation Using a Table in Memory
- I-Nodes

Refer them to my notes.
Q. 7)Explain different layers of UNIX Operation System with figure.

A UNIX system can be regarded as a kind of pyramid, as illustrated in the fig. below. At the bottom is the hardware, consisting of the CPU, memory,disks, terminals and other devices. Running on the bare hardware is the UNIX operating system. Its function is to control the hardware and provide a system call interface to all the programs. These system calls allows user programs to create and manage processes, files and other resources.

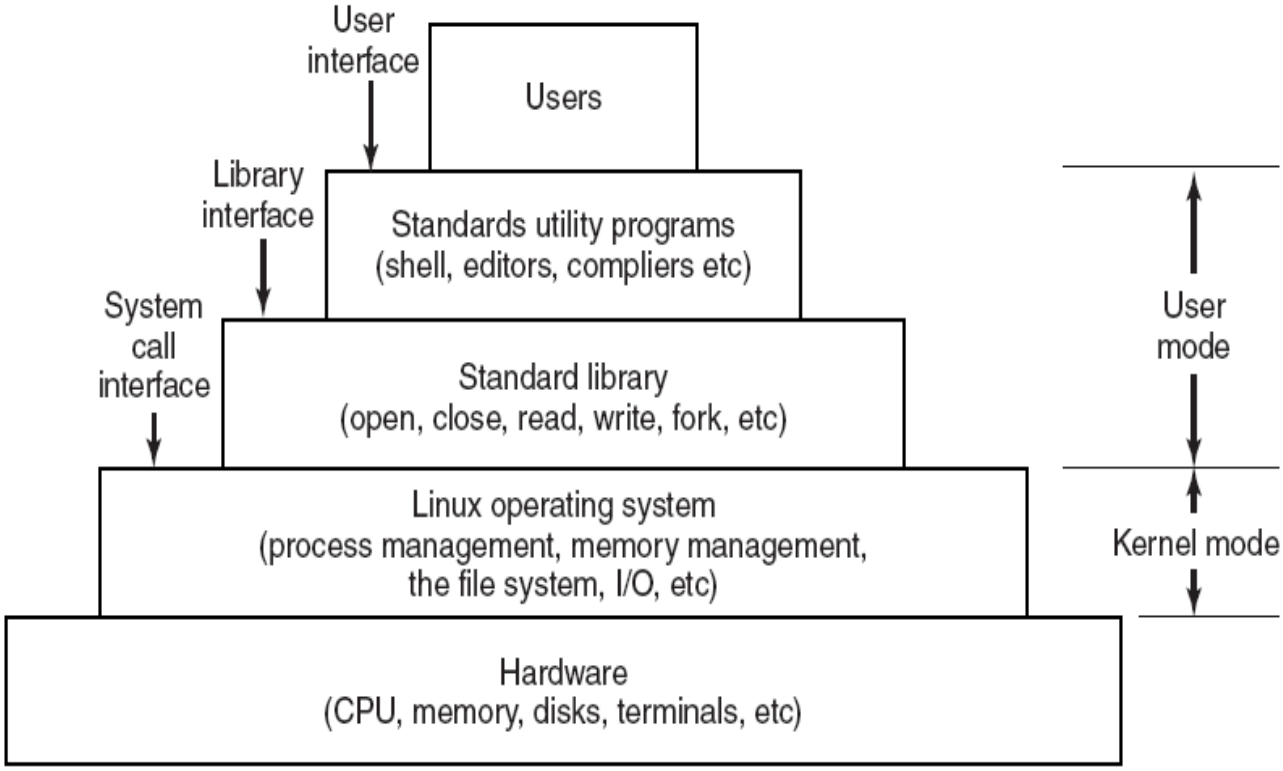

Fig: The layers in a Unix System.

Q. 8)multimedia files.

In most system, and ordinary file consists of a linear sequence of bytes without any structure that the operating system knows about or cares about. With multimedia, the situation is more complicated. Video and audio are completely different. They are captured by distinct devices, have a different internal structure and they are played back by different deices.

Key characteristics of multimedia:

Multimedia uses extremely high data rates.

Due to the nature of visual and acoustic information

The eye and the ear can process prodigious amounts of information per sec, and have to be fed at that rate to produce acceptable viewing experience

Therefore a video server requires huge amount of storage and must be supported by h/w and OS

Multimedia requires real-time playback.

Frames must be delivered at precise interval to prevent the movie from looking choppy

Based on standard such as NTSC, PAL and SECAM

Ear is more sensitive than eye, so variance of even a few ms in delivery times will be noticeable - jitter

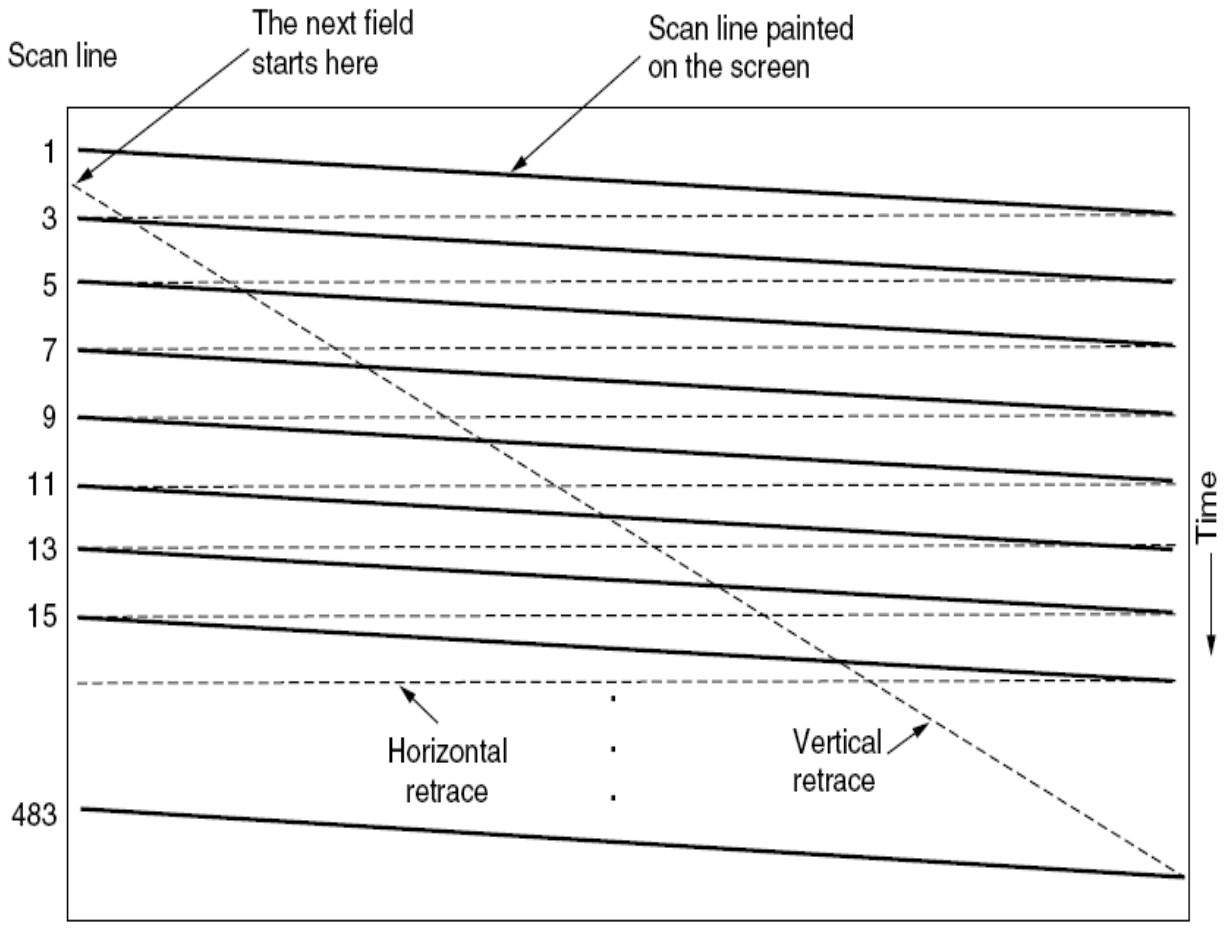

Fig: Video Encoding

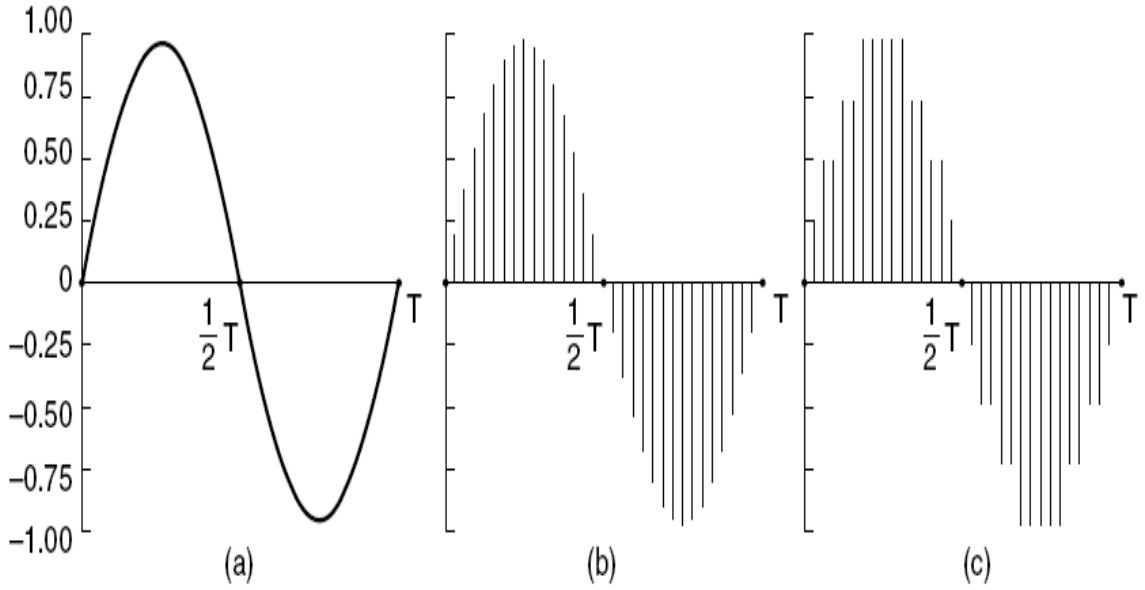

Fig: Audio Encoding:

(a) A sine wave. (b) Sampling the sine wave (c) Quantizing the samples to 4 bits.

Q. 9). File System in Linux Operating System.

Windows and Linux use different concepts for their file hierarchy. Windows uses a volumebased file hierarchy, Linux uses a unified scheme. Windows uses letters of the alphabet to represent different devices and different hard disk partitions. Under Windows, you need to know what volume (C:, D:,...) a file resides on to select it, the file's physical location is part of it's name. In Linux all directories are attached to the root directory, which is identified by a forwardslash, "/". For example, below are some secondlevel directories:

bin/ system binaries, user programs with normal user permissions
/sbin/ executables that need root permission
/data/ a user defined directory
/dev/ system device tree
/etc/ system configuration
/home/ users' subdirectories
/home/{username} home directory of users
/tmp/ system temporary files
/usr/ applications software
/usr/bin executables for programs with user permission
/var/ system variables
/lib/ libraries needed for installed programs to run

/root/ Root's home directory
/media and /mnt – Foreign filesystem mount point.
Every device and hard disk partition is represented in the Linux file system as a subdirectory of the root directory. The root directory lives in the root partition, but other directories (and the devices they represent) can reside anywhere. Removable devices and hard disk partitions other than the root are attached (i.e., "mounted") to subdirectories in the directory tree. This is done either at system initialization or in response to a mount command

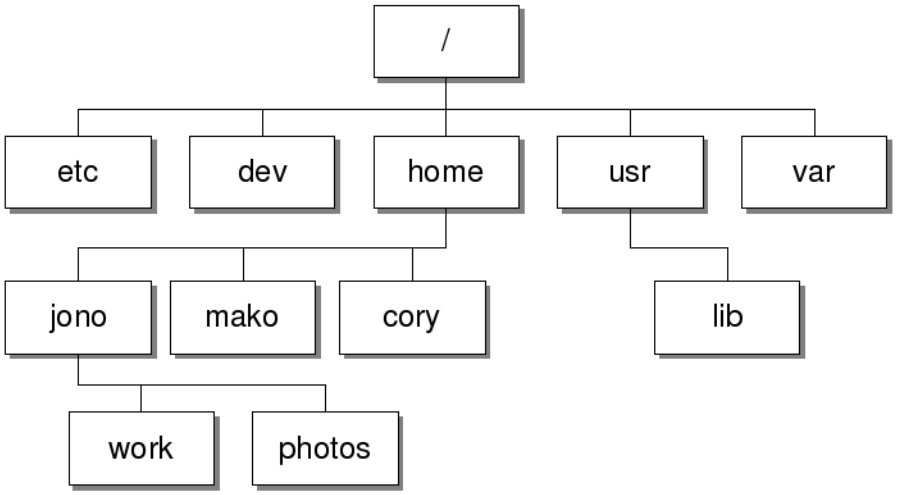

Fig: Linux file system hierarchy

What is a File?

File are collection of data items stored on disk. Or it's device which can store the information/data/music/picture/movie/sound/book; In fact what ever you store in computer it must be in the form of file. Files are always associated with devices like hard disk ,floppy disk, usb etc. File is the last object in your file system tree.

In Linux files have different types like

- Executable files (stored in /bin, /usr/bin)
- Special files (stored in /dev)
- Configuration files (stored in /etc)
- Directory
- User files or ordinary files etc

And the process which manage the Files i.e. creation, deletion, copying of files and giving read, write access of files to user is called File management.

Normally you can perform following operation on files

- Copy file

- Delete file
- Rename file
- Move file
- Changing file date & time stamp
- Creating symbolic link
- Changing file permission or ownership
- Searching files
- Compressing / Decompressing files
- Comparison between files
- Printing files on printer
- Sorting files etc

What directory is?

Directory is group of files. Directory is divided into two types as

(1) Root directory : Strictly speaking, there is only one root directory in your system, which is shown by / (forward slash), which is root of your entire file system. And can not be renamed or deleted.
(2) Sub directory : Directory under root (/) directory is subdirectory which can be created, rename by the user. For e.g. bin is subdirectory which is under / (root) directory.
Directories are used to organize your data files, programs more efficiently.

Linux File Permission:

Every file on your Linux system, including directories, is owned by a specific user and group. Therefore, file permissions are defined separately for users, groups, and others.

User: The username of the person who owns the file. By default, the user who creates the file will become its owner.

Group: The usergroup that owns the file. All users who belong into the group that owns the file will have the same access permissions to the file. This is useful if, for example, you have a project that requires a bunch of different users to be able to access certain files, while others can't. In that case, you'll add all the users into the same group, make sure the required files are owned by that group, and set the file's group permissions accordingly.

Other: A user who isn't the owner of the file and doesn't belong in the same group the file does. In other words, if you set a permission for the "other" category, it will affect everyone else by default. For this reason, people often talk about setting the "world" permission bit when they mean setting the permissions for "other."

The characters are pretty easy to remember.

r = read permission
w = write permission

x = execute permission
- = no permission

In the numeric mode, the file permissions aren't represented by characters. Instead, they are represented by a three-digit octal number.

4 = read (r)
2 = write (w)
1 = execute (x)
0 = no permission (-)

Example:
daya@daya-Aspire-4937:~$ ls -l test.py
-rw-r--r-- 1 daya daya 186 2011-11-11 07:55 test.py
Read and write for user (owner) daya:
Read only for group daya.
Read only for other

Filters and pipelining:

In Unix-like computer operating systems a **pipeline** is a set of processes chained by their standard streams, so that the output of each process (stdout) feeds directly as input (stdin) to the next one. Each connection is implemented by an anonymous pipe. Filter programs are often used in this configuration.
All widely used Unix and Windows shells have a special syntax construct for the creation of pipelines. In typical usage one writes the filter commands in sequence, separated by the ASCII vertical bar character **"|"** (which, for this reason, is often called "pipe character"). The shell starts the processes and arranges for the necessary connections between their standard streams (including some amount of buffer storage).

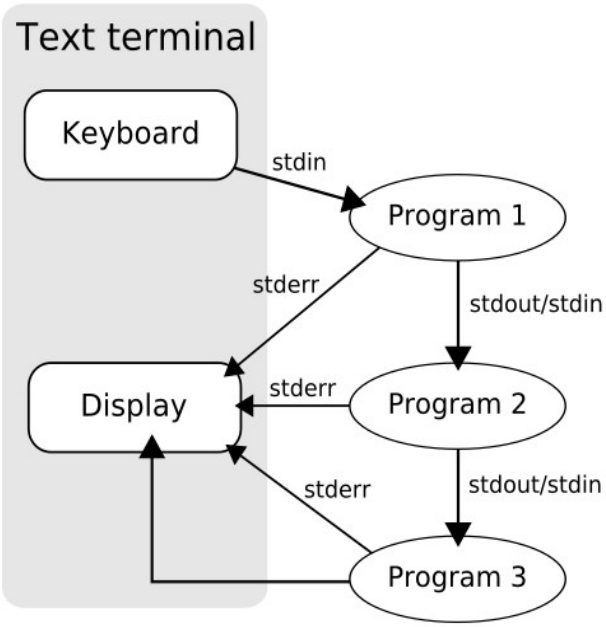

Fig:A pipeline of three programs run on a text terminal

In Unix and Unix-like operating systems, a filter is a program that gets most of its data from its standard input (the main input stream) and writes its main results to its standard output (the main output stream). Unix filters are often used as elements of pipelines. The pipe operator ("|") on a command line signifies that the main output of the command to the left is passed as main input to the command on the right.
The classic filter would be grep, which at it simplest prints to its output any lines containing a character

string. Here's an example:
cat /etc/passwd |grep foo
This finds all registered users that have "foo" as part of their username.
List of Unix filter programs

- awk
- cat
- comm
- cut
- expand
- compress
- fold
- grep
- head
- nl
- perl
- pr
- sed
- sh
- sort
- split
- strings
- tail
- tac
- tee
- tr
- uniq
- wc

Unix Shell:
A Unix shell is a command-line interpreter or shell that provides a traditional user interface for the Unix operating system and for Unix-like systems. Users direct the operation of the computer by entering commands as text for a command line interpreter to execute or by creating text scripts of one or more such commands.
The most influential Unix shells have been the Bourne shell and the C shell. The Bourne shell, sh, was written by Stephen Bourne at AT&T as the original Unix command line interpreter; it introduced the basic features common to all the Unix shells,
The most generic sense of the term shell means any program that users employ to type commands. A shell hides the details of the underlying operating system with the shell interface and manages the technical details of the operating system kernel interface, which is the lowest-level, or 'inner-most' component of most operating systems. In Unix-like operating systems users typically have many choices of command-line interpreters for interactive sessions. When a user logs in to the system, a shell program is automatically executed. The login shell may be customized for each user, typically in the `passwd` file, and can be customized via `passwd -e`, or on some systems via the `chsh` program. In addition, a user is typically allowed to execute another shell program interactively.