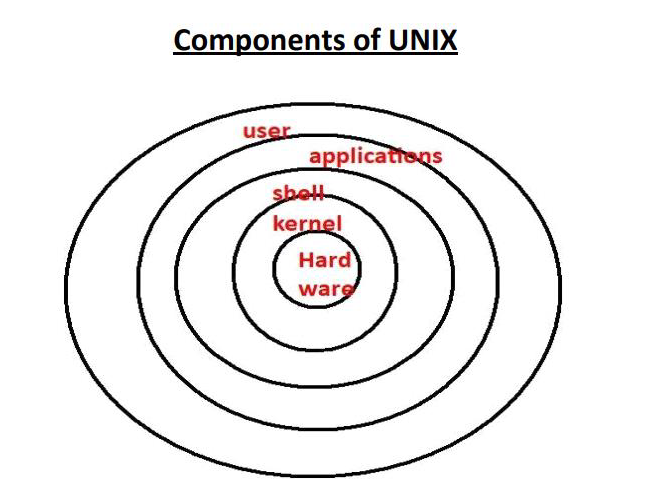
Cd :

1. Absolute path:exact path
2. Relative path:related path from the present directory

Cd /:command to go to root directory

Components of Unix



**Shell:**

✽It is the outer layer of UNIX operating System.

✽It reads our command, verify syntax and check whether the corresponding command related application is available or not.

✽If everything is proper, then shell interprets our command into kernal understandable form and handover to the kernal.

✽Shell acts as interface between user and kernal.

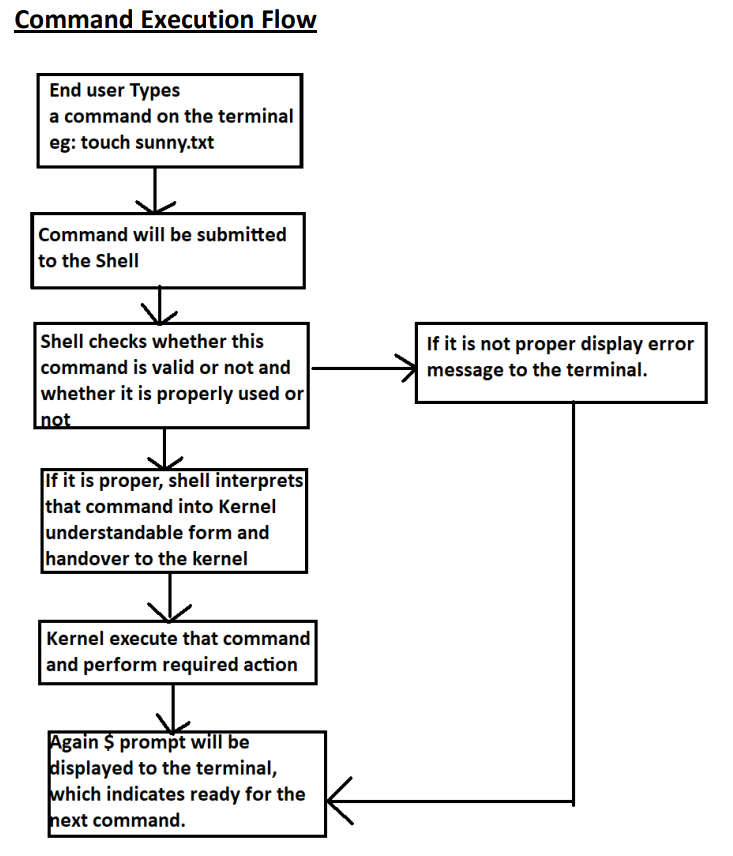
**Kernal:**

✽It is the core component of UNIX operating system.

✽It is responsible to execute our commands.

✽It is responsible to interact with hardware components.

✽Memory allocation and processor allocation will takes care by kernal



**Linux File System Structure :-**

**1. /bin**:-bin -stands for binary.s.

Binary is a file which contains the compiled source code.

We can also call it as executable, because it can be executed on the computer.

/bin is a sub-directory of the root directory in Unix/Linux OS.

this directory contains basic commands which is enough for the minimal system function

ex :-ls, cat, cp

2.**/sbin:** system binaries or super user binaries.

 This folder contains commands which are required for changing system properties.

Ex:-adduser,reboot,shutdown0

3**. /boot:**

The contents are mostly Linux kernel files and bootloaderfiles(files needed to start up the operating system)

4.**/dev :**

This contains device files

This file represents your speaker device,keyboard

5. **/etc:**-

it contains all system related configuration files in here or in its sub-directories

A "configuration file" is defined as a local file used to control the operation of a program;

it cannot be an executable binary.

Ex:-adduser.conf, theme config

**6. /cdrom:**-

directory is a standard practice to mount cd, but not necessary. We use media and mnt to mount anything these days

In Linux and other Unix-like operating systems, **/cdrom** is commonly used as the default directory where the contents of a CD or DVD inserted into the computer's optical drive are mounted.

When you insert a CD or DVD, the system may automatically mount it to the /cdrom directory (or it could be mounted manually).

You can view the contents of the disc by navigating to this directory in the terminal.

**7 ./home :**-

The home directory can be said as a personal working space for all the users except root.

There is a separate directory for every user.

 For example, two users ‘satyam' and ‘jack' will have directories like "/home/satyam“ and "/home/jack“

**8 ./lib :-**

The /lib directory contains **shared libraries** and **kernel modules** that are necessary to boot and run the system.

These libraries are used by essential system binaries located in directories like /bin and /sbin.

Shared libraries are files that contain code or data that multiple programs can use simultaneously.

Instead of having the same code duplicated in every executable, the system allows programs to load a shared library into memory when they need it. This improves efficiency and saves disk space.

The shared libraries in /lib are generally required by programs in /bin (basic user commands) and /sbin (system administration commands). Without these shared libraries, programs might fail to run

**Difference between lib, lib32, lib64, libx32**

lib :-architecture independent files.

lib32 :-for 32 bit architecture libraries

lib64 :-for 64 bit architecture libraries

libx32 :-for 64 bit architecture libraries but the pointer size is 32 bit,

Normally software using the x86-64 instruction set uses 64-bit pointer size.

**9 /media :**-

When you connect a removable media such as USB disk, SD card or DVD, a directory is automatically created under the /media directory for them. You can access the content of the removable media from this directory.

**10 /mnt –**

Mount directoryThis is similar to the /media directory but instead of automatically mounting the removable media, mnt is used by system administrators to manually mount a filesystem

**mounting** refers to the process of making a filesystem accessible to the operating system's directory structure. It is essentially the act of attaching an external storage device (like a hard drive, USB stick, or CD-ROM) or a remote filesystem to a specific location (a directory) within the main filesystem.

**11 /opt** –

Optional softwareTraditionally, the /opt directory is used for installing/storing the files of third-party applications that are not available from the distribution’s repository.

In the old days, "/opt" was used by UNIX vendors like AT&T, Sun, DEC and 3rd-party vendors to hold "Option" packages; i.e. packages that you might have paid extra money for.

**12 /proc** :-

It contains useful information about the processes that are currently running.It could be used for obtaining information about a system, we can also edit the config files related to kernel here.

**13 /root :-**

it works as the home directory of the root user. So instead of /home/root, the home of root is located at /root. root directory (/) is different from root user directory.

**14 /tmp :-**

this directory holds temporary files. Many applications use this directory to store temporary files. Even you can use directory to store temporary files.

the contains of the /tmp directories are deleted when your system restarts. Some Linux system also delete files old files automatically so don’ store anything important here.

**15 /var :**-

stores system-generated variable data files. This includes spool directories and files, administrative and logging data, cache, transient and temporary files.

Ex :-/var/spool contains data which is awaiting some kind of later processing/run :-runtime variable data.

 The purposes of this directory were once served by /var/run, system may use both

**16 /srv :**-

This directory gives users the location of data files for a particular service.

For example, if you run a HTTP or FTP server, it’s a good practice to store the website data in the /srv directory.

**17 /usr :-**

User System Resources. in ‘/usr’ go all the executable files, libraries, source of most of the system programs.

For this reason, most of the files contained therein is readonly (for the normal user).

**18 /sys :-**

allows you to get information about the system and its components (mostly attached and installed hardware) in a structured way.

Ex:-device, kernel, firmware

**19 /snap :-**

The /snap directory is, by default, where the files and folders from installed snap packages appear on your system.

**Types of Files in Linux:-**

In Linux everything is treated as File.

All files are divided into 3 types

**1) Normal or Ordinary files:**

These files contain data. It can be either text files (like abc.txt) OR binary files (like

images, videos etc).

**2) Directory Files:**

These files represent directories.

In windows, we can use folder terminology where as in linux we can use directory

terminology.

Directory can contains files and sub directories.

**3) Device Files:**

In Linux, every device is represented as a file. By using this file we can communicate

with that device.

**Note:** short-cut commands to open and close terminal

**ctrl+alt+t** ➔To open terminal

**ctrl+d** ➔To close terminal

**How to check File Type:**

In Ubuntu, blue color files represents directories and all remaining are considered as

normal files.

This color conventions are varied from flavour to flavour. Hence it is not

standard way to check file type.

The first character represents the type of file.

d = Directory File

- = Normal File

l = Link File

c = Character Special File

b = Block Special File

s = Socket File

Note: c, b, s are representing system files and mostly used by super user (also known as

root user or admin user)

**ls**

We can use ls command to listout all files and directories present in the given directory.

We can get manual documentation for any command by using man.

man ls

It provides complete information about ls command.

Various options of ls Command:

**1) ls**

It will display all files and directories according to alphabetical order of names.

**2) ls -r**

It will display all files and directories in reverse of aplhabetical order.

**3) ls | more**

To display content line by line

(To come out we have to use q)

**4)ls -l**

To display long listing of files

**5) ls -t**

To display all files based on last modified date and time. Most recent is at top and old

are at bottom.

**6) ls -rt**

To display all files based on reverse of last modified date and time. Old files are at top

and recent files are at bottom.

**7) ls -a**

a means all

To display all files including hidden files. Here . and .. also will be displayed.

**8) ls -A**

A means almost all

To display all files including hidden files except . and ..

**9) ) ls -h**

display in human readable format

**10) ls -R**

R means Recursive.

It will list all files and directories including sub directory contents also. By default ls

will display only direct contents but not sub directory contents.

Eg: All the following commands are equal

$ ls -l -t -r

$ ls -l -t -r

$ ls -t -r -l

$ ls -l -r -t

$ ls -ltr

$ ls -trl

Which Command will make a Long listing of all the Files in our System including Hidden

Files,

sorted by Modification Date (Oldest First)?

ls - lat

**6. mkdir:-**

We can create directories by using mkdir command.

1) mkdir dir1

To create a directory

2) mkdir dir1 dir2 dir3

To create multiple directories

3) mkdir dir1/dir2/dir3

To create dir3. But make sure dir1 and in that dir2 should be available already.

4) mkdir -p dir1/dir2/dir3

-p means path of directories.

All directories in the specified path will be created.

First dir1 will be created and in that dir2 will be created and within that dir3 will be created.

**7.  rmdir :-**

We can remove directories by using rmdir command.

1) $ rmdir dir1

To remove empty directory dir1

2) 2. $ rmdir dir1 dir2 dir3

To remove multiple empty directories

Note: rmdir command will work only for empty directories. If the directory is not empty then we

will get error. We cannot use rmdir for files. Hence the most useless (waste) command in linux is

rmdir.

8. rm :-

to delete files rm file1 file2

To delete non empty folders

$ rm -r mydir

$ rm -R folder

Note: In Linux operating system, there is no way to perform undo operation. Once we delete a file

or directory, it is impossible to retrieve that. Hence while using rm command we have to take

special care.

The following command is the most dangerous command in linux, because it removes

total file system.

Various options with rm Command:

1) interactive Option(-i)

While removing files and directories, if we want confirmation then we have to use -i

2)verbose Option(-v):

If we want to know the sequence of removals on the screen we should go for -v option.

**9.  Copy  (cp)**

1) To Copy from File1 to File2 (File to File)

$ cp source\_file destination\_file

$ cp file1 file2

Total content fo file1 will be copied to file2.

If file2 is not already available, then this command will create that file.

If file2 is already available and contains some data, then this data will be over write with file1

content.

2) To Copy File to Directory:

$ cp file1 file2 output

file1 and file2 will be copied to output directory.

Here we can specify any number of files, but last argument should be directory. output directory

should be available already.

3) To Copy all Files of One Directory to another Directory:

$ cp dir1/\* dir2

All files of dir1 will be copied to dir2 But dir2 should be available already.

4) To Copy Total Directory to another Directory:

$ cp dir1 dir2

cp: -r not specified; omitting directory 'dir1'

Whenever we are copying one directory to another directory,

compulsory we should use -r option. 🕐 $ cp -r dir1 dir2    total dir1 will be copied to

dir2

5) To Copy Multiple Directories into a Directories:

$ cp -r dir1 dir2 dir3 dir4 dir5

dir1,dir2,dir3 and dir4 will be copied to dir5

**10. mv  :-**

Moving and Renaming Directories:

Both moving and renaming activities can be performed by using single command: mv

**1) Renaming of files:**

$ mv oldname newname Eg: $ file1.txt file2.txt

file1.txt will be renamed to file2.txt

2) Renaming of Directories:

$ mv dir1 dir2

dir1 will be renamed to dir2

3) Moving files to directory:

$ mv a.txt b.txt c.txt output

a.txt,b.txt and c.txt will be moved to output directory.

4) Moving of all files from one directory to another directory:

$ mv dir1/\* dir2

All files of dir1 will be moved to dir2. After executing this command dir1 will become

empty.

5) Moving total directory to another directory:

$ mv dir1 dir2

Note: If dir2 is already available then dir1 will be moved to dir2.

If dir1 is not already available then dir1 will be renamed to dir2.