

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

1. **/sbin :**

* system binaries or super user binaries.
* This folder contains commands which are required for changing system properties.
* Ex:- adduser, reboot, shutdown

1. **/boot:**

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

1. /**dev :**

* This contains device files.
* This file represents your speaker device,keyboard

1. **/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

1. **/cdrom :-**

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

1. **/home :-**

* The home directory can be said as a personal working space for all the users except root.
* Thereis a separate directory for every user.
* For example, two users ‘jitendra' and ‘jack' will have directories like

"/home/jitendra“ and "/home/jack“

1. **/lib :-**

* directory contains those shared library files needed to boot the system and run the commands in the filesystem.
* ie. by binaries in /bin and /sbin.
* Only the shared libraries required to run binaries in /bin and /sbin will be here

Difference between lib, lib32, lib64:

lib x32 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.

1. **/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.

1. **/mnt – Mount directory**

* This 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.

1. **/opt – Optional software**

* Traditionally, 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.
* **/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.

1. **/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.

1. **/tmp :-**

* This directory holds temporary files.
* Many applications use this directory to store temporary files.
* You can even use a directory to store temporary files.
* the contents of the /tmp directories are deleted when your system restarts.
* Some Linux system also delete old files automatically,so don’t tore anything important here.

1. **/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.

1. **/run :-**

* runtime variable data.
* The purposes of this directory were once served by /var/run, system may use both.

1. **/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.

1. **/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 read only (for the normal user). why /usr/bin , /usr/sbin ?

1. **/sys :-**

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

Ex:- device, kernel, firmware

1. **/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 in 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).

1. **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.

1. **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

**Commands**

# **Cd :-**

* Every directory implicitly contains 2 directories . and ..

. Current Directory

.. Parent Directory

**$ cd .**

Changes to Current Directory (Useless)

**$ cd ..**

Changes to Parent Directory

**$cd**

If we are not passing any argument, then changes to user home directory.

1. **$ cd ~**

* ~ Means User Home Directory.

It will Changes to User Home Directory.

1. **$ cd –**

* - Means Previous Working Directory.

It will Changes to Previous Working Directory.

1. **$ cd /**

* It will change to the root directory.

1. cd “My Song” or cd My\ song

For Directories with

spaces.

1. cd dir1/dir2/dir3 or cd ../dir1/dir2 or cd /home/user/dir2 For moving multiple directories

# **Pwd :-**

* pwd stands for Print Working Directory.
* It prints the path of the working directory, starting from the root.

Options:

-L : Works the same as pwd and prints Symbolic Link.

-P: works the same as /bin/pwd and prints physical link

# **ls :-**

* We can use ls command to list out 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.
* Syntax: ls [option] [file/directory]

Various options of ls Command:

Options Description

**-l** known as a long format that displays detailed information about files and

directories.

**-a** Represent all files Include hidden files and directories in the listing.

**-t** Sort files and directories by their last modification time, displaying the

most recently modified ones first.

**-r** known as reverse order which is used to reverse the default order of listing.

**-S** Sort files and directories by their sizes, listing the largest ones first.

**-R** List files and directories recursively, including subdirectories.

-**i** known as inode which displays the index number (inode) of each file and

directory.

**-g** known as group which displays the group ownership of files and

directories instead of the owner.

**-h** Print file sizes in human-readable format (e.g., 1K, 234M, 2G).

**-d** List directories themselves, rather than their contents.

1. **ls**

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

1. ls **-r**

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

1. **ls | more or ls -1**

To display content line by line

(To come out we have to use q)

1. **ls -l**

* To display long listing of files.
* 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)

Field 1 – **File Permissions:**

* Next characters specifies the files permission.
* Every 3 characters specify read, write, execute permissions for user(root), group and others respectively in order.
* Taking the above example, -rw-rw-r– indicates read-write permission for user(root), read permission for group, and no permission for others respectively.
* If all three permissions are given to user(root), group and others, the format looks like -rwxrwxrwx

Field 2 – **Number of links:**

* Second field specifies the number of links for that file.
* In this example, 1 indicates only one link to this file.

Field 3 – **Owner:**

* Third field specifies owner of the file.
* In this example, this file is owned by username ‘maverick’.

Field 4 – **Group:**

* Fourth field specifies the group of the file.
* In this example, this file belongs to” maverick’ group.

Field 5 **– Size:**

* Fifth field specifies the size of file in bytes.
* In this example, ‘1176’ indicates the file size in bytes.

Field 6 – **Last modified date and time:**

* Sixth field specifies the date and time of the last modification of the file.
* In this example, ‘Feb 16 00:19’ specifies the last modification time of the file.

Field 7 – **File name:**

* The last field is the name of the file.
* In this example, the file name is 1.c.

1. **ls -t**

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

1. **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.

1. **ls -a**

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

1. **ls -A**

* A means almost all.
* To display all files including hidden files except . and ..

1. **) ls -h**

* display in human readable format

1. **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**

.