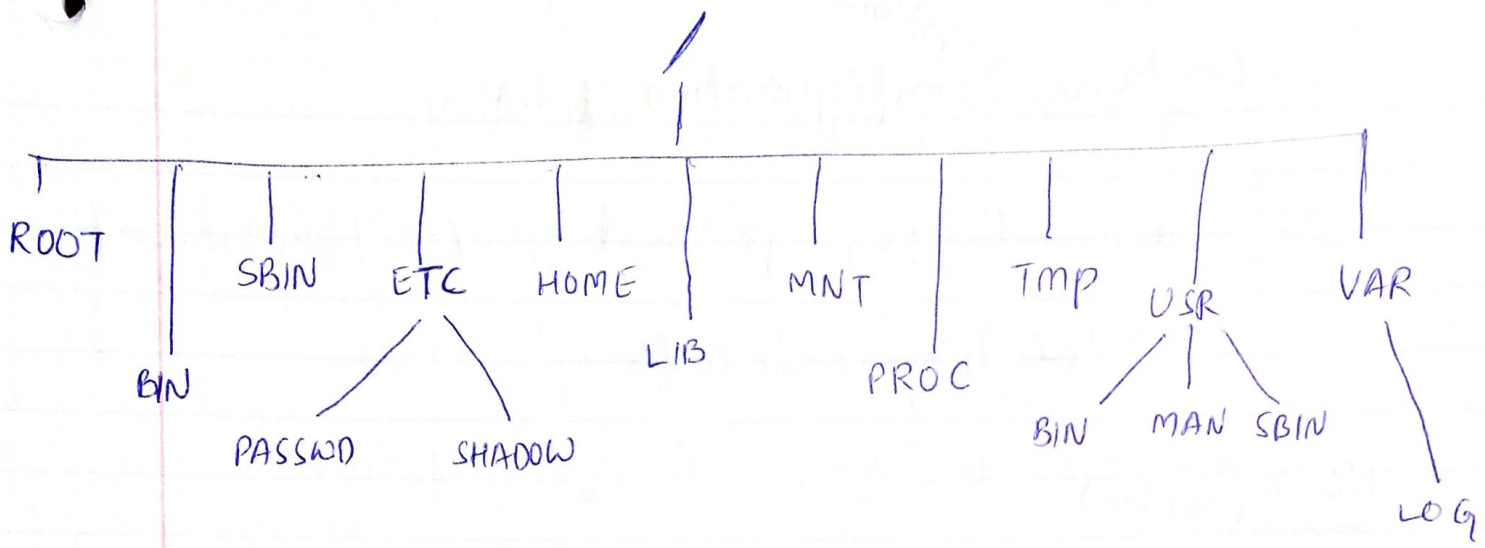
Linux Directory Structure :-



1) **ROOT:-**

- root account's home directory

2) **BIN: (/bin):-**

This directory contains executable programs which are needed in single user mode and to bring the system up (or) repair it. Ex: ls, ps, ping, grep, cp

3) **SBIN (/sbin):-**

Like /bin, this directory holds commands needed to boot the system, but which are usually not executed by normal users

Ex:- iptables, fdisk, reboot, ifconfig, init, route, swapon

etc (/etc):-

- Contains system-wide + user level configuration files

Ex:- account info, passwd, /etc/resolve.conf, /etc/logrotate.conf

HOME (/home)

- any user's home directory

LIB (/lib):-

- Libraries essential for the binaries in /bin, /sbin
- filenames are either, ld*, lib*.so.*

Ex:- ld-2.11.1.so, libncurses.so.5.7.

MNT:-

- Temporary file systems are attached like CDROM. (OR) USB DRIVE
- In this mount directory, sysadmins can mount file systems

PROC: (/proc):

- Virtual file system, providing process and kernel information as files.
- Generally automatically generated and populated by the system, on the fly
- contains information about system process
- This is a pseudo file system contains info about running process.

Ex:- /proc/<pid> → directory contains info about the process with the particular pid.

- This is a virtual file system with text information about system resources.

Ex. /proc/uptime

TMP (/tmp):

- Directory that contains temporary files created by system and users
- files under this directory are deleted when system is rebooted.

USR (/usr):

- Secondary hierarchy for read-only user data
- contains binaries, libraries, docs & source-code for second-level programs
- /usr/bin → binaries (files) for user programs

  Ex: at, awk, cc, less, scp
- /usr/sbin → binaries (files) for system administrators

  Ex: atd, cron, sshd, useradd, userdel
- /usr/lib → contains libraries for /usr/bin & /usr/sbin

/usr/local

VAR:

- Variable data where system must be able to write during operations

- logs -

ACCOUNT RELATED CMDs:

$ → normal user
# → root user

whoami → which user you logged in as

passwd <account_name>

passwd → root

id → id of the account # root → 0

→

locate 'filename' → locates a file ~~name~~ in entire f-s

- ls

  ls -l → description of files

  ls -a → hidden files

- mount

  mount cdrom

  eject cdrom

- scp :-

  - local machine ⇄ (TO / FROM) Remote server

  - secure copy — scp

COPYING TO :

- scp <file_name> <user_name>@<IP_addr>:<dest>

- scp hello.sh codebind@19.16.17.172 :/home/Desktop

→ scp a directory → (-r option)

- scp -r <dir_name> <user>@<IP-addr>:<dest>

Copying From :

I want to copy this file to

scp codebind@19.168.45.242:/home/codebind.txt <destination in local bs to copy to> /home/abc/

FILE SYSTEM:

- gedit file-name
- cat file-name
- less filename → when file is large, we can go through one line at a time.

→ RUNNING A PROGRAM:-

→ ./program

→ ps aux → all the process information

→ bg → background programs

→ jobs → to list the bg pgms

→ fg <job-number> → bring back-ground (bg) pgm to fore ground

Installing Software from Source in Linux:-

S/w Installation:

- Programming source code
  - C | C++
  - make
- RedHat package Manager (RPM)
- Advanced Packaging Tool (APT)
- Ports (BSDs)

Package Managers:

- Automate the installation, removal and management of s/w applns
- Similar to the Add/Remove pgms in Windows

→

Steps:

1- Unpack, "gzipped tar balls" of source code

2. Check readme instructions for specific install instructions

3. Usually the following commands in order

$ configure → local script inside the src directory which probes the system to find info about compiler, whether appropriate components installed to perform the installation

$ make ← does the compilation

$ make install → copies the binary file in the appropriate place of the system so that it can be found within the path & can be installed.

→ "run" ldconfig to load .so files after installation

SUID & SGID:

- These are "Special Permissions" in Linux
- setUID & setGID

- setUID is a special file permission for executable files which enables other users to run the file with effective permissions of the file owner. Instead of 'x', which represents execute permissions, you will see an 's' (SUID) special permission for the user.

```
rwx r-x r--   file.txt
↳ SUID
   ↓
rws r-x r--   file.txt
```

Ex:- passwd file in /etc

Everyuser can execute & change passwd eventhough there are not execute & write permissions

* It can be set using :

chmod 4777 file.txt

Similarly for SGID :-

4 - SUID
2 - SGID
} 6 - both.

STICKY BIT:

When this bit is set (or) permission is enabled, anyone can write (or) read this file (or) folder but cannot delete it.

Ex:

drwx rwx rwt 11 root root /tmp

↙

In temp directory anyone can write (or) read files but only root can delete the files.

→

sudo chmod 1777 abc.txt.

Users & Groups:

- ~~sudo~~

Create User:

$ sudo useradd -m "user_name" → add/create user

$ passwd <user_name> → set the passwd.

create group

$ sudo groupadd <students>

→ addUser to group

$ sudo user~~add~~mod -a (addition) -G (group) <group_name> <user_name>

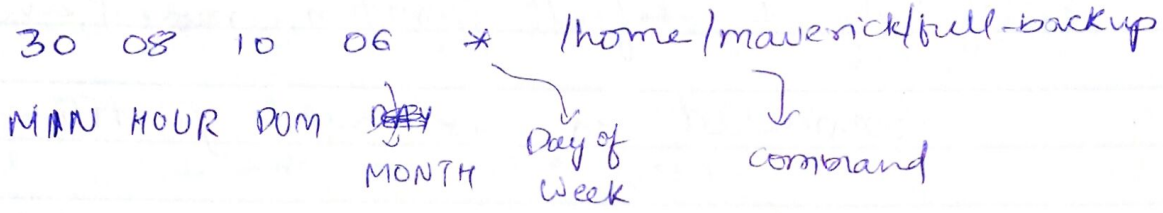
→ Give permissions to a specific group for a folder

$ sudo chown -R :<grap_name> <folder_name>

$ sudo chmod -R g+rwx <folder_name>

CRON JOB:

- Cron jobs are used to execute a job in a specific time as shown below



Field	Description	Allowed Value
MIN	minute field	0 to 59
HOUR	Hour field	0 to 23
DOM	Day of month	1-31
MON	Month field	0-12
DOW	Day of week	0-6
CMD	Command	Any command to be executed.

→ To list the crontabs for user:

- crontab -l

→ To edit a cron tab

- crontab -e

fdisk:

- It stands for "fixed disk" or "format disk".
- Needs "root" previlages
- Only '4' default partition, more than that are considered as extended partitions
- To perform operations on partitions like view, create, resize, delete, change, copy and move partitions
- <General syntax) - fdisk <operation> <partition-name>
- fdisk -l : →
  - list all existing disk partition on your system.
  - partitions are displayed by their device's names.

    /dev/sda

    /dev/sdb

    /dev/sdc

2) fdisk -l <particular_disk>

3) Create a new partition

$ fdisk <device_name>

you'll get available commands in fdisk

$ n → To add a new partition

$ first cylinder → default

$ second cylinder → assign size
↳ so the disk of particular size is created.

- After entering the size, enter 'w'
  $ w
- Now the ~~disk~~ partition is created & partition table is altered.

④ Delete a partition:

$ fdisk 'disk_name'

- It gives options, just enter 'd', to delete the partition.
- then asks for partition #, enter the same

Format the partition:

#mkfs.ext4 "partition_name"
(or)
: device_name"

=> Ex:
mkfs.ext4 /dev/sda7

Check the size of the partition:-

fdisk -s "partition_name"

Toggle the bool flag:-

fdisk :"device_name"

options :"a"

partition# :"2"

GREP

1) find content in a file.

- $ grep <parrots> <file_name>
(or)
text

Ex: $ grep 'pijush' student.txt

② exclude search string from o/p:

$ grep -v '<text>' <file_name>

③ Case insensitive

$ grep -i '<text>' <file_name>

④ Entire directory search:

$ grep -r '<text>' .

-r → flag to search the directory

. → current directory

→

Find:-

- find files and directories in linux system.
- find [area to look for] [parameter/action]
- find <directory> -name <file_name>

Ex: find / -name 'abc.txt

case insensitive

find / -iname 'abc.txt

Wild Characters for search completion

$find / -name pij*.txt

4) Search for a Directory:-

$find / -type d -iname pictures

→

Ps:- Used to monitor currently running processes and daemons.

1) View currently running daemons

ps -e

② To display processes running in our system

ps -f.

③ To run a process in background

use '&'

Ex: gedit &

4) Process run on a terminal

ps a

5) Non-CLI Process running in system

ps x

6) All processes running in a system

ps aux

Fields

Virtual mem size (amt of memory claimed by a process)

actual memory proc is using Resident memory size

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.1	19360	1512	?	Ss	9:26	0:01	/sbin/init

cmd used to run the process.

process starting using a terminal

7) pstree

↳ processes in tree structure

Systemd Basics

- It is init system which manages stop, start, restart of services

① Systemctl list-units

- lists all the background process & daemons

② systemctl status <process or service.name>

- $ systemctl status ssh.service

③ * If you disable a service, means it'll not come-up when you reboot the machine next time

* If a service is enabled it means, ~~the~~ it's started (or) it'll start when the system is up.

$ system disable <service>

④ $ systemctl stop <service>

5) $ systemctl --failed

lists all units, that failed to start when we started the computer.

6) systemctl restart <service>

7) systemctl reboot
(or)
power-off
(or)
suppend

NETSTAT :-

1) netstat -a :-

1. listing all the listening ports of TCP and UDP connections

2) netstat -at :

listing only TCP connections

3) netstat -au :

only UDP connections

4. netstat -l :-

listing all active listening ports connections.

5. netstat -lt :-

- active listening TCP connections

6. netstat -lu :-

- active listening UDP connections

7. netstat -lx

- all UNIX listening ports

8. netstat -s :-

- display statistics by protocol. shown ~~using~~ for the TCP, UDP, ICMP & IP protocols

$ netstat -s

IP:

2461 total pkts received

0 forwarded

:

ICMP:

TCP:

9) netstat -st

↳ statistics of only TCP protocol

10) netstat -su

statistics of only UDP protocol

11) netstat -p

displaying the service name with their PID

12) netstat -pt → TCP netstat -pu → UDP

13) Displaying kernel IP routing:

$ netstat -r

14) netstat -i

↳ network interface packet transactions including both transfering & receiving packets

15) netstat -ie ≅ ifconfig

- showing kernel interface table

Watch:

The command followed by "watch" will be executed for every '2s'

Ex: watch free -m

but you can also customize it, by setting the frequency

Ex: watch -n 1 free -m

- Now it'll be executed every second.

Top:

- To monitor process information
- ≡ ps aux

df:- (disk filesystem)

- To check how much storage space is available in total and to see the current utilisation of drives

$ df -h → (output in human readable units)

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	238M	0	238	0%	/dev

→ Finding Information about Storage Devices (Block Devices) with lsblk

$ sudo lsblk

O/P:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNT POINT
sda	8:0	0	100G	0	disk	
vda	253:0	0	20G	0	disk	
└vda1	253:1	0	20G	0	part	/

MOUNT:-

- Mounting the file system makes it available to the server at the selected mount point.
- A mount point is simply a directory under which the new filesystem can be accessed

device where it should be mounted

$ sudo mount /dev/sda1 /mnt

- pass the file system type using -t

$ sudo mount -t ext4 /dev/sda1 /mnt

Listing filesystem mount options:-

$ findmnt /mnt

Unmounting a filesystem:

$ sudo umount /mnt.