Computer:

- electronic device that performs tasks like browsing internet , playing videos, mp3

executing programs, applications etc. with some mathematical cal behind

- computer can understand binary lang (0,1)

computers -> CPU -> CHIPS -> ALU -> Gates (and,or,not,nand,nor) -> transistors (switch)

Operating System:

- software/ program which acts as a interface b/w the user and the computer hardware

- OS controls execution of all kinds of program's

- EG: windows, linux, macos, android

some of imp functions of OS

1. process management

2. memory management

3. Device management

4. security

5. Coordination b/w softwares

5. system performance

6. job accounting

7. scheduling

8. error detections

History:

- computers are as big as a room

- very difficult to operate

- every computer has a different OS

- every software is designed separately for different os

- expensive

- normal people cannot afford as these are expensive and difficult to understand

Linux History:

- linux is a free distributional version of unix

- unix is inspired by multics os (multi-user, multi-process, dynamic linking, hierarchical file system)

- Unix first version released in 1970 (assembly)

- later in 1972 C lang was developed by DR

- Unix is rewritten in C

- Unix gains popularity because of its portability, networking, availability

- linus torvald a finnish student started project LINUX

- first version was released in 1991

- he made linux as open source

- torvald only written kernel part of linux

Linux vs Unix:

- is both same..?

- linux and unix are different

- linux is derived from unix

- linux code was completely written from scratch not even a single line of code copied from unix

- linux is not an complete OS it's only kernel but Unix is a complete OS

- linux is packed into ditros to make it as an OS

- linux is an open course , Unix : closed source

- linux : C lang, other prog lang , Unix: C, assembly

- linux : opensource, Unix: mixed (open, closed)

- linux : monolithic , Unix : monolithic, micro, hybrid

- linux: Debian, ubuntu, fedora, redhat, android , unix : ibm aix, solaris, darwin, macos

- linux : patches are quick , Unix : slow

- linux : mostly free but some paid versions are also available (redhat), Unix : mostly paid but free versions are also available (FreeBSD, Solaris)

Distributions:

- other OS like microsoft, macOS combine each bit of code internally and release asa single package (windows 10,11,7,8.1)

- but linux is only a kernel

- different parts of linux are developed by different organizations

- distro combine those parts to name an OS

- Ubuntu, Linux Mint, Debian, Fedora, Redhat, Arch Linux

Where linux is used ?

- smart phones, supercomputers , desktop

- web servers, tablets, laptops

- washing machines, DVDs players , refrigerators

- routers, modems

- cars, airplanes

- rockets

Introductory terms in linux:

- binaries: executables in windows , all commands are binaries

- Case sensitive: linux is case sensitive

- directory : folder in windows

- home : one of linux directory where user data is stored

- root user : administrator in windows

- script : series of commands which gets executed in interpreter env

- shell : program that acts as an interface b/w the user and OS

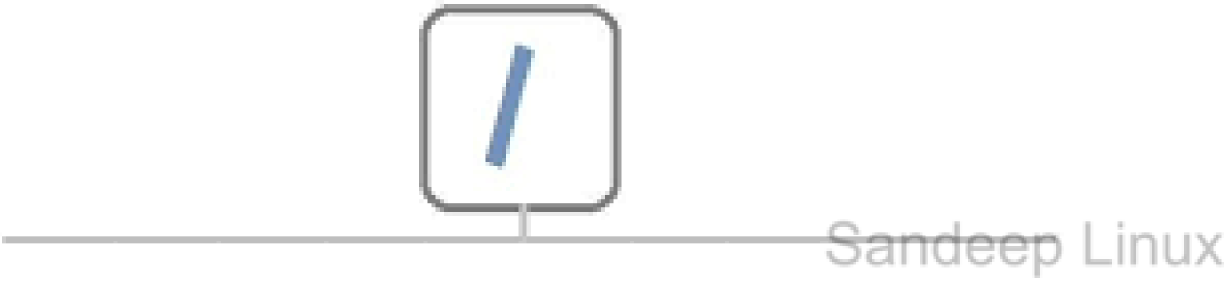
- terminal : CLI

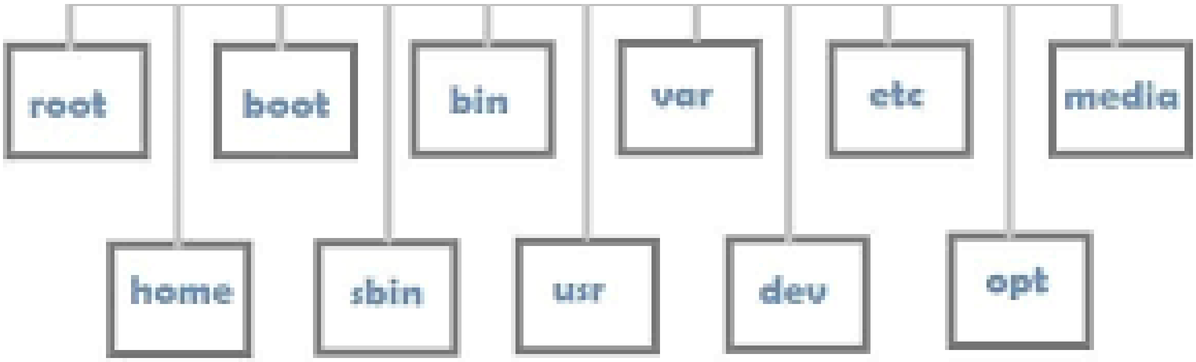
Linux File System:

- In linux everything is a file

- linux do not contain physical hard drive like windows

- linux has a logical hierarchical file structure





1)/bin - In this binaries folder we are having commands

2)/opt - If we download 3rd party s/w in linux that will store in this optional folder

3)/boot - whenever you on the system, booting data present. So that boot code will present in this folder

4)/root - Root user data present

5)/dev - device folder - External devices info stored here

6)/home - Every user data stored here

7)/tmp - we can store the temporary data

8)/usr - In linux kernel info and system binaries info present in user folder

9)/lib - shareable files present in libraries folder

10)/sbin - system binaries -extra commands present when we compared to bin

11)/mnt - mount - If we put pendrive, usb from external that data is coming into this directory

12)/var - In this variable folder, all logs stored.

i) System Commands

```
uname
```

used to get the Operating System of the server. Eg:- Linux, etc..,

```
uname  -r
```

Displays Linux kernel version

```
uname -a
```

Displays all information about Linux system information

```
uptime
```

Displays since how much time system has been running

```
uptime -s
```

Shows uptime in pretty format

```
hostname
```

Displays the hostname and IP address

```
hostname -i
```

Displays the IP address for hostname

```
ip route
```

Show table routes

```
date
```

Shows system date and time stamp

```
timedatectl
```

We will get the universal time

```
cal
```

Displays the current calendar, month and day

```
cal 2023
```

Displays the particular year calendar

```
cal 09 2024
```

Displays the particular month

```
last reboot
```

Shows system reboot history

```
who (or) w
```

Prints information about default user in our server

```
whoami
```

Prints information about all users who are currently logged in

```
top
```

List out the running processors in our system

```
hostnamectl
```

we will get the hostname details

```
hostnamectl set-hostname flm
```

Instead of IP address we get this name

```
ip address (or) ip addr
```

Shows IP addresses assigned to all network interfaces

```
timedatectl set-timezone Asia/Kolkata
```

Changing the universal to local time

ii) Hardware Commands

lscpu

Displays information about the CPU architecture

lsblk -a

Lists the information about all the block devices attached to the system

cat /proc/cpuinfo

Displays information about the CPU architecture

cat /proc/meminfo

Displays system memory(RAM) details

free

Displays system memory(RAM) details in KB

free -m

Displays system memory(RAM) details in MB

df -h

Report file system disk space usage i.e., how much memory we took and how much we consume, we will get all the details.

du filename

disk usage file name

```
du -sh filename
```

disk usage file name

```
fdisk -l
```

Getting the volume info

```
ps
```

To see all the running process

```
Kill -9 PID
```

To kill the process

```
dmesg
```

Bootup messages

iii) File Commands

```
touch filename
```

used to create a single file

```
touch f1 f2 f3
```

used to create multiple files

```
touch file{1..5}
```

create 5 files at a time

```
rm filename
```

used to remove single file

```
rm f1 f2 f3
```

used to remove multiple files

```
rm file{1..5}
```

used to remove 5 files

```
rm -f filename
```

used to remove a file without our permission

```
rm -f *
```

used to remove all files at a time

creating a folder:

using mkdir command:

- make directory

- used to create a new folder

- can create multiple folders at a time

- folders with hierarchy

Eg: mkdir folder

```
mkdir 1 2 3
```

mkdir a/b = create folder b inside folder a if folder a is present otherwise display error

```
mkdir -p a/b
```

create folder b inside folder a if a is present otherwise it will first create a then b inside a

```
mkdir folder1 folder2
```

used to create multiple folders

```
mkdir folder{1..7}
```

create 7 folders at a time

```
rmdir folder
```

used to remove empty directory

```
rmdir *
```

used to remove all empty directories

```
rmdir -rf folder/*
```

used to delete all the files inside the folder

```
rmdir -rf folder/filename
```

used to delete the particular file inside the folder

```
rmdir -rf folder
```

used to delete a particular folder along with inside the files

```
rm -rf *
```

used to remove all files and folders at a time

```
touch folder/filename
```

used to create a file inside the folder

Changing directories

```
cd foldername
```

used to change the directory

```
cd ..
```

used to go back to one step back

```
cd ../../..
```

used to go back to 3 folders back

```
cd -
```

used to go back to the previous directory

```
cd  (or) cd ~
```

used to go back to root directory at a time

```
cd /
```

To change the pwd to root directory which is the topmost/outermost parent   directory

```
pwd
```

present working directory. If you want to check the current path use this command

```
ll  (or) ls -l
```

used to see all the files along with the data (l - original list)

```
ls
```

used to see only file names

```
ls folder1
```

used to see the list of files present in folder1

```
ll -a (or) ls -al
```

used to see the both hidden and regular files

```
ll -r  (or) ls -r
```

used to see the files in reverse order

```
ll -t
```

used to see the latest files in top (t - time)

```
ll -ltr
```

To list the files in long listing format with sort by modification time, newest first and then in reverse order

COPY:-

```
cp file1 file2
```

used to copy the data from source file1 to destination file2

Note: Here, if file2 i.e.., destination file is not created also, it will create automatically

```
cp file1 folder1
```

used to copy the file to folder

MOVE:-

```
mv file1 file2
```

used to move/renaming the files i.e.., the data from file1 to file2

```
mv file1 folder1
```

used to cut the file and paste in folder cmp file1 file2

used to compare the 2 files

diff file1 file2

used to get the differences of a file b/w 2 files

using cat command:

- cat along with redirection operator
- we can add data into the file during creation
- press ctrl+d to come out of cat prompt

- redirection operator (>) redirects o/p of a command from standard file (file 1) to any arbitary file

double redirection(>>):

- > = clear previous data and enter new data into a file
- >> = it appends new data to old data

Eg: ls > cat.txt

cat cron.txt >> cat.txt

cat > filename

used to write/overwrite the data in a file

Note: Here, if we don't create file also, it will create automatically

cat >> filename

used to append/extend the data into a file

```
cat -n filename
```

used to read the data along with the line numbers

```
cat filename
```

used to read the data into a file

```
cat f1 f2 f3
```

used to see all the files data at a time

```
tac filename
```

reverse the data from top to bottom

```
rev filename
```

reverse the data from left to right

```
head filename
```

used to print first 10 lines of a file

```
tail filename
```

used to print last 10 lines of a file

```
sed -n '5,9p' filename
```

used to print the lines between 5 to 9

```
sed -n '7p' filename
```

used to print the 7th line

```
head -n 8 filename
```

prints 8 lines in a file

```
tail -n 4 filename
```

used to print last 4 lines in a file

```
wc filename
```

used to get the no of lines, words, letters in a file

```
wc -l filename
```

used to get only line numbers of a file

```
wc -w filename
```

used to get no of words in a file

```
wc -c filename
```

used to get no of characters in a file

```
cat aws | tee file file2 file3
```

copy one file data to multiple files at a time

tee - used to copy the data from one file to multiple files

| - used to perform two commands at a time

pipe parameter sends the o/p of the 1st command to the 2nd command

```
 cat file1 | tee -a file2 file3 file4
```

used to append/extends the data from file1 to file2 file3 file4

If you want to copy the folder data to serialized folders

#mkdir folder{1..7} > cd folder1 > touch file1

#echo folder{2..7} | xargs -n 1 cp -v folder1/*

xargs - used to copy from one folder to serialized folders

-v - verbose - If we use means we don't get permission denied

-n 1 -> used for copy to individual file

More:-

Used to display content of large/very large files, In form of pages

More command loads the entire file at a time into memory

Used to see the multiple files data at a time

Used to see all the files data at a time with %

press space enter or traverse

eg: more f1 f2

less:-

large/ very large display files in form of pages

It will loads the parts of the data into memory that is displayed

we do have search capability use /word name, Afte perform less command

eg: less filename

When compare with less/more user prefer less command because we got some free space in memory/ram

our ultimate aim is If ram is free means system is speed

locate:-

It is light weight command, used to find a file

It searches for given file in backend database which updates once in a day

problem when we search for a newly created file

```
#sudo updatedb - used to update the database
```

```
#locate filename
```

Find:-

used to search for a file/folder

Find do not depend on any database if actually searches on file system

Find provides multiple options for searching a file eg: name, size, group, user, date, permissions, etc..,

Note:- Some times you are trying to find some files you got permission denied errors, On that case use sudo before the command

```
# find path(or)pathType -name filename
```

```
# find / -name file
```

```
# find / -iname f1
```

```
# find /home/ec2-user -name file
```

```
# find / -type d -name fol-name
```

```
# find / -type f -perm 0644
```

Comparing files

```
cmp f1 f2
```

If it's same data in both files means, it gives nothing

If it's different means it will show error msg

```
diff f1 f2
```

It will prints the difference data which is present in a file

grep:-

Global Regular Expression print

It is used to search for a word inside a file, without going inside a file

```
grep "searchwordname" filename
```

```
 grep -n "wordname" aws

 grep -i "wordname" aws

 grep "wordname" aws -c  (wordcount)
```

Search multiple words inside a file

```
 grep -e "1stword" -e "2ndword" filename
```

search words in multiple files

```
grep -e "wordname" file1 file2
```

search multiple words and multiple files

```
grep -e "word" -e "word" file1 file2
```

Date filters

date +"%d" - give date (full details)

date +"%m" - give month

date +"%y" - give year

date +"%H" - give Hour

date +"%M" - give minute

date +"%S" - give Seconds

date +"%D" - give date (format change)

date +"%F" - give date " "   "

date +"%A" - give date in english

date +"%B" - give month in english

date +"%T" - give time

man pages:-

- manual pages

- display long description about the command

- If you don't know the command you can use map and find the details of the command

Eg: man command

        man nmap

        man ls

whereis:-

used to search for a binary file

It is also display manual page if available

Eg: whereis locate

which:-

used to search for a binary file but it is even more specific.

It's search for the binary file in PATH variable

        Eg: which locate

PATH:-

- env variable

- locations of binaries

Eg: echo "$PATH"

Types of files:

- : regular file

d : directory file

c : character file

b : blocked file

| : pipeline file

File Modifications:

Vim editor: It is used to modify the data in a file.

It has 3 modes

1. command mode

It is the default mode in vim editor. It is used for operations like copy, delete, paste the lines in a file

keys/commands:

In linux we can't press the keywords directly. you have to do in esc mode

yy : used to copy the entire of the line

3yy: copies 3 lines from our cursor

p : used to print the copied data

4p : it will print 4 times

dd : used to delete the entire line

5dd: used to delete 5 lines

gg : used to go to the 1st line of the file

G : used to go to the last line of the file

5gg: our cursor goes to the 5th line

shift:8 -> our cursor goes to the 8th line

u : used to undo the changes

ctrl+r : used to redo the changes

shift:set number -> used to give the numbers in a file

/word or ?word -> used to search a word

shift:%s/old word/new word/g -> to replace a word (g - global)

2. Insert mode

Used to insert the data in a file

esc: used to go to insert to command mode

i : used to go to command to insert mode

O : used to create a new line (up)

o : used to create a new line (down)

A : used to go to the end of the line

I : used to go to the starting of the line

3. save & quit mode

It is used to save the modified data in a file

And quit from the VI editor to root user

:w - used to save the data in a file

:q - used to quit from the vim editor

:wq - used to save & quit from the vim editor

:w! - forcefully save the data in a file

:q! - forcefully quit from the vim editor

:wq! - forcefully save & quit from the vim editor

Note: All these vim keys will work on command mode

4. visual, reverse, record modes - we don't use here

4)User Management:

Adding user:

using adduser/useradd command:

- used to add a new user into linux os

Eg: adduser username

```
    adduser -u 8765 pop

    adduser -u 6578 -gid 8765 bob
```

note:

- whenever we create user, one group also will gets created with the same
- whenever we create user, Inside /home automatically folder will gets created with same name

backend process:

- controller will edit the following files

/etc/passwd - contains info about users in linux OS. we can see the users in our servers. First, last are the actual users. In b/w there are system user accounts we called as daemons.

/etc/shadow - users password info present

/etc/group - contains info about groups in linux OS

/etc/gshadow - groups password info present

- creating a new directory in home folder
- assigning required default permissions to the new folder created

```
kali:x:1000:1000:,,,:/home/kali:/usr/bin/zsh
```

kali : username

x : password is stored in /etc/shadow

1000 : uid - unique identifying number assigned by the os to a user

1000 : gid

,,,, : info about user, shows additional info about the user. Using below command we can add the additional data

```
useradd -c "hi this is linux" username
```

/home/kali : path of directory creates a folder and contains user info folder

/usr/bin/zsh : shell associated with the user and user performed commands stored su - pop/ su pop

Login from root user into our created user

passwd pop

setting a password for user

cd /home -> ll

display the user name with folder

note:

- password will changed by users who have permissions/access (forgot the passwd)

- Root user can access/go to any user, no restrictions. Because it has all privileges.

useradd -M pop

Whenever if we create a user, folder will not create

Removing user:

using userdel command:

- user delete

- used to remove a user from linux os

Eg: userdel dhoni

userdel -r virat - delete user in both server & home directory

userdel -r -f rohit - delete user in both server & home directory and also deleted user related files

note:

- whenever we delete the user, only user is deleted not the folder

- At a time, we can delete single user only

Expiration:

If i want to give access to the user upto 2 months, then after user will automatically delete.

eg: useradd -e 2023-08-25 / 25-09-2023 filename

After this date user will delete

Adding a Groups:

whenever we create a user, automatically group created.

eg: useradd sandy -> cat /etc/group -It will visible here

Creating own/manual group

groupadd Sandy

Deleting group

groupdel sandy (x) - we can't delete here we can delete our manual groups

userdel -r sandy

id filename - we can see the id's of group & user

Changing the owners of a file:

why to change the owners means

Suppose, I'm having one file in root user. I want to access in jenkins user. But it's not possible, because owners are different. So, if we change the owners root to jenkins means then we can access

chown username filename

changing the owners in user

chgrp groupname filename

changing the owners in group

```
chown user file file1
```

changing the owners for multiple files

```
chown username:groupname filename
```

Changing the owners and groups at a time in file

```
chown -R username:groupname folder
```

Changing the owners and groups in folder and also folder inside files

5)File Permissions:

- If you want to give all permissions to the file
- very important
- directly impact on security of os

using chmod command:

- change mode
- used to change permissions for a file

permissions in os

read(r) : we can access data inside file/folder = 4

write(w) : we can change /alter data in file/folder = 2

execute(x): we can execute/ run file = 1

-        : type of a file = 0

permissions are given to three set of people

owner(u): who actually creates the file/folder

group(g): every user is associated with a group

others(o): everyone other than group and owner comes under others

```
ls -l

drwxrwxrwx
```

d : file / folder

rwx : permissions of owner

rwx : group permissions

rwx : other permissions

we can change permissions of a file/ folder using 2 ways

1. using operators

+ : add a permission

- : removes a permission

= : replicates permission

```
Eg: chmod u+x file

chmod u-x,g+w,o+r abc

chmod u=rwx abc

chmod a+x abc
```

2. using numeric codes

rwx

000 - 0 - no permissions

001 - 1 - execute

010 - 2 - write

011 - 3 - write and execute

100 - 4 - read

101 - 5 - read and execute

110 - 6 - read and write

111 - 7 - all permissions

Eg: chmod 123 abc

chmod 456 abc - for file

chmod 777 abc

chmod -R 777 folder1 - changing permissions in files inside the folder also

note: file name will change when we give/change the execute permissions

SSH:

- It is a protocol, nothing but secure shell

- used to remotely connect and control a system

- both systems should be in the same network

- Through ssh we can login into remote server to our computer

note: If both servers are using same n/w SSH will be possible otherwise don't

Eg: service ssh status

service ssh start

service ssh stop

service ssh restart

ssh username@serverpublicIP

Login through cmd from remote server

Standard file streams

we're having 3 types of files

file0 - input file

file1 - output file

file2 - error file

how exactly the command run in linux

terminal (open) --> command (typing) --> (It goes to file system) file0 --> (command need to execute that's why it's going to)shell prog --> path varia (find) -->(present means) execute --> file 1 --> monitor(we can see)

If it's can't find in path variable it goes to file2 -error file

using echo command:

- echo with >

Eg: echo "Hello world" > hello.txt

Software management:

- important feature

- we do install/uninstall third party software into OS

- we do update our software/os

- in debian systems software is managed by using apt tool

using apt tool:

- It is advanced packaging tool

- we can search, install, remove a software from linux os

Searching for a package:

- in linux os even before adding/installing a software we can search for particular package is available/not

- we have to used keyword search to search for a package

Eg: apt-cache search software

```
apt-cache search nmap
```

Adding / Installing a software:

- using install keyword along with apt/yum

Eg: apt-get install software

```
apt-get install nmap

nmap -v
```

Removing a software:

- we do have 2 kinds of removing keywords

i)using remove keyword

- it will be removing only the software but not it's config file

Eg: apt-get remove nmap

ii)using purge option

- It will removes software along with it's configuration files

Eg: apt-get purge nmap

updating system:

- For updating software/ app/ os is very important as updates brings

- add security and add new features to our software

- in linux, updates won't reach us automatically as in windows

- we have to manually update our system

- in linux os we do have two kinds of updates

update

upgrade

i) update

- it updates the list of software, applications that are ready to get update, But it doesn't update the actual s/w

- it is fast for updates

Eg: sudo apt-get update

ii)upgrade

- it actually updates every software that has updates

- it takes hours/days for upgrade

Eg: sudo apt-get upgrade

Repository:

- repo

- It is nothing but a server that holds the software for a particular distribution

- every distro will be having a separate repo

Eg: nano /etc/apt/sources.list

After this one, copy the link address paste in browser.

Process Management:

- process is nothing but a program in execution

- in linux, unlike windows all commands run on terminal/shell

- tuning or controlling execution of a process is process management

Any process can be run in 2 ways:

foreground process :

- by default every process run as foreground process

- process that runs on terminal

- until foreground process completes it's execution we cannot use terminal

background process :

- process which run on background

- we can use terminal even background process do not completes it's execution

- we have to manually create background process

So, overall whatever the commands we executed is in foreground process because everything is execute in terminal

Types of processes:

1)parent process:

- process created by the user

- all processes will have a parent process

- kernel process will be parent processes if process is called by user directly

2)child process:

- process created by another process

3)orphan process:

- if a parent process completes it's execution even before child process

- child process becomes orphan process

- init process will be new parent for orphan process

4)zombie process:

- if a process competes it's execution but shows up in process status then it is called zombie process

5)Daemon Process:

- system-related background processes

- if this process not there means our os will not work properly

init process:

- when you turn on your system, booting will happen. After one main process execution started i.e., called init process.

- So, init process is nothing but a parent process. Through that parent process whatever the process we're created it will be like child process

- If you study OS internally, every process will have a parent-child process relationship.

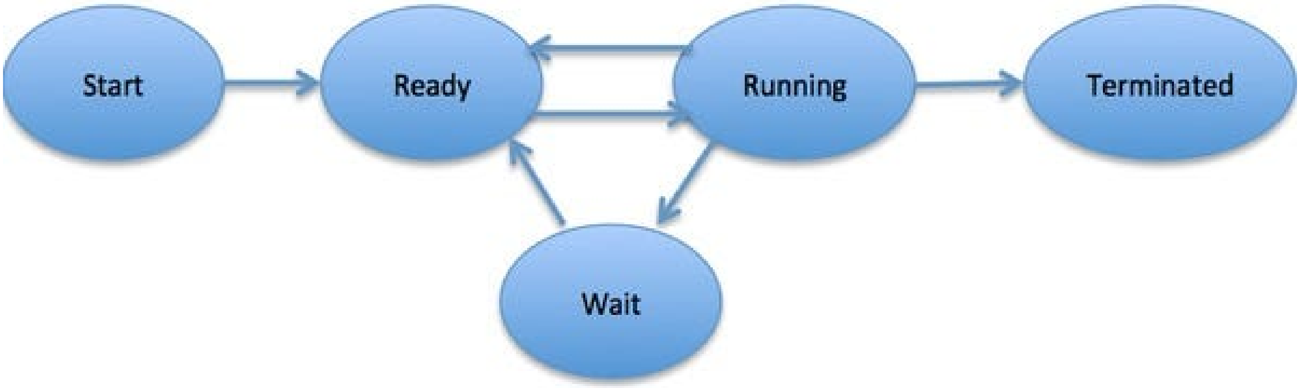
PCB: (Process Control Block)

- every process contains PCB

- It is nothing but a data structure like linked list.

- PCB contains below things

1) Process State:



2)PID - Unique Id given by OS to process

3)Program counter - contains the address of next instruction to be executed

4)Register information - cache memory

5)list of open files -

6)CPU scheduling information - allow process info stored

- Round Robin algorithm

- shortest job first

- largest job first

7)memory management - pagein, pageout

8)I/O Status and more.....

- Which input and output device using info present

process table: In this table we're having 2 columns (pid, pcb address)

commands:

ps :

- process status (display current processes info)

ps -e :

- display daemon processes. These are the system process which runs on background

ps -f :

- display process info with full options

- So, in this we can clearly see parent-child process relation.

ps aux :

- display all processes running in the linux OS

jobs:

- display jobs that are running or suspended in background

Creating a Background process:

- command -> (enter) = foreground process by default

- command & -> (enter) = background process

eg: cat > file.py -> python file.py -> ps - you can see python in cmd that's a foreground process

while true:

pass

(ctrl+z)

ii) python file.py & -> ps

So, this is way to create foreground and background process

fg % PID

in order to bring background process to foreground

top:

- display real time dynamic view of current running processes

- press "q" to quit from top prompt

PID : process id

PR : process priority (value lower, process having high priority)

NI : nice value (value negtive, high priority)

VIRT : something like virtual memory used by the process

USER : It represents the user who created the process

%CPU : display cpu consumed port

SHR : shared memory through IPC(Inter process communication)

RES : It displays consumed ram

%MEM : It displays consumed memory

Command : which command created the process

kill signals:

- way of communicating a message from one process to another is a signal
- kill signal command in linux does not only stop processes
- But also used to pause, restart, terminate, suspend, wait, continue
- Overall, we can control the execution of a process

Eg: kill -signal pid

kill -2 pid, kill -SIGINT pid (ctrl + c)

kill -3 pid, kill -SIGQUIT pid (ctrl + d)

kill -9 pid, kill -SIGKILL pid (used to stop/terminate process at any cost)

kill -15 pid, kill -SIGTERM pid (terminate)

kill -17,19,23 pid , kill -SIGSTOP pid (stop)

Networking:

Network means If group of devices are communicated through router is called networking.

- ifconfig -a :

command used to display network configuration of all available network interfaces and their settings

eth0 - ethernet/lan info - used to access the internet

lo - reperesents wifi info/ shows configuration

Overall, network interface info shows

hostname :

It will display machine hostname

hostname -i:

display machine ip address

host :

used to display some DNS lookup activity i.e., If you want to know any

website DNS information use host.

Eg: host instagram.com - show you the DNS ip address

host 134.443.222.133

netstat: (network statistics)

used to display all listening ports info

Eg: netstat -l

netstat -t - display all active TCP connections

port:

It is nothing but a s/w(or)n/w gate. If you want to communicate with internet our OS assign the ports for google. that means data out, data in

It contains 1- 65535 ports

1- 1024 default ports. Remaining ports we can use in our daily life

popular ports:

ssh 22

http 80

dns 53

https 443

ping :

used to check whether our system is able to reach destination i.e.., n/w reachable or not checked

eg: ping google.com - internally it uses ICMP packets

ping -c 4 google.com - how many requests you want ?

dig :

used to query DNS servers for information i.e.., get DNS information

eg: dig instagram.com

whois :

it displays website info

eg: whois instagram.com

ip:

- command line tool to perform network admin tasks

Eg:ip addr - same as if config

ip route - displays route table

ip address show eth0 - displays ethernet ip address

bmon:

- bandwidth monitor

- used to monitor traffic across interfaces

- used to display real-time bandwidth consumed by the n/w interface

Eg: bmon

ifstat:

display IN-OUT packet information of all n/w interfaces i.e.., how much data come inside and going outside info. it shows in kb

Eg: ifstat

ifdown lo:

used to take down a particular n/w interface

ifup lo:

used to turn on a n/w interface

traceroute in linux , tracert in windows:

- used for n/w trouble shoot

- display the information of intermediate routers that our request reach before going to destination

Eg: traceroute www.google.com

tracert www.google.com (try in cmd)

ethtool:

used to display detailed information about n/w interfaces

Eg: ethtool eth0

curl:

- used to communicate with internet through command prompt, terminal

Eg: curl https://www.facebook.com/

curl -h

- used to retrieves the data from external devices from internet. So, our log filepath we have to give

# curl "log path file" | grep ERROR

wget: (web get)

- downloaded from internet to local

eg: wget log link -> cat log | grep ERROR

diff b/w curl and wget:

In curl command we're able to do it in a single command.

Using wget, we're able to use it 2 different commands.

Based on your requirement use above ones.

alias:

- It's a temporary way to rename the command

- giving another name to a command

- If you need your alias permanent Go to ls -a > .zshrc (Zshell conf file)

- So, whenever you open the terminal one program is running because of zsh script

- So, go inside that .zshrc file and add that alias means it takes permanent

#nano .zshrc (open the file)

Note: If you want to do any changes in .zshrc take backup(using copy command)

eg: cp .zshrc zshrc.bak1

Eg: alias dhoni="ls -l"

crontab:

- used to schedule a task to get executed in future

Eg: crontab -l : Available crontabs displayed

crontab -e : Add a new/Edit a crontab

So, using this command we can create a file that means we can execute the task in future, just we have to schedule it

- How to schedule a task in crontab?

```
#crontab -e

18 11 * * * -> create at 11:18 am

* * * * * echo "hi" > file.txt
```

du:

- disk usage

- display the memory usage of a file

```
Eg: du /home/kali/Downloads

    du -h /home/kali/Downloads (h means human readable data o/p)
```

pipe(|):

- used to redirect output of a command as input to another command

```
Eg: command 1 | command 2

top | head
```

nproc :

gives the cpu of the current machine

ps -ef :

It provides the entire detail processes in the full format

i.e., start or stop or daemon processes we will get here

In every VM there is python by default running

awk:

- It is a powerful command

- It can filter out the information from your o/p

```
eg: awk -F" " '{print $n}'    n=column number
```

diff b/w grep & awk:

grep command gives the entire statements. but awk it can also gives a specific columns from the o/p