Linux :-

(Terminal Emulator)

Shell → {Command line Interface}

↓

interpret commands and tell

OS what to do.

How does shell run commands ?

'python' ....

↳ Runs an executable associated with python Stored in path.

ls → list Directories.

mkdir → make directory.

cd → Go to

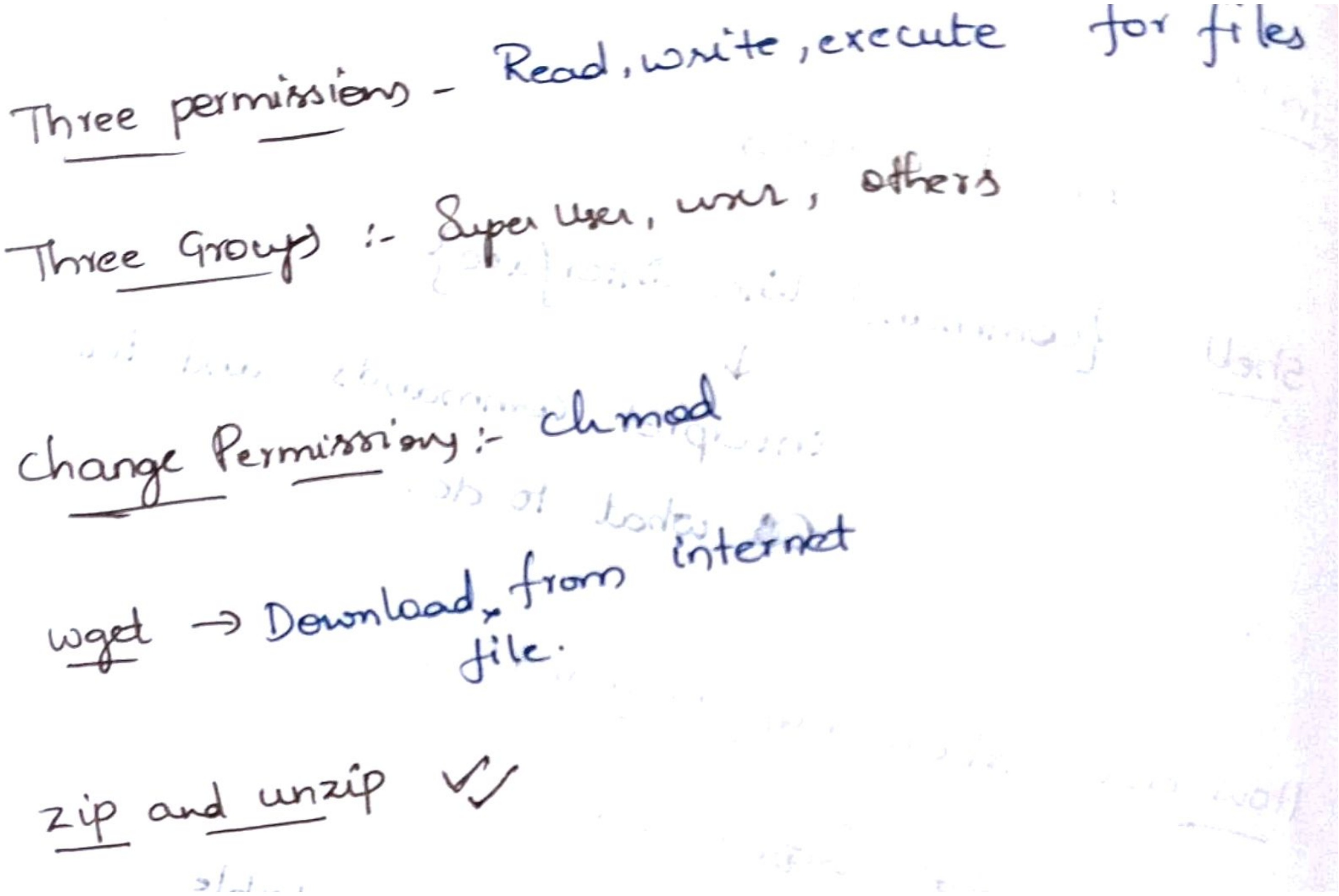
pwd → Print working directory

ls -a → lists hidden files also

ls -R → all Subdirectories too

cat → See what is in the file.

Path :- When a command is ran, program checks for the executable of that command in all the paths.



Three permissions - Read, write, execute for files

Three Groups :- Super User, user, others

change Permissions :- chmod

wget → Download file from internet

zip and unzip ✓✓

Linux :-

Kernel :- Brain of the Linux OS.

controls the hardware and makes the hardware interact with applications.

- * Core part of OS
- * Software, system process
- * Programs to machine code.

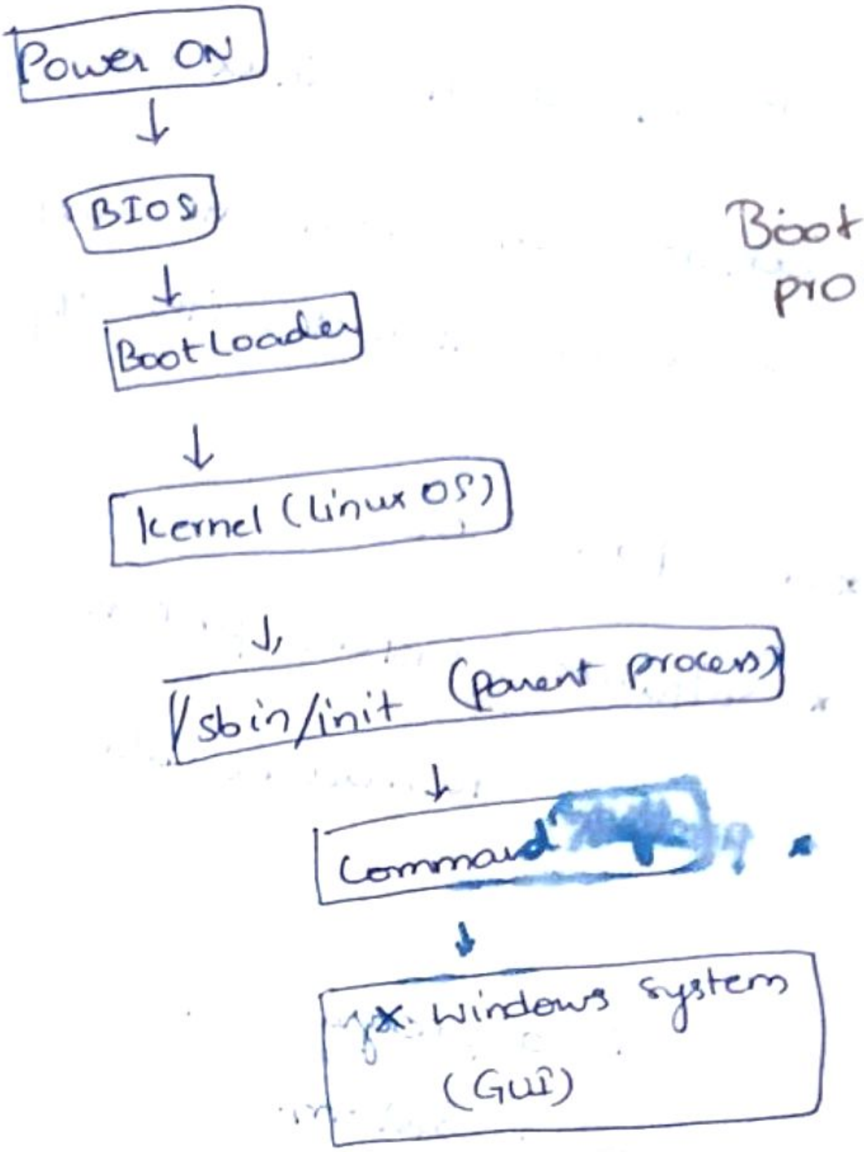
Bootloader :- Is a program that boots the operating system.

Service :- Program that runs as a background process

Shell :- Command Line INTERPRETER

Interprets the Command line Input and instructs the OS to do necessary operation

eg bash, zsh, etc



Boot process.

Partition :- A section of a disk

File System :- A method of storing/finding files on a hard disk.

/
Root
directory

/bin/ essential user command binaries

/dev/ device files

/etc/ host specified system Configurat

/home/ user home directories

/opt/ Add-on application software packages

/sbin/ System binaries.

In Linux, all application programs we are running are processes.

A Process is simply an instance of one or more related tasks (threads) executing on your system.

Interactive Processes :- Need to be started by the user, either through command line or GUI.

Batch Processes:- Automatic process which are scheduled These are implemented in FIFO order.

Daemons:- Server processes that run continuosly.

Threads :- Lightweight processes, Run under main process.

Kernel Threads :- Threads run by the kernel that user has little to no control over.

A critical kernel function called Scheduler constantly shifts processes on and off the CPU.

- * Running State
- * Sleep State.

* At any given time, there are multiple processes being executed. The OS keeps track of them using unique Process ID (PID) number.

* Terminating a process using "kill" command.

Load Averages :-

(0.45, 0.17, 0.12) → 12% utilized on average for the last 15 mins.

For last minute, 45% utilized on average

↓

For last 5 minutes, 17% utilized on average

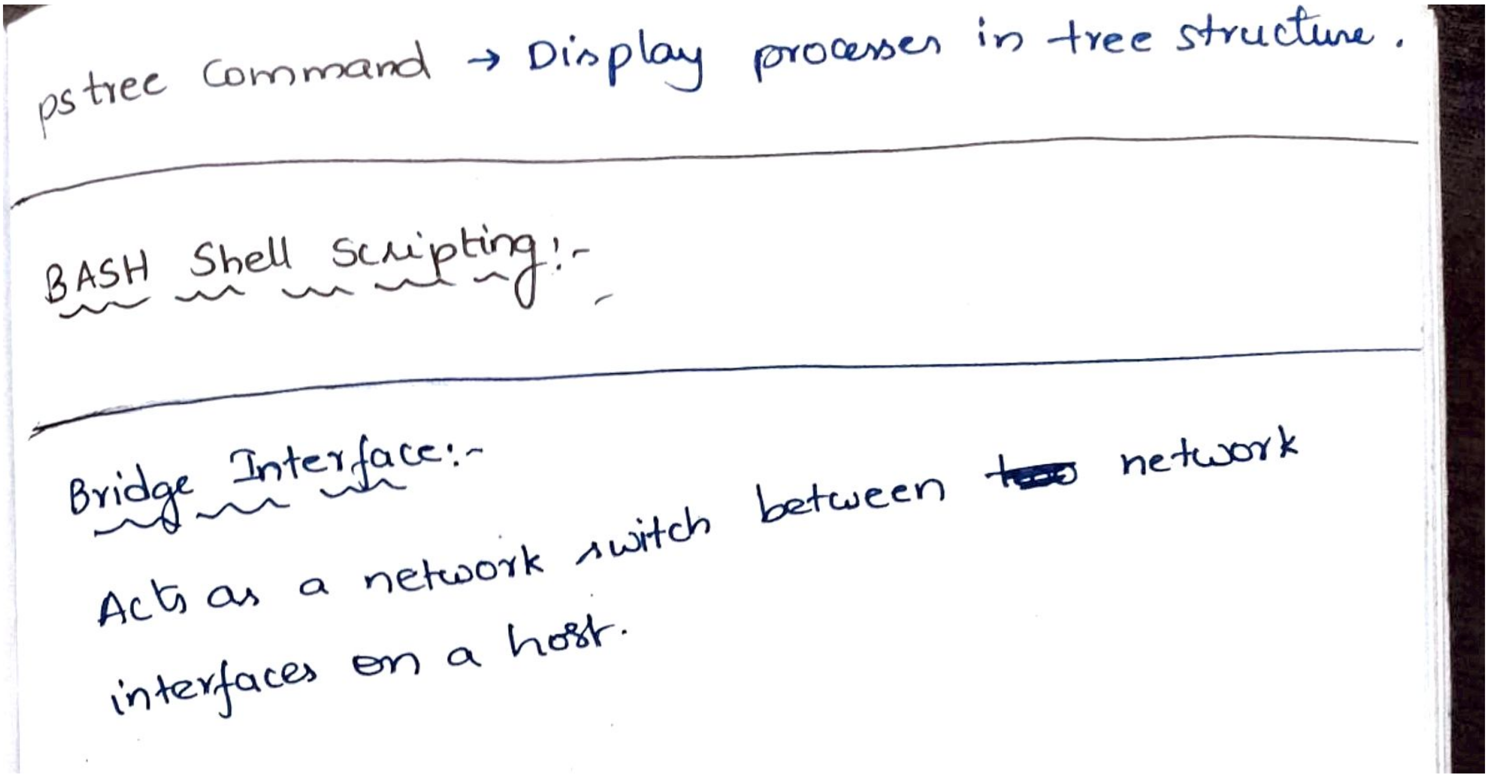
Background and Foreground processes :-

Job → Command launched from shell.

Foreground jobs → Run directly from shell → Only 1 at a time

Background jobs :- Push foreground to background to run on a low priority.

↳ Can push to background by suffixing '&' to the command or using "bg"



pstree Command → Display processes in tree structure.

BASH Shell Scripting:-

Bridge Interface:-

Acts as a network switch between ~~two~~ network interfaces on a host.