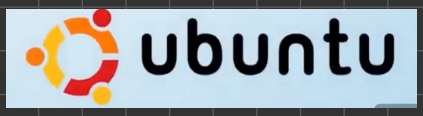
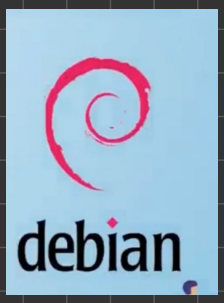
- A Linux Distribution is an Operating system made from Software collection i.e. based upon Linux Kernal.

* Some Linux Distributions :-

There are on an avg. more than 600 Linux Distributors providing diff. features.





* Linux Fundamentals :-

① KERNAL :-

Kernal is a computer program at the core of computers Operating system with complete control over everything in the system.

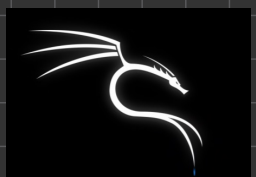
- It acts as a interface blw the hardware & software.
- Kernal takes responsibility for deciding at any time, which of many running programs should be allocated to the processor.
- RAM is used to store both program instruction & data. The kernal is responsible for deciding which memory each process can use.
- It handles peripherals like keyboards, monitors, printers & speakers.

⇒ Different types of Kernals in different O.s.

① Mac → XNU Kernal
② Windows → Windows NT Kernal 10
③ Android → LINUX.

* TYPES OF KERNAL :-

① Monolithic Kernal (Linux)
② Microkernal
③ Hybrid Kernal. (Windows)
[Hybrid Kernal = Monolithic + Microkernal]



- Difference in Monolithic & Microkernal:-

① for eg., if we want to open "paint", so in Monolithic, the sources/permission will be taken from kernal, but in Microkernal, the sources/permissions needed to open "paint", will be available in user space, so we don't have to go to kernal for resources.

② In Monolithic, kernal is used for each & every operation. But, in Micro kernal, only the major operations/major part is performed by kernal. & other operations are managed by user space.

③ In Microkernal, if any big operation will be performed, then it will first go to user space & then access kernal, which somewhat slows down the process. But, in Monolithic kernal, the operations will directly access kernal, which makes the process bit fast.

SHELL

- The shell & kernal is a part of o.s.
- When user gives command, then the request will go to shell parts.
- The shell part is also called as Interpreter., it translates the Human Program into Machine language. Then the req. will be transferred to Kernal.
- shell provides you with an Interface to Linux System.
- shell is an enviornment in which we can run our commands, programs, & shell scripts.

---

* Types of Shells :-

- The first shell to appear on UNIX system was Borne shell (sh)

① Borne Shell (sh)

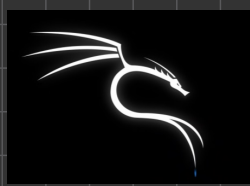
- Original unix shell written by Bill Joy at UC Berkeley.
- First shell to appear on Unix System.
- Usually installed as /bin/sh on most version of unix.

② Bash shell (Bourne again shell)

- Developed for GNU project.
- This is an actual standard Linux shell.
- The $ character is default prompt.

③ C shell (csh)

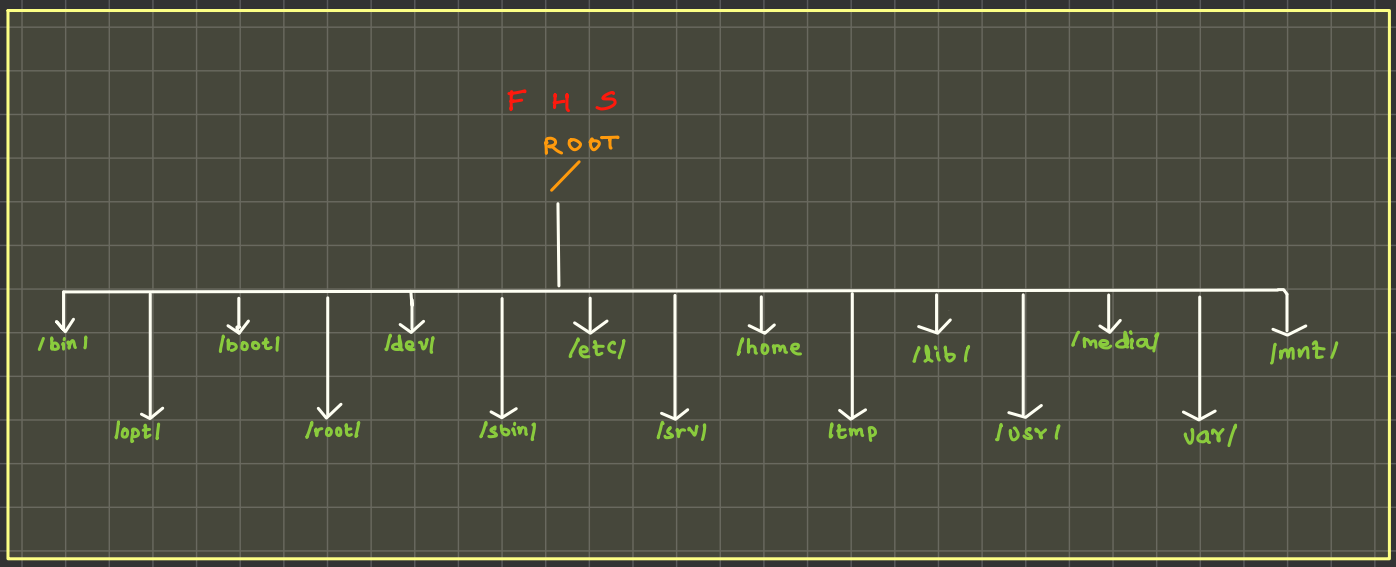
- The % character is default prompt.
- Ideal for learner comfortable with C.



- This shell is mostly used by Network Admin.

* Linux File System Hierarchy :

- The File Hierarchy System (FHS) project was began in 1993.
- The goal was to come to a consensus on how directories should be organized & which files should be stored where.
- so that distributions could have a single reference point from which to work.



① /bin/ :- contains commands that may be used by both, system administrator & by users.
eg. Applications.

② /boot/ :- This directory contains everything required for boot process.

- Boot stores data that is used before the kernal begins executing user-mode programs.
- The operating system kernal must be located in either / or /boot.

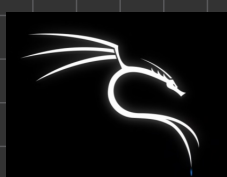
③ /dev/ :-

- The /dev directory is the location of special or device files.
- All hardware files are present in /dev (device) folder.

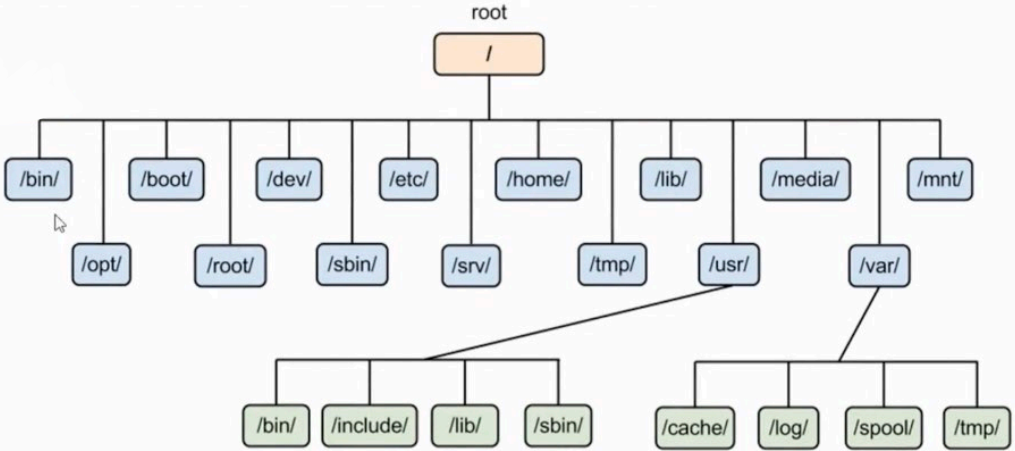
④ /etc/ :-

- The /etc hierarchy contains configuration files.
- configuration files for boot loaders that are not required at boot time must be placed in /etc.

⑤ /home/ :- user home directories.



Linux File System



⑥ /lib/ :-

The /lib contains those shared library images needed to boot the system & run the commands in the root filesystem.

⑦ /media/ :- Mount Point for removable media.
- This folder is managed by O.S. & not by administrator.

⑧ /mnt/ :- This directory is provided so that the system administrator may temporarily mount a filesystem as needed.
- This folder can be managed by administrator also.

⑨ /opt/ :- Add-on application software packages.
- eg. if we install printer, then it's files will be in this folder.

⑩ /root/ : Home directory for root user.
- eg. The administrator's account information will be present in this folder.

⑪ /sbin/ : (standard bin)
- Utilities used for system administration.
- This folder contains the files which can be runned by administrator only & not by all users.
- /sbin contains the binaries essential for booting, restoring, recovering, and/or repairing the system in addition to binaries in /bin.

- The files in /bin can be accessed & run by administrators as well as users, but files in /sbin can only be accessed & runned by administrators.

⑫ /srv/ : /srv contains site specific data which is served by this system.

If you were using the Apache HTTP server to serve a website, you'd likely store your website's files in a directory inside the /srv directory.

⑬ /tmp/ : The /tmp directory must be made available for programs that require temporary files.

- Files & directories located in /tmp will be deleted whenever the system is booted.

⑭ /usr/ : User shareable data.

- /usr is shareable, read only data.
- The /usr directory contains application & files used by user.

(15) /var/ : /var contains variable data files.

- This includes files, administrative & logging data & temporary files.

(16) /proc/ : It contains special files that represent system & process information.

(17) /lost+found/ : This is an important directory which is useful for recovering files which are not properly closed due to many reason such as power failure.

(18) /run/ : This directory contains system information data describing the system since it was booted.

Linux file types

- There are 7 different file types in Linux OS.

-	: Regular file	s	: Local socket file
d	: Directory	p	: Named Pipe
c	: Character Device file	l	: Symbolic Link.
b	: Block Device file		

---

① Regular Files :- · Most common file type on linux

- You can create regular file with touch command.

② Directory : · Second most common file type on linux.
· same as that of Directories in Windows.
· Can be created with mkdir.

③ Character & Block Device file:

- Character Device file refers to the peripheral files.
  eg. files required for mouse, keyboard etc.
- Block Device file refers to the files required by Removable disk or HDDs.
- character & block device files allows users & programs to communicate with hardware peripheral devices.

④ Local Socket File:

Local Domain sockets are used for communication between processes.
eg. If two or more than 2 processes are going on, & a process needs help of another process, then they communicate with the help of Local Socket file.

- Generally, they are used by the services such as X Windows, syslog, & etc.

⑤ Named Pipe :- • Named Pipe allows communication b/w two local processes.

- They can be created by mknod command & removed with rm command.

⑥ Symbolic Link :- With symbolic link, administrator can assign a file or directory multiple identities.

This means, we can link another file in a file.

- To see all the files present, just type ls.
- To see the files in list type ls -l.
- To see the hidden files also, type ls -la.

① bin → contains executable files, that can be executed by users as well as administrators.

② boot → contains boot loader related files.

③ dev → contains device files.
eg. terminal devices, usb, etc.

④ etc → contains configuration files req. by all programs.

⑤ home → Home directories for all user to store their files.

⑥ lib → System libraries.
eg. filenames → ld or lib or so.

⑦ media → Removable media devices.

⑧ mnt → Mount directory.
- Temporary mount directory where sysadmis can mount system files.

⑨ root → Every single file & directory starts from root directory.
- only root user can write on it.
- /root is root user's home, which is not same as /.

⑩ opt → optional add-on applications.

⑪ sbin → standard bin. (system binaries)
- But, linux commands located under this can be runned only by system administrators.

⑫ srv → service data
- server specific data

⑬ tmp → Temporary files
- files in this directory gets deleted when system is rebooted.

⑭ usr → user programs

9. /usr – User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin
- /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

⑮ var → variable files

includes →
- sys log files
- Packages & database files.
- emails
- print queues
- lock files
- temp. files needed for reboot.

⑯ proc → process information
- info about system process
- info about running process

FAT32

(File Allocation Table)

- Increases no. of bits used to address clusters.
- Reduces size of each cluster.
- supports larger disk. i.e. upto 2TB & better storage efficiency.
- 4 GB max file size.

NTFS

(New Technology File System)

- Windows NT O.S. uses it for storing & retrieving files on Hard disk.
- Linux & BSD have free & open source NTFS driver
- mac OS → Read only NTFS support.
- Max file size → 16 TB ; cluster size → 4 kb.

Ext4

(Extended File System 4)

- Ext 4 supports file-based encryption.
- Being used by linux kernel
- It supports journelling.

Other Commands :

- curl ⟶ Curl is a tool to transfer data from or to a server, using one of the supported protocols.

  To get all the commands in curl, type curl --help.

- ifconfig ⟶ gives network related info.

  If you want eth0 to be disconnected, or it should stop, then type ifconfig eth0 down.

  to turn it on again: ifconfig eth0 up

- hostname
- uname ⟶ gives kernal name
- uname -a ⟶ gives system & user info.
- who ⟶

```
root@techhacker: ~/Desktop
root@techhacker:~/Desktop# who --help
Usage: who [OPTION]... [ FILE | ARG1 ARG2 ]
Print information about users who are currently logged in.

  -a, --all         same as -b -d --login -p -r -t -T -u
  -b, --boot        time of last system boot
  -d, --dead        print dead processes
  -H, --heading     print line of column headings
      --ips         print ips instead of hostnames. with --lookup,
                    canonicalizes based on stored IP, if available,
                    rather than stored hostname
  -l, --login       print system login processes
      --lookup      attempt to canonicalize hostnames via DNS
  -m                only hostname and user associated with stdin
  -p, --process     print active processes spawned by init
  -q, --count       all login names and number of users logged on
  -r, --runlevel    print current runlevel
  -s, --short       print only name, line, and time (default)
  -t, --time        print last system clock change
  -T, -w, --mesg    add user's message status as +, - or ?
```

- id ⟶ gives userid, group id & groups.
- nano <file/directory name> ⟶ Text editor (vim can also be used)

  cntrl + o ⟶ Write

- cp ⟶ copies the file.

  eg. cp test.txt /root/desktop/copied.txt

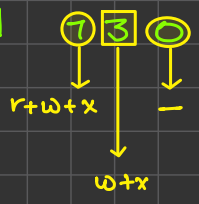
  cp → copy; test.txt → file to be copied; /root/desktop/copied.txt → copied file.

- mv ⟶ move file.

  eg. mv test1.txt renamed.txt

- If you wish to change the permission, then,

sudo chmod 7 3 0 user1.txt



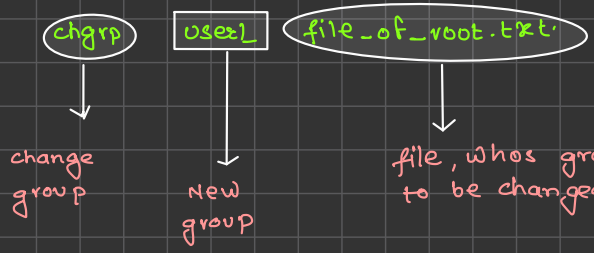
- chown → This command is used to change owner of the file.

chown user1 file_of_root.txt

```
root@techhacker:/home/user1# ls -l
total 0
-rwxr--r-- 1 root root 0 Dec 28 22:06 file_of_root.txt
-rwx-wx--- 1 root root 0 Dec 28 21:55 user1.txt
root@techhacker:/home/user1# chown user1 file_of_root.txt
root@techhacker:/home/user1# ls -l
total 0
-rwxr--r-- 1 user1 root 0 Dec 28 22:06 file_of_root.txt
-rwx-wx--- 1 root  root 0 Dec 28 21:55 user1.txt
root@techhacker:/home/user1#
```

- chgrp → This command is used to change group of a file.

```
root@techhacker:/home/user1# ls -l
total 0
-rwxr--r-- 1 user1 root 0 Dec 28 22:06 file_of_root.txt
-rwx-wx--- 1 root  root 0 Dec 28 21:55 user1.txt
root@techhacker:/home/user1# chgrp user1 file_of_root.txt
root@techhacker:/home/user1# ls -l
total 0
-rwxr--r-- 1 user1 user1 0 Dec 28 22:06 file_of_root.txt
-rwx-wx--- 1 root  root  0 Dec 28 21:55 user1.txt
root@techhacker:/home/user1#
```



COMMAND LINE

1. Working with files and file contents :-

1 Touch :-

- Touch command is used for creating files.
  - creating empty files.
  - create multiple empty files.
    - $touch file1 file2 file3 ⟶ eg. command ⟶ touch file {1..33}
    - Create files named A-Z.
    - specify multiple files extension.

- Change Modification Time of file.
  - $touch -m file1

```
root@kali:~/Desktop# ls -l
total 8
-rw-r--r-- 1 root root    0 agd 16 16:57 file1
-rw-r--r-- 1 root root    0 agd 16 16:57 file2
-rw-r--r-- 1 root root    0 agd 16 16:57 file3
-rw-r--r-- 1 root root    0 agd 16 16:57 file4
-rw-r--r-- 1 root root    0 agd 16 16:57 file5
-rw-r--r-- 1 root root    0 agd 16 16:55 file.txt
-rw-r--r-- 1 root root   33 agd 16 00:44 hash.txt
-rw-r--r-- 1 root root    0 agd 16 16:54 master
-rw-r--r-- 1 root root    0 agd 16 16:57 song10.mp3
-rw-r--r-- 1 root root    0 agd 16 16:57 song6.mp3
-rw-r--r-- 1 root root    0 agd 16 16:57 song7.mp3
-rw-r--r-- 1 root root    0 agd 16 16:57 song8.mp3
-rw-r--r-- 1 root root    0 agd 16 16:57 song9.mp3
-rw-r--r-- 1 root root    0 agd 16 16:55 song.mp3
drwxr-xr-x 9 root root 4096 nah 20  2019 vmware-tools-distrib
```

Enter the command

```
touch -am
```

```
-rw-r--r-- 1 root root    0 agd 16 16:57 file1
-rw-r--r-- 1 root root    0 agd 16 16:57 file2
-rw-r--r-- 1 root root    0 agd 16 16:57 file3
-rw-r--r-- 1 root root    0 agd 16 16:57 file4
-rw-r--r-- 1 root root    0 agd 16 16:57 file5
-rw-r--r-- 1 root root    0 agd 16 16:55 file.txt
-rw-r--r-- 1 root root   33 agd 16 00:44 hash.txt
-rw-r--r-- 1 root root    0 agd 16 17:00 master
-rw-r--r-- 1 root root    0 agd 16 16:57 song10.mp3
-rw-r--r-- 1 root root    0 agd 16 16:57 song6.mp3
-rw-r--r-- 1 root root    0 agd 16 16:57 song7.mp3
-rw-r--r-- 1 root root    0 agd 16 16:57 song8.mp3
-rw-r--r-- 1 root root    0 agd 16 16:57 song9.mp3
-rw-r--r-- 1 root root    0 agd 16 16:55 song.mp3
drwxr-xr-x 9 root root 4096 nah 20  2019 vmware-tools-distrib
```

- Display type of file
  - File _____

If you are removing / deleting a file from a GUI, then the file might go in recycle bin & can be restored.

But, if you are removing/deleting a file from command line, then the file cannot be restored.

2 RM :-

- RM is a command used to remove file & directories.
- The command line in general does not have a waste bin or trash can to recover files.
- To prevent yourself from accidently removing files, you can type `rm -i`, this reconfirms

  - rm -rf
    - The rm -rf statement is famous because it will erase anything.

3 Head (-head)

- This command is used to access / control the lines from Top to Bottom.

4 Tail (-tail)

- This command is used to access / control the lines from Bottom to Top.

5 Cat

- Cat command is one of the most universal tools, yet all it does is copy standard input to standard output.
- To see what is present in the file, we can use cat.

```
root@techhacker:~/Desktop/sample# cat >> test.txt
abcd new appended
^C
root@techhacker:~/Desktop/sample# cat test.txt
jfjewiohreowhoewhiohdsfdsaf
asf
sad
f
aewterteritjpidfds
fds
fds
fds
fsdf
abcd new appended
root@techhacker:~/Desktop/sample#
```

- Create files

  you can use it to create flat text files.

  - Type `cat > Winter.txt`

```
rahul@debian8:~$ cat > winter.txt
```

We can create a file at write in it at the same time using cat.

eg. cat > Winter.txt
abcdefghij
control +c.

2. Working with directories:-

1. pwd : print working directory
   It shows the name of the directory you are working on.
2. cd : change directory
   you can change your current- directory with cd.
3. cd ~ : cd is also a shortcut to get into your home directory.
4. cd .. : To go one directory back.

Special Keys Strokes

Ctrl+A	Moves the cursor to the beginning of the line.
Ctrl+B	Moves the cursor backward one character.
Ctrl+C	Cancels the currently running command.
Ctrl+D	Logs out of the current session.
Ctrl+E	Moves the cursor to the end of the line.
Ctrl+F	Moves the cursor forward one character.
Ctrl+H	Erase one character. Similar to pressing backspace.
Ctrl+P	Paste previous line(s).
Ctrl+R	Allows you to search for a previously used command
Ctrl+S	Stops all output on-screen (XOFF).
Ctrl+Q	loses an application window.
Ctrl+U	Erases the complete line.
Ctrl+W	Deletes the last word typed.

CONTROL OPERATIORS

With the help of control operatiors, we can put more than one command in command line.

① ; → semicolon

```
root@kali:~# echo this is masters in it
this is masters in it
root@kali:~# echo this is masters in it cyber security
this is masters in it cyber security
root@kali:~# echo this is masters in it cyber security ; echo rahul
this is masters in it cyber security
rahul
root@kali:~#
```

BSCRIBE PLEASE LIKE | SHARE | CC

② & → ampersand

- When a line ends with an ampersand &, the shell will not wait for the the command to finish.
- you will get your shell prompt back, & the command, is executed in background.
- You will get message when this command has finished executing in background.

③ && ⟶ double ampersand

The shell will interpret && as a logical AND. When using && the second command is executed only if the first one succeeds. (returns a zero exit status)

④ # ----> Pound Sign

- Everything written by after a pound sign (#), is ignored by the shell.
- This is useful to write a shell comment but has no influence on the command execution or shell expansion.

⑤ history

- If we want to see how much & which command we have entered, then history command is used.
- It basically displays all the previous commands given.

I/O REDIRECTION :-

This feature of command line enables you to redirect the input and/or output of commands from and/or to files.

① >> ⟶ append

- If we don't want to overwrite a file, then we use >>.

for eg.

```
echo   file4     folder  newnewfile.txt  rahul22.txt   sample      song6.mp3  song8.mp3  song.mp3
file3  file.txt  master  rahul           rahulnew.txt  song10.mp3  song7.mp3  song9.mp3  vmware-tools-distrib
root@kali:~/Desktop# cat echo
this is masters in it cyber security
root@kali:~/Desktop# echo this is it > /root/Desktop/echo
root@kali:~/Desktop# cat echo
this is it
```

In the above example,

"this is masters in it cyber security" is replaced by "this is it".
This has occurred because, ">" this command just overwrites the file.

But, if we want that the line should be added without getting overwritten, then we have to use ">>".

Filter & Pipes

When a program takes its input from another program, it performs some operations on that input, & writes the result to the standard output.
It is referred as filter.

① grep :- The grep command searches file or files for the lines that have a certain pattern.

- The name "grep" comes from the ed (a unix code editor) command g/re/p which means "globally search for a regular expression & print all lines containing it."

```
root@kali:~/Desktop# echo this is i gyugdyugeygcyuegugecugt >> /root/Desktop/echo
root@kali:~/Desktop# cat echo
this is it
this is i gyugdyugeygcyuegugecugt
root@kali:~/Desktop# touch filenewone.txt
root@kali:~/Desktop# cat tounewone.txt
cat: tounewone.txt: No such file or directory
root@kali:~/Desktop# cat > tounewone.txt
this is test file
file2
file3
file4
^C
root@kali:~/Desktop# cat tounewone.txt | grep this
this is test file
```

② pipe command: If we want that more than two commands will be linked up into a pipe.

Basic Linux Tools

① find: Find all files in /etc & put the list in etcfiles.txt

command: find /etc > etcfiles.txt.

② sleep: This command is used to suspend execution for atleast the integral no. of seconds specified by the time operand.

③ time

④ date

⑤ cal

Introduction to VI :-

- There are many ways to edit file in linux.
- This editor enables you to edit lines in context with other lines in the file.
- Imported version of VI editor is called VIM.

- vi filename :- creates a new file if it does not exist, otherwise open an existing one.
- vi -R filename :- opens an existing file in read only mode.
- view filename :- opens an existing file in read only mode.

eg. vi master opens the file "master"

i enables you to, Insert/edit.

"insert/edit the things you want".

→ :wq exit.

Process Management

Whenever you issue a command in UNIX, it creates or starts a new process.

- Starting a Process :-

When you start a process, there are two ways you can run it:

① Foreground Process : Process which we can see running.

② Background Process. : Processes which runs in background.

- List Running Processes:

You can see your own processes by running the ps (process status) command.

PID → Process ID

PPID → Parent Process ID

```
root@kali:~/Desktop# vim master
root@kali:~/Desktop# ps
    PID TTY          TIME CMD
   3124 pts/0    00:00:00 bash
   3422 pts/0    00:00:00 ps
root@kali:~/Desktop# kill
```

- Ending a running process:

If a process is running in background, you should get its process ID using "ps" command.

After, you can kill the process as follows:

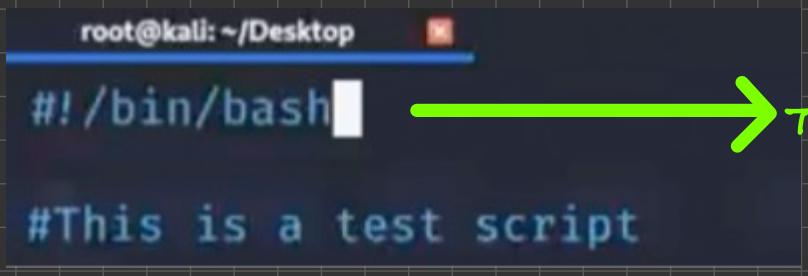
```
$ps -f
UID    PID  PPID C STIME   TTY   TIME CMD
amrood   6738 3662 0 10:23:03 pts/6 0:00 first_one
amrood   6739 3662 0 10:22:54 pts/6 0:00 second_one
amrood   3662 3657 0 08:10:53 pts/6 0:00 -ksh
amrood   6892 3662 4 10:51:50 pts/6 0:00 ps -f
$kill 6738
Terminated
```

Here, the kill command terminates the first_one process. If a process ignores a regular kill command, you can use kill -9 followed by the process ID as follows −

```
$kill -9 6738
Terminated
```

Scripting Introduction

- A shell script is a computer program designed to be run by the UNIX / Linux shell.
- It is a file that contains ASCII text. (Data files, Data set, etc.)
- The extension of script is .sh



This is also known as "shebang header".

Introduction to user and group management

There are 3 types of account on Linux system.

① Root Account: -This is also called a super user & would have complete & unfettered control of the system.

- A superuser can run any command without any restriction.
- This user should be assumed as a System Administrator.

② Service Account: service accounts are created by installing packages, when they are installed.

- These accounts are used by services to run processes & executes functions.
- These accounts are not used in routine work.

③ User Accounts: user accounts provide interactive access to the system for users & group of users.

Linux supports concept of Group account which logically groups a no. of accounts.

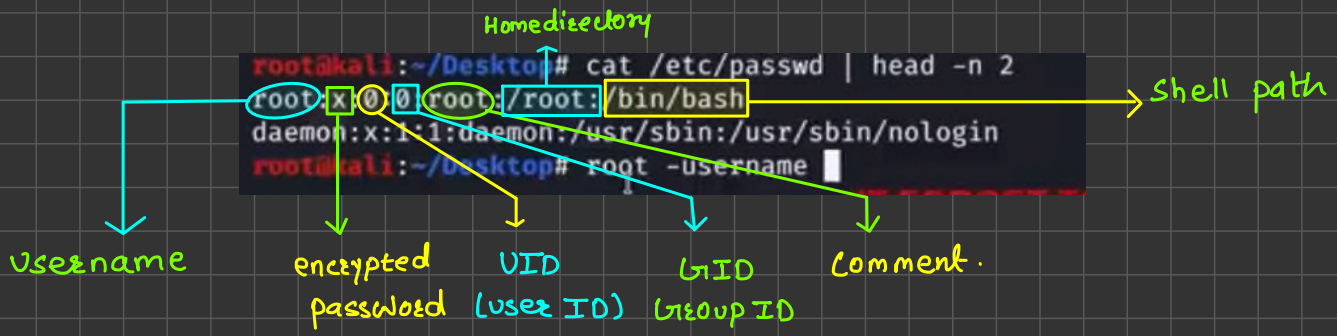
Every account would be a part of another group account.

A linux group plays important role in handling file permissions & process management.

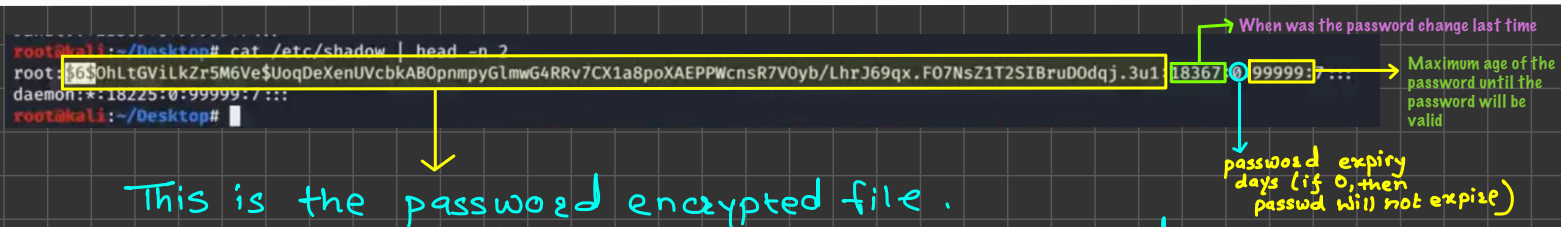
Managing users and groups

There are 4 main user administration files:

1. `/etc/passwd` — Keeps the user account & password information. This file holds the majority of information about accounts on UNIX system.
2. `/etc/shadow` — Holds the encrypted password of the corresponding account. Not all the systems support this file.
3. `/etc/group` — This file contains the group information for each account.
4. `/etc/gshadow` — This file contains secure group account information.



1. adduser : Adds account to the system
2. usermod : Modifies account attributes.
3. userdel : Deletes account from the system.
4. groupadd : Adds group to the system.
5. groupmod : Modifies group attributes.
6. groupdel : Removes group from the system.
7. useradd : This command gen. creates the user, but will not create the directory.



This is the password encrypted file.
To know, by which algorithm the file is encrypted, check first 3 alphabets. if :

1 : md5 algorithm i.e. Message digest

6 : sha 512 algorithm

5 : sha256 algorithm

$2a : Blowfish

- Add User :-

```
root@kali:~/Desktop# adduser linux
Adding user `linux' ...
Adding new group `linux' (1001) ...
Adding new user `linux' (1001) with group `linux' ...
Creating home directory `/home/linux' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for linux
Enter the new value, or press ENTER for the default
        Full Name []: Kali
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] Y
root@kali:~/Desktop#
```

- If you add a user, using useradd, then you have to do as follows:

```
root@kali:/home# useradd -c "testing" -m -d /home/kamal -s /bin/bash -p 12345 kamal
```

-c "testing" → comment
-m -d /home/kamal → Make directory
-s /bin/bash → change shell
-p 12345 → set password
kamal → username.

- To dete the user added, we have to use userdel.

```
root@kali:/home# ls -l
total 12
drwxr-xr-x 3  1003  1003 4096 agd 17 15:14 kamal
drwxr-xr-x 3 linux linux 4096 agd 17 15:08 linux
drwxr-xr-x 3 rahul linux 4096 agd 17 12:31 rahul
root@kali:/home# userdel -rf kamal
userdel: user 'kamal' does not exist
root@kali:/home# ls
kamal  linux  rahul
root@kali:/home# rm -rf kamal
root@kali:/home#
root@kali:/home# ls -l
total 8
drwxr-xr-x 3 linux linux 4096 agd 17 15:08 linux
drwxr-xr-x 3 rahul linux 4096 agd 17 12:31 rahul
root@kali:/home#
```

- To add a user in the group :-

```
root@kali:~/Desktop# useradd -m click
root@kali:~/Desktop# groupadd hotstar
root@kali:~/Desktop# groups click
click : click
root@kali:~/Desktop# usermod -g hotstar click
root@kali:~/Desktop# groups click
click : hotstar
```

OR

useradd click hotstar

useradd → command
click → user
hotstar → group.

- To add a user in Multiple groups :

```
root@kali:~/Desktop# usermod -G start,groupnew,rahul click
root@kali:~/Desktop# groups click
click : hotstar rahul groupnew start
root@kali:~/Desktop#
root@kali:~/Desktop#
```

command

To add in Groups

Group Names

Username

- Create a password for a file :-

```
root@kali:~/Desktop# gpasswd start
Changing the password for group start
New Password:
```

- We can check by the command "cat /etc/gshadow", whether the password is set or not.

- Delete user from group :-

```
root@kali:~# deluser user1 techhacker
Removing user `user1' from group `techhacker' ...
Done.
```

- Delete the whole group :-

To delete the group, we have to use command:

groupdel <group_name>.

Command	Description
sudo adduser username	Adds a user
sudo passwd -l 'username'	Disable a user
sudo userdel -r 'username'	Delete a user
sudo usermod -a -G GROUPNAME USERNAME	Add user a to a usergroup
sudo deluser USER GROUPNAME	Remove user from a user group
finger	Gives information on all logged in user
finger username	Gives information of a particular user

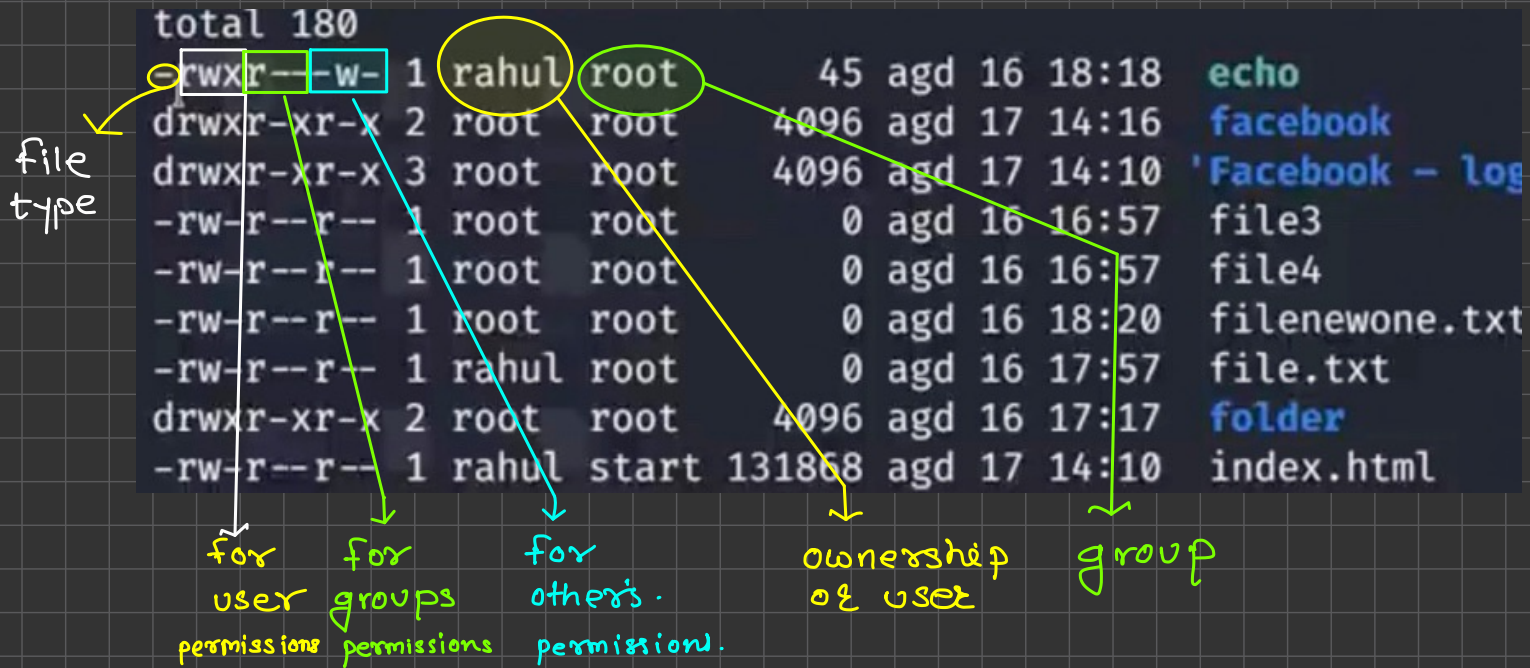
File Permissions

r : read — r - 4

w : write — w - 2

x : execute — x - 7

7 = read + write + execute
6 = read + write
4 = read.



- Symbolic Mode :

\+ ⟶ for adding permission.
\- ⟶ for removing permission.
= ⟶ sets & overrides permissions.

u ⟶ user/owner
g ⟶ group
o ⟶ other
a ⟶ all.

- To change the user/ownership of the file :

```
-rw-r--r-- 1 root  root      66 agd 16 17:15  rahulnew.txt
```

Now, if we want to change the user/ownership to rahul, then

⟶

```
root@kali:~/Desktop# chown rahul rahulnew.txt
root@kali:~/Desktop# ls -l
total 180
```

- To change the group:

```
-rw-r--r-- 1 root  root      46 agd 16 17:14  rahul22.txt
-rw-r--r-- 1 rahul root      66 agd 16 17:15  rahulnew.txt
```

→

```
root@kali:~/Desktop# chown :groupnew rahul22.txt
```

- To change the permissions:

```
-rw-r--r-- 1 root  root          61 agd 16 18:00  newnewfile.txt
```

To change these permissions

→

```
root@kali:~/Desktop# chmod a=rwx newnewfile.txt
```

```
-rwxrwxrwx 1 root  root          61 agd 16 18:00  newnewfile.txt
```

If you wish to change permissions only for user or group or other then,

```
root@kali:~/Desktop# chmod u+rwx,go+rw rahulnew.txt
```

group & others.

- To read a binary file:

We have to use strings command in order to do this.

```
root@kali:~/Desktop# strings /bin/pwd
```

Some Linux tools

- Quick stego → for stagano graphy.
- john the reaper → for password cracking / guessing.
- httrack.com → to duplicate a website.