Linux.

Linux is a Unix-like OS developed by Linus Torvalds and thousands of Open-source contributors.

Linux is an operating system. It is reliable and secure than others; also it is completely opensource. (Launched 17/Sept/1991)

-> Linux is used in our smart phones. 85% of all smartphones are based on linux.
-> In cars, specially self driving cars.
-> Refrigerators need linux to run.

Because the linux is faster, secure and it is reliable than other OS.

* Before 2000 Unix is dominating the super computer market & even the BSD (Berkeley Software distribution), Mac (Macintosh operating system), windows OS running on super computers but right now in 2020 top 500 super computers existing in the world are running on Linux OS. Because it is lightweight, faster, reliable

-> All developers prefer the linux OS.

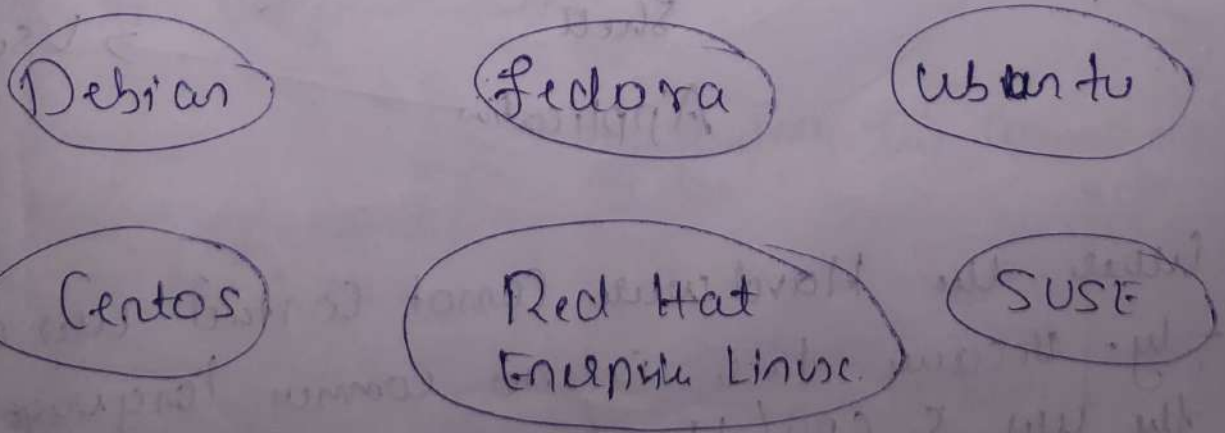
Comparision b/w Linux & windows

→ For a windows server, you need to purchase the licence & the expend-iture increases according to the no of users.	→ with linux, licence is free, installing software is free & you can install in any no of machines you want.
→ NOT many customizat-ion options available	→ There are a lot of linux distributions & you can choose one from it.
→ windows is vulnerable to virus and malware threats. A powerful anti virus is needed.	→ More secure than windows and virus can't easily break the kernel.

"Vulnerable - Exposed to the possibility of being attacked.

Kernel - is the essential center of a computer operating system.

Linux Distributions

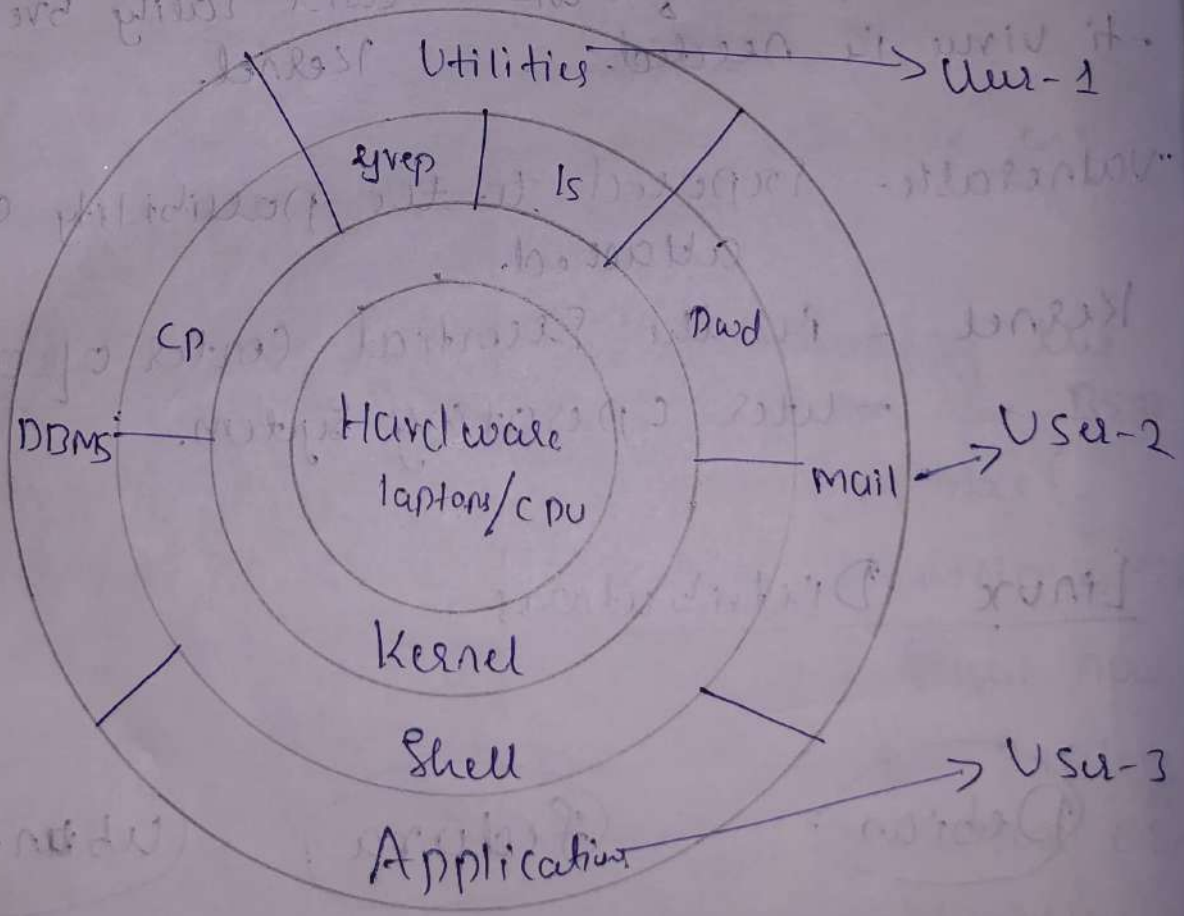


CentOS is the community version of the Red Hat Enterprise linux and CentOS as all the features of any other linux distribution.

But the ~~command~~ Basic linux commands are same in every single linux distribution.

Basics of Shell

Linux Architecture



When the Hardware cannot contact user directly. Because there is no common language b/w the user & CPU/Hardware. To Enable that com

Top Shells

Bash - The Bounce again Shell is the default shell in a lot of linux distributions & it is the most portable shell available. Bash is the default shell of centos, Ubuntu

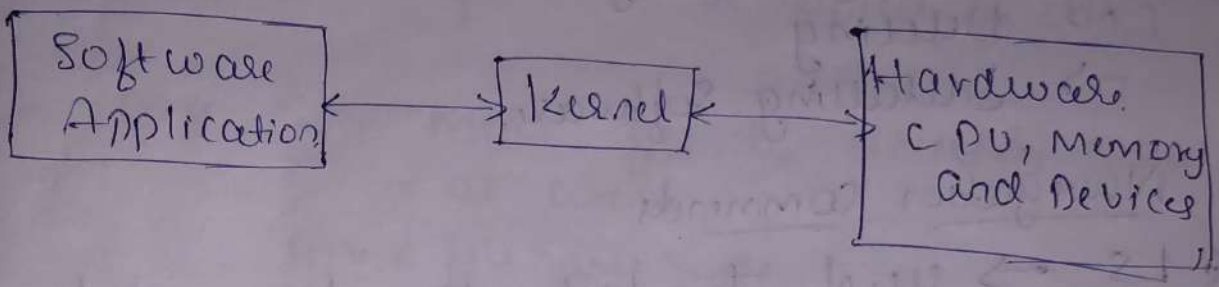
Zsh (Z Shell) - It is similar to bash or an extended version of it. It has a lot of useful features like sharing your command history across multiple terminals.

Fish - Friendly & interactive shell is again an extended version of the common sell that is Bash. It has a great feature of ability to autocompletion of commands.

tcsh - Tenex C Shell is an extended version of C Shell. The plus of tcsh is its scripting language, because it will be similar for user with experience with C program

Basic of Kernel

A linux Kernel is a UNIX-like OS Kernel. It is a computer program which is the core interface which connects the Hardware components to the Software to Processes.



Top operations performed by a Kernel

→ **Resource Management** - Decides which Process gets a resource for an operation.

→ **Memory Management** - Kernel has complete access to System memory and must efficiently manage it and allow memory access to Processes.

→ **Device management** - if we connect devices such as a Printer or a pen drive, Kernel detects it and helps the System establish connection with those peripherals.

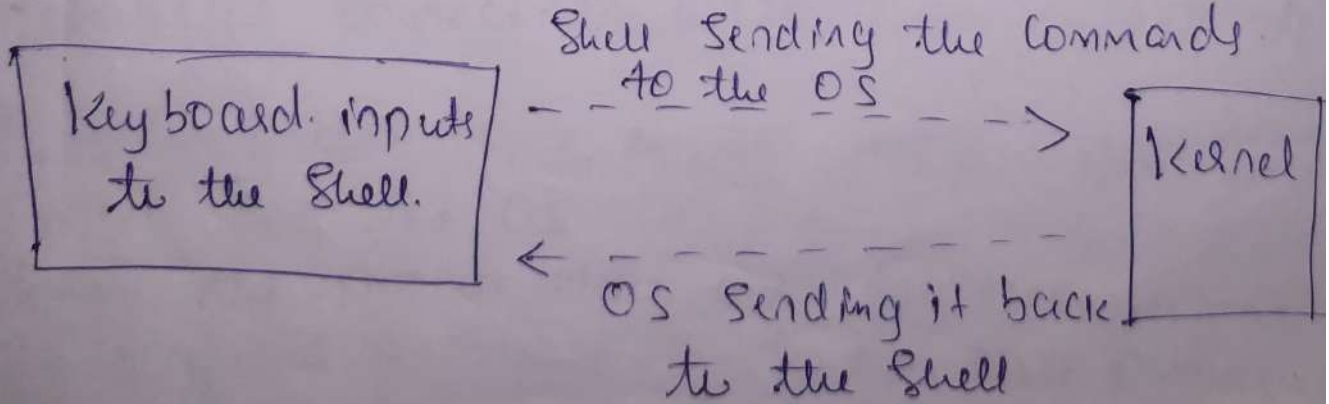
→ **System Calls** : This is an interface b/w a Process and the OS.
When the process does not have permissions to access a resource, a system call provides it without the process accessing the resource

the Kernel which is the middle man which Contacts the Hardware for the user. So the Commands with in the linux and these Commands are executed by the Entity Called Shell.

(Eg) If we are using the application as a End user if u open the application the first instruction is given to the Shell understands the command then Shell instructs the Kernal when it interprites the command & it instructs the Kernal to do the particular operation so Kernal will understand ~~the~~ what exactly the operation & it will pass that to the Hardware in the language the Hardware can understand ((Eg) Assembly level language/Micro Controller language) then Hardware will do the operation for user & send it to the Shell where we can view that.

SHELL

A Shell interprets the commands you have Entered using a keyboard and sends it to the OS to perform them.



Linux Commands - Karthik B.M

1) **Pwd** - path of our current working Directory.

Note- A path of all the directories that start with a forward slash (/).

Listing files and Directories

2) **ls** - ls command lists the files and Directories within a system. Running it without a flag will show the current working directory's content.

To see other directories content, **ls** followed by the desired path.

(a) **ls /** (Forward slash) - To list out all the contents of the root directory

(b) **ls ..** - Prints the one step Back Directory Structure like **ls ../..** for 2 step Back word.

(c) **ls -l** - list all the files and Directories in long format.

It will give the details like rights for a file, user, size, Date and Time

(d) **ls -a** - Shows hidden files in addition to the visible ones.

(e) **ls -al** - Hidden files & long list

Note - > - To save & output content in the file.

(f) ls -ls - Sort the directory by their size

(g) ls <Dir>*.ext - To list the .ext file from the Particular Directory.

(h) ls <Dir>*.* - To list all the ext and names.

(i) ~~ls -ld */~~ ls -d */ = To list only directories.

(j) ls <File Name>* - To list all the files starting with the letter particular <FN>

(J) ls p* - To list all the files start with the Alphabet 'P'

(k) ls -ld <DN> - To see the permissions of a particular Directory

(l) ls -l <FN> - To see the permission of a particular File

(m) ls -R - Directory structure & list all the files in sub Directory

(n) ls -lt - Sort To list out the files and Directories with time line format @ newest First & Recently created file & D at the first & oldest at the Bottom.

Note ls -lrt - B To list Recently created @ the Bottom.

(o) ls -lh - Shows the files sizes in easily readable format, such as MB, GB & TB.

man <command name> — provide guide for the particular command. Its information abot what it does & use of flags with command

whatis ls - used to display the short discription about le command which is their in man itself.

which - used to return the path name of a file & command which we use.

which <command> & Application. & ls - give the location of the ls command & wherever the location of the command is executable.

watch <cmd NAME> -

Note. ~ $ -> Home Dir (3)
/ $ -> Root Dir

Navigation Commands

cd - Change Directory & Go to Home Directory

(3) (a) cd / - Change the directory to the Root Directory.

(b) cd <DN> - To Change to the particular
or
cd <Path> Directory

(c) cd ~ - To Change to Home Directory

cd - = move to your previous ==,,==

(d) cd .. - To come out 1 step out of Directory

(e) cd ../.. - To come out 2 step out of Dir

(f) ~~cd~~ To Navigate to the directory with the space b/w the Name (Eg) My World

There are 3 options to Navigate

i) cd My\ Books

ii) cd " My Books"

iii) cd ' My Books'

Creating Folder

4) mkdir <Directory_Name> - Used to create a Directory.

a) mkdir d_1/d_2 - To create a sub directory inside the d_2 inside the directory d_1 which is already exist.

Note - mkdir -P $d_1/d_2/d_3$ - To create a directory & subdirectory which doesnt really exist

(b) mkdir d_1 d_2 d_3 - To create a multiple directories d_1 d_2 d_3

5) TOUCH < File Name > - Used to create an empty file.

(a) TOUCH < Directory_Name / < FN > - To create an empty file inside a directory

(b) TOUCH F1 F2 F3 - To create a Multiple file Empty file.

(c) TOUCH < Previously Created FN > - Used to modify a timestamp in the linux command line.

If the file doesn't exist then we run TOUCH command it will create a file & if already the file & run the touch command it will modify the timestamp of that file.

(6) Cat Command

Concatenate & cat, is one of the most frequently used linux command. It lists, combines and writes file content to the standard output

Cat <File Name> - Display the content of the file

(a) Cat > FN.txt - Create a new file.

(b) Cat <FN1.txt> <FN2.txt> > <FN3.txt> = used merge FN1.txt & FN2.txt and store in FN3.txt

(c) tac <FN.txt> - Display content in reverse order.

(d) Cat <FN1> <FN2> - It will combine the content of both the files and display on terminal.

(e) Cat -b <FN> - It will not add the number to non Blank line.

(f) Cat -n <FN> - It will add the number to non Blank line also.

(g) Cat -s <FN> - To remove the large line Brake to 1 line Brake.

(h) Cat -E <FN> - It add's the '$' symbol at the End of ~~line~~ Each line to know this is End of the line

Vi - Editer

Vi - editer opens in command mode and hear we can move the cursor and cut, copy, paste the text. This mode also saves the changes that made to the file.

Commands are case sensitive.

Vi Editing Commands.

Vi <File Name> -

a) i - Switch to the insert mode from the command mode.

b) Esc - To return to the command mode from the insert mode. & Terminate insert mode.

It will work only in vi editor

1) :set nu → To set numbers to the lines of a particular file.

2) :set → To remove the Number

3) :/<Pattern> → To search a particular pattern inside our vi editor

Note: Press 'n' - To move to the next accurance

4) :%s/<old Pattern>/<new pattern>/g — Used to Substitute the old pattern with the new pattern.
% → Every ware
%s → Substitute g → Globally

Casesenitive – Text or typed input is senitive to capitalization of letters. like Upper & lower case.
For case senitive – C s c are two different Letters.

Note: To make Incase Senitive

Eg for the above.

:%s/<OP>/<NP>/gi –

5) :1s/<OP>/<NP>/g – To replace & Substitute the pattern in the first line only.

6) :1,5s/<OP>/<NP>/g – To replace pattern from 1st to 5th line.

Note: $ → Till End of End of line.

7) :5,$s/<OP>/<NP>/g — To Substitute the pattern from the 5th line to the End of the line.

8) :1s/<OP>/<NP>/1 — To Replace the pattern from the first line & 1st occurance.

wc - word count

9) wc <FN> - Used to count the lines, words 1st column 2nd 3rd and ~~the characters (include letters~~ Bits/memory it consumed for an input file ~~number, special characters~~)

Line words ~~Characters~~ memory consumed

10) wc -l <FN> - Used to count only lines.

wc -w <FN> - Used to count only words

wc -m <FN> - only characters.

wc -c <FN> - To know how many bits it consumed. & memory consumed

wc -L <FN> - Length of longest line.

Saving and closing the File

a) `:w` - Save the file but keep it open

b) `:q!` - Quit vi and do not save changes

c) `:wq` - Save the file and Quit.

d) `:wq!` - Save the file & Quit Forcefully

w -> Save q -> Quit ! -> Forcefully

Displaying using Command - echo

- The command echo is used to display a line of text / string by passing it as an argument. or To print some statement with in my terminal.
& To print = ,, = . very quickly we use Echo

8) Echo " Content" — Print the Content in the terminal Quickly.

(a) Echo -n "Content" - Give the output without a newline

Note:- Echo -e - Enables the interpretation of the following Backslash Escapes.

Echo -E - displays the default option and @ vice versa of the above Case.

b) Echo -e " content \b Content" - Remove the space b/w the text.

c) Echo -e " Text \n text" - Prints a text in a new line (Separate & prints in next line)

(Eg) Text
text

d) echo -e " Text \t ext" - Does a Horizontal Tab

(Eg) Text tab ext

(f) `echo -e " One \v two \v three"` - Does a vertical Tab

(Ex) One
two
three

(g) Use of Expressions

`x = 7`

i) `echo $x` - Prints the value of x

`y = 6`

ii) `echo $((x+y))` - It adds the Expression and Prints the value 13

ii) `echo $x + $y` - Prints the content of x & y Seperately.

Filter Commands

Filter commands are used to filter the output so that the required things can easily be picked up. The commands which are used to filter the out put are

Less, more, tail, head, Sort, Cut, Sed.

9) More Command :

a) more < File Name > - Used to display the content of the file in a page wise, where press Enter to view more information.

Note: (It only Scroll's down)

b) Press Enter to scroll down line by line.

c) Q is used to come out of it.

d) d - Go to next page.

e) b - Go to previous page.

f) / - To search for a file word in a file.

g) v - To go to vi mode where we can Edit the file and once you save it back to more comm

10) Less command

Less <FN> - Less command is opposite of more command but it offers extra features and it's a lit-
-le faster with large file.

-> It is used to display the content of the file in page wise. Here we can scroll up & down the line using up & down arrow key where as in mo-
-re command we can only scroll down.

a) Shift + G - Browse to the End of the text file.

Shift + g & 1g - Go to the Begining of the ===

11) Head Command

head <FN> - used to display the first portion of the file and by default it will display first 10 lines.

(a) head -n <FN> - Use the num flag to specify the no of lines to be displayed.

(Eg) head -5 <FN> - Display first 5 lines.

12) Tail Command

tail <FN> - used to display the last portion of the file.

(Eg) Tail -6 <FN> - Display last 6 lines.

13) Grep Command

Global regular Expression print.

Grep "pattern" <FN> - It let us to find a word / Pattern by searching through all the texts in a specific file.

Grep is used to search a pattern in a file and if the pattern is found it will print the whole line.

Note: Grep command will work only outside the Vi editor

(12)
(11)

(a) grep -i "pattern" <FN> - To avoid case sensitivity of the pattern in side the file. (Eg) Sufix, Prefix, Upper, lower (Es) devops, devopsis

(b) grep -in "pattern" <FN> - It will print the line Number where the pattern is present.

(c) grep -ic "pattern" <FN> - It will count the how many patterns are present

(d) grep -iw "pattern" <FN> - It will search only the particular word and it will not consider any sufix and prefix.

(f) grep -il "pattern" * <FN> - It will print only the file name which contains the pattern in side the directory. Present directory

(g) grep -il -R "Pattern" * <FN> & -iLR - It will print only the file name which contains the pattern inside the present directory & the subdirectory.

(h) grep -v "pattern" <FN> - It will print /display the line expect the pattern line.

(i) grep -e "PN1" -e "PN2" <FN> - It will search the multiple pattern in the particular file.

(j) grep "pattern" <FN1> <FN2> - It will search the pattern in multiple files.

(k) grep --colour "PN" <FN> - It will display the searched word in colour.

(l) grep "^x" pattern" <FN> -> Search the line sta-
-rts with particular word.

(m) grep " pattern x$" <FN> -> Ends with particu-
-lar pattern

(n) grep -nA2 " pattern" <FN> -> To display the word line & 2 line after the word line.

(o) grep -nB2 " pattern" <FN> -> 2 line before the wo-
-rd line.

14) Uniq Command

Uniq is used to print only unique data & to to remove the duplicate entries.

Drawback is it will ~~check~~ Print only consecutive entries of duplication (sequential entries.)

(Eg) Set } Sequential Unique → Set
Set } Prd
Prd Set
Set

Uniq <FN> - Print only unique data from the spread sheet & column content.

Note: Uniq command always used along with the sort command if not will not get desired o/p.

Uniq & sort command only applied for spread sheet, Ex sheet, column wise content & only single pattern in a line.

Duplicate removed from the sheet will persist in the original file.

(14)
(13)

(15) Sort Command

Sort <FN> - It used to sort / list the data / content in the file. in Alphabetic Order of numerical Order. or In Ascending Order.

Sort is used to sort data of file in ascending orde

(a) Sort -r <FN> - Sort the data in the file in Decreending Order.

(b) Sort -U <FN> - TO remove the duplicate Entries from the output.

(c) Sort -n <FN> or -h <FN> - Sort the file according to number.

Sort -nr - reverse

Note - Sort should be used before the uniq comm- - and so to get the required output.

Sort <FN> | uniq - So the sort will list the content in the file in ascending & Alphabetical Orderde then the Ou using the pipe the out of this where the uniq will remove the Consecutive Entries and get the O/p.

(6) File Permissions

Ownership of linux file.

Permissions are applied on three levels

* Owner & user level
* Group level
* Others / process

The purpose of user and groups are for Access the
5 Every user has their own directory, usually within
/home/username

Even process need permissions to access specific
file and location

Access modes are of 3 types

* r - read only
* w - write / edit / delete / append
* x - Execute / run a command the .ext file

Access modes are different on files & Directories

Permissions	Files	Directories
r	Open the file	Is the content of dir
w	write, edit, append delete file	Add / Delete / Rename contents of dir
x	To run a command / shell script	To enter into dir using "cd"

Permissions can be set on any file / dir by two methods

1) Absolute method (number) $\overset{w}{6}\overset{r}{2}\overset{x}{1}$
2) Symbolic method (ugo) a ugo = a

1) Absolute method (number) - In this method we use numbers to give the permissions

a) Read - $2^2 = 4$
b) Write - $2^1 = 2$
c) Execute - $2^0 = 1$

rwx	rw-	r-x
4 2 1	4 2	4 1
7	6	5

2) By default the file permission will be for a file will be 644 (rw - r - r) also for a newley created file dir and 755 for dir

17) Chmod ~~Chown~~ command (To change the permision of file and directories)

Chmod stands for change mode ~~ownership~~ permision

Chmod Number <FN> - used to change & modify the user, groups and process ~~ownership~~ Permisions of file.

(a) Chmod 777 <FN> - Assigning full permision to the file for user, groups and process.

(b) Chmod 777 <dir N> - " = "

(c) Chmod 765 <F_1><F_2><F_3> - To change the Permisions of multiple files.

(d) Chmod 765 <d_1><d_2><d_3> - multiple directory

(e) Chmod -R 777 <DN> - To change the Permisions of present directory & subdirectories.

2) Symbolic Method (ugo) or a=ugo

General mode of form of Symbolic mode is

Chmod -[who][+/-/=][Permision] <FN>

who -> TO whom the permisions to be assigned user/owner (u), group (g), other (o)

(a) Assigning full permisions to the file

Chmod ugo=rwx <FN> &

Chmod a=rwx <FN>

(b) like wise we can add & remove permissi-ons from any file for anyone (user, group & ote)

(b) chmod u+x <FN> - Adding execute permissions to user only.

(c) chmod go-wx <FN> - Removing write & execute permissions from group & other.

(d) chmod go=r <FN> - Giving only read Permissions to group & other.

(e) chmod u+rw, g-w, o+w <FN> - Adding r & w Permission to user, removing write permission to the group & Add w Pr to the other.

18) Umask command

Umask used to set the default permission on the system used by only root user & admin. System admin.
Umask run opposite to the chmod.
Umask will decide what should be the default Permissions for a file and directory when it is created.

a) Umask 641 -> So the Per default permission after the creation of the file will be like
136 -> x-rx-rw

b) Umask 777 -> NO r,w & x permission

Note: In real time we wont use it is only the root & System level is used.

19) du Command (Disk Usage)

du <FN> & <DN> - used to check how much space a file & a directory takes up. It is used to identify which part of the system use the storage excessively. and du only show the Number. So for human readable format we use flag -sh
By using ls-ltr we can get detail of the Fl & Dir and the size but weather a KB, MB, it won't show but it displays only the Number. So in such case to find a B, KB, MB we use du command.

(a) -s offers the total size of a specified Folder

(b) du -sh <FN> & <DN> - Display the size information in Human readable format.

(c) du -sh * - To display size of all the files and directories in the present working directory.

(d) du -m <FN> & <DN> - Display the size of F & D in MB

(e) du -k <FN> & <DN> - in KB

(f) du -h < > - informs the last modification date of the displayed folder's file.

Note: We must provide the directory path when using the du command.

20) df - command

Available

df To report the system's disk space usage of Hard disk, shown in percentage and KB

& flag

df [option] <F> - used to see the current directory's system disk space usage in a Human readable format

-h - human readable

(a) df -h <FN> - It will list all the drive's usage space

(b) df - M <FN> - displays information on the file system usage in MBs.

(c) df - k <FN> - in kBs

(d) df - T <FN> - shows the file system type in a new column

(e) df - h <-> & <FN> - used to find the disk usage of the present working directory.

(f) free - g - used to check the ram size in giga B

(g) free - b - in Bite

free - tera - Tera bite

(h) free - m - in Mega B

fra - k - kilobite

Note :- If it is less than 1 GB it won't show

2) mv command (move & rename the file & dir)

mv command used to move and rename files and directories. It doesn't produce an out put upon execution.

(a) mv <Old_N> <New_N> (Source, Destination) - used to rename the file & directory will when the source & destination is on same directory / path.

(b) mv <source> <Destination / path> - used to move the file & directory when the source & destination is on different path / directory.

Note: mv command behave differently depending on the destination path, if the S & Destination is on the same directory it will acts as rename if S & D are in different path / directory this will act as cut & paste & move.

(c) mv <source> . - move the file from the source to the present working directory.

(d) mv <old name> <Path / destination / new name> (Source) - it is used to move the file name & from source to the destination and renaming the file.

Note: In destination if we give file Name it will act as rename with in the directory & if we give dir name it will accept as move.

(22) Cp command

Cp command is used to copy file & directories and their content to the destination.

(a) Cp <FN> <Destination> - To copy one file from the current directory to another directory.

(b) Cp <FN1 FN2 FN3> <path/Destination> - To copy files to a directory.

(c) Cp <FN> <FN> - To copy the content of the file to the new file in the same directory. in case the file dint exist it will create new file.

(d) Cp -R <path/soure/Directory> <path/Destination/Directory> - Used to copy the an entire directory from source to the destination.

(e) Cp -i <FN1> <DN1> - If u are coping file to the directory, if the file already exist in the Dir it will directly & with out intimation overwrite the content of the file.

Note: if we use the flag -i (interactive mode) then it will ask before overwriting & intimate and also we can know weather the file is exist Before.

If we type 'y' it will overwrite & 'n' it will copy

(f) Cp -R <DN1> <DN2> - If the Destination dir2 is doesn't exist it will create and copy only the content into the Dir2. If the Dir2 exist it will copy the entire dir1.

(ii)

Tee Command

The redirection will not display any output, but directly save the output to a file. But the tee command will not only redirect the output to new file, but it will also display the output.

Syn: cat <FN> | tee <FN> - the output of the command cat will save to the file and it will display the output.

cat (<FN1> <FN2>) > FN3 - used to transfer the content of the FN1 & FN2 in to the FN3 of new file.

(23) I/O Redirection

Redirection is a process where we can copy the output of any command's file's in to a new file.

(>) Redirect is used to store the output output to the any of the file.
Redirect is used to write output of the command to a file.
If the file does not exist then it will create it and write the content to the file.
If the file already exist it will overwrite the content of the file.

Append (>>) Append is used to attach the output of the command to the end of the file.
It will not overwrite the content of the file it will attach if the file does not present it will create if the file already exist it will append the content.

(a) ls -ltr > log - it will save the output of the ls-ltr command to the log file.

(b) wc file > log - output of the word cont count of the file to the log file.

(c) cat f1 f2 > f3 - Saves the content of two files in to a new file.

24) rm Command (remove & Delete

rm <FN> - used to delete files within a direct-ory.

Note:- make sure that the user performing this comm-and has write permissions.

(a) rm <F1><F2><F3> - used to delete the multi-ple files in a directory.

(b) rm -r <FN> - used to delete the file recur-sely in Pwd.

(c) rm -i <FN> - Prompts System confirmation before deleting a file.

(d) rm -f <FN> - remove without confirmation.

(e) rm -rf <Dir N> - used to delete the directory -r - recursely f - forcefully. & multiple directory.
Since directory contains Sub dir & files.

(f) rmdir <Empty Dir> - used to delete the Empty directory.

(g) rmdir -P <Path/Directory> - used to delete the an Empty Subdirectories

(h) rm -rf <dir1/dir2/dir3> used to delete the dir3.

(i) rmdir -P $d_1/d_2/d_3/d_4$ - used to delete the Empty directory Structure & Subdirectories.

Chown Command (Change Owner) file.
Chown cmd is used to change the ownership of the user or group.

Note: User must be the Owner of a file to change the Ownership for user or group.

Syn: Chown <option> <UN> <File N>

(a) Chown <Owner-Name> <FN> - is used to Change the Owner of the file to the user Karthik.

(b) Sudo chown root file.txt - used to change the ownership of the file file.txt as root to the root.

(c) Sudo chown -f Karthik file1 - used to supress the most of Error messages. When you are not permitted to change group permissions & show Errors, this flag forcefully/silently changes the Ownership.

(d) Sudo chown : <Group-Name> <file.txt> - used to change the Ownership of the file to the Group.

(e) Chown Karthik: Group Name <FN> - used Change the Owner as well as group Ownership of the file at the same time.

(f) Chown Karthik: Group Name <F1> <F2> for multiple files.

Sed Command (Stream line Editor)

-> Sed command is used to search, find, replace and also insertion & deletion.

-> In most common use of Sed Cand is for Substitution or for find and replace.

-> Using Sed we can edit file without opening file, which is much quicker way to find & replace something in file.

-> Sed is a stream line editor where we can do insertion & deletion, search, substitute the pattern into the file, & print.

Synt : Sed <option> < s/oldP/NP/g > < File N >

1) Replacing / Substituting the Pattern.

(a) Sed 's/old-N/New-N/' <FN> - used to replace the pattern in a file.

Note: By default, the Sed command replace the fired accurance of the pattern in each line and it won't replace the 2nd, 3rd accurance in the line.

b) Sed 's/old-N/New-N/2' <FN> - used to replace the 1st, 2nd Occurance of a pattern in a line.

c) Sed 's/ON/NN/g' <FN> - used to replace all the pattern in the line.

d) `Sed 's/ON/NN/3g' <FN>` - used to replace the pattern from the 3rd occurance to the last / all occurance in a line.
Where except the 1st & 2nd it will replace all the occurances.

(e) `Sed '3s/unix/linux/' <FN>` - used to replace the pattern in the specific line. (3rd line)

(f) `Sed '10,30s/ON/NN/gi' <FN>` - from 10th line to 30th line

(g) `Sed '10,$s/ON/NN/gi' <FN>` - from 10th line to end of line.

(h) `Sed -i 's/ON/NN/g' <FN>` -
i - used to save the changed pattern directly in to the file. (insertion)

Note: without -i it will take the content from the file and replace the substitute the pattern and prints on the terminal / streamline and in the file it doesn't change any thing.
If we give flag -i and all the pattern changed will save to that file. & it will not shown in the terminal.

(i) `Sed -i 's/ON/NN/gi' <FN>` - with case insentitive.

(j) `Sed -n 's/ON/NN/p' <FN>` - prints only the replaced lines.

② Deleting lines from a particular file (content).

Sed command can also be used for deleting line from a particular line.
And it performs the deletion operation without even opening the file.

(a) sed 'nd' <FN> - To delete the particular line (n) in the file.

(b) sed '5d' <FN> - Delete 5th line

(c) sed '$d' <FN> - Delete last line of the file.

(d) sed '3,6d' <FN> - Delete the lines from 3rd to 6th line in a file.

(e) sed '3,$d' <FN> - from 3rd to last line

(f) sed '/Pattern/d' <FN> - To delete the pattern matching line.

③ Print the particular line content in the file.

(a) sed -n '98p' <FN> - Print the 98th line.

(b) sed -n '10,20p' <FN> - Print the lines from 10th to 20th line.

Cut Command

Cut command is used to cut the content of the file in column wise. (file like excel sheet)

Note: It will work only if the columns are seperated with the same delimiter.

Syn: `cut <Option> <File N>`

Note: Cut without options specifies error and must specify a list of Bites, Characters & fields

1) `-b (byte)` used to extract the specific bites.
   Like Bike - Each alphabet is a Byte.

Syntax: `cut -b <Byte NO> <File N>`

(a) `cut -b 1,2,3 <FN>` - used to list the first 3 bits from each word (Ex) ~~Bike~~

(b) `cut -b 1-3,5-7 <FN>` - used to list the bites with ranges from 1-3 & 5-7

(c) `cut -b -3 <FN>` - used to list the bites from 1st bite to 3rd bite of a line

(d) `cut -b 3- <FN>` - used to list the byte from 3rd byte to end byte of a line

(Eg) For

	(a)	(b)	(c)	(d)
Bikesol	Bik	Biksol	Bik	esol
Carbon	Car	Caronl	Car	bonl
Cycling	Cyc	Cycing	Cyc	ling
~~Penciling~~ Pencing	Pen	Pening	Pen	cing

If Column are separated

(28) (29)

✱ ② field (-f) - used to cut the content in Column wise.

Synt cut -d "delimiter" -f <FIeld NO> <FN>

(field seperator) d - delimiter it may space, :, |

f - field or Column

(a) cut -d " " -f1 <FN> - used to display the the first fild (1)/Colum of the file.

(b) cut -d " " -f 1,2 <FN> - used to display the 1st & 2nd field of the file.

(c) cut -d " " -f 1-3 <FN> - used it print from 1st to 4th fild.

(Ex) A : B : C

21 24 27

22 25 28

23 26 29

->

(a)

A

21

22

23

(b)

A B

21 24

22 25

23 26

(c)

A B C

21 24 27

22 25 28

23 26 29

Note: If there is a change in delimiter it won't print that line. cut command won't detect that there is the issue with the cut command.

So we are using cut command our file should be seperated with the delimiters with proper manner if it found any mismatch it won't print that line.

If Column are seperated with 1 or more spaces or delim-ite awk will detect & give the exact output

awk Command

awk command is used to cut the content of the file in column wise.

Syntax awk -F " " '{print $n}' <FN>

Note:

(a) awk -F " " '{print $1}' <FN> - used to print the first column of the file.

Note: If their is change in the delimiter also it will print the content of the column without skiping.

(b) awk -F " " '{print $1,$3}' <FN> - Prints the first and 3rd column.

(c) awk -F " " '{print $NF}' <FN> - Prints the last column. $1, $NF - 1st & last

Note: NF (No of fields) - NF cmd keeps a count of the no of fields within the current input record.

(d) awk -F " " '{print $(NF-1)}' <FN> - Prints the last 2nd column.

Note: Diff B/w awk & cut

awk & cut both are used for cut the file content in the column wise. But in cut if the column content is seperated with 1 or more spaces it wont work properly & we may not get the desired output. SO to overcome that issue we can use awk Because awk will detect automatically the column and display.

It is also used to check weather the process is running or not

Ps command (process status)

Ps cmd is used to list all the running process information (PID's & other information) on the system.

Note: It reads the process information from the virtual files in /Proc file system.
/Proc contains the virtual file.

Syntax: Ps <option> ps -e All the process user & system
Ps -f used

(a) Ps -ef - used to list all the running process in a system

Note: ~~Linux is a~~ Since linux is a multitasking and multiprocessing system so it allows multiple processes to operate simultaneously without interfering with each other.

(b) Ps -T - used to list all the running process related to Terminal.
Associated

(c) Ps aux - list's more detailed.
Ps auxf - it will give tree structure where we can see the process from parent to child

(d) Ps -ef | grep -i "pattern" - used to list the particular pattern

(e) Ps -ef | grep -ie "Process 1" -ie "Process 2" used to list multiple process.

(f) PS -U <username> - used to list the Process run by particular user.

(g) PS -G <GROUP Name> - The Group ID identifies the group of the user who created the process.

(h) PS -g <GROUP ID> -

Kill Command

Kill command in linux, is a build in command which is used to terminate process manually. Kill Cmd sends a signal to a process which terminates the process. forcefully.

Syntax Kill <PID>

Note: (6) -9 is a specific signal to terminate the process.

(a) kill -9 <PID> & kill -9 1986 - is used to kill the process with the PID.

(b) kill -9 <PID1> <PID2> <PID3> - is used to kill the multiple process.

Locate Command

Locate command is used to find a file in the database System

Moreover, adding the "-i" argument will turn off case sensitivity, so we can search for a file even if you don't remember its exact name.

(a) locate <Name/FN/DN> - used to find the Name in database.

(b) locate -i <FN> - used to find the file with incase sentitive.

(c) locate -i <FN>* <FN> & <School>* Pad - used to search for the file that contains the words School and note with incase sensitive.

Find Command

Find command is used to find the & search the file with in the mentioned / specific directory used to find the location of a file and the find is automatic / defaultly recursive it will search all the sub directories.

Syntax find <option><path><expression>

(a) find . -name <"FN"> - used to find the location of the file in Pwd with the name FN

find . -iname

(b) find . -iname <"FN"> - Intensitive to the FN.

(c) find / -iname <"FN"> - Used to find the location of the file in the root directory.

(d) find ~~/dir~~ <path> -name <"FN"> - Used to find the file of ~~that~~ path by using path.

(e) find /home/kal -name text.* - Used to find the file when we dont no the extension of it in this case what ever the comes the text. it then it is going to search all those files.

(f) find /path/ -name *.text - Used to Search for all the .txt Extension file. & Pattern.

(g) find . -type f -iname <FN> - Used to list all the files in the Present Working directory.

(h) find . -type d -iname <FN> - Used to list only directories in the PWD.

(i) find . -type f -mtime -10 - Used to list the files which are created with in 10 days.. days

(g) find . -type f -mtime -6 - Before 10 days.

(k) find . -type d -mtime -3 - list directories.

(l) find . -type -mtime +4 - Before 4 days list files & directories.
Ily for minds

(m) find . -type -mmin -15 - with in 15 min.

xargs command (.arguments)

cmd is used to build and execute commands from standard output.

Note: Some commands like grep can accept input as parameters, but some commands accept arguments. this is place where xargs come into picture.

Syn: xargs <options> <command>

Any output from any command are treated as a parameter inorder to pass it as an argument to the next cmd we use xargs

(a) find . -type f -mtime +60 | xargs rm -
used to delete the list of all the files which are output of the find cmd so in this case the output (filenames) are treated as arguments

Note: xargs used to pass the output of particular command to the next cmd as an arguments..
If there more no of parameters it will pass 1 by 1. as an argument.

(b) find . -type f -mtime +365 | xargs rm -
Before 1 Year files

Link (link is a pointer to a file)

~~Link is a.~~

In a linux file system, a link is a connection b/w a file name and the actual data on the disk.

- ~~By~~ creating links is a kind of shortcuts to access a file.
- -> Link allows more than one file name to refer to the same file, executable.

There are two types of links.

1) Softlink or Symbolic link. or Sym link
2) Hard link.

1) Soft link - Soft link is a shortcut way file if we make any changes in the original file then it will get reflected in soft link.
- -> In case if we delete the original file then soft link won't work.

Note: We create the shortcuts for the to protect the files.
But the soft link doesn't contains any file it will just get the reference from the O File.
Usually to protect, manage & if we don't want to show the file where it is.

Syn - ln -s <FN> <Link Name>

(a) ln -s <FN> <LN> - Used to create a link for a file which is in Pwd & will create SL in the PWD.

Soft link

-> Each soft linked file contains a separate Inode value that points to the original file.

-> If the original file is deleted or moved, the the softlinked file ~~not~~ will not work correctly (called hanging link)

-> ~~ls -l command~~ Soft link contains the path of the original file not the content.

-> Removing soft link doesn't affect anything but removing the original file, the link becomes "dangling".

-> The size of the soft link is equal to the length of the path of the original file.

-> If we make any changes in the original file content immediately with in a fraction of time it will get reflected to the Soft link.

~~2) Hard link used~~

(b) ls -s Path/FN <LN> - Used to create the shortcut of the file which is present in that Path.

ls-l - cmd shows all links with first column value 2 if the link points to Original file SL --> <FN>

2) Hard link :

-> It is also a shortcut way of file if we make any changes in the O.F then it will get reflected to the shortcut file. / hard link also.

-> In this case if we delete the Original file the hard link will work Because it points to (i-node) inode of a file.

-> Each Hard linked file is assigned the same inode value as the original, therefore they refer-ences the same physical file location. Hard links are more flexible & remain linked even if the ori-ginal or linked files are moved throughout the file system; although hard links are unable to cross different file system.

Syn: ln <FN> <HL>

(a) ln /home/dir1/test <Hard link Name> - Used to create shortcut / Hard link for a file.

(b) ls -i <FN> - Used to display the inode of a file or directory.

HD link

-> Links have Actual file content.

-> Removing any link, Just reduces the link count, but doesn't Effect the Other link.

-> Even if we Change Name of OF & In also the HL works properly.

-> We cannot create the Hardlink for a directory to avoid recursive loop.

-> The size of any of the HL file i.e. same as O.F & if we change the content in any of the HL then size of all HD HL files are updated.

Note: ① When ever we create a new file the file will be attached to a inode (Just like a Badge) and inode will be unique through the File System and it is by default bit i.e system generated

② Soft link will refer the file & HL will refer the inode if we delete the O.F the SL will won't work since it is pointing to O.F.
But if we delete the O.F the H.D link will work since it refers to inode. Because if we delete the file also the inode will be present.

Uname Command

Uname cmd will print detailed information about our linux system and Hardware. This includes the machine name, Operating System, & kernel.

Syntax Uname <OPTIONS>
Complete information ast linux.

(a) Uname -a - Prints all the system information in the order, Kernel Name, network node hostname, kernel release date, Kernel version, machine hardware name, hardware platform, O.S.

(b) Uname -s - Prints the kernel name.

(c) Uname -n - Prints the hostname of the network node (current computer)

(d) Uname -v - Prints Check Prelent linux version.

Hostname Command

Hostname cmd is used to know the system's Hostname.

Also used to KNOW Obtain the DNS (Domain name System) name & Set the System's hostname.

(Note) A Hostname is a name which is given to a computer & it attached to the network. Its main purpose is to uniquely identify over a network.

Syn: hostname -<OPTION><File>

HD book

(a) hostname - displays the host name

(b) hostname -a - used to get alias name of the Host system. (If any)
It will return an Empty line if no alias name is set.

(c) hostname -A - used to get all Fully Qualified domain name of the Host system.

(d) hostname -b - used to set a hostname.

(e) hostname -i - Displays machine ip address.

nohup command (No Hang up)

nohup cmd is used to run the process even after logging out from shell / terminal.

Note: Every cmd in linux starts a process at the time of its execution, which automatically gets terminated upon exiting the terminal. Suppose, you are executing programs over SSH and if the connection drops, the session will be terminated, all the executing process will stop, In such cases, running cmd in the Background can be very helpful to the user & this is where nohup cmd comes in to picture.

Usually, every process in linux system is sent a SIGHUP (Signal Hang Up) which is responsible for terminating the process after closing / exiting the terminal. Nohup cmd prevents the process from receiving this signal upon closing or exiting the terminal / shell. Once the job is started or executed nohup cmd., stdin/tty will not be available to the user and nohup.out file is used as the default file for stdout & stderr.

If the output of the nohup command is redirected to some other file, nohup.out file is not generated.

Syn: nohup command [cmd - arguments]

(a) nohup bash <FN> (a script) - Runs the cmd in the foreground and redirects the cmd output to the nohup.out file if any redirecting file name is not mentioned.

(b) nohup bash <cmd> > Output.sh - redirect the Output to Output.sh

(c) nohup bash <cmd> & - To run the cmd in the background, the '&' symbol is appended at the end of the cmd. After running the cmd in B.G it doesn't return to the shell cmd prompt.
It can be brought back to the F.G. with the fg cmd.

(d) To run multiple cmds in B.G

1) nohup bash -c 'commands'
2) nohup bash -c 'cal && ls' - Output To run multiple cmds in B.G & Output will be by default stored in nohup.out.
3) nohup bash -c 'cal && ls' > Output.txt redirected to Output.txt file.

Note: nohup bash <cmds> & - it will run a cmd in BG and generate a process id & Job id
fg <Job id> - used to Bring the process from BG to FG if it is still running.
Jobs - used to list the running Jobs in BG

Top Command

Top cmd will display all the running process & a dynamic real-time view of the current system.

This cmd shows the summary info of the system & the list of process & threads which are currently managed by the linux kernel.

After executing this cmd the output half portion will contain the statistics of process & resource usage. And lower half half contains a list of the currently running process. q will exit the cmd

Usually to know which process is taken the CPU & memory usage of the system.

By default the Top cmd refreshes the usage of the process for every 3 seconds. and to change the refresh time type s & change on the terminal

If with the Top cmd list the info like PID, PR (process priority), VIRT (Total virtual memory used by the task, User (User name of owner of Task), %CPU (%. usage), %MEM (Memory usage of Task) RES (How much physical RAM the process is using, measured in KB) SHR (Shared memory size (KB) used by the task, TIME (CPU time)

(41) a

Uptime Command

Uptime cmd is used to find out how long the system is up and running.

And also returns set of values that involve, the current time, and the amt of time the system is in running state, no of users currently logged into, and the ~~load~~ load time for the past 1, 5 & 15 min respectively.

< Synt - Uptime <options>

load average : 0.00 , 5.000 , 15.000
1min 5min 15min -> System load / time

System load - No of process waiting to get executed

(a) Uptime -p - Show uptime in pretty format.

(b) Uptime -s - System up since.

(41) b

SSH Command (Secure Shell)

SSH Command is used to connect to the remote machine & virtual machine & for executing command in Remote machine it use only port 22

Username-IP
ssh <username@server> - used to login to a remote server
Password

SCP Command (Secure Copy)

SCP cmd is used to copy files & directories b/w the local & a remote system & b/w 2 remote servers

Username@IP Path
Syntax SCP <FN> <Source Server>:<Destination server>
Destination

(a) SCP FN1 user@9945server : /home/temp -
Server 1 Server 2 Path
used to copy the files from local server 1 to the server 2 remote server.

(b) SCP -P <=,,=> - Preserve the same permissions of a file and from Server 1 to Server 2

(c) SCP -r <=,,=> - Copy directory from S1 to S2 recursively.

(d) SCP user@IP: path of file . Path of local server - used to copy files from the remote server to the local server where the remote server ID & path of the file were it exist <space> to the path of the local server.

Netstat command (network statistics)

Netstat cmd is used to display various network related information such as network connections, routing tables, interface statistics, masquerade connections, multicast membership in addition it can used to view network protocol statistics.

(a) netstat -a - used to show all the connections which are available on system whether it is a tcp, udp & unix connection also it will display the status connected, listening & established & diff kind of connection state.

(b) netstat -a | less - to see the terminal more flexible. & q to quit less

(c) netstat -at - to list the tcp connection only

(d) netstat -au - udp

(e) netstat -l - used to see the ports which are listening.

(f) netstat -lt - listening tcp connection

(g) netstat -lu - listening udp lly x for unix

(h) netstat -s - show statistics we can see ip total packets received & out, which type of connection it is, protocol of packets.

(i) netstat -st - statistic of tcp connection

(j) netstat -su - —— , udp

who command

who cmd is used to get information about curr-ently logged in user on the system.

Syn: who [option] <F.N>

Note: who cmd also used to know current run level of the system and Time of last system boot.

By running this cmd default by display the information like :

1) login name of the users.
2) Terminal line numbers.
3) Login time of the user into system.
4) Remote host name of the user.

a) who -a - displays all details of current logged in user.

b) whoami - displays system user name.

c) who -w - displays list of user and their activities.

d) who -u - displays list of users logged into system.

Diff B/w who & whoami - Both the cmd's are used to display info about the user. whoami display the name of the current user & who cmd will display all the users.

(44) use net-tools instead

ifconfig command (Interface configuration)

-> ifconfig is used to change and view the network configuration on the system.

-> used to configure the kernel-resident network interfaces. It is used at the Boot time to setup the interfaces as necessary.

-> It is usually needed during debugging or when you need system tuning.

-> Also this cmd is used to assign the IP address and netmask to an interface or to enable or disable a given interface.

Synt ifconfig <option> <interface>

Note: eth0 (interface) - it means the wired connected.

WLAN0 - " wireless connected.

lo interface - loopback interface used to communicate itself.

Note: There can be multiple interface eth0, eth1, eth2

(a) ifconfig -a - used to display all the interfaces available, even if they are down.

(b) ifconfig eth0 up - used to activate the driver for the given interface or connect to net

(c) ifconfig eth0 down - Deactivate / disconnect to the network.

Sudo Command (superuser DO)

Sudo is one of the most popular basic linux commands that lets you perform tasks that require administrative or root permissions.

When using sudo, the system will prompt user to authenticate themselves with a password. Then, the linux system will log a timestamp as a tracker. By default every root user can run sudo commands for 15 min/session.

If you try to run sudo in the command lin without authenticating ourself, the system will log the activity as a security event.

Sudo allow us to do extra previliages as an administrator & power user and sudo is used in the situation like permission denied error by default some system doesn't have administrati & root previliage.

Note: It will ask the login password during execution of sudo but the password will not shown on the terminal. & it pass will not ask again in particular session.

Syn sudo (command) - to run the command with admini & root previliage.

a) sudo -s - to switch to the root user

b) sudo -k (comm) - invalidate the timestamp file

c) sudo -g ( ) - runs commands as a specified group name & ID

d) sudo -h ( ) - host run the command on the host

User Administration

-> User is anyone who use a computer.
-> The system has a name for each account it creates & it is this name by which a person gain access to use it.
-> Linux supports multiple users to be logged in the system.
-> Managing user is done for the purpose of security by limiting access in certain specific ways.
-> A user account of a system is uniquely identified by a numerical NO called the User ID (UID)
-> There are some reserved names which may not be used such as "root", "home".

User Account

-> One user can create multiple accounts for logging in the linux.
-> Each user in the computer can have diff previleges.
-> The user having the highest authority / root user can restrict the other user to add new user & the data he has access to.
-> By default, one account named root has access to all the files & perform every action.

Switch User

Syn SU <OPTION> <User Name> - used to switch the user from one user to another or from the default root user to the other particular user.

→ su <op><UN> - Once the command is given, it asks for the password of the user. If correct password is provided, it opens the shell for that user

→ exit - To return back to original user.

Note - Root doesn't require password to switch to other user.

(a) su -P <UN> - keeps the same shell environment, consisting HOME, SHELL, USER & LOGNAME.

(b) su -S <UN> - lets you specify a different shell environment to run.

(c) su -l <UN> - runs a login script to switch to a different username.

Note: When execute su without any option or argument, the su command runs through root previlege. It will prompt you to authenticate & use the sudo previlege temporarily.

(a) Add New User

Syn useradd <User Name> - used to add (create) the new user

Configuration to add new user is present in /etc/default/useradd

(a) sudo useradd -m <username> - it is used to create the default home directory for that particular user.

(b) sudo useradd -m -d <User Name> - used to create the user with predefined hom directory.
Ex: useradd -m -d /home/linux/user <UN>

(c) useradd -m -e <YYYY-MM-DD> <User Name>
used to create the user with an expiry date or to change the expiry date of a user.

(d) useradd -s <User Name> - used to allow the user to use the shell & specified the default shell for the user.

NOte - After Flag -s - After providing the flag it will allow to use the shell after flag we provide. /bin/bash.

(e) useradd -g <User Name> <Group Name> - it is used to ~~add the user~~ create the new user and add to existing group / assign to group.

(f) sudo useradd <User Name> -c "My Comment" - it will used to provide the comment to the user

(g) sudo useradd - <User Name> - used to create the user as a root user.

(h) sudo passwd - used to change the Password of the user.

(i) sudo passwd <User Name> - used to change the Pwd of Partic user.

(j) usermod -c - to add a comment for a user.

(b) Remove User or Delete User:

Syn userdel <username> - Used to delete the user from the system

Note: By default, the home directory of the user is not deleted. and it will delete only the user-name of the user & passwords. Because sometim-es the organization needs the data like, whatever the work he done.

(a) sudo userdel -r <user Name> - Used to delete the username, password, data and the home directory of the user at the same time.

(b) sudo rm -r <user directory> - Used to delete the user home directory after the removal of the user from the computer.

Note: we should be careful Because it will remove all the data of the user work.

(c) Modify A User

Modify user is used to modify the user credent-ials of after the creation of the user like expira-tion time, password of the user, login Dir

Syntax: sudo usermod <options> <username>

(a) sudo usermod -P <username> - & Used to specifie a password for user.

Note: usermode cmd needs to be executed only as a root user.

Note: The flags for the useradd and usermod are same

(b) sudo usermod -d /home/dir1 <user N> - used to change the home directory of a user

(c) usermod -e 2020-05-29 <UN> - used to change the expiry date of a user.

(d) usermod -g <Group-N> <User-N> - used to change the group of the user.

(e) user mod -l <old-N> <New-N> - used to change login name of the user.

(f) usermod -s /bin/sh <User-N> - create a shell for the user.

(g) usermod -U 1234 <User-N> - To change the user id of a user.

Password Aging Policy

After we provide the password the minimum days to change the password is 0 & maximum was 99,999 Days But their was no way of the password was aging.

To be in the greater side of security it is advisable to change our PSW time to time. we have Password Aging POLICY.

Syntax: Chage <option> <UN>

View s

Chage command is used to change password related information. This information is used by the system to determine when a user must change his/her password.

It is also used to change the no of days b/w pws changes & the date of the last pws change.

(a) Chage -d <UN> - used to set the last password change date to our specified date or new date.
(Eg) sudo chage -d 2019-12-01 <UN>

(b) sudo chage -l <UN> - used to view the account aging information.

(c) sudo chage -E <UN> - used to specify the date when the account should expire.

(d) sudo chage -M <UN> - used to set the maximum NO of days b/w the password change

(e) sudo change -m <UN> - minimum days.
(Eg) sudo chage -m 6 <UN>

(f)

(f) sudo chage - <UN> - to set no of days the account should be inactive after it expiry.

Note: It is necessary that the user should change the password after it expires, this command is useful when the user does not login after it's expiry. Even after this inactivity period if the password is not changed then the account is locked & user should approach the admin to unlock it.

Group Management

Groups in linux refer to the user groups. a collection of users In real time there are many users, there might be some privileges that some users have & some don't, & it becomes difficult to manage all the permissions at the individual user level. So using groups, we can group together a no of users, & set previleges & permi-ssions for the entire group

-> Each group associated with a unique ID called the group ID (GID)

-> There are 2 type of group: primary & supplementary group & each user is a member of primary group & of zero or 'more than zero' supplementary group.

-> Group ID start with 1000 & there is no limit of user in group.

-> It will allow us to see who are the members of the group.

Syntax: groupadd <option> <group-Name>

(a) Sudo groupadd <group-name> - used to create the new group

Note: Every new group created is registered in the file "etc/group". To verify that the group is has been created, enter the cmd.

Sudo tail /etc/group - since the group created recently will add at the end.

Ping command (Packet internet Groper)

Ping command is used to check the network connectivity b/w the host and server/host. & speed of connection.
This cmd take as input the IP address of the URL and send a data packet to the specified address with the message ping and get a response from the server/host this time is recorded which is called latency.

Note: Fast ping low latency means faster connection.
Ping time is generally measured in millisecond.

Syntax - Ping <I.P & URL>

Note: Ctrl + C - To stop the ping request.

a) Ping -c 5 <URL> - Used to send the No of packets / Ping req to the Server & Host.

b) Ping -i 2 <URL> - By default ping wait 1sec to send next packet we can change this time by using -i.

c) Ping -w 3 <URL> - This will stop pinging after 3 sec

Zip = tar + gzip

tar Command (Tape archive)

tar cmd is used to create Archive and extract the Archive files.

tar cmd can also used for to create compressed & uncompressed Archive files and also maintain and modify them.

Syntax: tar [option] <test.tar> <file & directory to be archived>

a) tar -cvf <test.tar> <file1 f2 f3 f4> & <dir>

c - Creating an archive / tar file.

v - Verbose to display progress.

f - Allows us to specify a file name.

Note: If we wont specify the file name linux will create a zip file which is system dependent whose name is not defined by you.

b) tar -xvf <test.tar> - used to extract the test file from the test.tar

c) gzip <file.txt> <file.gz> - used to compress the files archive files.

d) gunzip <file.gz> - uncompress the zip file.

e) tar -cvzf <file.tar.gz> <files> - used to archive and compress the files at a time.

z - zip the file.

f) tar -xvzf <file.tar.gz> - uncompress & un archive a file

Archiving is the process of collecting and storing a group of files and directories into one file. The tar utility performs this Action.

Compression is the act of shrinking the size of a file, which is quite useful in sending large files over the internet. The gzip utility performs this action.

Zip command

Zip is a compression and file packaging utility for Unix. Each file is stored in single.zip file with the extension .zip.

-> Zip is used to compress the files to reduce file size and also used as file package utility.

-> If you have a limited bandwidth b/w two servers and want to transfer the files faster, then zip the file and transfer.

-> Zip program puts one & more compressed files into a single zip archive, along with the information abt the files. An entire directory structure can be packed into a zip archive with a single cmd.

Syntax: zip [option] <zipfile.zip> <file_list>
& file2 file3

a) zip myfile.zip file.txt - used to create the archive and compress the file to myfile.zip.

b) unzip myfile.zip - used to list, test and extract files from a zip Archive to the cPwd.

c) zip -r myfile.zip directory_name - it will recursively zips the files in a directory

Run Multiple Terminal Commands

Used to combine the commands and run

; - Semi colone

(a) ls ; Pwd - used to first it give the out for the ls command then execute the Pwd cmd

(b) date ; Cal ; Pwd - first date, Cal & Pwd

If one of the command is wrong or Fails then it will execute the remaining command.

(c) && - Double Amp Ampacent Simbol

This is also used for the same purpose to combine the multiple cmds & execute in sequence.

ls && Pwd && date && cal

Note: The main difference b/w the ';' & "&&" is if the previous command fails using && the next command wont execute but in case of ';' if the previous command fails or success it will execute the remaining commands in sequentical.

(d) ls || Pwd - Double pipe if the first command execute then it will go to the next command.