**---------------------------- Why Command Use Line ? --------------------------------**

- More control over the machine :

Mastering the command line gives us greater control over the machine. We can run commands to change permission, view hidden files, interact with databases, start servers, manage processes etc.

- It’s faster :

We could perform tasks much faster than could using a GUI.

- You can automate many tasks :

Can make 10,000 new files with different names in a single line using a single command.

- It’s available everywhere

These can also be used on Linux on Mac OS and a little bit of them on Windows too.

- It’s basically a requirement :

We need to use the command line whether we become web developer, data scientist, system admin, security admin or ml engineer.

- Needed for cloud computing :

Most could services are operated via command line interface.

**----------------------------------- Ubuntu Basic Commands --------------------------------**

Ctrl + Alt + T - Open terminal/command/shell/prompt

Ctrl + D - Exit the terminal

exit - Exit the terminal

pwd - Print/Present working directory of the system : relative command

cd / - Change directory to root ( / ) directory : absolute command

ls - List of all directory inside root directory or pwd

ls -l - List of files with extra details

ls -lh - List of files with extra details in human readable format

cd dirname - Move to the child or next directory

cd - Quickly move to the home directory ( home/user ) : absolute command

cd ~ - Quickly move to the home directory ( home/user ) : absolute command

cd ~ / Music - Quickly move to the home directory then Music : absolute command

cd . . - Move to the parent or previous directory : relative command

cd . . / . . - Move to the parent ( home ) / parent ( root ) directory

cd . . / . . / etc - Move to the etc directory inside parent ( home ) / parent ( root )

cd /etc - Move to root then change to etc directory

cd /home/user/Music - Move to the root/home/user/Music directory

Absolute path - Any path that starts with forward slash ( / )

Relative path - That user cd ( next ) or . . ( previous ) command

whoami - Tells the current user of the system

clear - Clear terminal

clear -x - Not clear but scroll so it seems like clear

Ctrl + L - Clear terminal

reset - Clear terminal

man any-command - View the manual of the specific command

man man - View the manual of the man command

Ctrl + Shift + + - Zoom in

Ctrl + - - Zoom out

**-------------------------------- Directory/Folder Creation ----------------------------------**

mkdir /temp/practice - Make directory named practice inside temp inside root

mkdir practice - Make directory named practice

cd practice - Move to practice directory

mkdir dir1 dir2 dir3 - Make dir1, dir2, dir3 directory inside pwd

Mkdir Command accept at least one argument or parameter

Cd command can accept one or zero argument but not more than that

mkdir -p dir4/dir5/dir6 - Make dir6 inside dir5, dir5 inside dir4, dir4 inside pwd

The -p is called as an option or switch ( Here p means create parent directories too )

mkdir --parents --verbose dir4/dir5 - Make dir5 inside dir4, dir4 inside pwd

mkdir -p --verbose dir4/dir5 - Make dir5 inside dir4, dir4 inside pwd

mkdir -p -v dir4/dir5 - Make dir5 inside dir4, dir4 inside pwd

mkdir -pv dir4/dir5 - Make dir5 inside dir4, dir4 inside pwd

mkdir folder-0 - Make a directory named folder-0

mkdir folder\_0 - Make a directory named folder\_0

mkdir "folder 1" - Make a directory folder 1 having space in the name

mkdir 'folder 2' - Make a directory folder 2 having space in the name

mkdir folder\ 3 - Make a directory folder 3 having space in the name

mkdir "folder 4" "folder 5" - Make a directory folder 4, folder 5 with space in the name

mkdir -p "folder 6"/"folder 7" - Make a directory folder 7 inside folder 6 with space

**----------------- File Create/Read/Write/Override/Append Data -------------------**

touch demo.txt - A demo.txt file created inside pwd

touch f1.txt f2.pdf f3.png - Create three empty files, if exists update the timestamp

cat demo.txt - Prints the demo file content ( all directories and files )

echo “Message” - Prints the message on terminal

echo “Message” > new.txt - Creates a file if not exist & override the file content

cat demo.txt new.txt - Concatenate all the file outputs

cat -n demo.txt - Print all the file outputs along with the numbers

cat demo\_?\_file.txt - The ? indicates any single character in the filename

cat demo\_\*\_file.txt - The \* indicates zero or more characters in the filename

cat t\* - Concatenate all the files starts with t + zero or more chars

cat t\* > combined.txt - Copy all files content name starts from t, in combined.txt

cat t\* >> combined.txt - Append file content in combined.txt

echo “Append” >> combined.txt - Append content in combined.txt

cat combined.txt - Print content of combined.txt on terminal

more combined.txt - Print with a more option to view the large file content

less combined.txt - Print file content on a pager ( Quit with q or Q )

xdg-open . - Open pwd

xdg-open file1.txt - Open file1.txt

xdg-open dir1 - Open dir1 folder

head file1.txt - Print first 10 lines

head file.txt -n 100 - Print first 100 lines

tail file1.txt - Print last 10 lines

tail file.txt -n 100 - Print last 100 lines

tail -f log.txt - Print data but don’t stop at EOF, quit with Ctrl + C

**----------------------------------- Case-Sensitive File Name --------------------------------**

Unix OS is case-sensitive so the below names are different for Unix OS

ls > a.txt

ls > A.txt

ls > a.TXT

ls > A.Txt

**--------------------- Move/Copy/Rename/Delete A File & Folder --------------------**

mv demo.txt dir1 - Move demo.txt from pwd to dir1 directory

ls dir1 - Quickly show the files & folder inside dir1 directory

mv demo.txt . . - Move demo.txt to the parent directory ( cmd execute in child )

mv dir/\* . - Move any file to the parent directory ( cmd execute in parent )

mv a.txt test\_\*.txt dir1 - Move a.txt and all the test\_\*.txt file in the dir1 directory

mv new.txt dir4/dir5/dir6 - Move new.txt inside dir6 that’s inside dir5 that’s inside dir4

ls dir4/dir5/dir6 - Quickly show the files & folder inside dir6 directory

cp dir4/dir5/dir6/new.txt . - Copy new.txt from dir6 to inside pwd

cp new.txt backup\_new.txt - Copy new.txt from pwd to inside pwd

mv backup\_new.txt n\_b.txt - Rename backup\_new.txt to new\_backup.txt

mv ‘dir1’ dir-1 - Rename dir1 to dir-1

mv dir2 dir1 - Move dir2 inside dir1

rm new.txt demo.txt - Delete file named new.txt and demo.txt

rm dir1/new.txt - Delete file named new.txt inside dir1 that’s inside pwd

rmdir dir-\* - Delete all empty folders named dir-\*

rm -r dir-\* - Delete all files and folders whether it’s empty or not

rmdir -p dir4/dir5/dir6 - Delete dir5 then dir6 then dir4 ( All must be empty )

rm -i new—\*.txt - Ask confirmation before deleting files

**------------------------------------------- Plumbing ---------------------------------------------**

wc demo.txt - Count total line, word, characters including \n in demo.txt

wc -l demo.txt - Count total lines in demo.txt

wc -w demo.txt - Count total words in demo.txt

wc -c demo.txt - Count total characters including \n in demo.txt

ls ~ | wc -l - Count lines or folders in home directory

ls /etc | wc -l - Count lines or folders in root/etc directory

ls /etc | less - List all the lines and quit with q or Q

uniq demo.txt - Removes only adjacent duplicate lines

sort demo.txt | uniq - Here all are adjacent lines

sort -u uniq - Does the same as above

sort demo.txt | uniq -d - Only prints duplicate values one for each

sort demo.txt | uniq -u - Only prints single or unique values

sort demo.txt | uniq -c - Print counts along with every value

sort demo.txt | uniq -c | sort -n - Print counts along with every value in a sorted manner

sort demo.txt | uniq -c | sort -nr - Print counts along with every value in reverser sorted

cat demo.txt | uniq | wc -l - Count total unique lines in demo.txt

cat demo.txt | uniq | less - Print total unique lines in demo.txt

cat f1.txt f2.txt | wc -l > f3.txt - Concatenate f1 and f2, count lines and write in f3.txt

sort demo.txt - Print sorted content from file

sort demo.txt | less - Print sorted content from file and quit with q or Q

sort demo.txt | uniq | less - Print sorted and unique content not sorted in file

sort demo.txt > demo.txt - Sort content and save it in itself

sort -n nums.txt - Sort numeric content in file

sort -nr nums.txt - Sort in reversed order

sort -nu nums.txt - Sort only unique numbers

sort -nu nums.txt | wc -l - Count total unique lines in a file

cat f1.txt f2.txt | sort - Concatenate and sort content

**------------------------------------ Hidden Files & Folders ---------------------------------**

Hidden file & folders name starts with a dot ( . ) followed by file name. These files store configuration and settings of the user or software or a system.

ls > .demo.txt - Make a hidden file with name .demo.txt

mkdir .hidden - Make a hidden folder with name .hidden

ls -a - Shows all hidden files in pwd

ls -a .demo.txt - Prints content

ls -a .hidden - Prints content

ls .demo.txt - Prints content

tree -a - Prints the tree structure of pwd including hidden files & folders

ls -a | wc – l - Count total file & folder including hidden ones

cd .hidden - Change current directory to .hidden

rm -r .hidden\* - Delete all hidden files name starts with .hidden

**------------------------------------------- Super User -------------------------------------------**

If anyone asks you to enable the root account or log in as root, be very suspicious of their intentions.

su - Switch user or switch to super user ( root if without name : disabled by default )

su username - Switch to the specific user

logout - logout from the system or Ctrl + D

sudo - Switch user and do this command ( Request superuser rights per command )

sudo su - It will give you the root shell even if the root account is disabled

Sudo is used to prefix a command that has to be run with superuser privileges. Configuration file is used to define which users can do sudo and which commands they can use. Authentication will be done at the time of using the superuser rights.

cat /etc/shadow - Permission denied

sudo cat /etc/shadow - Enter password then access if authenticated

reset - Clear the terminal

sudo cat /etc/shadow - No password needed this time ( 15 minutes session )

sudo apt install tree - Install tree program

sudo apt-get install tree - Install tree program

tree - Prints the tree structure of current directory

If the instruction require you to first add a new software repository to your system using the apt-add-repository command by editing files in /etc/apt or by using a PPA ( Personal Package Archive ), you should be careful.

Use of curl/wget/pip/npm/make/others commands indicates that file are coming from outside distribution repositories tells us to change files permission to make it executable.

**--------------------------------------- More Commands ---------------------------------------**

date - Show current date and time

touch today.txt - Create an empty file

date > today.txt - Redirect and write current date, time inside today.txt

date >> today.txt - Redirect and append current date, time inside today.txt

head today.txt - Prints first 10 lines

ls > today.txt - Redirect and write all listed files inside today.txt, if not exists then create it

ls -l > today.txt - Redirect and write all listed files inside today.txt, if not exists then create it

echo ~ - Prints home directory

echo ‘ ~ ’ - Prints ~ character

echo $PATH - Path environment variable

echo $USER - User name variable

echo \* - Everything ( any file, folder or path ) printed

echo F\* - Prints all the files and folder whose name starts with “ F “

diff f1.txt f2.txt - Nothing will be printed if they’re having same data

diff f1.txt f2.txt - 2a3 > text ( in file 2 at line 3 text is appended )

diff f2.txt f1.txt - 3d2 < text ( in file 1 at line 2 text is deleted )

diff f1.txt f2.txt - 3c3 < text --- > new text ( in file 1 at line 3 text is changed to new text )

diff -y f1.txt f2.txt - Show f1 and f2 side by side

diff -u f1.txt f2.txt - Git like difference, ++ and –

find . - Find every file, folder, nested file and nested folder in pwd

find . -name ‘ f\* ’ - Find everything in pwd that starts with ‘ f ‘

find . -name ‘ \*2\* ‘ - Find everything in pwd that has 2 in it’s name

find . -name ‘ \*.txt ‘ - Find everything in pwd whose extension is .txt

find dir1 -name ‘ \*.js ‘ - Find everything is dir1 whose extension is .js

find . -type d - Find only directories in pwd along with the nested one’s

find . -type f - Find only files in pwd along with the nested one’s

find . -type d -name ‘F\*’ - Find only directories whose name starts with F

find . -type d -iname ‘F\*’ - Find only directories whose name starts with F or f ( case-insensitive )

find . -name ‘E\*’ or -name ‘F\*’ - Find everything whose name starts with E or F

find . -type -f -size +100c - Find file of size more than 100 characters ( bytes )

find . -type f -size +1k -size -1M - Find files of size that bigger than 1kb but less than 1MB

find . -type f -mtime +3 - Find files edited more than 3 days ago

find . -type f -mtime -1 - Find files edited in the last 24 hours

find . -type f -exec ls -l \; - Exec is a command that allows us to run a command on each file result

find . -type f -exec cat { } \; - Find all the files and print out the content, { } is a placeholder for a file

grep hello demo.txt - Find hello inside demo.txt, highlighted in red color

grep -n hello demo.txt - Find hello inside demo.txt along with the line number

grep -nC 2 hello demo.txt - Find hello with the line no. & 2 lines before and after each match

grep -r ‘hello’ - Find hello in all files along with the nested one’s

grep -rn ‘hello’ - Find hello in all files and line no.’s along with the nested files

grep -ri ‘hello’ - Case-insensitive search

du - Prints disk size along with the name of the directory

du dir1 - Tells size of the dir1 directory

du demo.txt - Tells size of the demo.txt file

du -m - Tells the size in megabytes or MB

du -h - Tells the size in human readable format

du -b - Tells the size in bytes

du -h | sort -h - Find and sort in the human readable form

du -h | sort -h - Find and sort in the human readable form in reverse order

df - Tells the allocated, used and free space of the file system

df -h - Tells the file system information in human readable form

df -h dir1 - Tells the file system information about dir1 directory

history - Tells the history of previously executed command with the line number

history | less - Show on less pager

!776 - Re-executed command present at line 776

history | grep ‘test’ - Find commands having test in it

ps - Tells the processes started by the current user

ps ax - Tells all the process status

ps ax | grep ‘user’ - Find those status having user in it

top - Tells us the real time process information

kill -l - Lists the different signals with a no. that we can send

kill -n <PID> - Kill process running with the PID with n signal

killall -9 node - Kill ( -9 is a signal for kill ) all node processes

top - Display dynamic real-time linux processes

jobs - Tells stopped jobs along with the job number

fg 2 - Re-execute the job having Id as

fg - If there is only job re-execute that

bg 1 - Re-execute in background ( & or Running status is the identity of bg jobs )

sleep 2 - Sleep for 2 seconds in foreground

sleep 50 & - Sleep for 50 seconds in background

gzip demo.txt - Zip demo.txt with extension .gz

gzip -c demo.txt > demo.gz - Make a new zip file for existing file

gzip -k demo.txt - Make a new zip file for existing file

gzip -kv demo.txt - Make a new zip file for existing file and tell the reduced size in %

gzip -kv f1.txt f2.txt f3.txt - Make a new and individual zip file for all the existing files

gzip -d demo.gz - Decompress demo.gz

gunzip demo.gz - Unzip demp.gz

gunzip -c demo.gz > un.txt - Make a new unziped demo.gz

tar -cf arch.tar f1.txt f2.txt - Group file f1.txt and f2.txt inside arch.tar

tar -xf arch.tar - Extract all the files from the archive file

tar -xf arch.tar -C dir1 - Extract files and move in dir1 directory

gzip arch.tar - Compress archive file

tar -czf arch.tar.gz f1.txt f2.txt - Group all the files in .tar and compress it

tar -xf arch.tar.gz - Extract all the files or gipped and archived files

nano demo.txt - Open demo.txt in nano editor or create if not exists

Ctrl + X - Quit nano editor

Ctrl + W - Search for a word

Ctrl + S - Save content in a file

Ctrl + G - Help menu for nano editor

Ctrl + K - Cut text

Ctrl + U - Paste text

alias myls=’ls -la’ - Treat myls as ls -la ( temporary alias )

alias - All the listed aliases

cd - Quickly move to the home directory

ls -a - List all the hidden files in home

nano .bashrc - Open .bashrc in nano editor

alias count=’echo {1..99}’ - Write in file and save ( permanent alias )

source .bashrc + count - To run in the existing window

count - To run on a new terminal window

alias lsthis=”ls $PWD” - Prints that content where the alias is made

alias lscurrent=’ls $PWD’ - Print the content of current directory

ls | sort - Here output of the ls is passed as input to sort

sort demo.txt - Here argument is passed to the sort

cat demo.txt | xargs rm - Xargs is used provide

find . -size +1M - List of files greater than 1MB

find . -size +1M | xargs ls -lh - List of files in human readable format

find . -size +1M | ls -lh - List only folders

ln is a link ( hard and soft link ) pointer to another file like a shortcut

Hard links are rarely used. They can’t be linked to directories or disk ( external fs )

Soft links are different. They can be linked to other fs and directories but when the original file is removed, the link will be broken

ln demo.txt hard-link.txt - demo.txt and hard-link.txt links to each other

cat hard-link.txt - Write content of demo.txt

echo “More” >> demo.txt - Append content in demo.txt

cat hard-link.txt - Write updated content of demo.txt

echo “text” >> hard-link.txt - Append content in hard-link.txt

cat demo.txt - Write updated content of demo.txt/hard-link.txt

rm demo.txt - Deleted demo.txt but not hard-link.txt

ln demo.txt soft-link.txt - Created a bolded and light blue color file as link to demo.txt

rm demo.txt - Deleted demo.txt along with the soft-link.txt

who - Tells about all the users logged in the system

passwd - To change to current password of the system

sudo passwd sahil - You can change others password too

sudo chown rob demo.txt - Change ownership of the file to rob

sudo chown -R rob demo.txt - Change ownership to all the nested files

groups - List all the group owner

sudo chown sahil:rob Img/ - Change group owner name ( read-write operations only )

**--------------------------------------- File Permissions ---------------------------------------**

ls -la : List all the file details along with the user, group and world permission

-rw-rw-r— A 10 byte character

File Type : The very first character indicates the type of the file

* Regular File

d Directory

c Character Special File

l Symbolic Link

Then we have 3 group of rest 9 characters :

rw- Permission for owner ( read, write )

rw- Permission for Group ( read, write )

r-- Permission for World ( read )

r ( read ), w ( write ), x ( execute ), - ( no permission ),

That means no one has execute permission

chmod is used to change the permission of a file or directory

u - User or owner of the file

g - Group members

o - Others or world

a - All of the above

+ - Grant a permission

- - Removes permission

= - Set a permission and removes others

r - Read permission

w - Write permission

x - Execute permission

chmod g+w file.txt - Add write permission to all members of the group on file.txt

chmod a-w file.txt - Remove write permission from everyone