Linux commands basic

1. whoami :Find the user name of the machine.
2. printenv SHELL :Checking the name of the Shell
3. bash : change zsh to bash
4. Id : Find the details of the machine.
5. clear :To clean in
6. pwd :Present working directory
7. cd / :to go to the root directory.
8. cd folder name : Change directory. If we want to enter any folder. Exam cd Desktop/folder/
9. cd /folder1/folder2/folder3
10. cd.. :To step back one folder/parent directory.
11. cd ../folder1/folder2 : Step back from one folder then enter folder2
12. cd :To step back the whole directory. cd ~ : home directory
13. ls : To see all file and directories
14. ls -al : to see the hidden file. Or, ls -l
15. mkdir folder :To make a folder/directory
16. mkdir -p folder1/folder2/folder3 : To make more than one folder one inside another.
17. mv oldfolder newfolder : To rename a folder
18. rm -r folder name: To remove single folder
19. rm -rf folder name: To remove consecutive folders. Or, rm -r -f
20. touch filename.txt :To make a file
21. cat file name : To read a file
22. wc -m filename : To count number of words
23. How to change the permission of the file and directory?
    chmod<whom and what permission><in which file>
    Example:permission for current user:
    chmod u+x avatar.js(filename)
    chmod u-x filename (executive permission remove)
    Permission for group:
    chmod ug+x avatar.js(filename)
    :permission for other user:
    chmod ugo+wx avatar.js(filename)
    chmod ugo-wx filename (executive permission remove)

24. Permission with octal number
    Read =2
    Write=4
    Executable=1
    Current user permission of read+write+execute
    chmod 700 filename
    Group permission; chmod 710 filename

Group permission for folder: chmod 760 hello/(folder name)

25. Permission in recursive folder: chmod 760 -R foldername

Some common shell commands for getting information include:

1) whoami – which returns the user's username,
2) id – which returns the current user and group IDs,
3) Id -u – only return user id
4) Id -u -n –return user name also
5) uname – returns the operating system name,
   Uname -r –returns kernel version
   Uname -s -r —returns name and version. OR, uname -v
   Uname -a —prints all the system information in the following order: Kernel name, network node hostname, kernel release date, kernel version, machine hardware name, hardware platform, operating system.
6) ps – displays running processes and their IDs or, ps -u root
   Ps -e display all of the processes running on the system. The includes processes owned by other users.
7) top – displays running processes and resource usage including memory, CPU, and IO.If you want to exit automatically after a specified number of repetitions, use the `-n` option as follows:
   top -n 3
8) df – shows information about mounted file systems or shows disk usages,
   df=disk free
9) df -h ~ —here -h turns it human readable and ~ is for home directory Or, df -h
10) man – fetches the reference manual for any shell command
    man id
11) date – prints today's date.
    date "+%D" displays the current date as mm/dd/yy
    date “+%j day of %Y –returns days of a years
    date “+It’s %A, the %j day of %Y!”
    date -r filename –Find the last modified date

Specifier	Explanation
%d	Display the day of the month (01 to 31)
%h	Displays the abbreviated month name (Jan to Dec)
%m	Displays the month of year (01 to 12)
%Y	Displays the four-digit year
%T	Displays the time in 24 hour format as HH:MM:SS
%H	Displays the hour

Some common shell commands for working with files include:

1) cp – copy file
   To copy file : cp/source/file/dest/filename
   Or, cp/source/file/dest
   To copy directory: cp -r /source/dir/dest/dir/
   Example:
   >ls returns file folder
   >cp file folder
   >ls folder returns file
   >cp -r folder folder1
   >ls returns file folder folder1
   >ls folder1 returns file
   Question: Create a copy of display.sh called report.sh.
   cp display.sh report.sh
2) mv – change file name or path, /move file or directory
   Rename file and folder:
   mv users.txt user-info.txt –rename a file name to user-info.txt
   Move file and folder:
   >mv file folder
   >ls file —-returns empty
   >ls folder –returns file
   Moving 2 folders in 1 folder:
   > mv folder1 folder2 folder3
   >ls returns folder3
   >ls folder3 returns folder1 folder2
3) rm – remove file,
4) touch – create empty file, update file timestamp,
5) chmod – change/modify file permissions,
   File permission check: ls -l filename
   Folder permission check: ls -ld foldername

>ls -l filename returns -rw -r - -r– filename
>execute command: ./filename returns permission denied
>chmod +x filename
>ls -l filename returns -rwx -xr -x– filename
>Now execute command: ./filename
chmod -r usdoi.txt   –remove read permission for all users on the file usdoi.txt
ls -l usdoi.txt –verify the changed permission
chmod +r usdoi.txt —To add read access to all users.
chmod o-r usdoi.txt   —To remove the read permission for ‘all other users’

6) wc filename — get count of lines, words, characters in file,
   wc -l filename —get only number of line
   wc -w filename—get only number of words
   wc -c filename—get only number of characters
7) grep —return lines in file matching pattern,

Very common shell commands for navigating and working with directories include:

1) ls – lists the files and directories in the current directory,
   ls /home –display the content of /home directory
   ls Downloads –returns the download folder
   ls -l –return details of file, permissions,owner
   ls -R —Using the LS command with the –R option, you can recursively list all the directories and files in your current directory tree. You can see the correspondence with the graphical representation of the tree.

Here are some popular options that you can try with the `ls` command.

Option	**Description**
`-a`	list all files, including hidden files
`-d`	list directories only, do not include files
`-h`	with `-l` and `-s`, print sizes like 1K, 234M, 2G
`-l`	include attributes like permissions, owner, size, and last-modified date
`-S`	sort by file size, largest first
`-t`	sort by last-modified date, newest first
`-r`	reverse the sort order

   ls -la /etc –returns a long listing of all files in /etc, including any hidden files.
2) find – used to find files matching a pattern in the current directory tree,
   find . -name “filename”

find .-iname “a.text” –find insensitive file

3) pwd – prints the current, or ‘present working,’ directory,
4) mkdir – makes a new directory,
5) cd – changes the current directory to another directory,
   cd .. –return parent directory
   cd ~ –return home directory
6) rmdir – removes an entire directory, For printing file contents or strings, common commands include:
   rm folder –remove folder
   rm -r folder –remove the folder with all child file, folder
   rm -i myfile.txt — The `rm` command is used to delete files, and is ideally used along with the `-i` option, which makes it ask for confirmation before every deletion.To remove the file `myfile.txt`, enter the following command and press `y` to confirm deletion, or `n` to cancel:

For printing file contents or strings, common commands include:

1) cat filename – which prints the entire contents of a file,
2) More filename – used to print file contents one page at a time,
3) head filename– for printing just the first ‘N’ lines of a file,
   head -n 3 filename —for 3 lines
4) tail filename – for printing the last ‘N’ lines of a file,
   tail -n 3 filename –print last 3 lines or, tail -3 filename
5) echo – which 'echoes' an input string by printing it. It can also ‘echo’ the value of a variable.
   echo hello —returns hello. Echo $PATH– returns the path
   echo "Welcome to the linux lab" returns Welcome to the linux lab
   echo -e "This will be printed \nin two lines" for special character(\n. \t) we have to use -e

Shell commands related to file compression and archiving applications include:

1) tar – which is used to archive a set of files, tar:bundle=>compress
   tar -cf folder.tar folder—-create folder or file in archive
   tar -czf notes.tar.gz notes—create a compressed folder in archived.
   tar -xzf notes.tar.gz notes—-create an uncompressed/extract folder notes.tar.gz
   tar -tf notes.tar —-find the list archive contents
   tar -xf notes.tar notes—- -x tells to extract the file notes.tar from archive then check by ls -R

2) zip – which compresses a set of files, zip: compress=>bundle
   zip notes.zip notes—-compressed and archive as notes.zip
   Check it by ls.
   zip top-sites.zip top-sites.txt —Zip the file top-sites.txt into a file called top-sites.zip
3) unzip – which extracts files from a compressed or zipped archive.
   unzip notes.zip —extract and decompress. Check it by ls -R

Networking applications include the following:

1) hostname —-prints the host name
   hostname -s —drop the local domain
   hostname -i —hostname provide ip address
2) ping —-sends ICMP packets to a URL and prints the response
   ping www.google.com
   ping -c 5 www
3) ifconfig —-(Interface configuration) displays or configures network interfaces on the system,
   Ifconfig etho —provide ethernet HWaddr
4) curl – displays the contents of a file located at a URL,
   curl www.google.com
   curl www.google.com -o google.text —scraping google page's HTML to file
   Check by: head -n 1 google.txt
5) wget command can be used to download a file from a URL
   wget https://www.w3.org/TR/PNG/iso_8859-1.txt
   Check by: head -n 12 iso_8859-1.txt

Shell commands related to Viewing File content:

1) sort filename —arrange the text of the file in ascending order
   sort -r filename —-arrange the text of the file in reverse order
2) uniq filename —-drop the consecutive duplicates lines
3) grep ch filename —matching ch pattern
   grep -i ch filename—matching insensitive ch pattern

Some of the frequently used options for `grep` are:

Option	Description
`-n`	Along with the matching lines, also print the line numbers

`-c`	Get the count of matching lines
`-i`	Ignore the case of the text while matching
`-v`	Print all lines which do not contain the pattern
`-w`	Match only if the pattern matches whole words

4) cut -c 2-9 filename—-separate 2-9 characters from each line
   cut -c -2 zoo.txt —to view first two characters of each line
   cut -c 2- zoo.txt—-each line starting from the second character
   cut -d ‘ ‘ -f2 filename —-separate field 2 where delimiter space from each line
   cut -c -3 top-sites.txt —print the first 3 characters of each line from top-sites.txt
5) paste 1st.txt 2nd txt 3rd.txt—-merge line from different files
   paste -d ‘,’ 1st.txt 2nd.txt 3rd.txt —merge lines with a de
6) jjjj

Shell Scripting

1) which bash —The which command helps to find out the path of bash
2) Command1 | command 2 –For chaining filter commands. Pipe command - |
   Example : ls | sort -r
   set | head -4 —set: list all shell variable
3) var_name=value –Defining shell variables. No spaces around ‘=’
   Example: GREETING=”Hello”
   echo $GREETING
4) unset var_name —deletes var_name
   Example: unset GREETING
5) Environment Variable
   export var_name
   Example: export GREETING
   Env | grep “GREE   —env ==list all environment variable
6) $ echo "Linux and shell scripting are awesome\!" | tr "aeiou" "_" —tr replace the a,e,i,o,u by ‘_’ in the first command.
   Output: L_n_x _nd sh_ll scr_pt_ng _r_ _w_s_m_!
7) Metacharacters:
   # –precedes a comment
   ;   — command separator.example: echo “Hello”; whoami
   * —filename expansion wildcard. Example: ls /bin/ba* output: /bin/bash

? –single character wildcard in filename expansion. Example: ls /bin/?ash output: /bin/bash /bin/dash

8) Quoting:
   \ —escape special character interpretation. Exam: echo “\$1 each” output: $1 each
   “ “ – interpret literally, but evaluate metacharacters. Exam: echo “$1 each” Output: each
   ‘ ‘ –interpret literally. Example: echo ‘$1 each’ Output: $1 each
9) I/0 redirection:
   > —-redirect output to file
   Example: echo “line1” > eg.txt
   >> —Append output to file
   Echo “line2”>eg.txt
   2> —redirect standard error to file
   < —Redirect file contents to standard input
10) Cron, Crond, Crontab
    - Cron is the general name of the tool that runs scheduled jobs consisting of shell commands or shell scripts.
    - Crond is the daemon or service that interprets “crontab files” every minute and submits the corresponding jobs to cron at scheduled times.
    - A crontab, short for “cron table,” is a file containing jobs and schedule data. Crontab is also a command that invokes a text editor to allow you to edit a crontab file.
11) crontab -e —add/remove cron job with editor
12) Job syntax:
    m h dom mon dow command
    Example: 30 15 * * 0 date >> sundays.txt
13) crontab -l —to check if there any crontab or not.if not then add
14) crontab -e —to add a cron job.
    0 21 * * * echo "Welcome to cron" >> /tmp/echo.txt —at 9pm it will run
    Control + X then enter
    crontab -l —-check crontab added or not.
15) crontab -r —-removes all your cron jobs
    Then check : crontab -l
16) Problem: *Create a cron job that runs the task date >> /tmp/everymin.txt every minute.*

    crontab -e —-edit the crontab file.
    * * * * * date >> /tmp/everymin.txt —-add the line at the end of the file.
    Save the file(control + X) and quit the editor.(Y then enter)