**Linux Commands**

1. **lsblk** – To check partition summary.

-------------------------------------------------------------------------------------------

**Working with Directories**

-------------------------------------------------------------------------------------------

1. **ls -al** – To list all files, including hidden files.
2. **ls -alh** – To list all files, including hidden files. It will be in human readable format.
3. **ls -ltr** – To list files in such a way that latest files will be towards bottom.
4. **cd -** -- This will take us to last working directory.
5. **rm -rf folder\_name** – This will remove non-empty directory.
6. **mkdir -p parent/child1/child2/child3** – This is will create directories in parent-child order in one go.

-------------------------------------------------------------------------------------------

**Working with Files**

-------------------------------------------------------------------------------------------

1. **Touch file\_name** – To create an empty file.
2. **file file\_name/folder\_name** – To check file type.
3. **touch file\_name/folder\_name** – To modify any file. It will show latest date & time in modification history.
4. **touch -t yearmonthdayhourminute file\_name/folder\_name** – To modify any file/folder to old or new date & time.
5. **touch f1 f2 f3 f4 f5** – To create multiple files at once.
6. **touch f{1..10}.txt** – To create n no. of files at once.
7. **alias** – It will list all the alias.
8. **rm -rf** – Remove any file without giving any prompt.
9. **cp f1.txt f2.txt f3.txt /tmp** – Copy multiple files at once in /tmp directory.
10. **mv f1.txt f2.txt** – Use to rename a file. It is also used to cut-paste operation of a file from one location to another.

-------------------------------------------------------------------------------------------

**Working with File Contents**

-------------------------------------------------------------------------------------------

1. **cat f1.txt** – To read a file content.
2. **head /etc/passwd** – It will display first 10 lines of the passwd file content.
3. **head -20 /etc/passwd** – It will display first 20 lines of the passwd file content.....and so on.
4. **tail /etc/passwd** – It will display last 10 lines of the passwd file content.
5. **head -20 /etc/passwd** – It will display last 20 lines of the passwd file content….and so on.
6. **tac f1.txt** – It will display content of file in reverse order.
7. **cat > f1.txt** – It is used to create a file. Once content is written, press CTRL+D to save & quit.
8. **cat > f2.txt** <<stop – It is used to create a file using custom & marker sign. Once content is written, type stop & enter to save & quit.
9. **cat f1.txt > f2.txt** – Copy a file content to new file.
10. **cat >> f1.txt** – Used to add new lines, keeping previous lines as it is.
11. **echo God bless you > blessing.txt** – Used to create new file using given content.
12. **more & less** (doesn’t show percentage like more command) are used to see file content as one page at a time.
13. **strings /bin/who** – To see content of any binary files like who. It doesn’t show using cat command. Only shows junk characters.

-------------------------------------------------------------------------------------------

**Commands & Arguments**

-------------------------------------------------------------------------------------------

1. **echo $SHELL** – To check type of shell we are using.
2. **cat /etc/shells** – List the available shells in OS.
3. **echo Hello** World – Output will remove extra space between Hello & World. To keep it, we need to use single or double quotes. Example – echo “Hello World” or echo ‘Hello World’.
4. **echo -e “I Love\nMy India”** – It will print I Love in first line & My India in second line due to enabling special characters using -e.
5. **echo -e “I Love\tMy India”** – It will put space between I Love & My India due to enabling special characters using -e.
6. **type command\_name** – It will give idea about whether its shell builtin command or external (Binary).
7. **alias dog=cat** – To use cat command with different names.
8. **unalias dog** – To remove alias.

-------------------------------------------------------------------------------------------

**Control Operators in Linux**

-------------------------------------------------------------------------------------------

1. **pwd ; ls -ltr ; cal** – Use semicolon (;) to execute multiple commands serially at once.
2. **sleep 20 & pwd & cal** – Use ampersand (&) to execute multiple command parallelly at once. Starting command will go into background & execute the next one and so on. It generates process ID which can also be seen by ‘jobs’ command. Use ‘kill process ID’ to kill that background process.
3. **echo $?** – Used to check whether previous command was executed correctly or not. Zero means successful & other value means unsuccessful.
4. **pwd && ls -ltr** – This logical AND (&&) will execute both commands one by one. If previous command fails, it will halt the execution of other later command. Like, if pwd is wrong, it will not execute ls -ltr.
5. **pwd || ls -ltr** – This logical OR (||) will execute only first command if it is correct. If previous command fails, it will execute other later commands. Like, if pwd is wrong, it will not execute this, but will execute next one i.e ls -ltr. Ex – Input -> rm abhay.txt && echo File removed successfully || echo File removal failed

Output -> File removed successfully

1. **#** symbol is used to ignore the line. So it won’t give any output if any command is executed with this symbol.
2. **echo Abhay ; Singh** – It will halt the execution of characters after ; as it consider this as a command which is not. So use **\** before **;** to execute whole line. i.e **echo Abhay \; Singh**
3. **echo Hello \**

>Abhay \

>Singh!

Now output: Hello Abhay Singh!

Here \ is used to combine split lines into one.

-------------------------------------------------------------------------------------------

**Shell Variables**

-------------------------------------------------------------------------------------------

1. Few **systems defined variables** (**All Caps letter**): HOSTNAME, USER, UID, SHELL, HOME, PATH.
2. Few **user defined variables**: x=10, Abhay=100, sport=cricket etc. We call those variables using echo $variable name. It shouldn’t start with number.
3. **echo “$HOSTNAME”** – rhel8-server. echo ‘$HOSTNAME’ - $HOSTNAME. Single quote give variable name as output & double quote gives variable value as output.
4. **set** – Check list of variables.
5. **unset variable name** – To unset variable. Or variable name= will also remove variables.
6. **su** – We are gonna use environment variable of our own user account. su – username, then we are gonna use environment variable of that user account.
7. **env** – Show list of exported variables.
8. **X=10, Y=20 – echo $Y $Y** will give output as 10 20. To export variable to new shell, type **export X**. Use **bash** to go to new shell & type **echo $X $Y** & will get output as 10, not 20 as only X variable is exported in new shell. Use **exit** to go to previous shell.
9. **set -u** – It is used to set unbound variable & check whether any variable is unbound or not. Ex – echo $x will result output as ‘-bash: x: unbound variable. Cause it is not defined. Set +u will disable it.

-------------------------------------------------------------------------------------------

**Shell Embedding & Options**

-------------------------------------------------------------------------------------------

1. **echo $(x=100; echo $x)** – This will give output 100. It is defined in temporary shell. If we run echo $x, then it won’t give any value in output.

-------------------------------------------------------------------------------------------

**Shell History**

-------------------------------------------------------------------------------------------

1. **!!** – Use to execute previous command. And ! with few letters of the command will execute that previous command.
2. **history** – To see history of all commands that we executed so far in terminal. Use !serial no of command to execute it.
3. **CTRL+R** – To search & execute a command from history.
4. **echo $HISTSIZE** – Will show no. of maximum commands it will show in history.
5. **echo $HISFILE** – Show the location & name of history file. It save currently executed commands only after exiting terminal.
6. To **avoid any command to go in history, add one space before** that command.

-------------------------------------------------------------------------------------------

**File Globbing**

-------------------------------------------------------------------------------------------

1. Asterisk **(\*)** is used to find any files in directory. It will allow any no. of characters before or after \*.
2. Question mark **(?)** is used to find any files in directory with limitation. It will allow one character per ?.
3. Square bracket (**[]**) which contain some characters, it will show file name having any or all characters per square bracket. If there is multiple square brackets, then perform AND operation kind of action.

Ex- **ls File[a4], ls File[a4][c3]** . To exclude any character in file listing, use ! before that character.

1. **ls file[a-z]\*** - It will list all files starting with “file” and have any characters from a-z & then any no of characters after that. Similar for numbers as well. Ex-ls file[1-9]
2. **echo \*** - Will result in \* as output in case of empty directory, else it will shoe files or folders in it. Use echo \\* to print \* in output.

-----------------------------------------------------------------------------------------------

**Input Output Re-direction**

-----------------------------------------------------------------------------------------------

1. To print any error after executing wrong command: **zpwd 2> error.txt**. Here 2 represent error signal. 1 represent correct executions of output.
2. **set -o noclobber** -> It is used to erase the file content accidently when we are appending new data on file using redirect (>). If it is set & you still want to override it, then you can either use “>|” or set +o noclobber.
3. Use **2>&1** if redirecting output to a file & we are not sure if command is correct or not. Ex- **zpwd > test.txt 2>&1**
4. Nullifying or erasing data from a file quickly – use “**>filename**”.

-------------------------------------------------------------------------------------------

**Pipes & Commands (Filters)**

-------------------------------------------------------------------------------------------

1. **cat test.txt | cat | cat | cat | cat** – This will show same output as executed in cat test.txt, cause its output will be input for next pipe.

Or, **tac test.txt | cat | cat | cat | cat** – It will print output in reverse order.

1. **tac test.txt | tee mod.txt | tac** – It will show same output which was originally in test.txt. First one reversed it, second one has copied data in new file & third one again reversed it.
2. **grep Abhay test.txt** – It will find “Abhay” in file test.txt & will print in o/p if it is present in file. Grep -I Abhay test.txt – Here -i is used to case insensitivity (Upper or lower).
3. **grep -v Abhay test.txt** – It will exclude letter Abhay from test.txt in o/p. grep -vi Abhay test.txt – It will use for insensitivity.
4. **grep -A1 Abhay test.txt** – It will print line Abhay as well as very next line of this file. Similarly A2 for two lines, A3 for three lines & so on.
5. **grep -B1 Abhay test.txt** – It will print line Abhay as well as previous line of this file. Similarly B2 for two lines, B3 for three lines & so on.
6. **grep -C1 Abhay test.txt** – It will print line Abhay as well as one previous line & one next line of this file. Similarly C2 for two lines, C3 for three lines & so on.
7. **cut -d: -f1,3 /etc/passwd** – It will show only first & third column of passwd file. Similarly -f1,3,5 will show 1st, 3rd, 5th column. -f2-6 will do from 2nd to 5th column.
8. **cut -d” ” -f1 /etc/passwd** – It will remove space after 1st column.
9. **cut -c2-7 /etc/passwd** – Will show only first 2-7 characters of each column in the file.
10. **cut -c2-7 /etc/passwd | tail -5** – Will print only last five lines of this file.
11. **cat test.txt | tr ‘c’ ‘C’** – It will translate all small case c to capital C in entire file. Cat test.txt | tr ‘a-z’ ‘A-Z’ – It will make all caps. Cat test.txt | tr ‘t’ ‘x’ – Translate all t to x.
12. **cat test.txt | tr ‘a-z’ ‘fgrughrg’** – To encrypt file content.
13. **cat test.txt | tr -d a** – It will remove alphabet ‘a’ in printed o/p.
14. **wc f1.txt** – Gives total count of lines, words & characters count in this file. Wc -l f1.txt will give line count alone, wc -c f1.txt gives word count alone & wc -c f1.txt gives character count alone.
15. **cat f1.txt | wc**
16. **sort f1.txt** – List the content of file in alphabetical order.
17. **sort -k1 f1.txt** – Sort the content as per first column. Similarly -k2 for second column & so on.
18. **uniq** – Will remove duplicate entry while listing content of a file. It must be used with sort command. Ex – **sort duplicate.txt | uniq**
19. **sort duplicate.txt | uniq -c** – Total count of duplicate entries.
20. **comm f1.txt f2.txt** – compare two files content.
21. **od -t x1 f1.txt** – Print content in hexadecimal format.
22. **od -b f1.txt** - Print content in octal format
23. **od -c f1.txt** - Print content in ASCII format
24. **cat f1.txt | sed 's/x/y/'** – Using sed command, we can replace characters or words with other characters or words.
25. **echo Abhay Abhay | sed ‘s/Abhay/Singh/g’** – It will replace all **Abhay** with **Singh** globally after using **g** at the end, else it will replace first one only.
26. **cat f1.txt | sed '/x/d/'** – It will remove all words in output which are starting with **x**.
27. **who | wc -l** – To display no. of users logged in.
28. **grep bash /etc/passwd** – Will list users who are using bash shell in this system.
29. **who | cut -d' ' -f1 | sort | uniq** – Will represent different users logged in currently.

-------------------------------------------------------------------------------------------

**Basic UNIX Tools**

-------------------------------------------------------------------------------------------

1. **find /etc > etcfile.txt** – Copy all files of /etc directory into text file.
2. **find /etc -type f -name "\*.conf"** – Find all files in etc directory with .conf extension.
3. **find . -newer duplicate.txt** – Find all new files created after duplicate.txt files in present working directory.
4. **locate filename** – To find a file in entire system. We need to run **updated** command to update index as **locate** command works on index which stay outdated if we are creating new files. It is faster than **find** command.
5. **timedatectl** – More detail about date & time. Time command is used to find time taken for execution of any command. Ex – **time find /etc**
6. **sleep 60** – It will wait for 60 second before accepting any input.

-------------------------------------------------------------------------------------------

**File Compression, Decompression, Zipping & Unzipping**

-------------------------------------------------------------------------------------------

1. **gzip f1/txt** – Used to zip a file with **.gz** extension. To see content of this zip file, use **zcat** in place of cat, use **zmore** or **zless** in place of more or less.
2. **gunzip f1.txt** – To unzip .gz files.
3. **Bzip2 f1.txt** - Used to zip a file with .bz2 extension. To see content of this zip file, use **bzcat** in place of cat, use **bzmore** or **bzless** in place of more or less.
4. **bunzipz f1.txt** – To unzip .bz2 files.
5. **tar cf test.tar test1.txt test2.txt** – To make tar file of two or more files.
6. **tar cf test.zip test1.txt test2.txt** – To make zip file of two or more files.
7. **tar czf test.tar.gz test1.txt test2.txt** – To make compressed tar file of two or more files with **gz compression**.
8. **tar cjf test.tar.gz2 test1.txt test2.txt** – To make compressed tar file of two or more files with **gz2 compression**.
9. **tar cjpf test.tar.gz2 test1.txt test2.txt** - To make compressed tar file of two or more files with gz2 compression while **keeping files permission preserve.**
10. **tar tvf compressed\_filename** – Used to see content of compressed files. Ex – tar tvf test.zip
11. **tar tvf compressed\_filename text\_file** – To check whether a file is present inside compressed file or not. Ex- tar tvf test.zip test1.txt
12. **tar xvf compressed\_filename** – Used to unzip compressed file using tar.

-------------------------------------------------------------------------------------------

**Regular Expressions** -------------------------------------------------------------------------------------------