**Linux Fundamentals**

**Option are always used with hyphen(-)**

**To make new directory(folder):**

mkdir EI\_demo(to create folder)

cat -n temp.txt(to display the file)

ls (list to display content)

man ls(possible commands similar to help)// q: to exit

cd(change directory)- cd E <and double tab>// shows all the directories with E

pwd(to check current directory)

touch demo1.txt(We can create new file using touch, nano, gedit,vi,cat)

cd ..(to get back to previous parent directory)

cd ~ (home directory)

cd / <back>

$ touch f1.txt f2.txt f3.txt f4.txt

Ls -la

Edit content in a file

Vi <filename> creates file if not already created.

After pressing ‘i’ we can add content and esc :wq to save the content.(first esc and :wq and end of line)

Cat <filename> to open file

Rm <filename> to delete files

Rmdir – delete only empty directories

Rm -r p1 (if you want to delete empty directory – rmdir and non empty rm -r)

Cat – top down

Tac- bottom up

Date: to print date and time

Useradd user1 (to add new user to be performed in root with #)

Passwd (apply to existing user)user1 and then enter.

Now add password and then to confirm password.

To switch user use su user1(current user is user1)

After typing command by mistake type Control + z to stop the process.

In case control+z doesn’t work use control+c

More(to open file) command(spacebar- page down , b-page up , enter- line by line down)

Less command// find from home

Wc <filename> file shows a b c//a- no of lines, b- no of words, c- no of characters

Wc -l <filename> to count no of lines

Wc -w<filename> to count no of words

Wc -m <filename> to count no of characters

$ head <filename> - return 10 lines from top

$ tail <filename> - returns 10 lines from bottom//by default head and tail return 10 lines

$ head -5 <filename>- returns first 5 lines

$ head -n 5 <filename> returns first 5 lines

$ tail -n 3 <filename> return last 3 lines

$ cp <filename>/demo2.txt <filename> demo3.txt(if file is not there, it creates it)

$ mv(to move and rename) <existing filename> <new filename>

$ mv <filename>with path if file is at diff location source path(if it is in current dir the {.})

$ mv ~/EI\_demo/demo31.txt ~/ - move to home dir

$mv EI\_demo.txt EI\_demo/

$ mv EI\_demo/demo31.txt . (other to current)

!155(runs the number mentioned command.)

**Shortcuts for CLI:**

Control + A - jump to the beginning of command line

Control + E – jump to the end of command line

Control + U – clear from the cursor to the beginning of the command line

Control + k – clear from the cursor to the end of the command line

Control + leftarrow – jump to the beginning of the previous word of command line

Control + rightarrow – jump to the end of the next word on command line

Control + R – search the history list of commands for a pattern.

**7/2/2023**

To switch alternate directory

Cd –

To create multiple files together

Touch temp{1..5}.txt

To create multiple dir as well

Mkdir demo{a.c}

Touch temp{a,b}{1,2}.txt//to create tempa1.txt,tempa2.txt etc

To create parent and child dir using single command.

Mkdir -p demo/{music,videos,pictures}

Friends// cp ~/demo/music/song{1..2}.mp3

Recursively delete dir

rm -R{family,friends}

ls [t]\*//shows all the file in dir starting with t

Pattern matching

Ls s\* t\*// shows the files starting with s and t

{}- takes input as a string

[]- takes input as character

Dynamic display:

Echo today is $(date +%A)// today is Tuesday

To substitute blank lines:

After editing press- :%s/\n\n\*/\r

To delete single space in vi using file, type ‘dd’ in visual mode

TO dynamically link two files:

Ln xyz,txt sl\_xyz.txt(change in any file will reflect in both files)

Ls -ila (it shows unique id of all files and linked files have same inode number)

We can link file on one dir to file with another directory

Hard link not allowed for directories..

Ln -s <source> <destination>.//soft link

Ln <source><destn>

In hard link, if we delete child/parent the other one is accessible

In soft link, if we delete source file, then child will not be accessible and if we delete the child then the source file will be accessible.

Echo Hello> file.txt//output is stored in the file

Echo hello>>file.txt// added in bottom prevents overwriting.

Echo hello 2> file.txt //redirect error in file.

To store error not in file and terminal

Ls a\*t\* 2> /dev/null

&s- to redirect the output in file discards an error

Comment &> filename (for output) //&>> for append

**Difference between ‘<’ and ‘>’ and syntax of ‘<’**

**8.2.23**

Pipeline commands:

It combines two commands at the same time. In this, the output of 1st command is input of second and the output of 2nd command will be displayed to the terminal.

Eg head -5 file.txt | wc

2. To display in terminal as well as to input in the file

We can use ‘tee’ command.

Eg. Head -5 file.txt | wc|tee <new file>

3. 1. how to select, delete content in **visual mode**

First: ls -la

Then shift+v – to select the content in row wise

X- to delete the selected row

U- undo content

3.2 In column wise

Ctrl +v – to select in column wise.

4 :q!- to exit a file without saving anything.

To fetch the input from file we use:

‘<’

5. To display a content with matching regular expression from a file we use ‘grep’

Eg. Grep “Temp” <filename>

And using pipe—cat<filename> | grep “temp”

And to ignore the case, we use’-i’ eg: grep -i “temp” <filename>

And ‘-v’ shows inverse of output(except matching)

Grep -A 1 “temp” <filename>// returns 1 line after the temp matching line

Grep -B 1 “temp”<filename>// returns 1 lines before the temp matching file

Grep -C num “temp”<filename>// return both after and before.

6. Sed:

Used to replace something with new keyword.

Echo hello | sed ‘s/hello/yo/’—instead of hello, yo will be displayed on terminal.

-It will not store the changes in the files.