Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 1

File and Directory Related Commands

THEORY:

File Related Commands:

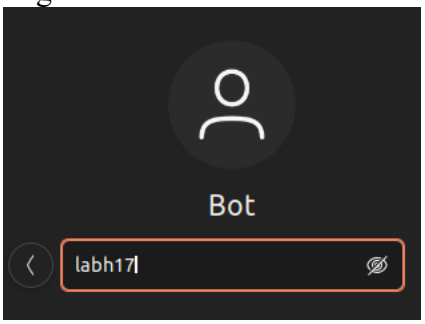
Command	Syntax	Description
touch	touch **<file name>**	**Used to create empty files.**
cat	cat > **<file name>**	**It is a multi-purpose utility. It can be used to create a file, display content of the file, copy the content of one file to another file, etc.**
rm	rm **<file name>**	**Used to remove a file.**
cp	cp **<existing file name> <new file name>**	**Used to copy a file or directory.**
mv	mv **<file name> <directory path>**	**Used to move a file or a directory form one location to another location.**

Directory Related Commands:

Command	Syntax	Description
pwd	pwd	**Used to display the location of the current working directory.**
mkdir	mkdir **<directory name>**	**Used to create a new directory under any directory.**
rmdir	rmdir **<directory name>**	**Used to delete a directory.**
ls	ls	**Used to display a list of content of a directory.**
cd	cd **<directory name>**	**Used to change the current directory.**

PRACTICAL:

1. Login as "Bot" user.



2. Create five empty files with the name a1, b2, c3, d4 and e5 in current working directory.
   tycs69@tycs:~$ touch a1 b2 c3 d4 e5

3. **Create three files with the name f6, g7, h8 using cat command with some meaningful contents with at least five lines each.**
   tycs69@tycs:~$ touch f6 g7 h8

   tycs69@tycs:~$ cat >f6

   line1

   line2

   line3

   line4

   line5

   tycs69@tycs:~$ cat >g7

   line uno

   line dos

   line tres

   line cuatro

   line cinco

   tycs69@tycs:~$ cat >h8

   line un

   line des

   line trois

   line quatra

   line cinq

4. **Display the contents of files f6, g7 and h8.**
   tycs69@tycs:~$ cat f6 g7 h8

   line1

   line2

   line3

   line4

   line5

   line uno

   line dos

line tres

line cuatro

line cinco

line un

line des

line trois

line quatra

line cinq

5. **Display the contents of files f6, g7 and h8 along with line numbers.**
   tycs69@tycs:~$ cat -n f6 g7 h8

```
     1  line1
     2  line2
     3  line3
     4  line4
     5  line5
     6  line uno
     7  line dos
     8  line tres
     9  line cuatro
    10  line cinco
    11  line un
    12  line des
    13  line trois
    14  line quatra
    15  line cinq
```

6. **Copy contents of f6 to a1, g7 to b2, h8 to c3.**
   tycs69@tycs:~$ cp f6 a1

   tycs69@tycs:~$ cp g7 b2

   tycs69@tycs:~$ cp h8 c3

7. **Add the contents of a1, b2 and c3 into "testfile".**
   tycs69@tycs:~$ cat a1 b2 c3 > testfile

8. Create three empty directories with names dd1, dd2 and dd3 in current working directory.
   tycs69@tycs:~$ mkdir dd1 dd2 dd3
   tycs69@tycs:~$ ls
   a1 c3 dd1 dd3 Documents e4 g7 Music Public Templates Videos
   b2 d4 dd2 Desktop Downloads f6 h8 Pictures snap testfile

9. **Copy the files a1 and b2 to the directories dd1.**
   tycs69@tycs:~$ cp a1 b2 dd1

10. **Copy the files f6 and g7 to the directory dd2.**
    tycs69@tycs:~$ cp f6 g7 dd2

11. **Display the contents of directory dd1 and dd2.**
    tycs69@tycs:~$ ls dd1 dd2

    dd1:

    a1 b2

    dd2:

    f6 g7

12. **Copy directory dd1 into dd3.**
    tycs69@tycs:~$ cp -r dd1 dd3

13. **Display contents of directory dd3.**
    tycs69@tycs:~$ ls dd3

    dd1

14. **Copy directory dd2 into dd4.**
    tycs69@tycs:~$ cp -r dd2 dd4

15. **Display contents of directory dd4.**
    tycs69@tycs:~$ ls dd4

    f6 g7

16. **Remove the directory dd2 along with its contents.**
    tycs69@tycs:~$ rm -r dd2

    tycs69@tycs:~$ ls

a1 c3 dd1 dd4 Documents e4 g7 Music Public Templates Videos

b2 d4 dd3 Desktop Downloads f6 h8 Pictures snap testfile

17. Rename the file a1 by newa1 and file b2 by newb2.
tycs69@tycs:~$ mv a1 newa1

tycs69@tycs:~$ mv b2 newb2

tycs69@tycs:~$ ls

c3 dd1 dd4 Documents e4 g7 Music newb2 Public Templates Videos

d4 dd3 Desktop Downloads f6 h8 newa1 Pictures snap testfile

18. Rename files d4 and e5 in directory dd4.
tycs69@tycs:~$ mv d4 e5 dd4

19. Display contents of directory dd4.
tycs69@tycs:~$ ls dd4

d4 e5 f6 g7

20. Rename directory dd4 with dd5.
tycs69@tycs:~$ mv dd4 dd5

21. Display contents of directory dd5.
tycs69@tycs:~$ ls dd5

d4 e5 f6 g7

22. Rename directory dd3 into dd5.
tycs69@tycs:~$ mv dd3 dd5

23. Display contents of directory dd5.
tycs69@tycs:~$ ls dd5

d4 dd3 e5 f6 g7

24. Create a directory tree d1/d2/d3 in one command.
tycs69@tycs:~$ mkdir -p d1/d2/d3

25. Change to directory tree d1/d2/d3 in one command.
tycs69@tycs:~$ cd d1/d2/d3

26. Display the present working directory.
tycs69@tycs:~/d1/d2/d3$

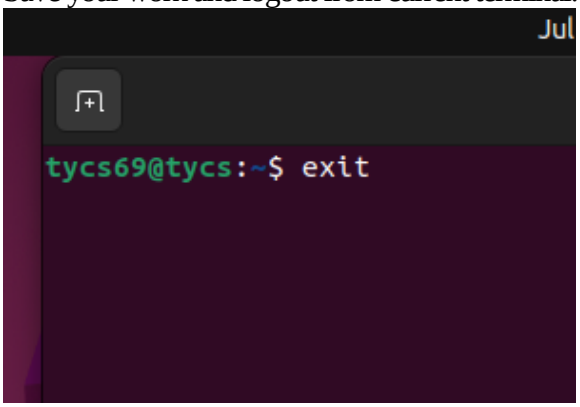
27. Create files x1, x2 and x3 with some contents in this directory.
tycs69@tycs:~/d1/d2/d3$ touch x1 x2 x3
tycs69@tycs:~/d1/d2/d3$ cat >x1

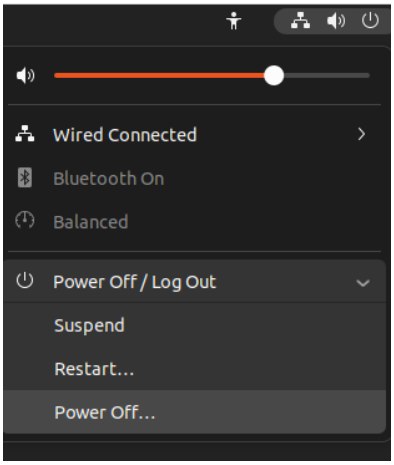
hello i am a cat
tycs69@tycs:~/d1/d2/d3$ cat >x2
hello i am a kangaroo
tycs69@tycs:~/d1/d2/d3$ cat >x3
hello i am a dinasaur

28. Copy file x3 into directory d2.
    tycs69@tycs:~/d1/d2/d3$ cp x3 ..
29. Display the contents of directory d2.
    tycs69@tycs:~/d1/d2/d3$ cd ..
    tycs69@tycs:~/d1/d2$ ls
    d3 x3
30. Rename file x1 into directory d1.
    tycs69@tycs:~/d1/d2/d3$ mv x1 ../..
31. Display the contents of directory d1.
    tycs69@tycs:~/d1/d2/d3$ ls ../..
    d2 x1
32. Change from current working directory to home directory in one command.
    tycs69@tycs:~/d1/d2/d3$ cd ../../..
33. Remove the directory tree d1/d2/d3 in one command. Is directory removed? If no, Why?
    NO Because Directory is NOT empty

34. Remove your all files and directories.
    tycs69@tycs:~$ rm -r *

35. Save your work and logout from current terminal.





Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 2

ls Command

THEORY:

The **ls** is the list command in Linux. It will show the full list or content of your directory. Just type ls and press the enter key. The whole content will be shown.

ls commands:

Ls options	Description
ls -a	Lists all files including hidden file starting with '.'
ls -A	Lists all executable files and not hidden ones.
ls -l	It will show the list in a long list format.
ls -r	Display in reverse alphabetical order.
ls -R	Lists files recursively and its subdirectories.
ls -lh	This command will show you the file sizes in human readable format(in terms of Mb, Gb, Tb, etc.).
ls -lhS	Displays files in descending order.
ls -g or ls -lG	It excludes column of group information and owner.
ls -n	Display group and user using id.
ls -li	Displaying Inode number of file or directory. (Inode : location where file/dir is saved)
ls -lx	It sorts by extension
ls ~	It gives the contents of home directory.
ls --version	It checks the version of ls command.
ls -lu	It sorts based on access time
ls -c	It sorts by ctime, newest first
ls -lt	It sort by time & date
ls ../	It give the contents of parent directory.
ls -C	Lists entries by columns
ls *	Lists all subdirectories:

PRACTICAL:

Creating Files:

labh@labh-virtual-machine:~$ mkdir dd1 dd2 dd3

labh@labh-virtual-machine:~$ cd dd1

labh@labh-virtual-machine:~/dd1$ touch Abc Axy Ak owo uwu awe axe abc add Oa1 pab bcd

labh@labh-virtual-machine:~/dd1$ cd ../

labh@labh-virtual-machine:~$ cd dd2

labh@labh-virtual-machine:~/dd2$ touch wow mom dad mad pasd pa99 agb5 atb9 Ak5 K78 j77 o78

labh@labh-virtual-machine:~/dd2$ cd ../

labh@labh-virtual-machine:~$ cd dd3

labh@labh-virtual-machine:~/dd3$ touch aa bb cc dd 8a a8 eg5ui du6pg zg5fj demo example labh@labh-virtual-machine:~/dd3$ touch aa bb cc dd 8a a8 eg5ui du6pg zg5fj demo example test

labh@labh-virtual-machine:~/dd3$ cd ../

1. **List all files and directories.**

   labh@labh-virtual-machine:~$ ls

   dd1 dd2 dd3 Desktop dir1 Documents Downloads Music Pictures Public snap Templates Videos

   labh@labh-virtual-machine:~$ ls dd1 dd2 dd3

   dd1:

   abc Abc add Ak awe axe Axy bcd Oa1 owo pab uwu

   dd2:

   agb5 Ak5 atb9 dad j77 K78 mad mom o78 pa99 pasd wow

   dd3:

8a a8 aa bb cc dd demo du6pg eg5ui example test zg5fj

2. List all files and directories in multicolumn format.

labh@labh:~$ls -C

labh@labh-virtual-machine:~$ ls -C

8a aa Abc agb5 Ak5 awe Axy bcd dad dd1 dd3 Desktop Documents du6pg example K78 mom o78 owo pab Pictures snap test uwu wow

a8 abc add Ak atb9 axe bb cc dd dd2 demo dir1 Downloads eg5ui j77 mad Music Oa1 pa99 pasd Public Templates testdemo Videos zg5fj

3. List all files identifying directories and executable files.

labh@labh-virtual-machine:~$ ls -nm

8a, a8, aa, abc, Abc, add, agb5, Ak, Ak5, atb9, awe, axe, Axy, bb, bcd, cc, dad, dd, dd1, dd2, dd3, demo, Desktop, dir1, Documents, Downloads, du6pg, eg5ui, example, j77, K78, mad, mom, Music, o78, Oa1,

owo, pa99, pab, pasd, Pictures, Public, snap, Templates, test, testdemo, uwu, Videos, wow, zg5fj

4. List the contents of the directory along with all hidden files.

labh@labh-virtual-machine:~$ ls -a

. .bash_history .bashrc .config dd2 Desktop Documents .gnupg .local Pictures Public .ssh Templates

.. .bash_logout .cache dd1 dd3 dir1 Downloads .lesshst Music .profile snap .sudo_as_admin_successful Videos

5. List recursive list of all files and directories.

labh@labh-virtual-machine:~$ ls -R

.:

dd1 dd2 dd3 Desktop dir1 Documents Downloads Music Pictures Public snap Templates Videos

./dd1:

abc Abc add Ak awe axe Axy bcd Oa1 owo pab uwu

./dd2:

agb5 Ak5 atb9 dad j77 K78 mad mom o78 pa99 pasd wow

./dd3:

8a a8 aa bb cc dd demo du6pg eg5ui example test zg5fj

./Desktop:

./dir1:

./Documents:

./Downloads:

./Music:

./Pictures:

./Public:

./snap:

firefox snapd-desktop-integration snap-store

./snap/firefox:

1635 1670 common current

./snap/firefox/1635:

./snap/firefox/1670:

./snap/firefox/common:

./snap/snapd-desktop-integration:

10 14 common current

./snap/snapd-desktop-integration/10:

./snap/snapd-desktop-integration/14:

./snap/snapd-desktop-integration/common:

./snap/snap-store:

575 common current

./snap/snap-store/575:

./snap/snap-store/common:

./Templates:

./Videos:

6. Give the file listing in reverse order.

labh@labh-virtual-machine:~$ ls dd1 -r

uwu pab owo Oa1 bcd Axy axe awe Ak add Abc abc

labh@labh-virtual-machine:~$ ls dd2 -r

wow pasd pa99 o78 mom mad K78 j77 dad atb9 Ak5 agb5

labh@labh-virtual-machine:~$ ls dd3 -r

zg5fj test example eg5ui du6pg demo dd cc bb aa a8 8a

7. Give long listing of all files and directories.
(List all files and directories with their attributes and file permissions.)

labh@labh-virtual-machine:~$ ls -o

total 64

-rw-rw-r-- 1 labh 0 Aug 12 15:02 8a

-rw-rw-r-- 1 labh 0 Aug 12 15:02 a8

-rw-rw-r-- 1 labh 0 Aug 12 15:02 aa

-rw-rw-r-- 1 labh 0 Aug 12 15:02 abc

-rw-rw-r-- 1 labh 0 Aug 12 15:02 Abc

-rw-rw-r-- 1 labh 0 Aug 12 15:02 add

-rw-rw-r-- 1 labh 0 Aug 12 15:02 agb5

-rw-rw-r-- 1 labh 0 Aug 12 15:02 Ak

-rw-rw-r-- 1 labh 0 Aug 12 15:02 Ak5

-rw-rw-r-- 1 labh 0 Aug 12 15:02 atb9

-rw-rw-r-- 1 labh 0 Aug 12 15:02 awe

-rw-rw-r-- 1 labh 0 Aug 12 15:02 axe

-rw-rw-r-- 1 labh 0 Aug 12 15:02 Axy

-rw-rw-r-- 1 labh 0 Aug 12 15:02 bb

-rw-rw-r-- 1 labh 0 Aug 12 15:02 bcd

-rw-rw-r-- 1 labh 0 Aug 12 15:02 cc

-rw-rw-r-- 1 labh 0 Aug 12 15:02 dad

-rw-rw-r-- 1 labh 0 Aug 12 15:02 dd

drwxrwxr-x 2 labh 4096 Aug 12 15:00 dd1

drwxrwxr-x 2 labh 4096 Aug 12 15:00 dd2

drwxrwxr-x 2 labh 4096 Aug 12 15:01 dd3

-rw-rw-r-- 1 labh 10 Aug 12 20:49 demo

drwxr-xr-x 2 labh 4096 Aug 7 15:59 Desktop

drwxrwxr-x 2 labh 4096 Aug 9 18:13 dir1

drwxr-xr-x 2 labh 4096 Aug 7 15:59 Documents

drwxr-xr-x 2 labh 4096 Aug 7 15:59 Downloads

-rw-rw-r-- 1 labh   0 Aug 12 15:02 du6pg

-rw-rw-r-- 1 labh   0 Aug 12 15:02 eg5ui

-rw-rw-r-- 1 labh   12 Aug 12 20:49 example

-rw-rw-r-- 1 labh   0 Aug 12 15:02 j77

-rw-rw-r-- 1 labh   0 Aug 12 15:02 K78

-rw-rw-r-- 1 labh   0 Aug 12 15:02 mad

-rw-rw-r-- 1 labh   0 Aug 12 15:02 mom

drwxr-xr-x 2 labh 4096 Aug 7 15:59 Music

-rw-rw-r-- 1 labh   0 Aug 12 15:02 o78

-rw-rw-r-- 1 labh   0 Aug 12 15:02 Oa1

-rw-rw-r-- 1 labh   0 Aug 12 15:02 owo

-rw-rw-r-- 1 labh   0 Aug 12 15:02 pa99

-rw-rw-r-- 1 labh   0 Aug 12 15:02 pab

-rw-rw-r-- 1 labh   0 Aug 12 15:02 pasd

drwxr-xr-x 2 labh 4096 Aug 7 15:59 Pictures

drwxr-xr-x 2 labh 4096 Aug 7 15:59 Public

drwx------ 5 labh 4096 Aug 8 22:25 snap

drwxr-xr-x 2 labh 4096 Aug 7 15:59 Templates

-rw-rw-r-- 1 labh   18 Aug 12 20:49 test

-rw-rw-r-- 1 labh   0 Aug 12 21:15 testdemo

-rw-rw-r-- 1 labh   0 Aug 12 15:02 uwu

drwxr-xr-x 2 labh 4096 Aug 7 15:59 Videos

-rw-rw-r-- 1 labh   0 Aug 12 15:02 wow

-rw-rw-r-- 1 labh   0 Aug 12 15:02 zg5fj

8. List all filenames sorted by last modification time.

labh@labh-virtual-machine:~$ ls dd1 -lt

total 0

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 bcd

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 Oa1

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 pab

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 abc

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 Abc

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 add

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 Ak

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 awe

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 axe

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 Axy

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 owo

-rw-rw-r-- 1 labh labh 0 Aug 12 13:19 uwu

labh@labh-virtual-machine:~$ ls dd2 -lt

total 0

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 Ak5

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 j77

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 K78

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 o78

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 agb5

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 atb9

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 dad

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 mad

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 mom

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 pa99

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 pasd

-rw-rw-r-- 1 labh labh 0 Aug 12 13:21 wow

labh@labh-virtual-machine:~$ ls dd3 -lt

total 0

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 8a

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 a8

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 aa

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 bb

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 cc

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 dd

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 demo

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 du6pg

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 eg5ui

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 example

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 test

-rw-rw-r-- 1 labh labh 0 Aug 12 13:23 zg5fj

9. Give listing sorted by last modification time.

labh@labh-virtual-machine:~$ ls --sort time

testdemo example 8a aa Ak5 bb dd eg5ui K78 o78 pasd abc add awe Axy dad Oa1 pab wow dd2 dir1 Desktop Downloads Pictures Templates

test demo a8 agb5 atb9 cc du6pg j77 mad pa99 zg5fj Abc Ak axe bcd mom owo uwu dd3 dd1 snap Documents Music Public Videos

10. Displays all files sorted according to the extension.

labh@labh-virtual-machine:~$ ls dd1 -x

abc Abc add Ak awe axe Axy bcd Oa1 owo pab uwu

labh@labh-virtual-machine:~$ ls dd2 -x

agb5 Ak5 atb9 dad j77 K78 mad mom o78 pa99 pasd wow

labh@labh-virtual-machine:~$ ls dd3 -x

8a a8 aa bb cc dd demo du6pg eg5ui example test zg5fj

11. Displays all files sorted according to the extension in reverse order.

labh@labh-virtual-machine:~$ ls dd1 -x -r

uwu pab owo Oa1 bcd Axy axe awe Ak add Abc abc

labh@labh-virtual-machine:~$ ls dd2 -x -r

wow pasd pa99 o78 mom mad K78 j77 dad atb9 Ak5 agb5

labh@labh-virtual-machine:~$ ls dd3 -x -r

zg5fj test example eg5ui du6pg demo dd cc bb aa a8 8a

12. Displays all files sorted according to the last modification time in reverse order.

labh@labh-virtual-machine:~$ ls dd1 -t -r

uwu owo Axy axe awe Ak add Abc abc pab Oa1 bcd

labh@labh-virtual-machine:~$ ls dd2 -t -r

wow pasd pa99 mom mad dad atb9 agb5 o78 K78 j77 Ak5

labh@labh-virtual-machine:~$ ls dd3 -t -r

zg5fj test example eg5ui du6pg demo dd cc bb aa a8 8a

13. Display all files and directories including hidden files and executable files.(F)

labh@labh-virtual-machine:~$ ls -F

dd1/ dd2/ dd3/ Desktop/ dir1/ Documents/ Downloads/ Music/ Pictures/ Public/ snap/ Templates/ Videos/

14. Display detailed listing including hidden files sorted in reverse order.

Videos .sudo_as_admin_successful snap .profile Music .lesshst Downloads dir1 dd3 dd1 .cache .bash_logout ..

Templates .ssh Public Pictures .local .gnupg Documents Desktop dd2 .config .bashrc .bash_history .

15. List the contents of the directory “dd1”.

labh@labh-virtual-machine:~$ ls dd1

abc Abc add Ak awe axe Axy bcd Oa1 owo pab uwu

16. Give listing of “dd1” with their attributes and file permissions.

labh@labh:~$ls dd1 -lt

total 0

-rw-rw-r-- 1 labh labh 0 Jul 29 12:57 a4k

-rw-rw-r-- 1 labh labh 0 Jul 29 12:57 B5m

-rw-rw-r-- 1 labh labh 0 Jul 29 12:57 demo

-rw-rw-r-- 1 labh labh 0 Jul 29 12:57 Ow1

-rw-rw-r-- 1 labh labh 0 Jul 29 12:44 a

-rw-rw-r-- 1 labh labh 0 Jul 29 12:44 b

-rw-rw-r-- 1 labh labh 0 Jul 29 12:44 ccc

-rw-rw-r-- 1 labh labh 0 Jul 29 12:44 d

-rw-rw-r-- 1 labh labh 0 Jul 29 12:44 ef

17. List all filenames with one screen at a time.

labh@labh-virtual-machine:~$ ls |more

8a

a8

aa

abc

Abc

add

agb5

Ak

Ak5

atb9

awe

axe

Axy

bb

bcd

cc

dad

dd

dd1

dd2

dd3

demo

Desktop

dir1

Documents

Downloads

du6pg

eg5ui

example

j77

K78

mad

mom

Music

o78

Oa1

owo

pa99

pab

pasd

Pictures

Public

snap

Templates

test

testdemo

uwu

--More--

18. List all filenames with 2 characters.

labh@labh-virtual-machine:~$ cd dd1

labh@labh-virtual-machine:~/dd1$ ls [a-z][a-z]

ls: cannot access '[a-z][a-z]': No such file or directory

labh@labh-virtual-machine:~/dd1$ cd ../

labh@labh-virtual-machine:~$ cd dd2

labh@labh-virtual-machine:~/dd2$ ls [a-z][a-z]

ls: cannot access '[a-z][a-z]': No such file or directory

labh@labh-virtual-machine:~/dd2$ cd ../

labh@labh-virtual-machine:~$ cd dd3

labh@labh-virtual-machine:~/dd3$ ls [a-z][a-z]

aa bb cc dd

19. List all filenames with 3 characters.

labh@labh-virtual-machine:~$ cd dd1

labh@labh-virtual-machine:~/dd1$ ls [a-z][a-z][a-z]

abc add awe axe bcd owo pab uwu

labh@labh-virtual-machine:~/dd1$ cd ../

labh@labh-virtual-machine:~$ cd dd2

labh@labh-virtual-machine:~/dd2$ ls [a-z][a-z][a-z]

dad mad mom wow

labh@labh-virtual-machine:~/dd2$ cd ../

labh@labh-virtual-machine:~$ cd dd3

labh@labh-virtual-machine:~/dd3$ ls [a-z][a-z][a-z]

ls: cannot access '[a-z][a-z][a-z]': No such file or directory

20. List all filenames starting with a lowercase letter ‘a’.

labh@labh-virtual-machine:~$ cd dd1

labh@labh-virtual-machine:~/dd1$ ls a*

abc add awe axe

labh@labh-virtual-machine:~/dd1$ cd ../

labh@labh-virtual-machine:~$ cd dd2

labh@labh-virtual-machine:~/dd2$ ls a*

agb5 atb9

labh@labh-virtual-machine:~/dd2$ cd ../

labh@labh-virtual-machine:~$ cd dd3

labh@labh-virtual-machine:~/dd3$ ls a*

a8 aa

labh@labh-virtual-machine:~/dd3$ cd ../

21. List all filenames starting with an uppercase letter ‘A’.

labh@labh-virtual-machine:~$ cd dd1

labh@labh-virtual-machine:~/dd1$ ls A*

Abc Ak Axy

labh@labh-virtual-machine:~/dd1$ cd ../

labh@labh-virtual-machine:~$ cd dd2

labh@labh-virtual-machine:~/dd2$ ls A*

Ak5

labh@labh-virtual-machine:~/dd2$ cd ../

labh@labh-virtual-machine:~$ cd dd3

labh@labh-virtual-machine:~/dd3$ ls A*

ls: cannot access 'A*': No such file or directory

labh@labh-virtual-machine:~/dd3$ cd ../

22. List all filenames starting with vowel.

labh@labh-virtual-machine:~/dd1$ rm -r Abc Axy Ak owo uwu awe axe abc add Oa1 pab bcd

labh@labh-virtual-machine:~/dd1$ cd ../

labh@labh-virtual-machine:~$ cd dd2

labh@labh-virtual-machine:~/dd2$ rm -r wow mom dad mad pasd pa99 agb5 atb9 Ak5 K78 j77 o78

labh@labh-virtual-machine:~/dd2$ cd ../

labh@labh-virtual-machine:~$ cd dd3

labh@labh-virtual-machine:~/dd3$ rm -r aa bb cc dd 8a a8 eg5ui du6pg zg5fj demo example test

labh@labh-virtual-machine:~/dd3$ cd ../

labh@labh-virtual-machine:~$ touch Abc Axy Ak owo uwu awe axe abc add Oa1 pab bcd wow mom dad mad pasd pa99 agb5 atb9 Ak5 K78 j77 o78 aa bb cc dd 8a a8 eg5ui du6pg zg5fj demo example test

---

23. List all filenames with the last character as ‘a’ or ‘b’ or ‘c’ or ‘d’.

labh@labh-virtual-machine:~$ ls *[abcd]

8a aa abc Abc add bb bcd cc dad dd mad pab pasd

24. List all filenames with exactly three characters in which the second character is a vowel.

labh@labh-virtual-machine:~$ ls [a-z][aeiou][a-z]

dad mad mom pab wow

25. List all filenames with three characters in which first character is an uppercase letter and second character is a digit.

labh@labh-virtual-machine:~$ ls [A-Z][0-9][a,1-z,9]

K78

26. List all filenames starting with an uppercase letter vowel and ends with a digit.

labh@labh-virtual-machine:~$ ls [AEIOU]*[1-9]

Ak5 Oa1

27. List all filenames in which last two characters is a digit in ranger 1 to 9.

labh@labh-virtual-machine:~$ ls *[1-9][1-9]

j77 K78 o78 pa99

28. List all filenames with four characters in which first character is 'a' and third character is 'b'.

labh@labh-virtual-machine:~$ ls a*[a,1-z,9]b*[a,1-z,9]

agb5 atb9

29. List all filenames whose first character is any thing other than a small case letter vowel.

labh@labh-virtual-machine:~$ ls [!aeiou]*

8a Abc Ak Ak5 Axy bb bcd cc dad dd demo du6pg j77 K78 mad mom Oa1 pa99 pab pasd test testdemo wow zg5fj

dd1:

dd2:

dd3:

Desktop:

dir1:

Documents:

Downloads:

Music:

Pictures:

Public:

30. List all files and directories with five characters whose first character is in the range a to m, second character is in the range c to z and third character is a digit in the range 4 to 9.

labh@labh-virtual-machine:~$ ls [a-m][c-z][4-9][a,1-z,9][a,1-z,9]

du6pg eg5ui

OR

labh@labh-virtual-machine:~$ ls [a-m][c-z][4-9]??

du6pg eg5ui

31. Find the type of all files.

labh@labh-virtual-machine:~$ ls -n

total 64

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 8a

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 a8

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 aa

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 abc

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 Abc

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 add

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 agb5

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 Ak

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 Ak5

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 atb9

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 awe

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 axe

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 Axy

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 bb

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 bcd

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 cc

-rw-rw-r-- 1 1000 1000 0 Aug 12 15:02 dad

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 dd

drwxrwxr-x 2 1000 1000 4096 Aug 12 15:00 dd1

drwxrwxr-x 2 1000 1000 4096 Aug 12 15:00 dd2

drwxrwxr-x 2 1000 1000 4096 Aug 12 15:01 dd3

-rw-rw-r-- 1 1000 1000   10 Aug 12 20:49 demo

drwxr-xr-x 2 1000 1000 4096 Aug 7 15:59 Desktop

drwxrwxr-x 2 1000 1000 4096 Aug 9 18:13 dir1

drwxr-xr-x 2 1000 1000 4096 Aug 7 15:59 Documents

drwxr-xr-x 2 1000 1000 4096 Aug 7 15:59 Downloads

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 du6pg

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 eg5ui

-rw-rw-r-- 1 1000 1000   12 Aug 12 20:49 example

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 j77

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 K78

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 mad

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 mom

drwxr-xr-x 2 1000 1000 4096 Aug 7 15:59 Music

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 o78

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 Oa1

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 owo

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 pa99

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 pab

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 pasd

drwxr-xr-x 2 1000 1000 4096 Aug 7 15:59 Pictures

drwxr-xr-x 2 1000 1000 4096 Aug 7 15:59 Public

drwx------ 5 1000 1000 4096 Aug 8 22:25 snap

drwxr-xr-x 2 1000 1000 4096 Aug 7 15:59 Templates

-rw-rw-r-- 1 1000 1000   18 Aug 12 20:49 test

-rw-rw-r-- 1 1000 1000   0 Aug 12 21:15 testdemo

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 uwu

drwxr-xr-x 2 1000 1000 4096 Aug 7 15:59 Videos

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 wow

-rw-rw-r-- 1 1000 1000   0 Aug 12 15:02 zg5fj

32. Display and count the number of lines, words and characters of file “demo”.

labh@labh-virtual-machine:~$ cat -n demo

1 a

2 b

3 c

4 d

5 e

labh@labh-virtual-machine:~$ wc demo

5 5 10 demo

33. Display and count the number of lines of file “demo”.

labh@labh-virtual-machine:~$ wc -l demo

5 demo

34. Display and count the number of words of file “demo”.

labh@labh-virtual-machine:~$ wc -w demo

5 demo

35. Display and count the number of characters of file “demo”.

labh@labh-virtual-machine:~$ wc -c demo

10 demo

36. Display and count the number of lines, words and characters of file “demo” and “example” and “test”.

labh@labh-virtual-machine:~$ wc demo

5 5 10 demo

labh@labh-virtual-machine:~$ wc example

3 3 12 example

labh@labh-virtual-machine:~$ wc test

3 3 18 test

37. Count the number of all files and directories.

labh@labh-virtual-machine:~$ ls | wc -l

50

38. Count the number of all users currently logged in to the system.

labh@labh-virtual-machine:~$ who | wc -l

1

39. Store the number of users currently logged in to the system in file “testdemo”.

labh@labh-virtual-machine:~$ who testdemo | wc -l

0

40. Display and store the list in of all filenames in file “list”.

labh@labh-virtual-machine:~$ ls >list

labh@labh-virtual-machine:~$ cat list

8a

a8

aa

abc

Abc

add

agb5

Ak

Ak5

atb9

awe

axe

Axy

bb

bcd

cc

dad

dd

dd1

dd2

dd3

demo

Desktop

dir1

Documents

Downloads

du6pg

eg5ui

example

j77

K78

list

mad

mom

Music

o78

Oa1

owo

pa99

pab

pasd

Pictures

Public

snap

Templates

test

testdemo

uwu

Videos

wow

zg5fj

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:03-08-2022**

US-TCS-501 Linux Practical 3

General Purpose Utility and More Bash Shell Commands

THEORY:

Command	Syntax	Description
date	date	Used to display date, time and time zone.
cal	cal <year>	Used to display the current month's calendar with the current date highlighted
who	who	To display current logged in users
whoami	whoami	To display current user.
echo	echo "<material to print>"	To print statements.
w	w	Displaying all user data with details
history	history	List of all commands executed in the shell from first login.
find	find	The find command is used to find a particular file within a directory. It also supports various options to find a file such as byname, by type, by date, and more.
locate	locate <file name>	The locate command is used to search a file-by-file name. It searches the file in the database, whereas the find command searches in the file system. It is faster than the find command.
time	time	Used to display the time to execute a command.

PRACTICAL:

36. Display the calendar of the current month of the current year.

tycs69@tycs:~$ cal August 2022

```
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

37. Display the calendar of the year 2009.

tycs-123@tycs:~$ cal 2009

```
tycs-123@tycs:~$ cal 2009
                              2009
      January               February                March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
             1  2  3   1  2  3  4  5  6  7   1  2  3  4  5  6  7
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   8  9 10 11 12 13 14
11 12 13 14 15 16 17  15 16 17 18 19 20 21  15 16 17 18 19 20 21
18 19 20 21 22 23 24  22 23 24 25 26 27 28  22 23 24 25 26 27 28
25 26 27 28 29 30 31                        29 30 31


      April                 May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4            1  2     1  2  3  4  5  6
 5  6  7  8  9 10 11   3  4  5  6  7  8  9   7  8  9 10 11 12 13
12 13 14 15 16 17 18  10 11 12 13 14 15 16  14 15 16 17 18 19 20
19 20 21 22 23 24 25  17 18 19 20 21 22 23  21 22 23 24 25 26 27
26 27 28 29 30        24 25 26 27 28 29 30  28 29 30
                      31


      July                August               September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4            1        1  2  3  4  5
 5  6  7  8  9 10 11   2  3  4  5  6  7  8   6  7  8  9 10 11 12
12 13 14 15 16 17 18   9 10 11 12 13 14 15  13 14 15 16 17 18 19
19 20 21 22 23 24 25  16 17 18 19 20 21 22  20 21 22 23 24 25 26
26 27 28 29 30 31     23 24 25 26 27 28 29  27 28 29 30
                      30 31


     October             November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
             1  2  3   1  2  3  4  5  6  7         1  2  3  4  5
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   6  7  8  9 10 11 12
11 12 13 14 15 16 17  15 16 17 18 19 20 21  13 14 15 16 17 18 19
18 19 20 21 22 23 24  22 23 24 25 26 27 28  20 21 22 23 24 25 26
```

25 26 27 28 29 30 31 29 30 27 28 29 30 31

38. Display the calendar of the month July of the year 1998.

tycs69@tycs:~$ cal 07 1998
July 1998
Su Mo Tu We Th Fr Sa
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

39. Display today's date.

tycs69@tycs:~$ date
Wednesday 03 August 2022 10:49:49 AM IST

40. Display date as mm/dd/yy.

tycs69@tycs:~$ date +" %n %D"
08/03/22

41. Display current hour, minutes and seconds separately.

tycs69@tycs:~$ date +"%H %M %S"
11 02 02

42. Display time in hours, minutes and seconds as HH:MM:SS.

tycs69@tycs:~$ date +"%T"
10:55:57

43. Display abbreviated month name.

tycs69@tycs:~$ date +"%h"
Aug

44. Display day of the week.

tycs69@tycs:~$ date +"%A"
Wednesday

45. Display full weekday.

Tycs69@tycs:~$ date +%A
Wednesday

46. Display abbreviated weekday.

Tycs69@tycs:~$ date +%a
Wed

47. Display time in AM/PM notation.

tycs69@tycs:~$ date +"%T %p"
11:11:53 AM

48. Display last two digits of the year.

tycs69@tycs:~$ date +"%y"
22

49. Display full year.

tycs69@tycs:~$ date +"%Y"
2022

50. Display the date and time on two different lines.

tycs69@tycs:~$ date +"%D %n %r"
08/03/22
11:21:39 AM IST

51. Calculate the following using bc command:-

tycs69@tycs:~$ bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2012-2017 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.

i) 6 + 7
13
ii) 12 * 12
144
iii) 144 / 3
48
iv) scale = 2 25 / 7
.0110101
v) (8 + 2) * 5
50
vi) (2 + 3) + 8 - 4 * (6 – 3)
12
vii) 2 ^ 6
64
viii) 12*12; 12 / 6
144
2
ix) ibase=2
11001010
202
x) obase=2
14
11

52. Display all the users currently logged in the system.

tycs69@tycs:~$ who
tycs69 tty2 2022-08-03 10:45 (tty2)

53. Display the login name of the current terminal.

tycs69@tycs:~$ whoami
tycs69

54. Display all the users currently logged in the system with heading.

tycs69@tycs:~$ who -Hu
NAME LINE TIME IDLE PID COMMENT
tycs69 tty2 2022-08-03 10:45 00:55 1686 (tty2)

55. Display the terminal file of the current terminal.

tycs69@tycs:~$ tty
/labh/pts/0

56. Display the message on to the terminal as “Hello World”.

tycs69@tycs:~$ echo "Hello World"
Hello World

57. Display the message on to the terminal as “Hello World”.

tycs69@tycs:~$ echo -e "Hello \t World"
Hello World

58. What will be the output of the following:-x = 10 echo The value of x is $x

tycs69@tycs:~$ x=10
tycs69@tycs:~$ echo The value of x is $x
The value of x is 10

i) str=”Linux Practical 1”
echo The string is $str
tycs69@tycs:~$ str="Linux Practical 3"
tycs69@tycs:~$ echo the string is $str
the string is Linux Practical 1

ii) d=`date`
echo “Today’s date is $d”
tycs69@tycs:~$ d='date'
tycs69@tycs:~$ echo "Today's date is $d"
Today's date is date

iii) echo “The number of users currently logged in are `who`”
tycs69@tycs:~$ echo "The number of users currently logged in are 'who' "
The number of users currently logged in are 'who'

iV) echo “The number of users currently logged in are `who` ”
tycs-123@tycs:~$ echo The number of users currently logged in are 'who'
The number of users currently logged in are who

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:05-08-2022**

US-TCS-501 Linux Practical 4
File and Simple Filter Commands

THEORY:

Filter commands:

Command	Syntax	Description
head	head <file name>	It displays the first 10 lines of a file.
Tail	tail <file name>	The difference between both commands is that it displays the last ten lines of the file content. It is useful for reading the error message.
tac	tac <file name>	It displays the file content in reverse order

More	more <file name>	The more command is quite similar to the cat command, the only difference between both commands is that, in case of larger files, the more command displays screenful output at a time.
Less	less <file name>	shows a file's contents one screen at a time.
sort	sort <file name>	The sort command is used to sort files in alphabetical order.
Tr	command | tr <'old'> <'new'>	used to translate the file content like from lower case to upper case.
uniq	command <fileName> | uniq	used to form a sorted list in which every word will occur only once.
cut	cut -d(delimiter) -f(columnNumber) <fileName>	used to select a specific column of a file.
paste		
tee	cat <fileName> | tee <newFile>	To store the file as well as show the output
od	od -b <fileName> // Octal format od -t x1 <fileName> // Hexa decimal format od -c <fileName> // ASCII character format	To display data in binary, hex or octal values of file content

PRACTICAL:

I) Create the following file with name "west" with the contents:-

California
Washington
Oregon
Nevada
Utah

tycs-123@tycs123-virtual-machine:~$ cat >west
California
Washington
Oregon
Nevada
Utah

II) Create the following file with name "coast" with the contents:-

Florida
Washington

Maine
Oregon
California
Georgia

tycs-123@tycs123-virtual-machine:~$ cat >coast
Florida
Washington
Maine
Oregon
California
Georgia

1. **Sort the above two files and store the output in "sor_west" and "sor_coast".**
   tycs-123@tycs123-virtual-machine:~$ sort west>>sor_west
   tycs-123@tycs123-virtual-machine:~$ sort coast>>sor_coast

2. **Compare two files "sor_west" and "sor_coast" and check whether they differ.**
   tycs-123@tycs123-virtual-machine:~$ cmp sor_west sor_coast
   sor_west sor_coast differ: byte 12, line 2

3. **Compare two files "sor_west" and "sor_coast" byte by byte and display its octal value.**
   tycs-123@tycs123-virtual-machine:~$ cmp -l sor_west sor_coast
   12 116 106
   13 145 154
   14 166 157
   15 141 162
   16 144 151
   17 141 144
   18 12 141
   19 117 12
   20 162 107
   22 147 157
   23 157 162
   24 156 147
   25 12 151
   26 125 141
   27 164 12
   28 141 115
   29 150 141
   30 12 151
   31 127 156
   32 141 145
   33 163 12
   34 150 117
   35 151 162
   36 156 145
   38 164 157
   39 157 156
   40 156 12

41 12 127
cmp: EOF on sor_west after byte 41

4. **Display the lines unique to file “sor_west”, lines unique to file “sor_coast” and lines common to both.**
tycs-123@tycs123-virtual-machine:~$ comm sor_west sor_coast
		California
	Florida
	Georgia
	Maine
Nevada
		Oregon
Utah
		Washington

5. **Display the lines which are unique to file “sor_west”.**
tycs-123@tycs123-virtual-machine:~$ comm -23 sor_west sor_coast
Nevada
Utah
6. **Display the lines which are unique to file “sor_coast”.**
tycs-123@tycs123-virtual-machine:~$ comm -13 sor_west sor_coast
Florida
Georgia
Maine

7. **Display the lines which are common to both files.**
tycs-123@tycs123-virtual-machine:~$ comm -12 sor_west sor_coast
California
Oregon
Washington

8. **Display lines unique to file “sor_west“ and lines unique to file “sort_coast”.**
tycs-123@tycs123-virtual-machine:~$ comm -3 sor_west sor_coast
	Florida
	Georgia
	Maine
Nevada
Utah

9. **Display lines unique to file “sor_west“ and lines common to both files.**
tycs-123@tycs123-virtual-machine:~$ comm -2 sor_west sor_coast
	California
Nevada
	Oregon
Utah
	Washington

10. **Display lines unique to file “sor_coast“ and lines common to both files.**
tycs-123@tycs123-virtual-machine:~$ comm -1 sor_west sor_coast
	California

Florida
Georgia
Maine
Oregon
Washington

11. Display the differences of lines between two files "sor_west" and "sor_coast" in context form.

```
tycs-123@tycs123-virtual-machine:~$ diff sor_west sor_coast
2c2,4
< Nevada
---
> Florida
> Georgia
> Maine
4d5
< Utah
```

III) Create the file with the name "testu1" with the following contents:-

DCNII
ADVJAVAII
LINUXOS
SEII
WEBNETII
CPROG
DM
CG
COREJAVA
DBMSI
DCNII
LINUXOS
CPROG
COD
MICROPROC
WEBNETII
SEII
CG

```
tycs-123@tycs123-virtual-machine:~$ cat >testu1
DCNII
ADVJAVAII
LINUXOS
SEII
WEBNETII
CPROG
DM
CG
COREJAVA
DBMSI
DCNII
LINUXOS
```

CPROG
COD
MICROPROC
WEBNETII
SEII
CG

IV) Create the file with the name "testu2" with the following contents:-

01:accounts:6213:a
01:accounts:6213:a
02:admin:6403:b
03:marketing:6521:c
03:marketing:6521:c
04:personnel:7630:d
05:production:8589:e
05:production:8589:e
06:sales:9876:f

tycs-123@tycs123-virtual-machine:~$ cat >testu2
01:accounts:6213:a
01:accounts:6213:a
02:admin:6403:b
03:marketing:6521:c
03:marketing:6521:c
04:personnel:7630:d
05:production:8589:e
05:production:8589:e
06:sales:9876:f

V) Create the file with the name "testu3" with the following contents:-

50
20
10
0
-1
5
2
100
10

tycs-123@tycs123-virtual-machine:~$ cat >testu3
50
20
10
0
-1
5
2
100
10

VI) Create the file with the name "testu4" with the following contents:-
01:accounts:6213:a
06:sales:6213:f
05:production:5489:e
04:personnel:7630:d
02:admin:6521:b
03:marketing:6521:c

tycs-123@tycs123-virtual-machine:~$ cat >testu4
01:accounts:6213:a
06:sales:6213:f
05:production:5489:e
04:personnel:7630:d
02:admin:6521:b
03:marketing:6521:c

Write the commands for the following using simple filter commands:-

1. **Display first ten lines of file "testu1**".
   tycs-123@tycs123-virtual-machine:~$ head testu1
   DCNII
   ADVJAVAII
   LINUXOS
   SEII
   WEBNETII
   CPROG
   DM
   CG
   COREJAVA
   DBMSI

2. **Display first seven lines of file "testu1".**
   tycs-123@tycs123-virtual-machine:~$ head -7 testu1
   DCNII
   ADVJAVAII
   LINUXOS
   SEII
   WEBNETII
   CPROG
   DM

3. **Display last ten lines of file "testu1".**
   COREJAVA
   DBMSI
   DCNII
   LINUXOS
   CPROG
   COD
   MICROPROC
   WEBNETII

SEII
CG

4. **Display the last five lines of file "testu1".**
   tycs-123@tycs123-virtual-machine:~$ tail -n 5 testu1
   COD
   MICROPROC
   WEBNETII
   SEII
   CG

5. **Display from line number three onwards of file "testu1".**
   tycs-123@tycs123-virtual-machine:~$ tail -n +3 testu1
   LINUXOS
   SEII
   WEBNETII
   CPROG
   DM
   CG
   COREJAVA
   DBMSI
   DCNII
   LINUXOS
   CPROG
   COD
   MICROPROC
   WEBNETII
   SEII
   CG

6. **Sort the file "testu1".**
   tycs-123@tycs123-virtual-machine:~$ sort testu1
   ADVJAVAII
   CG
   CG
   COD
   COREJAVA
   CPROG
   CPROG
   DBMSI
   DCNII
   DCNII
   DM
   LINUXOS
   LINUXOS
   MICROPROC
   SEII
   SEII
   WEBNETII
   WEBNETII

7. **Sort the file “testu3” in ascending order.**

```
tycs-123@tycs123-virtual-machine:~$ sort -n testu3
-1
0
2
5
10
10
20
50
100
```

8. **Sort the file “testu3” in descending order.**

```
tycs-123@tycs123-virtual-machine:~$ sort -n -r  testu3
100
50
20
10
10
5
2
0
-1
```

9. **Remove repeated lines and sort the file “testu2”.**

```
tycs-123@tycs123-virtual-machine:~$ sort -u testu2
01:accounts:6213:a
02:admin:6403:b
03:marketing:6521:c
04:personnel:7630:d
05:production:8589:e
06:sales:9876:f
```

10. **Sort on 4th and 5th column of second field of file “testu4”.**

```
tycs-123@tycs123-virtual-machine:~$ sort -t: -k 2.4,2.5 testu4
05:production:5489:e
06:sales:6213:f
02:admin:6521:b
03:marketing:6521:c
01:accounts:6213:a
04:personnel:7630:d
```

11. **Sort on second field of file “testu4”.**

```
tycs-123@tycs123-virtual-machine:~$ sort -t":" -k 2 testu4
01:accounts:6213:a
02:admin:6521:b
03:marketing:6521:c
04:personnel:7630:d
05:production:5489:e
```

06:sales:6213:f

12. Sort on third field of file “testu4”.

```
tycs-123@tycs123-virtual-machine:~$ sort -t":" -k 3 testu4
05:production:5489:e
01:accounts:6213:a
06:sales:6213:f
02:admin:6521:b
03:marketing:6521:c
04:personnel:7630:d
```

13. Check whether file “testu2” is already sorted. Observe the output?

```
tycs-123@tycs123-virtual-machine:~$Desktop$ sort -c testu2
sort: testu2:2: disorder: 01:accounts:6213:a
```

14. Check whether file “testu4” is already sorted. Observe the output?

```
tycs-123@tycs123-virtual-machine:~$Desktop$ sort -c testu4
sort: testu4:3: disorder: 05:production:5489:e
```

15. Check whether second field of “testu2” is already sorted. Observe the output?

```
tycs-123@tycs123-virtual-machine:~$Desktop$ sort -k 2 -c testu4
sort: testu4:3: disorder: 05:production:5489:e
```

16. Check whether second field of “testu4” is already sorted. Observe the output?

```
tycs-123@tycs123-virtual-machine:~$Desktop$ sort -k 2 -c testu4
sort: testu4:3: disorder: 05:production:5489:e
```

17. Store the sorted output of file “testu1” in “sort_testu1”.

```
tycs-123@tycs123-virtual-machine:~$Desktop$ sort testu1 >sort_testu1
tycs-123@tycs123-virtual-machine:~$Desktop$ cat sort_testu1
ADVJAVAII
CG
CG
COD
COREJAVA
CPROG
CPROG
DBMSI
DCNII
DCNII
DM
LINUXOS
LINUXOS
MICROPROC
SEII
SEII
WEBNETII
WEBNETII
```

18. Store the sorted output of file “testu3” in “sort_testu3”.

```
tycs-123@tycs123-virtual-machine:~$Desktop$ sort testu3 >sort_testu3
tycs-123@tycs123-virtual-machine:~$Desktop$ cat sort_testu3
0
-1
10
10
100
2
20
5
50
```

19. Merge the sorted output of file “sort_testu1” and “sort_testu3”.

```
tycs-123@tycs123-virtual-machine:~$Desktop$ paste sort_testu1 sort_testu3
ADVJAVAII  0
CG      -1
CG      10
COD   10
COREJAVA   100
CPROG         2
CPROG         20
DBMSI         5
DCNII 50
DCNII
DM
LINUXOS
LINUXOS
MICROPROC
SEII
SEII
WEBNETII
WEBNETII
```

20. Display unique lines only in file “testu1”.

```
tycs-123@tycs123-virtual-machine:~$Desktop$ uniq testu1
DCNII
ADVJAVAII
LINUXOS
SEII
WEBNETII
CPROG
DM
CG
COREJAVA
DBMSI
DCNII
LINUXOS
CPROG
COD
MICROPROC
```

WEBNETII
SEII
CG

21. Display unique lines only in file “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ uniq testu2
01:accounts6213:a
01:accounts:6213:a
02:admin:6403:b
03:marketing:6521:c
04:personnel:7630:d
05:production:8589:e
06:sales:9876:f

22. Display only non-repeated lines in file “testu1”.

tycs-123@tycs123-virtual-machine:~$Desktop$ uniq -u testu1
DCNII
ADVJAVAII
LINUXOS
SEII
WEBNETII
CPROG
DM
CG
COREJAVA
DBMSI
DCNII
LINUXOS
CPROG
COD
MICROPROC
WEBNETII
SEII
CG

23. Display only non-repeated lines in file “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ uniq -u testu2
02:admin:6403:b
04:personnel:7630:d
06:sales:9876:f

24. Display one copy of duplicate lines in files “testu1”.

tycs-123@tycs123-virtual-machine:~$Desktop$ uniq -d testu1

25. Display one copy of duplicate lines in files “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ uniq -d testu2
03:marketing:6521:c
05:production:8589:e

26. Count the frequency of occurrence of all lines in file “testu1”.

tycs-123@tycs123-virtual-machine:~$Desktop$ uniq -c testu1

```
1 DCNII
1 ADVJAVAII
1 LINUXOS
1 SEII
1 WEBNETII
1 CPROG
1 DM
1 CG
1 COREJAVA
1 DBMSI
1 DCNII
1 LINUXOS
1 CPROG
1 COD
1 MICROPROC
1 WEBNETII
1 SEII
1 CG
```

27. Count the frequency of occurrence of all lines in files “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ uniq -c testu2

```
1 01:accounts6213:a
1 01:accounts:6213:a
1 02:admin:6403:b
2 03:marketing:6521:c
1 04:personnel:7630:d
2 05:production:8589:e
1 06:sales:9876:f
```

28. Translate a character ‘a’ with ‘A’ in files “testu1” and “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ tr 'a' 'A' <testu1

```
DCNII
ADVJAVAII
LINUXOS
SEII
WEBNETII
CPROG
DM
CG
COREJAVA
DBMSI
DCNII
LINUXOS
CPROG
COD
MICROPROC
WEBNETII
```

SEII
CG

tycs-123@tycs123-virtual-machine:~$Desktop$ tr 'a' 'A' <testu2
01:Accounts6213:A
01:Accounts:6213:A
02:Admin:6403:b
03:mArketing:6521:c
03:mArketing:6521:c
04:personnel:7630:d
05:production:8589:e
05:production:8589:e
06:sAles:9876:f

29. Translate characters ‘x’, ’y’, ‘z’ with ‘X’,’Y’,’Z’ respectively in files “testu1” and “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ tr 'xyz' 'XYZ' <testu1
DCNII
ADVJAVAII
LINUXOS
SEII
WEBNETII
CPROG
DM
CG
COREJAVA
DBMSI
DCNII
LINUXOS
CPROG
COD
MICROPROC
WEBNETII
SEII
CG

tycs-123@tycs123-virtual-machine:~$Desktop$ tr 'xyz' 'XYZ' <testu2
01:accounts6213:a
01:accounts:6213:a
02:admin:6403:b
03:marketing:6521:c
03:marketing:6521:c
04:personnel:7630:d
05:production:8589:e
05:production:8589:e
06:sales:9876:f

30. Translate all lowercase letters ‘a’ to ‘z’ with ‘A’ to ‘Z’ in files “testu1” and “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ tr '[a-z]' '[A-Z]' <testu1
DCNII
ADVJAVAII

LINUXOS
SEII
WEBNETII
CPROG
DM
CG
COREJAVA
DBMSI
DCNII
LINUXOS
CPROG
COD
MICROPROC
WEBNETII
SEII
CG

tycs-123@tycs123-virtual-machine:~$Desktop$ tr '[a-z]' '[A-Z]' <testu2
01:ACCOUNTS6213:A
01:ACCOUNTS:6213:A
02:ADMIN:6403:B
03:MARKETING:6521:C
03:MARKETING:6521:C
04:PERSONNEL:7630:D
05:PRODUCTION:8589:E
05:PRODUCTION:8589:E
06:SALES:9876:F

31. Translate all occurrences of ":" with "|" in "testu2".

tycs-123@tycs123-virtual-machine:~$Desktop$ tr ':' '|' <testu2
01|accounts6213|a
01|accounts|6213|a
02|admin|6403|b
03|marketing|6521|c
03|marketing|6521|c
04|personnel|7630|d
05|production|8589|e
05|production|8589|e
06|sales|9876|f

32. Delete all occurrences of character 'b' from the file "testu1".

tycs-123@tycs123-virtual-machine:~$Desktop$ tr -d 'b' <testu2
01:accounts6213:a
01:accounts:6213:a
02:admin:6403:
03:marketing:6521:c
03:marketing:6521:c
04:personnel:7630:d
05:production:8589:e
05:production:8589:e

06:sales:9876:f

tycs-123@tycs123-virtual-machine:~$Desktop$ tr -d 'b' <testu1
DCNII
ADVJAVAII
LINUXOS
SEII
WEBNETII
CPROG
DM
CG
COREJAVA
DBMSI
DCNII
LINUXOS
CPROG
COD
MICROPROC
WEBNETII
SEII
CG

33. Display only last field of file “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ cut -d ":" -f 4 testu2
a
b
c
c
d
e
e
f

34. Display second and third field of file “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ cut -d ":" -f 2,3 testu2
accounts6213:a
accounts:6213
admin:6403
marketing:6521
marketing:6521
personnel:7630
production:8589
production:8589
sales:9876

35. Display the characters from 1 to 4 and 12 to 14 of file “testu2”.

tycs-123@tycs123-virtual-machine:~$Desktop$ cut -c 1-4,12-14 testu2
01:a621
01:a:62
02:a03:

03:mg:6
03:mg:6
04:pl:7
05:pon:
05:pon:
06:s76:

36. Save the second and fourth field of file “testu2” in file “cut_testu21”.

tycs-123@tycs123-virtual-machine:~$Desktop$ cut -d ":" -f 2,4 testu2 >cut_testu21
tycs-123@tycs123-virtual-machine:~$Desktop$ cat cut_testu21
accounts6213
accounts:a
admin:b
marketing:c
marketing:c
personnel:d
production:e
production:e
sales:f

37. Save the first field of file “testu2” in file “cut_testu22”.

tycs-123@tycs123-virtual-machine:~$Desktop$ cut -d ":" -f 1 testu2 >cut_testu22
tycs-123@tycs123-virtual-machine:~$Desktop$ cat cut_testu22
01
01
02
03
03
04
05
05
06

38. Paste “cut_testu22” and “cut_testu21” vertically.

tycs-123@tycs123-virtual-machine:~$Desktop$ paste cut_testu22 cut_testu21
01 accounts6213
01 accounts:a
02 admin:b
03 marketing:c
03 marketing:c
04 personnel:d
05 production:e
05 production:e
06 sales:f

39. Paste “cut_testu22” and “cut_testu21” vertically, fields should be separated by pipe.

tycs-123@tycs123-virtual-machine:~$Desktop$ paste -d "|" cut_testu22 cut_testu21
01|accounts6213
01|accounts:a
02|admin:b

03|marketing:c
03|marketing:c
04|personnel:d
05|production:e
05|production:e
06|sales:f

40. Paste "cut_testu22" and "cut_testu21" sequentially.

tycs-123@tycs123-virtual-machine:~$Desktop$ paste -s cut_testu22 cut_testu21
01 01 02 03 03 04 05 05 06
accounts6213 accounts:a admin:b marketing:c marketing:c personnel:d production:e production:e sales:f

41. Save the contents of "cut_test21" and "cut_test22" in "cut_test23".

tycs-123@tycs123-virtual-machine:~$Desktop$ paste cut_testu22 cut_testu21 >cut_testu23
tycs-123@tycs123-virtual-machine:~$Desktop$ cat cut_testu23
01 accounts6213
01 accounts:a
02 admin:b
03 marketing:c
03 marketing:c
04 personnel:d
05 production:e
05 production:e
06 sales:f

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:05-08-2022**

US-TCS-501 Linux Practical 5

Searching data in Files

THEORY:

Command:

grep : Searches a file for given pattern and given options. Options as mentioned below. It can also work with regular expressions.

Options for grep	Description
-c	No of lines that match the pattern

-i	Ignores case for matching-n
-n	Displays matched lines and line numbers
-v	Prints all lines that do not match a pattern
-e	Used when multiple patterns are to be matched
-f	Takes patterns from a file
-E	Takes pattern as an extended regular expression
-o	Print only matched parts of a line

PRACTICAL:

Commands used in this practical are as follows:-

(grep, egrep and its options, character class of grep and egrep)

Create a file **with the name Student** with the following fields separated by a **blank space** having the below mentioned values:

Field	**Roll No**	**First Name**	**Last Name**	**Date of Birth**	**Marks (out of 600)**
Values	**Numeric**	**Character**	**Character**	**dd-mm-yy**	**Numeric**

Insert at least 5 records in this file.

```
tycs-123@tycs:~$ gedit student
tycs-123@tycs:~$ cat student
1 Aryan Chilly 19-07-91 555
2 Krasia Nohara 05-12-94 567
3 Labjesh Joshi 17-11-92 599
4 Sagun Chatri 06-12-93 559
5 Aryan Noobikal 01-11-96 592
6 Devasheep Subbha 04-03-93 567
8 Pracheese Chavan 10-06-93 582
9 Anish Patel 10-02-99 590
10 Nidhi Gain 10-12-95 559
11 Tumlol Guptil 11-02-96 561
12 Hana Montana 12-04-91 550
```

Write and execute the commands for the following using grep with character class:

1. **Search for the First Name as "Tumlol".**

tycs-123@tycs123-virtual-machine:~$ grep -i Tumlol student

11 Tumlol Guptil 11-02-96 561

2. **Search for the four letter First Name only.**
   tycs-123@tycs123-virtual-machine:~$ cut -d" " -f 2 student|grep ^....$ > pat2

   tycs-123@tycs123-virtual-machine:~$ grep -f pat2 student

   12 Hana Montana 12-04-91 550

3. **Search for the Roll No starting with "1".**
   tycs-123@tycs123-virtual-machine:~$ grep ^1 student

   1 Aryan Chilly 19-07-91 555

   10 Nidhi Gain 10-12-95 559

   11 Tumlol Guptil 11-02-96 561

   12 Hana Montana 12-04-91 550

4. **Search for the marks with last digit as "0".**
   tycs-123@tycs123-virtual-machine:~$ cut -d" " -f 5 student|grep [*0] >> pat3

   tycs-123@tycs123-virtual-machine:~$ grep -f pat3 student

   9 Anish Patel 10-02-99 590

   12 Hana Montana 12-04-91 550

5. **Search for the name with first letter in First Name is 'A' and last character is 'l' in Last Name.**
   tycs-123@tycs123-virtual-machine:~$ cut -d" " -f 2,3 student|grep ^[A].*[l]$ >> pat4

   tycs-123@tycs123-virtual-machine:~$ grep -f pat4 student

   5 Aryan Noobikal 01-11-96 592

   9 Anish Patel 10-02-99 590

6. **Search for the Roll No that are not stating with "2".**
   tycs-123@tycs123-virtual-machine:~$ grep ^[^2] student

   Aryan Chilly 19-07-91 555

   3 Labjesh Joshi 17-11-92 599

   4 Sagun Chatri 06-12-93 559

   5 Aryan Noobikal 01-11-96 592

6 Devasheep Subbha 04-03-93 567

8 Pracheese Chavan 10-06-93 582

9 Anish Patel 10-02-99 590

10 Nidhi Gain 10-12-95 559

11 Tumlol Guptil 11-02-96 561

12 Hana Montana 12-04-91 550

7. **Search for the date of birth in which year ends with "2".**
   tycs-123@tycs123-virtual-machine:~$ cut -d" " -f 4 student|grep .*[2]$ > pat5

   tycs-123@tycs123-virtual-machine:~$ grep -f pat5 student

   3 Labjesh Joshi 17-11-92 599

8. **Search for the First Name which starts with an uppercase letter vowel**.
   tycs-123@tycs123-virtual-machine:~$ cut -d" " -f 2 student|grep ^[AEIOU] > pat6

   tycs-123@tycs123-virtual-machine:~$ grep -f pat6 student

   1 Aryan Chilly 19-07-91 555

   5 Aryan Noobikal 01-11-96 592

   9 Anish Patel 10-02-99 590

Create the file with the name "test_u4" with the following contents:-

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

tycs-123@tycs123-virtual-machine:~/Desktop/prac5$ gedit test_u4

Write and execute the commands for the following using grep:-

1. **Display the records containing "accounts".**
   tycs-123@tycs123-virtual-machine:~/Desktop/prac5$ grep accounts test_u4

   01:accounts:6213:a

   01:accounts:6003:h

2. **Display all records containing "personnel".**
   tycs-123@tycs123-virtual-machine:~$ grep personnel test_u4

   04:personnel:7630:a

3. **Display the record of "Admin". Observe the output?**
   tycs-123@tycs123-virtual-machine:~/Desktop/prac5$ grep Admin test_u4

   tycs-123@tycs123-virtual-machine:~/Desktop/prac5$

4. **Count the occurrence of "production".**
   tycs-123@tycs123-virtual-machine:~$ grep -c production test_u4

   2

5. **Display all the records of "marketing" along with line numbers.**
   tycs-123@tycs123-virtual-machine:~$ grep -n marketing test_u4

   7:01:marketing:6521:c

   12:10:marketing:6215:j

6. **Display all the records except "sales".**
   tycs-123@tycs123-virtual-machine:~$ grep -v sales test_u4

   01:accounts:6213:a

   05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

01:accounts:6003:h

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

7. **Display the records of "accounts" and "admin".**
   tycs-123@tycs123-virtual-machine:~$ grep -e accounts -e admin test_u4

   01:accounts:6213:a

   02:admin:6521:b

   01:accounts:6003:h

8. **Display the filenames containing "marketing".**
   tycs-123@tycs123-virtual-machine:~$ grep -l marketing *

   test_u4

9. **Store the patterns "admin", "production" and "sales" in new file and display the records containing these patterns.**
   tycs-123@tycs123-virtual-machine:~$ cat > pat1

   admin

   production

   sales

   tycs-123@tycs123-virtual-machine:~$ grep -f pat1 test_u4

   06:sales:6213:f

   05:production:5489:e

   02:admin:6521:b

   07:sales:6135:a

11:production:3480:i

10. Display all the records not containing “marketing” along with line numbers.

tycs-123@tycs123-virtual-machine:~$ grep -nv marketing test_u4

1:01:accounts:6213:a

2:06:sales:6213:f

3:05:production:5489:e

4:

5:04:personnel:7630:a

6:02:admin:6521:b

8:

9:07:sales:6135:a

10:01:accounts:6003:h

11:

13:11:production:3480:i

14:14:Personnel:7306:k

15:

11. Count all the records not containing “production”.

tycs-123@tycs123-virtual-machine:~$ grep -cv production test_u4

13

12. Display all the records of “admin” along with line numbers.

tycs-123@tycs123-virtual-machine:~$ grep -n admin test_u4

6:02:admin:6521:b

13. Display all the record except “admin” along with line numbers.

tycs-123@tycs123-virtual-machine:~$ grep -nv admin test_u4

1:01:accounts:6213:a

2:06:sales:6213:f

3:05:production:5489:e

4:

5:04:personnel:7630:a

7:01:marketing:6521:c

8:

9:07:sales:6135:a

10:01:accounts:6003:h

11:

12:10:marketing:6215:j

13:11:production:3480:i

14:14:Personnel:7306:k

15:

14. Display all records starting with “1”.

tycs-123@tycs123-virtual-machine:~$ grep ^1 test_u4

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

15. Display all records not starting with “1”.

tycs-123@tycs123-virtual-machine:~$ grep ^[^1] test_u4

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

16. Display all records starting with “1” along with line numbers.

tycs-123@tycs123-virtual-machine:~$ grep -n ^1 test_u4

12:10:marketing:6215:j

13:11:production:3480:i

14:14:Personnel:7306:k

17. Display all records not starting with “1” along with line numbers.

tycs-123@tycs123-virtual-machine:~$ grep -n ^[^1] test_u4

1:01:accounts:6213:a

2:06:sales:6213:f

3:05:production:5489:e

5:04:personnel:7630:a

6:02:admin:6521:b

7:01:marketing:6521:c

9:07:sales:6135:a

10:01:accounts:6003:h

18. Display all records end with “a”.

tycs-123@tycs123-virtual-machine:~$ grep a$ test_u4

01:accounts:6213:a

04:personnel:7630:a

07:sales:6135:a

19. Display all records not end with “a”.

tycs-123@tycs123-virtual-machine:~$ grep [^a]$ test_u4

06:sales:6213:f

05:production:5489:e

02:admin:6521:b

01:marketing:6521:c

01:accounts:6003:h

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

20. Display all records containing nothing along with line numbers.

tycs-123@tycs123-virtual-machine:~$ grep -n ^$ test_u4

4:

8:

11:

15:

21. Display all the lines with first character as “6” followed by any five characters at the end.

tycs-123@tycs123-virtual-machine:~$ grep 6.....$ test_u4

01:accounts:6213:a

06:sales:6213:f

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

10:marketing:6215:j

22. Display all the records starting with “0” followed by any three characters

tycs-123@tycs123-virtual-machine:~$ grep ^0... test_u4

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

23. Display all the records with first character not as “6” followed by any five characters at the end.

tycs-123@tycs123-virtual-machine:~$ grep [^6]..... test_u4

01:accounts:6213:a

06:sales:6213:f

05:production:5489:e

04:personnel:7630:a

02:admin:6521:b

01:marketing:6521:c

07:sales:6135:a

01:accounts:6003:h

10:marketing:6215:j

11:production:3480:i

14:Personnel:7306:k

Write and execute the commands for the following using egrep and fgrep :

1. **Display all the records with the names "sengupta", "dasgupta" and "ramgupta" using egrep command.**
   tycs-123@tycs123-virtual-machine:~$ grep -E '(sengupta|dasgupta|ramgupta)' demo1

   1 Piyusha sengupta 52

   2 Parth dasgupta 12

   3 Riddhi ramgupta 7

   4 Harsh Sengupta 22

2. **Display all the records of "ramgupta" and "dasgupta" using egrep command.**
   tycs-123@tycs123-virtual-machine:~$ egrep '(das|ram)gupta' demo1

   2 Parth dasgupta 12

   3 Riddhi ramgupta 7

3. **Display all the records of "parth", "riddhi" and "harsh" using fgrep command**.
   tycs-123@tycs123-virtual-machine:~$ fgrep 'Parth

   > Riddhi

   > Harsh demo1

   2 Parth dasgupta 12

   3 Riddhi ramgupta 7

   4 Harsh Sengupta 22

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:05-08-2022**

US-TCS-501 Linux Practical 6

Stream Editor – sed Commands

THEORY:

The sed utility is an "editor," but it is unlike most others. In addition to not being screen-oriented, it is also noninteractive. This means you have to insert commands to be executed on the data at the command line or in a script to be processed. When you visualize it, forget any ability to interactively edit files as you would do with Microsoft Word or most other editors. sed accepts a series of commands and executes them on a file (or set of files) noninteractively and unquestionably. As such, it flows through text as water would through a stream, and thus sed fittingly stands for stream editor.

- SED is a powerful text stream editor. Can do insertion, deletion, search and replace(substitution).
- SED command in unix supports regular expression which allows it perform complex pattern matching.

The syntax for the utility is:

sed [options] '{command}' [filename]

PRACTICAL:

1. Create the file with the name "sedfile" with the following contents:-
Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

Write and execute the commands for the following using sed:

1. **Quits after printing first six lines.**
   labh@labh:~$ sed '6q' sedfile

   Carrots:veg:1.39:1:n

   Milk:Dairy:1.89:2:n

   Magazine:Sundry:3.50:1:y

   Cheese:Dairy:4.39:1:n

   Sandwich:Deli:3.89:2:y

   Onions:Veg:0.89:6:n

2. **Display line numbers from four to seven.**
   labh@labh:~$ cat -n sedfile | sed -n '4,7p'

   4 Cheese:Dairy:4.39:1:n

   5 Sandwich:Deli:3.89:2:y

   6 Onions:Veg:0.89:6:n

   7

3. **Display only last line.**
   labh@labh:~$ sed -n '$p' sedfile

   End:of:Data in Sed Editor

   or

   labh@labh:~$ cat -n sedfile | sed -n '$p'

   18 End:of:Data in Sed Editor

4. **Display only line number 8.**
   labh@labh:~$ cat -n sedfile | sed -n '8p'

   8 Chicken:Meat:4.89:2:n

5. **Display lines containing “Onions”.**
   labh@labh:~$ sed -n '/Onions/p' sedfile

   Onions:Veg:0.89:6:n

6. **Display all lines containing “Dairy”.**
   labh@labh:~$ sed -n '/Dairy/p' sedfile

   Milk:Dairy:1.89:2:n

   Cheese:Dairy:4.39:1:n

   Dairy:Dairy:Dairy

7. **Delete line 2.**
   labh@labh:~$ cat -n sedfile | sed -n '2!p'

    1 Carrots:veg:1.39:1:n
    3 Magazine:Sundry:3.50:1:y
    4 Cheese:Dairy:4.39:1:n
    5 Sandwich:Deli:3.89:2:y
    6 Onions:Veg:0.89:6:n
    7
    8 Chicken:Meat:4.89:2:n
    9 Newspaper:Sundry:1.00:1:y
   10 Fish:Meat:3.79:3:n
   11
   12 Floorwax:Hshld:4.65:1:y
   13 Melon:Fruit:1.98:3:n
   14
   15 Celery:Veg:1.79:1:n
   16 Dairy:Dairy:Dairy
   17 Veg:Veg:Veg
   18 End:of:Data in Sed Editor

8. **Delete lines 7 to 10.**
   labh@labh:~$ cat -n sedfile | sed -n '7,10!p'

1 Carrots:veg:1.39:1:n

2 Milk:Dairy:1.89:2:n

3 Magazine:Sundry:3.50:1:y

4 Cheese:Dairy:4.39:1:n

5 Sandwich:Deli:3.89:2:y

6 Onions:Veg:0.89:6:n

11

12 Floorwax:Hshld:4.65:1:y

13 Melon:Fruit:1.98:3:n

14

15 Celery:Veg:1.79:1:n

16 Dairy:Dairy:Dairy

17 Veg:Veg:Veg

18 End:of:Data in Sed Editor

9. Delete lines containing "Meat".

labh@labh:~$ sed -n '/Meat/!p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Newspaper:Sundry:1.00:1:y

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

10. Display lines start at 3 through the first line matching the string "Meat".

labh@labh:~$ sed -n '3,/Meat/p' sedfile

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

11. Delete lines start at 4 through the first line matching the string "Fruit".

labh@labh:~$ sed -n '4,/Fruit/!p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

12. Display lines number at interval of 2 i.e. starting at 1, every other line is output.

labh@labh:~$ cat -n sedfile | sed -n '1~2p'

1 Carrots:veg:1.39:1:n

3 Magazine:Sundry:3.50:1:y

5 Sandwich:Deli:3.89:2:y

7

9 Newspaper:Sundry:1.00:1:y

11

13 Melon:Fruit:1.98:3:n

15 Celery:Veg:1.79:1:n

17 Veg:Veg:Veg

13. Display lines starting at 2 and outputs every third line from there.
labh@labh:~$ cat -n sedfile | sed -n '2~3p'

2 Milk:Dairy:1.89:2:n

5 Sandwich:Deli:3.89:2:y

8 Chicken:Meat:4.89:2:n

11

14

17 Veg:Veg:Veg

14. Display lines starting with “S”.
labh@labh:~$ sed -n '/^S/p' sedfile

Sandwich:Deli:3.89:2:y

15. Display lines ends with “n”.
labh@labh:~$ sed -n '/n$/p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Cheese:Dairy:4.39:1:n

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Fish:Meat:3.79:3:n

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

16. Display lines that are not starting with “C”.
labh@labh:~$ sed -n '/^C/!p' sedfile

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

17. Display lines that are not ending with "y".

labh@labh:~$ sed -n '/y$/!p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Cheese:Dairy:4.39:1:n

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Fish:Meat:3.79:3:n

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Veg:Veg:Veg

End:of:Data in Sed Editor

18. Display all lines that are not starting with "v" or "V" in second field.

labh@labh:~$ cut -d":" -f 2 sedfile | sed -n '/[v,V]/!p'

Dairy

Sundry

Dairy

Deli

Meat

Sundry

Meat

Hshld

Fruit

Dairy

of

19. Display all lines that are not ending with “y” in second field.

labh@labh:~$ cut -d":" -f 2 sedfile | sed -n '/[y$]/!p'

veg

Deli

Veg

Meat

Meat

Hshld

Fruit

Veg

Veg

of

20. Delete lines containing nothing.

labh@labh:~$ sed '/^$/d' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

21. Delete all lines containing “2” in fourth field.

labh@labh:~$ cut -d":" -f4 sedfile | sed '/2/d'

1

1

1

6

1

3

1

3

1

22. Delete all lines containing “y” in fifth field.

labh@labh:~$ cut -d":" -f5 sedfile | sed '/y/d'

n

n

n

n

n

n

n

n

23. Delete all lines that do not start with C.

labh@labh:~$ sed '/^C/d' sedfile

Carrots:veg:1.39:1:n

Cheese:Dairy:4.39:1:n

Chicken:Meat:4.89:2:n

Celery:Veg:1.79:1:n

24. Display all lines that are not empty.

labh@labh:~$ sed -n '/^$/!p' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

25. Delete any other line that does not contain the string "Fish".

labh@labh:~$ sed '/Fish/d' sedfile

Fish:Meat:3.79:3:n

26. Display lines between the strings "Milk" and "Newspaper".

labh@labh:~$ sed -n '/Milk/,/Newspaper/p' sedfile

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

27. Display lines between the string "Sundry" and line number 11.

labh@labh:~$ cat -n sedfile | sed -n '/Sundry/,11p'

3 Magazine:Sundry:3.50:1:y

4 Cheese:Dairy:4.39:1:n

5 Sandwich:Deli:3.89:2:y

6 Onions:Veg:0.89:6:n

7

8 Chicken:Meat:4.89:2:n

9 Newspaper:Sundry:1.00:1:y

10 Fish:Meat:3.79:3:n

11

28. Replace only first occurrence of “Dairy” with “DAIRY” in all lines.

labh@labh:~$ sed 's/Dairy/DAIRY/' sedfile

Carrots:veg:1.39:1:n

Milk:DAIRY:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:DAIRY:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

DAIRY:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

or

labh@labh:~$ sed -n 's/Dairy/DAIRY/p' sedfile

Milk:DAIRY:1.89:2:n

Cheese:DAIRY:4.39:1:n

DAIRY:Dairy:Dairy

29. Replace only first occurrence of “3” with “4” in all lines.

labh@labh:~$ sed 's/3/4/' sedfile

Carrots:veg:1.49:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:4.50:1:y

Cheese:Dairy:4.49:1:n

Sandwich:Deli:4.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:4.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:4:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

or

labh@labh:~$ sed -n 's/3/4/p' sedfile

Carrots:veg:1.49:1:n

Magazine:Sundry:4.50:1:y

Cheese:Dairy:4.49:1:n

Sandwich:Deli:4.89:2:y

Fish:Meat:4.79:3:n

Melon:Fruit:1.98:4:n

30. Replace blank lines with “BLANK LINE” in all lines.

labh@labh:~$ sed 's/^$/BLANK LINE/' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

BLANK LINE

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

BLANK LINE

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

BLANK LINE

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

31. Replace all occurrences of "Dairy" with "DAIRY" in all lines.

labh@labh:~$ sed -n 's/Dairy/DAIRY/gp' sedfile

Milk:DAIRY:1.89:2:n

Cheese:DAIRY:4.39:1:n

DAIRY:DAIRY:DAIRY

32. Replace all occurrences of "3" with "4" in all lines.

labh@labh:~$ sed -n 's/3/4/gp' sedfile

Carrots:veg:1.49:1:n

Magazine:Sundry:4.50:1:y

Cheese:Dairy:4.49:1:n

Sandwich:Deli:4.89:2:y

Fish:Meat:4.79:4:n

Melon:Fruit:1.98:4:n

33. Replace only first occurrence of “Veg” with “VEG” in lines 5 to 17.
labh@labh:~$ sed -n '5,17s/Veg/VEG/p' sedfile

Onions:VEG:0.89:6:n

Celery:VEG:1.79:1:n

VEG:Veg:Veg

34. Replace all occurrences of “Veg” with “VEG” in lines 5 to 17.
labh@labh:~$ sed -n '5,17s/Veg/VEG/gp' sedfile

Onions:VEG:0.89:6:n

Celery:VEG:1.79:1:n

VEG:VEG:VEG

35. Replace all occurrences of “9” with “1” in lines 3 to 8 and print only modified lines.
labh@labh:~$ sed -n '3,8s/9/1/gp' sedfile

Cheese:Dairy:4.31:1:n

Sandwich:Deli:3.81:2:y

Onions:Veg:0.81:6:n

Chicken:Meat:4.81:2:n

36. Replace only first occurrence of “e” with”EEE” only in line 4 and print only this line.
labh@labh:~$ sed -n '4s/e/EEE/p' sedfile

ChEEEese:Dairy:4.39:1:n

37. Replace all occurrences of “:” with”|” only in last line and print only this line.
labh@labh:~$ sed -n '$s/:/|/gp' sedfile

End|of|Data in Sed Editor

38. Replace all occurrences of “y” with “yes” on lines between “Deli” and “Fruit” and print only modified lines.

labh@labh:~$ sed -n '/Deli/,/Fruit/s/y/yes/gp' sedfile

Sandwich:Deli:3.89:2:yes

Newspaper:Sundryes:1.00:1:yes

Floorwax:Hshld:4.65:1:yes

39. Replace all occurrences of “n” with “NO” in lines containing “Meat”.
labh@labh:~$ sed -n '/Meat/s/n/NO/gp' sedfile

ChickeNO:Meat:4.89:2:NO

Fish:Meat:3.79:3:NO

40. Replace all occurrences of “y” with “YES” except lines between “Cheese” and “Newspaper”.
labh@labh:~$ sed -n '/Cheese/,/Newspaper/!s/y/YES/gp' sedfile

Milk:DairYES:1.89:2:n

Magazine:SundrYES:3.50:1:YES

Floorwax:Hshld:4.65:1:YES

CelerYES:Veg:1.79:1:n

DairYES:DairYES:DairYES

41. Replace all occurrences of “:” with “~” in all other lines except lines containing “Veg”.
labh@labh:~$ sed -n '/Veg/!s/:/~/gp' sedfile

Carrots~veg~1.39~1~n

Milk~Dairy~1.89~2~n

Magazine~Sundry~3.50~1~y

Cheese~Dairy~4.39~1~n

Sandwich~Deli~3.89~2~y

Chicken~Meat~4.89~2~n

Newspaper~Sundry~1.00~1~y

Fish~Meat~3.79~3~n

Floorwax~Hshld~4.65~1~y

Melon~Fruit~1.98~3~n

Dairy~Dairy~Dairy

End~of~Data in Sed Editor

42. Print the line number along with the line that contains the string “Floorwax”.

labh@labh:~$ sed '/Floorwax/=' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

12

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

43. Print only line number for a line that contains the string “Floorwax”.

labh@labh:~$ sed -n '/Floorwax/=' sedfile

12

44. Inserts line “Linux Practical” before first line.

labh@labh:~$ sed '1 i Linux Practical' sedfile

Linux Practical

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

45. Insert line "********" before every line.**

labh@labh:~$ sed 'i ********** ' sedfile

****

Carrots:veg:1.39:1:n

****

Milk:Dairy:1.89:2:n

****

Magazine:Sundry:3.50:1:y

****

Cheese:Dairy:4.39:1:n

****

Sandwich:Deli:3.89:2:y

****

Onions:Veg:0.89:6:n

****

****

Chicken:Meat:4.89:2:n

****

Newspaper:Sundry:1.00:1:y

****

Fish:Meat:3.79:3:n

****

****

Floorwax:Hshld:4.65:1:y

****

Melon:Fruit:1.98:3:n

****

****

Celery:Veg:1.79:1:n

****

Dairy:Dairy:Dairy

****

Veg:Veg:Veg

****

End:of:Data in Sed Editor

46. Appends line "Sed Editor" after fourth line.

labh@labh:~$ sed '4 a Sed Editor' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sed Editor

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

47. Append after each line this line "----------".

labh@labh:~$ sed 'a ----------' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

---

48. Inserts three lines “Sed Editor...”, “This is Linux Practical…” and “This is seventh practical…” before fifth line.

labh@labh:~$ sed '5 i Sed Editor...\

> This is Linux Practical…\

> This is seventh practical…' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sed Editor...

This is Linux Practical…

This is seventh practical…

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

49. Appends three lines "Sed Editor…", "This is Linux Practical…" and "This is seventh practical…" after seventh line.

labh@labh:~$ sed '7 a Sed Editor…\

> This is Linux Practical…\

> This is seventh practical…' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Sed Editor…

This is Linux Practical…

This is seventh practical…

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

50. Insert line “Linux OS” before last line.

labh@labh:~$ sed '$ i Linux OS' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

Linux OS

End:of:Data in Sed Editor

51. Append line “Linux OS” after last line.

labh@labh:~$ sed '$ a Linux OS' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

Linux OS

52. Translate characters ‘D’,’O’,’V’ with ‘d’,’o’,’v’ respectively.

labh@labh:~$ sed 'y/DOV/dov/' sedfile

Carrots:veg:1.39:1:n

Milk:dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:dairy:4.39:1:n

Sandwich:deli:3.89:2:y

onions:veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:veg:1.79:1:n

dairy:dairy:dairy

veg:veg:veg

End:of:data in Sed Editor

53. Print both the text and nonprintable ASCII characters.

labh@labh:~$ sed -n 'l' sedfile

Carrots:veg:1.39:1:n$

Milk:Dairy:1.89:2:n$

Magazine:Sundry:3.50:1:y$

Cheese:Dairy:4.39:1:n$

Sandwich:Deli:3.89:2:y$

Onions:Veg:0.89:6:n$

$

Chicken:Meat:4.89:2:n$

Newspaper:Sundry:1.00:1:y$

Fish:Meat:3.79:3:n$

$

Floorwax:Hshld:4.65:1:y$

Melon:Fruit:1.98:3:n$

$

Celery:Veg:1.79:1:n$

Dairy:Dairy:Dairy$

Veg:Veg:Veg$

End:of:Data\tin \tSed Editor$

54. Write line 12 into new file "new1".

labh@labh:~$ sed -n '12 w new1' sedfile

labh@labh:~$ cat new1

Floorwax:Hshld:4.65:1:y

55. Write line 2 to 4 into new file "new2".

labh@labh:~$ sed -n '2,4 w new2' sedfile

labh@labh:~$ cat new2

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

56. Write lines containing text "Celery" into new file "new3".

labh@labh:~$ sed -n '/Celery/ w new3' sedfile

labh@labh:~$ cat new3

Celery:Veg:1.79:1:n

57. Write lines containing text between "Onions" and "Melon" into new file "new4".

labh@labh:~$ sed -n '/Onions/,/Melon/ w new4' sedfile

labh@labh:~$ cat new4

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

58. Write lines containing text between lines 2 to “Onions” into new file “new5”.

labh@labh:~$ sed -n '2,/Onions/ w new5' sedfile

labh@labh:~$ cat new5

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

59. Write lines containing text between lines “Chicken” to 15 into new file “new6”.

labh@labh:~$ sed -n '/Chicken/,15 w new6' sedfile

labh@labh:~$ cat new6

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

60. Read the contents of file “data1 “after line 6.

labh@labh:~$ sed '6 r data1' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

abd

asd

ASDF

ASfff

sdfs

sdg

1ds1

s3d4

222

43sf

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

61. Read the contents of file "data2 "after last line.

labh@labh:~$ sed '$ r data2' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1323s

df2

Linux

Python

ubuntu

aed

132e

de3

d32r

62. Read the contents of file “names “after pattern “End”.

labh@labh:~$ sed '/End/ r data2' sedfile

Carrots:veg:1.39:1:n

Milk:Dairy:1.89:2:n

Magazine:Sundry:3.50:1:y

Cheese:Dairy:4.39:1:n

Sandwich:Deli:3.89:2:y

Onions:Veg:0.89:6:n

Chicken:Meat:4.89:2:n

Newspaper:Sundry:1.00:1:y

Fish:Meat:3.79:3:n

Floorwax:Hshld:4.65:1:y

Melon:Fruit:1.98:3:n

Celery:Veg:1.79:1:n

Dairy:Dairy:Dairy

Veg:Veg:Veg

End:of:Data in Sed Editor

1323s

df2

Linux

Python

ubuntu

aed

132e

de3

d32r

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 7

Awk editor

THEORY:

AWK is an interpreted programming language. It is very powerful and specially designed for text processing. Its name is derived from the family names of its authors − **Alfred Aho, Peter Weinberger, and Brian Kernighan.** The awk command programming language requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators.

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line.

- **Typical Uses of AWK**

Myriad of tasks can be done with AWK. Listed below are just a few of them −

- Text processing
- Producing formatted text reports
- Performing arithmetic operations
- Performing string operations

PRACTICAL:

A)Create a file empdata, which contains the following fields:-

Fieldname	**Datatype**	**Value**
1. Employee name	character	
2. Employee code	numeric	starts with letter 'E'
3. Department code	character	MKT, HRD, PUR
4. Grade	character	A-C
5. Designation	character	manager, director, gm, executive
6. Years of experience	numeric	
7. Date of birth	dd-mm-yy	
8. Region	character	Pune, Mumbai etc…
9. Basic pay	numeric	

Insert at least five records in above file; character fields in each record may not be same in the same case. '~' is used as a field separator. Give commands for the following:-

cat empdata

```
labh@labh-virtual-machine:~$ cat empdata
Labh~E1~HRD~A~gm~5~10-10-1986~Pune~15000
Aryan~E2~MKT~B~manager~6~1-2-1981~Mumbai~10000
```

Dev~E3~PUR~C~director~7~22-1-1985~Pune~10000
Krasia~E4~MKT~A~executive~6~10-9-1988~Mumbai~130000
Nidhi~E5~HRD~B~gm~9~3-2-1982~Pune~13000
Prachi~E6~HRD~B~manager~10~30-11-1985~Pune~20000
Anish~E7~HRD~C~gm~12~5-5-1989~Delhi~17000
Tumul~E8~PUR~A~executive~17~22-12-1972~Mumbai~120000
Chile~E9~MKT~C~manager~4~12-12-1991~Mumbai~1800
Smit~E10~HRD~A~executive~15~19-3-1973~Mumbai~180000
Pratham~E11~HRD~A~executive~9~30-11-1981~Pune~30000

1. **Display all employees who are manager in office.**
   labh@labh-virtual-machine:~$ awk -F"~" '$5 =="manager" {print $0}' empdata
   Aryan~E2~MKT~B~manager~6~1-2-1981~Mumbai~10000
   Prachi~E6~HRD~B~manager~10~30-11-1985~Pune~20000
   Chile~E9~MKT~C~manager~4~12-12-1991~Mumbai~1800

2. **List the details of an employee “Nidhi” in Pune office.**
   labh@labh-virtual-machine:~$ awk -F"~" '$1 =="Nidhi" && $8 =="Pune" {print $0}' empdata
   Nidhi~E5~HRD~B~gm~9~3-2-1982~Pune~13000
3. **Display all employees who are not in the department MKT. display the output sorted on department code.**
   labh@labh-virtual-machine:~$ awk -F"~" '$3 !="MKT" {print $0}' empdata | sort -t"~" -k 3

   Smit~E10~HRD~A~executive~15~19-3-1973~Mumbai~180000
   Pratham~E11~HRD~A~executive~9~30-11-1981~Pune~30000
   Labh~E1~HRD~A~gm~5~10-10-1986~Pune~15000
   Nidhi~E5~HRD~B~gm~9~3-2-1982~Pune~13000
   Prachi~E6~HRD~B~manager~10~30-11-1985~Pune~20000
   Anish~E7~HRD~C~gm~12~5-5-1989~Delhi~17000
   Tumul~E8~PUR~A~executive~17~22-12-1972~Mumbai~120000
   Dev~E3~PUR~C~director~7~22-1-1985~Pune~10000
4. **Display all employees whose years of experience are more than 5.**
   labh@labh-virtual-machine:~$ awk -F"~" '$6 > 5 {print $1}' empdata
   Aryan
   Dev
   Krasia
   Nidhi
   Prachi
   Anish
   Tumul
   Smit
5. **List only employee name, department code and basic pay of employees who are executive.**
   labh@labh-virtual-machine:~$ awk -F"~" '$5 == "executive" {print $1,$3,$9}' empdata
   Krasia MKT 130000
   Tumul PUR 120000
   Smit HRD 180000
   Pratham HRD 30000
6. **Display all employees having grade ‘A’.**
   labh@labh-virtual-machine:~$ awk -F"~" '$4 == "A" {print $1}' empdata
   Labh

Krasia
Tumul
Smit
Pratham

7. **Count total number of employees whose department code is HRD.**
   labh@labh-virtual-machine:~$ awk -F"~" '$3 == "HRD"' empdata | wc -l
   6

8. **Display employee's names with salary above 10,000.**
   labh@labh-virtual-machine:~$ awk -F"~" '$9 > 10000 {print $1}' empdata
   Labh
   Krasia
   Nidhi
   Prachi
   Anish
   Tumul
   Smit
   Pratham
9. **Display only designation and basic pay of employees having number of experience between 3 and 5.**
   labh@labh-virtual-machine:~$ awk -F"~" '$6 >= 3 && $6 <= 5 {print $5,$9}' empdata
   gm 15000
   manager 1800
10. **Find the number of employees in Pune office.**
    labh@labh-virtual-machine:~$ awk -F"~" '$8 == "Pune"' empdata | wc -l
    5
11. **Display employees who get basic pay less than 15000, also calculate and display average basic pay.**
    labh@labh-virtual-machine:~$ awk -F"~" '$9 < 15000 {print $0} {total +=$9;count++} END {print "Average salary="total/count}' empdata
    Aryan~E2~MKT~B~manager~6~1-2-1981~Mumbai~10000
    Dev~E3~PUR~C~director~7~22-1-1985~Pune~10000
    Nidhi~E5~HRD~B~gm~9~3-2-1982~Pune~13000
    Chile~E9~MKT~C~manager~4~12-12-1991~Mumbai~1800

    Average salary=45566.7
12. **Locate the employees with same date of birth in Pune office.**
    labh@labh-virtual-machine:~$ awk -F"~" '$8=="Pune" && n=x[$7]{print n"\n"$0;} {x[$7]=$0}' empdata
13. **Locate all for Labh, Aryan & Dev as employee name.**
    labh@labh-virtual-machine:~$ awk -F"~" '$1=="Labh"||$1=="Aryan"||$1=="Dev" {print $0}' empdata
    Labh~E1~HRD~A~gm~5~10-10-1986~Pune~15000
    Aryan~E2~MKT~B~manager~6~1-2-1981~Mumbai~10000
    Dev~E3~PUR~C~director~7~22-1-1985~Pune~10000
14. **Locate all except for Labh, Aryan & Dev as employee name.**
    labh@labh-virtual-machine:~$ awk -F"~" '$1!~/Labh|Aryan|Dev/ {print $0}'
    empdataKrasia~E4~MKT~A~executive~6~10-9-1988~Mumbai~130000
    Nidhi~E5~HRD~B~gm~9~3-2-1982~Pune~13000
    Prachi~E6~HRD~B~manager~10~30-11-1985~Pune~20000
    Anish~E7~HRD~C~gm~12~5-5-1989~Delhi~17000
    Tumul~E8~PUR~A~executive~17~22-12-1972~Mumbai~120000

Chile~E9~MKT~C~manager~4~12-12-1991~Mumbai~1800
Smit~E10~HRD~A~executive~15~19-3-1973~Mumbai~180000

15. **Find the employees who have designation as director and find the 40% of basic pay as da and 15% of basic pay as hra.**

```
labh@labh-virtual-machine:~$ awk -F"~" 'BEGIN{printf
"Name\tECode\tDCode\tGrade\tDesig\t\tEXP\tDOB\t\tRegion\tPay\t\tDA\t\tHRA\t\n"}$5=="director"
{da=0.4*$9;hra=0.15*$9}{printf $1"\t"$2"\t"$3"\t"$4"\t"$5"\t"$6"\t"$7"\t"$8"\t"$9"\t\t"da"\t\t"hra"\n"}'
empdata
```

Name	ECode	DCode	Grade	Desig	EXP	DOB	Region	Pay	DA	HRA
Labh	E1	HRD	A	gm 5	10-10-1986	Pune	15000			
Aryan	E2	MKT	B	manager	6	1-2-1981	Mumbai	10000		
Dev	E3	PUR	C	director	7	22-1-1985	Pune	10000	4000	1500
Krasia	E4	MKT	A	executive	6	10-9-1988	Mumbai	130000		
4000		1500								
Nidhi	E5	HRD	B	gm 9	3-2-1982	Pune	13000	4000	1500	
Prachi	E6	HRD	B	manager	10	30-11-1985	Pune	20000	4000	1500
Anish	E7	HRD	C	gm 12	5-5-1989	Delhi	17000	4000	1500	
Tumul	E8	PUR	A	executive	17	22-12-1972	Mumbai	120000		
4000		1500								
Chile	E9	MKT	C	manager	4	12-12-1991	Mumbai	1800	4000	
1500										
Smit	E10	HRD	A	executive	15	19-3-1973	Mumbai	180000		
4000		1500								
Pratham	E11	HRD	A	executive	9	30-11-1981	Pune	30000	4000	
1500										
								4000	1500	

16. **Store employee name and date of birth in a file' nbdata'**.

```
labh@labh-virtual-machine:~/Desktop$ awk -F"~" '{print $1,$7}' empdata > nbdata
labh@labh-virtual-machine:~/Desktop$ cat nbdata
labh@labh-virtual-machine:~$ awk -F"~" '{print $1,$7}' empdata > nbdata
labh@labh-virtual-machine:~$ cat nbdata
Labh 10-10-1986
Aryan 1-2-1981
Dev 22-1-1985
Krasia 10-9-1988
Nidhi 3-2-1982
Prachi 30-11-1985
Anish 5-5-1989
Tumul 22-12-1972
Chile 12-12-1991
Smit 19-3-1973
Pratham 30-11-1981
```

B)

Create a file student with following fields:-

Fieldname	**Datatype**	**Value**
Student code	numeric	

Student name	character	
Batch code	character	B11-B15
No. of modules	numeric	1-5
Average marks	numeric	

Insert at least five records in above file; ‘:’ is used as a field separator. Give commands for the following:-

1. **Display the details of student in order of their names ignoring case.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" '{print $0}' student | sort -t":" -k 2 -f
   S12:Abel:B11:4:50
   S5:Abel:B14:2:45
   S9:alLy:B13:4:77
   S1:An:B11:5:70
   S4:anika:B13:1:30
   S7:Hana:B13:5:90
   S2:Joe:B12:2:40
   S10:Liya:B11:1:50
   S11:Naina:B14:3:65
   S3:Naina:B15:4:60
   S6:Sandra:B12:3:66
   S8:Yukta:B15:3:57

2. **Display the details of students whose number of modules is greater than 3.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" '$4>3 {print $0}' student
   S1:An:B11:5:70
   S3:Naina:B15:4:60
   S7:Hana:B13:5:90
   S9:alLy:B13:4:77
   S12:Abel:B11:4:50

3. **Store the list of rank holders in file ‘merit’ along with student code and student name, and marks & display its contents.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" '{print $1,$2,$5}' student | sort -t" " -k 3 -nr > merit
   labh@labh-virtual-machine:~/Desktop$ cat merit
   S7 Hana 90
   S9 alLy 77
   S1 An 70
   S6 Sandra 66
   S11 Naina 65
   S3 Naina 60
   S8 Yukta 57
   S12 Abel 50
   S10 Liya 50
   S5 Abel 45
   S2 Joe 40
   S4 anika 30

4. **Count the number of students in batch B13.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" '$3=="B13"{count++}END{printf "The number of students in batch 13 is %d\n",count}' student
   The number of students in batch 13 is 3

5. **Display the names of students with same names.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" 'n=x[$2] {print n"\n"$2}{x[$2]=$2}' student
   Naina
   Naina
   Abel
   Abel

6. **Display the students belonging to batch codes B12 or B15.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" '$3~/B12|B13/ {printf "Student: %s\tBatch:%s\n",$2,$3}' student
   Student: Joe Batch:B12
   Student: anika Batch:B13
   Student: Sandra Batch:B12
   Student: Hana Batch:B13
   Student: alLy Batch:B13

7. **Display all the names not starting with ‘a’ or ‘A’.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf "Names not starting with 'a' or 'A':\n"} $2~/^[^aA]/ {printf "%s\n",$2}' student
   Names not starting with a or A:
   Joe
   Naina
   Sandra
   Hana
   Yukta
   Liya
   Naina

8. **Display all the names starting with ‘a’ or ‘A’.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf "Names starting with 'a' or 'A':\n"} $2~/^[aA]/ {printf "%s\n",$2}' student
   Names starting with a or A:
   An
   anika
   Abel
   alLy
   Abel

9. **Display and count the number of students having marks in the range 40 to 60. Also display the total and average marks.**
   labh@labh-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf"Students who scored marks between 40 and 50:\n"}$5>=40 && $5<=50 {print $0;count++;total+=$5;avg=total/count}END{printf "Count: %d\nTotal marks: %d\nAverage Marks: %d\n",count,total,avg}' student
   Students who scored marks between 40 and 50:
   S2:Joe:B12:2:40

S5:Abel:B14:2:45
S10:Liya:B11:1:50
S12:Abel:B11:4:50
Count: 4
Total marks: 185
Average Marks: 46

10. Display the student's records from line number 2 to 4.

labh@labh-virtual-machine:~/Desktop$ awk -F":" 'FNR>=2 && FNR<=4 {print $0}' student
S2:Joe:B12:2:40
S3:Naina:B15:4:60
S4:anika:B13:1:30

11. Display the student's records that are having number of fields 5.

labh@labh-virtual-machine:~/Desktop$ awk -F":" 'NF==5 {print $0}' student
S1:An:B11:5:70
S2:Joe:B12:2:40
S3:Naina:B15:4:60
S4:anika:B13:1:30
S5:Abel:B14:2:45
S6:Sandra:B12:3:66
S7:Hana:B13:5:90
S8:Yukta:B15:3:57
S9:alLy:B13:4:77
S10:Liya:B11:1:50
S11:Naina:B14:3:65
S12:Abel:B11:4:50

12. Display the student's records that are having number of fields less than or equal to 4.

labh@labh-virtual-machine:~/Desktop$ awk -F":" 'NF<=4 {print $0}' student
S13:Jane:4
S14:Rene:B12:1

13. Display the student code, student name and marks that are having number of fields greater than 5.

labh@labh-virtual-machine:~/Desktop$ awk -F":" 'NF>5 {print $1,$2,$5}' student
S15 Jenna 65

14. Display the student's name having the length greater than 3.

labh@labh-virtual-machine:~/Desktop$ awk -F":" '$2~/..../ {print $2}' student
or
labh@labh-virtual-machine:~/Desktop$ awk -F":" 'length($2)>3 {print $2}' student
Naina
anika
Abel
Sandra
Hana
Yukta
alLy

Liya
Naina
Abel
Jane
Rene
Jenna

15. Display the student's records having the length of student name less than or equal to 3.

```
labh@labh-virtual-machine:~/Desktop$ awk -F":" 'length($2)<=3 {print $0}' student
S1:An:B11:5:70
S2:Joe:B12:2:40
```

16. Display the student's records having the length less than or equal to 15.

```
labh@labh-virtual-machine:~/Desktop$ awk -F":" 'length($0)<=15 {print $0}' student
S1:An:B11:5:70
S2:Joe:B12:2:40
S13:Jane:4
S14:Rene:B12:1
```

17. Display the student's records having the length greater than or equal to 15.

```
labh@labh-virtual-machine:~/Desktop$ awk -F":" 'length($0)>=15 {print $0}' student
S2:Joe:B12:2:40
S3:Naina:B15:4:60
S4:anika:B13:1:30
S5:Abel:B14:2:45
S6:Sandra:B12:3:66
S7:Hana:B13:5:90
S8:Yukta:B15:3:57
S9:alLy:B13:4:77
S10:Liya:B11:1:50
S11:Naina:B14:3:65
S12:Abel:B11:4:50
S15:Jenna:B14:1:65:Marbles
```

18. Display the student's records having the length is in the range 5 to 15.

```
labh@labh-virtual-machine:~/Desktop$ awk -F":" 'length($0)>=5 && length($0)<=15 {print $0}'
student
S1:An:B11:5:70
S2:Joe:B12:2:40
S13:Jane:4
S14:Rene:B12:1
```

19. Display the line number and index having character 'b' in student name.

```
labh@labh-virtual-machine:~/Desktop$ awk -F":" '$2~/b/ {print NR" "$0" Index:" index($2,"b")}'
student
5 S5:Abel:B14:2:45 Index:2
12 S12:Abel:B11:4:50 Index:2
```

20. Display the line number, student name and index having character 'b' in student name.

```
labh@labh-virtual-machine:~/Desktop$ awk -F":" '$2~/b/ {print NR" "$2" Index:" index($2,"b")}'
student
```

5 Abel Index:2
12 Abel Index:

21. **Display the index, student name and marks having character 'b' in student name.**

```
labh@labh-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf "Line Number:\t Name:\t
Marks:\t\n"}$2~/b/ {print NR" \t\t "$2,"\t",$5}' student
```

Line Number:	Name:	Marks:
5	Abel	45
12	Abel	50

22. **Display the line number, index, student name and marks having character 'b' in student name.**

```
labh@labh-virtual-machine:~/Desktop$ awk -F":" 'BEGIN{printf "Line Number:\t Name:\t Marks:\t
Index of 'b' in Name:\n"}$2~/b/ {print NR" \t\t "$2,"\t",$5,"\t",index($2,"b")}' student
```

Line Number:	Name:	Marks:	Index of b in Name:
5	Abel	45	2
12	Abel	50	2

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 8

Shell scripting

Theory:

What is a Shell?

A shell is a special user program which provides an interface to a user to use operating system services. Shell accept human readable commands from a user andconvert them into something which the kernel can understand. It is a command language interpreter that executes commands read from input devices such as keyboards or from files.

Shell can be accessed by user using a command line interface. A special programcalled Terminal or Command Prompt is provided to type in the human readable commands such as "cat", "ls" etc. and then it is being execute. The result is then displayed on the terminal to the user.

Usually shells are interactive, but we may need to execute multiple commands together. One can write commands in a file and execute it to avoid repetitive work.These files are shell scripts.
Shell scripts are similar to the batch file in MS-DOS. Each shell script is saved with .sh file extension eg. myscript.sh

It has a syntax just like any other programming language. A shell script comprisesfollowing elements:

1. Shell Keywords – if, else, break etc.

2. Shell commands – cd, ls, echo, pwd, touch etc.
3. Functions
4. Control flow – if..then..else, case and shell loops etc.

To execute a file (say file.sh) one can use any of the two commands:

1. ./file.sh
2. Sh file.sh

Operations that can be performed in the shell scripts:

1. Creating variables (x=10)
2. Reading values from user via "read" command
3. Comparison using -lt, -gt, -le,-ge,-eq, -ne. Eg [$x -lt $y]. Used in loops andconditional statements.

And more….

1. Shell script file for mulitple commands Creating a shell file containing multiple commands

Commands:

```
cal
date
echo Hello World
who
tty
wc kc11
```

Command Prompt –

```
labh@labh-virtual-machine:~/Desktop$ gedit first.sh
labh@labh-virtual-machine:~/Desktop$ ls -l first.sh
-rw-rw-r-- 1 pranav pranav 43 Jul 11 11:50 first.sh
labh@labh-virtual-machine:~/Desktop$ ./first.sh bash: ./first.sh: Permission denied
labh@labh-virtual-machine:~/Desktop$ chmod u+x first.sh
labh@labh-virtual-machine:~/Desktop$ ls -l first.sh
-rwxrw-r-- 1 labh labh 43 Jul 11 11:50 first.sh
labh@labh-virtual-machine:~/Desktop$ ./first.sh
July 2022
Su Mo Tu We Th Fr Sa 1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
Monday 11 July 2022 11:52:11 AM IST
Hello World
labh   tty2         2022-07-11 10:07 (tty2)
/dev/pts/0
3 4 24 kc11
```

Q.2 Write a Shell script file to perform Addition, Subtraction, Multiplication and division of 2 integer numbers

Solution:

File:

```
#Q.2 Write a Shell script file to perform Addition, Subtraction, Multiplication and division of 2 integer numbers
x=50 y=20
echo The Addition of $x and $y is `expr $x + $y` echo The Subtraction of $x and $y is `expr $x - $y` echo The Mul of $x and $y is `expr $x \* $y`
echo The Div of $x and $y is `expr $x / $y`
```

Command Prompt:

```
labh@labh-virtual-machine:~/Desktop$ gedit second.sh
labh@labh-virtual-machine:~/Desktop$ chmod u+x second.sh
labh@labh-virtual-machine:~/Desktop$ ./second.sh

The Addition of 50 and 20 is 70
The Subtraction of 50 and 20 is 30
The Mul of 50 and 20 is 1000
The Div of 50 and 20 is 2


labh@labh-virtual-machine:~/Desktop$ sh second.sh The Addition of 50 and 20 is 70
The Subtraction of 50 and 20 is 30
The Mul of 50 and 20 is 1000
The Div of 50 and 20 is 2
```

*******Taking input from the user**

```
#Q.2 Write a Shell script file to perform Addition, Subtraction, Multiplication and division of 2 integer numbers
echo Enter number 1 read x
echo Enter number 2 read y
echo The Addition of $x and $y is `expr $x + $y` echo The Subtraction of $x and $y is `expr $x - $y` echo The Mul of $x and $y is `expr $x \* $y`
echo The Div of $x and $y is `expr $x / $y`


labh@labh-virtual-machine:~/Desktop$ sh second.sh Enter number 1
12
Enter number 2
6
The Addition of 12 and 6 is 18
The Subtraction of 12 and 6 is 6
The Mul of 12 and 6 is 72
The Div of 12 and 6 is 2
```

Q.3 Same operations as above with decimal numbers. Assign values and take from user

Assigning values:

File:

```
x=12.6 y=6.3
echo Addition of $x and $y : `echo $x + $y | bc` echo Subtraction of $x and $y : `echo $x - $y | bc` echo
```

Multiplication of $x and $y : `echo $x \* $y | bc` echo Division of $x and $y : `echo $x / $y | bc`

Command prompt:
labh@labh-virtual-machine:~/Desktop$ gedit third.sh
labh@labh-virtual-machine:~/Desktop$ sh third.sh Addition of 12.6 and 6.3 : 18.9
Subtraction of 12.6 and 6.3 : 6.3
Multiplication of 12.6 and 6.3 : 79.3
Division of 12.6 and 6.3 : 2

Taking values from user:
File:
echo "Enter value for x: " read x
echo "\nEnter value for y:" read y
echo Addition of $x and $y : `echo $x + $y | bc` echo Subtraction of $x and $y : `echo $x - $y | bc` echo Multiplication of $x and $y : `echo $x \* $y | bc` echo Division of $x and $y : `echo $x / $y | bc`

Command Prompt:
labh@labh-virtual-machine:~/Desktop$ gedit third.sh
labh@labh-virtual-machine:~/Desktop$ sh third.sh Enter value for x:
12.6

Enter value for y:
6.3
Addition of 12.6 and 6.3 : 18.9
Subtraction of 12.6 and 6.3 : 6.3
Multiplication of 12.6 and 6.3 : 79.3
Division of 12.6 and 6.3 : 2

********** **Conditional Statements: For if condition**
Ex:
if test condition then
cmds fi

for else:
if test condition then
cmds else

cmds fi

Q.4 Write a shell script to read a number from the user and check if it's values if >=20
Text editor:
#Conditional Statements echo Enter value for x read x
if test $x -ge 20 then
echo "$x variable value is greater than 20" fi

Command prompt:
-lt, -gt, -le, -ge, -eq, -n

Q.5 Opposite of the above. Greater than 20 or not

Text editor:

```
#Conditional Statements echo Enter value for x read x
if [ $x -ge 20 ] then
echo "$x variable value is greater than 20" else
echo "$x variable is less than 20" fi
```

Commands Prompt:

```
labh@labh-virtual-machine:~/Desktop$ gedit fifth.sh
labh@labh-virtual-machine:~/Desktop$ sh fifth.sh Enter value for x
18
18 variable is less than 20
labh@labh-virtual-machine:~/Desktop$ sh fifth.sh Enter value for x
22
22 variable value is greater than 20
```

Q.6 Shell script to check if number is even or odd

Text editor:

```
#Even odd
echo "Enter the value for x: " read x
k=`expr $x % 2` if [ $k -eq 0 ] then
echo "Number is Even" else
echo "Number is odd" fi
```

Command prompt:

```
labh@labh-virtual-machine:~/Desktop$ gedit sixth.sh
labh@labh-virtual-machine:~/Desktop$ sh sixth.sh Enter the value for x:
12
Number is Even
labh@labh-virtual-machine:~/Desktop$ sh sixth.sh Enter the value for x:
7
Number is odd
```

Q.6 Check number is divisble by 7 or not Text editor:

```
#Div by 5
echo "Enter the value for x: " read x
k=`expr $x % 7` if [ $k -eq 0 ] then
echo "Number is divisble by 7" else
echo "Number is not divisible by 7" fi
```

Command Prompt:

```
echo "Enter the value for x: " read x
k=`expr $x % 2` if [ $k -eq 0 ] then
```

```
echo "Number is Even" else
echo "Number is odd" fi
```

****

Q.7 Write a shell script to read number from user and check if it is positive, negative or equal to 0.

Text editor:

```
echo "Enter the value for x: " read x
if [ $x -gt 0 ] then
echo "$x is positive" else if [ $x -lt 0 ] then
echo "$x is negative" else
echo "$x is equal to 0" fi
fi
```

Commmand prompt:

```
echo "Enter the value for x: " read x
k=`expr $x % 2` if [ $k -eq 0 ] then
echo "Number is Even" else
echo "Number is odd" fi
```

To evaluate multiple comditions:

&& : -a

|| : -o

Q.8 Write a shell script to read number from user and check if it is divisible by 5 and 7 both or only by 5 or by 7 or neither of them

Text editor:

```
echo "Enter a number: " read x
k=`expr $x % 5` m=`expr $x % 7`
if test $k -eq 0 -a $m -eq 0 then
echo "$x is divisible by 5 and 7" elif test $k -eq 0
then
echo "$x is divisble by 5 only" elif test $m -eq 0

then
echo "$x is divisible by 7 only" else
echo "$x is not divisible by 5 or 7" fi
```

Command Prompt:

```
labh@labh-virtual-machine:~/Desktop$ gedit nine.sh
labh@labh-virtual-machine:~/Desktop$ sh nine.sh Enter a number:
35
35 is divisible by 5 and 7
labh@labh-virtual-machine:~/Desktop$ sh nine.sh Enter a number:
15
15 is divisble by 5 only
labh@labh-virtual-machine:~/Desktop$ sh nine.sh Enter a number:
14
```

14 is divisible by 7 only
labh@labh-virtual-machine:~/Desktop$ sh nine.sh Enter a number:
9
9 is not divisible by 5 or 7

Q.9 Display grade of student by taking marks for 3 subjects, calculate percent and print grade A, B, C

Text editor:

```
#grades
echo "Enter value for Maths: " read x
echo "Enter value for Science: " read y
echo "Enter value for English: " read z

k=`expr $x + $y + $z` p=`echo $k / 3 | bc` echo "k : $k"
echo "p: $p"

if test $p -gt 75 then
echo "Grade is A"
elif test $p -gt 50 -a $p -lt 75 then

echo "Grade is B" else
echo "Grade is C" fi
```

Command Prompt:

labh@labh-virtual-machine:~/Desktop$ sh ten.sh Enter value for Maths:
55
Enter value for Science:
55
Enter value for English:
55
k : 165
p: 55
Grade is B
labh@labh-virtual-machine:~/Desktop$ sh ten.sh Enter value for Maths:
35
Enter value for Science:
35
Enter value for English:
35
k : 105
p: 35
Grade is C

Q. 11 Find larget of 3 numbers using nested if else Text editor:

```
echo "Enter x: "
read xecho "Enter y:" read y
```

```
echo "Enter z:" read z

if test $x -gt $y then
if test $x -gt $z then
echo "$x is greatest" else
echo "$z is greatest" fi
elif test $y -gt $z

then
echo "$y is the greatest" else
echo "$z is the greatest" fi
```

Command prompt

```
@labh-virtual-machine:~/Desktop$ gedit eleven.sh
labh@labh-virtual-machine:~/Desktop$ sh eleven.sh Enter x:
23
Enter y:
32
Enter z:
55
55 is the greatest
labh@labh-virtual-machine:~/Desktop$ sh eleven.sh Enter x:
55
Enter y:
32
Enter z:
14
55 is greatest
labh@labh-virtual-machine:~/Desktop$ sh eleven.sh Enter x:
32
Enter y:
55
Enter z:
12
55 is the greatest
```

Q.12 calculate the value of y using following formulas y=2x+5, for x> 0 y=0, for x=0 y=2x-5, for x<0

Input:

```
#Nested conditions echo "Enter x: " read x

mul=`expr $x \* 2` if [ $x -ge 0 ]
then
if [ $x -gt 0 ] then

y=`expr $mul + 5` else
y=0 fi else
y=`expr $mul - 5` fi
```

```
echo "the value of y is $y"
```

Output:

```
labh@labh-virtual-machine:~/Desktop$ gedit twelve.sh
labh@labh-virtual-machine:~/Desktop$ sh twelve.sh Enter x:
2
the value of y is 9
labh@labh-virtual-machine:~/Desktop$ sh twelve.sh Enter x:
-2
the value of y is -9
labh@labh-virtual-machine:~/Desktop$ sh twelve.sh Enter x:
0
the value of y is 0
```

Method 2:

Text Editor:

```
#evaluate y
echo 'Enter the value of x: ' read x
k=`expr 2 \* $x` if test $x -ne 0 then
if test $x -gt 0 then
y=`expr $k + 5` else
y=`expr $k - 5` fi
else y=0 fi
echo value of y: $y
```

Comand prompt:

```
labh@labh-virtual-machine:~/Desktop$ sh twelve1.sh Enter the value of x:

2
value of y: 9
labh@labh-virtual-machine:~/Desktop$ sh twelve1.sh Enter the value of x:
0
value of y: 0
labh@labh-virtual-machine:~/Desktop$ sh twelve1.sh Enter the value of x:
-4
value of y: -13 labh@labh-virtual
```

Q.13 Write a shell script to print month name for corresponding month number from user using case..in structure.

Text editor:

```
echo "Enter moth number: " read m

case $m in
1)echo "January";;
2)echo "February";;
4) echo "April";;
12) echo "December";;
7) echo "July";;
```

```
6) echo "June";;
11) echo "November";;
3) echo "March";;
5) echo "May";;
8)echo "August";;
9)echo "September";;
10)         echo "October";;
*) echo "Please enter a valid month number" esac
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit thirt.sh
labh@labh-virtual-machine:~/Desktop$ sh thirt.sh Enter moth number:
2
February
labh@labh-virtual-machine:~/Desktop$ sh thirt.sh Enter moth number:
13
Please enter a valid month number
labh@labh-virtual-machine:~/Desktop$ sh thirt.sh

Enter moth number:
12
December
```

Q.14 Print weekday name for weekday number:

Text editor:

```
echo "Enter weekday number: " read m

case $m in
1)echo "Monday";;
2)echo "Tuesday";;
4) echo "Thursday";;
7) echo "Sunday";;
3) echo "Wednesday";;
6) echo "Saturday";;
5) echo "Friday";;
*) echo "Please enter a valid weekday number" esac
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit fourteen.sh
labh@labh-virtual-machine:~/Desktop$ sh fourteen.sh Enter weekday number:
99
```

Please enter a valid weekday number
labh@labh-virtual-machine:~/Desktop$ sh fourteen.sh Enter weekday number:
2
Tuesday
labh@labh-virtual-machine:~/Desktop$ sh fourteen.sh Enter weekday number:
1
Monday

Q.15 Write a shell script to perform 1 of the following operations on 2 numbers:
a)add
b)subtract
c)multiply
d)divisible
e)Invalid

Text Editor:

```
echo "Enter value of a: " read a
echo "Enter value of b:" read b
echo "Enter your choice:" read ch

case $ch in
A|a) echo Addition: `expr $a + $b` ;; B|b) echo Subtraction: `expr $a - $b`;; c|C) echo Multiplication: `expr $a \* $b`;; d|D) echo Division: `expr $a / $b`;;
*) echo "Please enter a valid option";; esac
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit fifteen.sh
labh@labh-virtual-machine:~/Desktop$ sh fifteen.sh Enter value of a:
12
Enter value of b:
6
Enter your choice:
a
Addition: 18
labh@labh-virtual-machine:~/Desktop$ sh fifteen.sh Enter value of a:
12
Enter value of b:
6
Enter your choice:
b
Subtraction: 6
labh@labh-virtual-machine:~/Desktop$ sh fifteen.sh Enter value of a:
12
Enter value of b:
6
Enter your choice:
c
```

Multiplication: 72
labh@labh-virtual-machine:~/Desktop$ sh fifteen.sh Enter value of a:
12
Enter value of b:
6
Enter your choice:

d
Division: 2
labh@labh-virtual-machine:~/Desktop$ sh fifteen.sh Enter value of a:
12
Enter value of b:
6
Enter your choice:
f
Please enter a valid option

labh@labh-virtual-machine:~/Desktop$ sh fifteen.sh Enter value of a:
12
Enter value of b:
6
Enter your choice:
A
Addition: 18
labh@labh-virtual-machine:~/Desktop$ sh fifteen.sh Enter value of a:
12
Enter value of b:
6
Enter your choice:
D
Division: 2

Q. 16 Write a shell script to perform one of the following commands, take option from user
a)Create a demo file
b)append content to demo file
c)Display content of demo file
d)Display no of line, words and ch of demo file
Text editor:

```
echo "Enter a choice: " read ch

case $ch in
a|A) touch demo.txt;;
b|B) cat >> demo.txt;;
c|C) cat demo.txt;;
D|d) wc demo.txt;;
*) echo "Please enter a valid option";; esac
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit sixteen.sh
labh@labh-virtual-machine:~/Desktop$ sh sixteen.sh Enter a choice:
a
labh@labh-virtual-machine:~/Desktop$ sh sixteen.sh Enter a choice:
b Hello There
This is the demo file
labh@labh-virtual-machine:~/Desktop$ sh sixteen.sh Enter a choice:
c Hello There
This is the demo file
labh@labh-virtual-machine:~/Desktop$ sh sixteen.sh Enter a choice:
d
3 7 34 demo.txt
```

Q.17 Write a shell script to read a pattern from user and check whether it is
a)starting with lowercase letter
b)ending with lowercase letter
c)starts with a digit
d)ends with a digit
e)ends with a 3 letter word
TE:

```
echo "Enter pattern: "
read p
case $p in
[a-z]*) echo $p pattern starts with a lowercase letter;;
*[a-z]) echo $p pattern ends with a lowercase letter;; [0-9]*) echo $p pattern starts with a digit;;
*[0-9]) echo $p pattern ends with a digit;;
???) echo $p pattern ends with a 3 letter word;;
*) echo "$p does not match any condition";; esac
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit seventeen.sh
labh@labh-virtual-machine:~/Desktop$ sh seventeen.sh Enter pattern:
hello
hello pattern starts with a lowercase letter
labh@labh-virtual-machine:~/Desktop$ sh seventeen.sh Enter pattern:
Hello
Hello pattern ends with a lowercase letter
labh@labh-virtual-machine:~/Desktop$ sh seventeen.sh Enter pattern:
HellOO
HellOO does not match any condition
labh@labh-virtual-machine:~/Desktop$ sh seventeen.sh Enter pattern:
123 abcs
123 abcs pattern ends with a lowercase letter
labh@labh-virtual-machine:~/Desktop$ sh seventeen.sh Enter pattern:
12A
12A pattern starts with a digit
```

labh@labh-virtual-machine:~/Desktop$ sh seventeen.sh Enter pattern:
A12
A12 pattern ends with a digit
labh@labh-virtual-machine:~/Desktop$ sh seventeen.sh Enter pattern:
AAA
AAA pattern ends with a 3 letter word

Q.18Shell script to check favourite color
1.red
2.green
3.blue
4.yellow
5.black
6.white

TE:
echo "Enter your favorite color: " read c

case $c in
[rR][eE][dD]) echo "Your favorite color is red";; [gG][rR][eE][eE][nN]) echo "Color is green";;
[bB][lL][uU][eE]) echo "Color is blue";;
[yY][eE][lL][lL][oO][wW]) echo "Color is yellow";;
[bB][lL][aA][cC][kK]) echo "Color is black";;
[wW][hH][iI][tT][eE]) echo color is White;;

*)echo "Your favourite color is something else. Sorry";; esac

CMD:
labh@labh-virtual-machine:~/Desktop$ sh eighteen.sh Enter your favorite color:
Red
Your favorite color is red
labh@labh-virtual-machine:~/Desktop$ sh eighteen.sh Enter your favorite color:
BluE
Color is blue
labh@labh-virtual-machine:~/Desktop$ sh eighteen.sh Enter your favorite color:
BlAcK
Color is black
labh@labh-virtual-machine:~/Desktop$ sh eighteen.sh Enter your favorite color:
WhItE
color is White
labh@labh-virtual-machine:~/Desktop$ sh eighteen.sh Enter your favorite color:
Purple
Your favourite color is something else. Sorry

***8**
Loops: for, while, until

Q. 19 Write a shell script to print 1st 10 natural numbers using for loop
TE:
```
echo "First 1st natural numbers: " for i in 1 2 3 4 5 6 7 8 9 10
do echo $i done
```

CMD:
```
labh@labh-virtual-machine:~/Desktop$ gedit 19.sh
labh@labh-virtual-machine:~/Desktop$ sh 19.sh First 1st natural numbers:
1
2
3
4
5
6
7
8
9
10
```

TE:
```
echo "First 1st natural numbers: " for i in Hi 10 20.5 Linux
do echo $i done
```

CMD:
```
labh@labh-virtual-machine:~/Desktop$ sh 19.sh First 1st natural numbers:
Hi 10
20.5
Linux
```

TE:
```
echo "First 1st natural numbers: " for((i=1;i<=10;i=i+1))
do echo $i done
```

CMD:
```
labh@labh-virtual-machine:~/Desktop$ gedit 19.sh
labh@labh-virtual-machine:~/Desktop$ bash 19.sh First 1st natural numbers:
1
2
3
4
5
6
7
8
9
10
```

Q.20 display first n natural numbers. Take n from user.

TE:

```
echo Enter x read x
i=1
while [ $i -le $x ] do
echo $i i=`expr $i + 1` done
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 20.sh Enter x
5
1
2
3
4
5
```

******Until**

TE:

```
echo Enter x read x
i=1
until [ $i -gt $x ] do
echo $i i=`expr $i + 1` done
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit 20.sh
labh@labh-virtual-machine:~/Desktop$ sh 20.sh Enter x
5
1
2
3
4
5
```

Q.21 Factorial of number

TE:

```
#factorial
echo "Enter number: " read x
res=1

for((i=2;i<=x;i++)) do
res=`expr $res \* $i` done
echo "Factorial : $res"
```

Command Prompt:

```
labh@labh-virtual-machine:~/Desktop$ bash 21.sh Enter number:
5
Factorial : 120
```

```
labh@labh-virtual-machine:~/Desktop$ bash 21.sh Enter number:
3
Factorial : 6
```

Q.22 Table of factorials upto n

```
#factorial
echo "Enter number: " read x
res=1

for((i=2;i<=x;i++)) do for((j=2;j<=i;j++)) do
res=`expr $res \* $j` done
echo "Factorial of $i : $res" res=1
done

labh@labh-virtual-machine:~/Desktop$ gedit 22.sh
labh@labh-virtual-machine:~/Desktop$ bash 22.sh Enter number:
5
Factorial of 2 : 2
Factorial of 3 : 6
Factorial of 4 : 24
Factorial of 5 : 120
```

Q. 23 Odd and even series up to n

```
echo "Enter limit: " read x
i=1
echo "Even numbers: " while [ $i -le $x ]
do
k=`expr $i % 2` if test $k -eq 0 then
echo $i fi
i=`expr $i + 1` done

i=1
echo "\nOdd numbers: " while [ $i -le $x ]
do
k=`expr $i % 2` if test $k -eq 1 then
echo $i fi
i=`expr $i + 1` done

labh@labh-virtual-machine:~/Desktop$ sh 23.sh Enter limit:
6
Even numbers:
2
4
6

Odd numbers:
```

1
3
5

Q. 24 Fibonacci upto n
TE:
#fibonaci Series echo "Value of n: " read x

a=1 b=1

echo "Fibonacci Series: " echo "$a"
echo "$b"

for((i=3;i<=x;i++)) do
fib=`expr $a + $b` echo "$fib"
a=$b b=$fib done

Command Prompt:
labh@labh-virtual-machine:~/Desktop$ bash 24.sh Value of n:
4
Fibonacci Series:
1
1
2
3
labh@labh-virtual-machine:~/Desktop$ bash 24.sh Value of n:
10
Fibonacci Series:
1
1
2
3
5
8
13
21
34
55

Q.25 Armstrong number TE:
#Armstrong number program echo "Enter a number: " read x
y=$x
while test $x -gt 0 do
val=`expr $x % 10` cu=`expr $val \* $val \* $val` sum=`expr $sum + $cu` x=`expr $x / 10`
done

if test $y -eq $sum then
echo "Given number is Armstrong number" else
echo "Given number is not an Armstrong number" fi

Command Prompt:

```
labh@labh-virtual-machine:~/Desktop$ sh 25.sh Enter a number:
153
Given number is Armstrong number
labh@labh-virtual-machine:~/Desktop$ sh 25.sh Enter a number:
123
Given number is not an Armstrong number
```

Q. 26 Table of squares and cubes upto n TE:

```
#Table of squares and cubes echo "Enter a limit: "
read n

echo -e "\nSquares till $n : " for((i=1;i<=$n;i++))
do
sq=`expr $i \* $i` echo "$sq"
done

echo -e "\nCubes till $n: "

for((i=1;i<=$n;i++)) do
cu=`expr $i \* $i \* $i` echo "$cu"
done
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit 26.sh
labh@labh-virtual-machine:~/Desktop$ bash 26.sh Enter a limit:
6

Squares till 6 :
1
4
9
16
25
36

Cubes till 6:
1
8
27
64
125
216
```

Q.27 Prime or not

TE:

```
#prime check
```

```
echo "Enter a number: " read n
flag=0

for((i=2;i<n;i++)) do
f=`expr $n % $i` if test $f -eq 0 then
flag=1 break fi done

if test $flag -eq 0 then
echo "It is a prime number" else
echo "Not a prime number" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ bash 27.sh Enter a number:
13
It is a prime number
labh@labh-virtual-machine:~/Desktop$ bash 27.sh Enter a number:
14
Not a prime number
labh@labh-virtual-machine:~/Desktop$ bash 27.sh Enter a number:
29
It is a prime number
labh@labh-virtual-machine:~/Desktop$ bash 27.sh Enter a number:
20
Not a prime number
```

Q.28 Prime number series

TE:

```
#prime number series echo "Enter a limit: " read n

echo -e "\nPrime number series: " for((i=4;i<=n;i++))
do flag=0
for((j=2;j<i;j++)) do
f=`expr $i % $j` if test $f -eq 0 then
flag=1 break fi done

if test $flag -eq 0 then

echo "$i" fi
done
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ bash 28.sh Enter a limit:
29

Prime number series:
5
```

7
11
13
17
19
23
29

Q. 29 GCD and LCM of 2 numbers:
TE:

```
#GCD and LCM echo "Enter A: " read a
echo "Enter B: " read b

if test $b -lt $a then
m=$b else m=$a fi

while test $m -ne 0 do
x=`expr $a % $m` y=`expr $b % $m`
if test $x -eq 0 -a $y -eq 0 then
echo "GCD of $a and $b: $m" break
fi
m=`expr $m - 1` done

#lcm
mul=`expr $a \* $b` res=`expr $mul / $m`
echo "LCM of $a and $b: $res"
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit 29.sh
labh@labh-virtual-machine:~/Desktop$ sh 29.sh Enter A:
15
Enter B:
20
GCD of 15 and 20: 5
LCM of 15 and 20: 60
```

********** File commands**

Q.1 shell script to read file from user and check if it exists. If exists, display contents, else display an error msg. “Invalid file”

TE:

```
labh@labh-virtual-machine:~/Desktop$ sh 30.sh Enter filename:
kc9
kc9 exists Contents:
this is some other content Some other content!!
A new Line here!
```

labh@labh-virtual-machine:~/Desktop$ sh 30.sh Enter filename:
employee
File does not exists

CMD:

```
echo "Enter filename: " read x
if [ -f $x ] then
echo "$x exists" echo "Contents: " cat $x
else
echo "File does not exists" fi
```

Q.2 Same for directory

TE:

```
echo "Enter folder name: " read x
if [ -d $x ] then
echo "$x exists" echo "Contents: " ls $x
else
echo "Folder does not exists" fi
```

CMD:

Enter folder name:
d
Folder does not exists
labh@labh-virtual-machine:~/Desktop$ sh 31.sh Enter folder name:
d2
d2 exists Contents:
a aa ac e2 f1 months.txt pat1
a1 aa1 d11 e3 f2 mya1 za
a2 aaa1 e1 employee1.txt kc2 numbers zc a3 ab e1.txt employee.txt kc3 p1
labh@labh-virtual-machine:~/Desktop$ sh 31.sh Enter folder name:
kc9
Folder does not exists

Q.3 Check if ordinary file or directory file.
If ordinary, check words and other count. Else check directory content.

TE:

```
echo "Enter folder/file name: " read x
if [ -d $x ] then
echo "$x is a folder" echo "Contents: "
ls $x
elif [ -f $x ]

then
echo "$x is a file" cat $x
else
```

```
echo "Folder/File does not exists" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 32.sh Enter folder/file name:
kc9
kc9 is a file
this is some other content Some other content!!
A new Line here!
labh@labh-virtual-machine:~/Desktop$ sh 32.sh Enter folder/file name:
d
Folder/File does not exists
labh@labh-virtual-machine:~/Desktop$ sh 32.sh Enter folder/file name:
d2
d2 is a folder Contents:
a   aa      ac      e2      f1 months.txt pat1
a1 aa1 d11 e3    f2   mya1       za
a2 aaa1 e1       employee1.txt kc2 numbers zc a3 ab e1.txt employee.txt kc3 p1
```

Q.4 Check if file has read permission or not. If read, then print content, else print invalid. ?

TE:

```
#Read file
echo "Enter filename: " read x

if [ -f $x ] then
if [ -r $x ] then
cat $x else
echo "File does not have read permission" fi
else
echo "File does not exists" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 33.sh Enter filename:
kc9
this is some other content Some other content!!
A new Line here!
labh@labh-virtual-machine:~/Desktop$ sh 33.sh Enter filename:
kc
File does not exists
```

********** -w = To check for write permission.**

Q.3 Check write permission, and append

TE:

```
echo "Enter the filename: " read x
```

```
if [ -f $x ] then
if [ -w $x ] then
echo "File exists with write permission" cat $x
cat >> $x cat $x else
echo "No write permission" fi
else
echo "The file does not exists" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 34.sh Enter the filename:
kc10
File exists with write permission 1231232
LSA
Linux commands

wc uniq
DevOps Project dev SPM
DWM KNN
this is some other content Some other content!!
A new Line here!
this is some other content Some other content!!
A new Line here!
This is some appended content to the original text in the file. Enjoy! 1231232
LSA
Linux commands wc
uniq DevOps Project dev SPM
DWM KNN
this is some other content Some other content!!
A new Line here!
this is some other content Some other content!!
A new Line here!
This is some appended content to the original text in the file. Enjoy!
```

******** -x : Check execute permission.**

Q.4 Same with execute permission

TE:

```
echo "Enter file: " read x

if [ -f $x ] then
if [ -x $x ] then
echo "File exists with the execute permission" gcc $x
./a.out else

echo "File has no execute permission" fi
else
echo "File does not exists" fi
```

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 35.sh Enter file:
p1.c
File exists with the execute permission Todays LSA Lecture

Q.5 Shell script to display the list of all ordinary files in currency directory
TE:
#Display all ordinary files. echo "List of all ordinary files: "

for i in * do
if [-f $i] then echo "$i" fi
done

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 36.sh List of all ordinary files:
19.sh 20.sh 21.sh 22.sh 23.sh 24.sh 25.sh

Q.6 check directory files in curr dir
TE:
#Display all ordinary files. echo "List of all directories: "

for i in *
do
if [-d $i]
then echo "$i"
fi
done

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 37.sh
List of all directories:
d2
d3
d4
d5
d7

Q.7 Count no of files and dir
TE:
x=0 y=0

for i in * do
if [-f $i] then
x=`expr $x + 1` else

```
y=`expr $y + 1` fi
done

echo "No of files: $x"
echo "No of directories: $y"
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 38.sh No of files: 55
No of directories: 5
```

Q.8 Write a shell script 2 filenames and check if the contents are same or not TE:

CMD:

Q.9 Shell script to read filename from user and check if it exists or not, and is non-empty. Count no of words, lines and character of file else error

TE:

```
echo "Enter a filename: " read x
if [ -f $x ] then
if [ -s $x ] then
echo `wc $x` else
echo "File is empty" fi
else
echo "File does not exists" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 40.sh Enter a filename:
kc22
File is empty
labh@labh-virtual-machine:~/Desktop$ sh 40.sh Enter a filename:
kc9
3 12 65 kc9
```

Q.10 Write a shell Script to read directory from user and check whether it is empty or not. If not empty then display it's contents with long listing.

TE:

```
echo "Enter directory: " read x


if [ -d $x ] then

y=`ls $x | wc -l` if [ $y -eq 0 ] then
echo "Directory is empty" else
ls -l $x fi
else
echo "Not a directory" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 41.sh Enter directory:
d2
total 84
-rw-rw-r-- 1 labh labh     0 Jun 29 12:23 a

labh@labh-virtual-machine:~/Desktop$ ls d3 kc2 kc3
labh@labh-virtual-machine:~/Desktop$ mkdir d8
labh@labh-virtual-machine:~/Desktop$ sh 41.sh Enter directory:
d8
Directory is empty
```

********** **String commands**

-n : Length = 0
-z : String is empty
-s : File is empty

Q. Read 2 strings from user and check if same. TE:

```
echo "String 1: " read x
echo "String 2: " read y

if [ "$x" = "$y" ] then
echo "Strings are same" else
echo "Strings are different" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 42.sh String 1:
Hello String 2:
Hello
Strings are same
labh@labh-virtual-machine:~/Desktop$ sh 42.sh String 1:
Hi String 2:
Bye
Strings are different
```

Q.2 Check if strings are greater than, less than or equal to each other

TE:

```
echo "String 1: " read x
echo "String 2: " read y

if [ "$x" != "$y" ] then
echo "Strings are different" if [ "$x" \> "$y" ]
then
echo "String 1 greater than String 2" else
```

```
echo "String 1 less than String 2" fi
else

echo "Strings are same" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 43.sh String 1:
hu String 2:
hello
Strings are different
String 1 greater than String 2
labh@labh-virtual-machine:~/Desktop$ gedit 43.sh
labh@labh-virtual-machine:~/Desktop$ sh 43.sh String 1:
hello String 2:
hi
Strings are different String 1 less than String 2
```

Q.3 Write a script to read string and check if it is empty or not

TE:

```
echo "Enter a string: " read x

if [ -z $x ] then
echo "String is empty" else
echo "String is not empty" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ gedit 44.sh
labh@labh-virtual-machine:~/Desktop$ sh 44.sh Enter a string:

String is empty
labh@labh-virtual-machine:~/Desktop$ gedit 44.sh
labh@labh-virtual-machine:~/Desktop$ sh 44.sh Enter a string:
hello
String is not empty
```

Q.4 Check if string not empty

TE:

```
echo "Enter a string: " read x

if [ -n "$x" ] then
echo "String is not empty" else
echo "String is empty"
fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 45.sh Enter a string:
```

String is empty
labh@labh-virtual-machine:~/Desktop$ sh 45.sh Enter a string:
h
String is not empty

Q.5 Shell script to read string and print length of the string
TE:
echo "Enter a String: " read x
len=${#x}
echo "Length of string: $len"

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 46.sh Enter a String:
hello
Length of string: 5
labh@labh-virtual-machine:~/Desktop$ sh 46.sh Enter a String:

Length of string: 0

Q.6 Concatenate “n” strings taken from the user.
TE:
echo "String 1: " read x

echo "String 2: " read y

echo "String 3: " read z

a=xy$z b="$x $y $z"

echo "Concatenated String via Method 1: $a" echo "Concatenated Strings via Method 2: $b"

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 47.sh String 1:
hi
String 2:
hi
String 3:
hi
Concatenated String via Method 1: hihihi Concatenated Strings via Method 2: hi hi hi

****
${x:7} : Print after 7th character.

Q.7 Read string and print extracted character
TE:

```
echo "String: " read x
echo "Output 1" ${x:7} echo "Output 2" ${x:0:7}
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ bash 48.sh String:
Hello World!!
Output 1 orld!!
Output 2 Hello W
```

Q.8 Array of Strings.

TE:

```
a=("Hello" "Wolrd" "Bye") echo ${a[@]}
echo ${a[1]}

s=(20.5 "Linux" 100)
echo ${s[@]} b="Linux"

c=("Welcome" 2022 $b "Fun") echo ${c[@]}
echo ${c[2]}
echo ${s[1]}
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ bash 49.sh Hello Wolrd Bye
Wolrd
20.5 Linux 100
Welcome 2022 Linux Fun Linux
Linux
```

Q. 9 Read a string from user and print all even and odd characters

TE:

```
echo "Enter string: " read a

echo "Even characters: " for((i=0; i<${#a}; i++))
do
m=`expr $i % 2` j=`expr $i + 1 ` if [ $m -eq 0 ] then b=${a:$i:$j}
echo ${b:0:1} fi
done

echo -e "\nOdd characters: " for((i=0; i<${#a}; i++))
do
m=`expr $i % 2` j=`expr $i + 1 ` if [ $m -eq 1 ] then b=${a:$i:$j}
echo ${b:0:1} fi
done
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ bash 50.sh Enter string:
Hello World Even characters:
```

H
l
o
W
r
d

Odd characters:
e
l

o
l

Q. 10 Read a string and delete the shortest substring.
TE:
echo "String: " read x

echo "Deleting "
echo ${x#*.} echo ${x%.*}
echo "Deleting longest substring" echo ${x##*.}
echo ${x%%.*}

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 51.sh String:
Hello World. I Robot. Longest Subtringss Deleting ....
I Robot. Longest Subtringss Hello World. I Robot Deleting longest substring Longest Subtringss Hello World

Q. 11 Count no of users logged in
TE:
echo "Users logged in: "
echo `who | wc -l`

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 52.sh
Users logged in:
1

****
Command line args . Can take 9 values only, max
$0 - filename
$1 - 1st command line arg
$* - All argos
$# - No of argos

$@
${11} : To take other than 9 args.

Q. 12 Display name of file using command line args
TE:
```
echo "Name of file: $0"
echo "1st cmd args: $1"
echo "2nd cmd args: $2"
echo "3rd cmd args: $3"
echo "4rth cmd args: $4"
echo "5th cmd args: $5"
echo "6th cmd args: $6"
echo "7th cmd args: $7"
echo "8th cmd args: $8"
echo "9th cmd args: $9"
echo "No of cmd args: $#"
echo "All cmd args: $*"
```

CMD:
```
labh@labh-virtual-machine:~/Desktop$ sh 53.sh 1 2 3 4 5 6 7 8 9 10 122
231232
Name of file: 53.sh 1st cmd args: 1 2nd cmd args: 2 3rd cmd args: 3

4rth cmd args: 4 5th cmd args: 5 6th cmd args: 6 7th cmd args: 7 8th cmd args: 8 9th cmd args: 9
No of cmd args: 12
All cmd args: 1 2 3 4 5 6 7 8 9 10 122 231232
```

Q.13 Rename file to another filename using cmd args
TE:
```
echo "Source file: $1" mv
$1 $2
echo "New filename: $2"
cat $2
```

CMD:
```
labh@labh-virtual-machine:~/Desktop$ sh 54.sh kc19 kc9 Source file: kc19
New filename: kc9
this is some other content
Some other content!!
A new Line here!
```

Q.14 Menu driven program to perform operations on files:
1. **Copy source file to target**
2. **rename file**
3. **delete file**
4. **exit**

TE:

```
while [ true ] do
echo "1. Copy Source to target file" echo "2. Rename source to target file" echo "3. Delete a file"
echo "4. Exit"
echo -e "\nEnter your choice: " read c
case $c in 1)

while [ true ] do
echo Enter source file read s
if [ -f $s ] then
echo $s source file exists while [ true ]
do
echo Enter target file read t
if [ -f $t ] then
echo "Target file already exists. Enter a new file" else
cp $s $t
echo "File is copied. Displaying contents: " cat $t
break fi

done
break else
echo "Source file does not exists please enter a valid filename: " fi
done;;
2)
while [ true ] do
echo Enter source file read s
if [ -f $s ] then
echo $s source file exists while [ true ]
do
echo Enter target file read t
if [ -f $t ] then
echo "Target file already exists. Enter a new file" else
mv $s $t
echo "File is renamed. Displaying contents... " cat $t
break fi

done
break else
echo "Source file does not exists please enter a valid filename: " fi
done;;

3)
while [ true ] do
echo Enter source file read s
if [ -f $s ] then
echo $s File exists echo "Deleting file   "
rm $s
break else
echo "File does not exists please enter a valid filename. " fi
done;;
```

4) echo "Exiting "
exit;; esac done

CMD:

labh@labh-virtual-machine:~/Desktop$ sh 55.sh
1. Copy Source to target file
2. Rename source to target file
3. Delete a file
4. Exit
-e
Enter your choice:
3
Enter source file kc22
kc22 File exists Deleting file .....
1. Copy Source to target file
2. Rename source to target file
3. Delete a file
4. Exit
-e
Enter your choice:

4
Exiting ....

****

$? - exit status of last command 0 - Success
1 - Error

$$ - The process ID of the current shell. For shell scripts, this is the process ID under which they are executing.

$! - The process number of the last background command.

labh@labh-virtual-machine:~/Desktop$ gedit 55.sh
labh@labh-virtual-machine:~/Desktop$ echo $?
0
labh@labh-virtual-machine:~/Desktop$ cat kc22 cat: kc22: No such file or directory
labh@labh-virtual-machine:~/Desktop$ echo $? 1
labh@labh-virtual-machine:~/Desktop$ who labh tty2 2022-07-26 10:00 (tty2)
labh@labh-virtual-machine:~/Desktop$ who -HU who: invalid option -- 'U'
Try 'who --help' for more information.
labh@labh-virtual-machine:~/Desktop$ who -Hu
NAME LINE TIME IDLE PID COMMENT labh tty2 2022-07-26 10:00 00:44
1756 (tty2)
labh@labh-virtual-machine:~/Desktop$ echo $$
2482
labh@labh-virtual-machine:~/Desktop$ echo $!

******************************(()) - Used for integer operations**

```
labh@labh-virtual-machine:~/Desktop$ a=5
labh@labh-virtual-machine:~/Desktop$ echo 5 + 9
5 + 9
labh@labh-virtual-machine:~/Desktop$ a+9 Command 'a+9' not found, did you mean:
command 'a+' from deb aplus-fsf (4.22.1-10.1ubuntu2) Try: sudo apt install <deb name>
labh@labh-virtual-machine:~/Desktop$ echo 5 + 9 $a + 9
5 + 9 5 + 9
labh@labh-virtual-machine:~/Desktop$ echo $((5+9)) 14

labh@labh-virtual-machine:~/Desktop$ echo $(($a+9)) 14
labh@labh-virtual-machine:~/Desktop$ echo $(($a-9))
-4
labh@labh-virtual-machine:~/Desktop$ echo $(($a*9)) 45
labh@labh-virtual-machine:~/Desktop$ echo $(($a*1.1))
bash: 5*1.1: syntax error: invalid arithmetic operator (error token is ".1")
```

****************** Using Logical operators**

TE:

```
read -p "Enter First Numeric Value: " first
read -p "Enter Second Numeric Value: " second

if [ $first -le 10 ] || [ $second -gt 20 ] then
echo "Atleast one conditions is true" else
echo "Both conditions are failed"
fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 56.sh Enter First Numeric Value: 9
Enter Second Numeric Value: 12 Atleast one conditions is true
```

CMD:

```
read -p "Enter First Numeric Value: " first
read -p "Enter Second Numeric Value: " second

if [ $first -le 10 ] && [ $second -gt 20 ] then
echo "Both conditions are true" else
echo "Atleast one conditions is false"
fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 57.sh Enter First Numeric Value: 9

Enter Second Numeric Value: 12 Atleast one conditions is false
```

TE:

```
# A sample shell script to take input a number from the user # Check if the number is between 10 - 20
# Or number is between 100 - 200 read -p "Enter a number: " num
```

```
if ([ $num -ge 10 ] && [ $num -le 20 ]) || ([ $num -ge 100 ] && [ $num -le 200 ]) then
echo "Input number ($num) is between 10-20 or 100-200" else
echo "Input number ($num) is neither between 10-20 nor 100-200"
fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 58.sh Enter a number: 15
Input number (15) is between 10-20 or 100-200
```

****

Functions TE:

```
#Functions: find_average() { sum=0
i=1 len=$#
x=$((len + 1)) while [ $i -lt $x ] do
arg=${!i} sum=$((sum + arg)) i=$((i + 1 ))
done
avg=$((sum / len)) return $avg
}

find_average 10 20 30 40 echo $?
```

CMD:

```
25
```

TE:

```
check_negative()
{
a=5 while :
do
if [ $a -lt 0 ]; then
echo "terminating the script" exit 5
fi
a=$((a - 1)) done
}

a=0 add(){ sum=0 i=1 len=$#
x=$((len + 1)) while [ $i -lt $x ] do
arg=${!i} sum=$((sum + arg)) i=$((i + 1 ))
done a=$sum
}
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ bash 59.sh terminating the script
labh@labh-virtual-machine:~/Desktop$ echo $? 5
```

TE:

```
a=0 add(){

sum=0 i=1 len=$#
x=$((len + 1)) while [ $i -lt $x ] do
arg=${!i} sum=$((sum + arg)) i=$((i + 1 ))
done a=$sum
}

#Calling function add add 5 4 9 1
echo $a

add1(){ sum=0 i=1 len=$#
x=$((len + 1)) while [ $i -lt $x ] do
arg=${!i} sum=$((sum + arg)) i=$((i + 1 ))
done
echo $sum
}

#Calling function add1 add1 10 15 25

echo $?
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ bash 59.sh 19
50
0
```

TE:

```
i=1
while [ $i -le 5 ]

do
echo "Linux Shell Script $i" i=$(($i+1))
sleep 5 done echo Bye
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 60.sh Linux Shell Script 1
Linux Shell Script 2 Linux Shell Script 3 Linux Shell Script 4 Linux Shell Script 5 Bye
```

****

String concatenations methods:

TE:

```
echo "**Example of String concatenation by placing side by using +=**" var_4="Now we will use += to
add the next line."
var_4+=" The next line got added along with the previous one!" echo $var_4
echo ""

echo "**Example of String concatenation by using printf**"
```

```
printf -v printf_variable "You can access more articles on bash and shell script at https://www.$1.com"
echo $printf_variable

echo ""
echo "**Example of String concatenation by placing side by use of double quotes**" echo "Input by user as name of website is: $1"
var_3="The website you have reached is: https://www.$1.com" echo $var_3
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ bash 61.sh Hello
**Example of String concatenation by placing side by using +=**
Now we will use += to add the next line. The next line got added along with the previous one!

**Example of String concatenation by using printf**
You can access more articles on bash and shell script at https://www.Hello.com

**Example of String concatenation by placing side by use of double quotes** Input by user as name of website is: Hello
The website you have reached is: https://www.Hello.com
```

****

Aliases in shell

alias -p : List of all aliases

CMD:

```
labh@labh-virtual-machine:~/Desktop/d2$ alias x='cut -f 2,3 employee.txt | sort
-n -k 1 | head -3'
labh@labh-virtual-machine:~/Desktop/d2$ x
3000    f
4000    f
4500    f
```

****

Removing whitespaces in string

TE:

```
echo "Exmaple to show how to remove trailing or leading whitespaces" var="Text with whitespace "
echo "Method 1: Parameter Expansion" var_1=$var
echo "Print with both lead and trail whitespace $var_1 <fullstop>"
var_1_1="${var_1#"${var_1%%[![:space:]]*}"}"
var_1_2="${var_1%"${var_1##*[![:space:]]}"}"
echo "Remove leading whitespace $var_1_1 <fullstop>" echo "Remove trailing whitespace $var_1_2
<fullstop>"
```

CMD:

labh@labh-virtual-machine:~/Desktop/d2$ bash 62.sh Exmaple to show how to remove trailing or leading whitespaces Method 1: Parameter Expansion
Print with both lead and trail whitespace Text with whitespace <fullstop> Remove leading whitespace Text with whitespace <fullstop>
Remove trailing whitespace Text with whitespace <fullstop>

***************Replacing strings:**

TE:

```
echo "****Example to show use of IFS to split a string****" IFS='-'
str="Learn-Bash-From-EduCBA"
echo "The string we are going to split by hyphen '-' is: $str" read -rasplitIFS<<< "$str"
echo "Print out the different words separated by hyphen '-'" for word in "${splitIFS[@]}"; do
echo $word done
echo "Setting IFS back to whitespace" IFS="
echo " "
echo "****Example to show split a string without IFS****" str="Learn,Bash,From,EduCBA"
echo "The string we are going to split by comma ',' is: $str" readarray -d , -t splitNoIFS<<< "$str"
echo "Print out the different words separated by comma "","" for word in "${splitNoIFS[@]}"; do
echo $word done

echo ""
echo "****Example to show split a string without IFS****" str="Learn||Bash||From||EduCBA"
echo "The string we are going to split by double pipe '||' is: $str" delimiter="||"
conCatString=$str$delimiter splitMultiChar=()
while [[ $conCatString ]]; do
splitMultiChar+=( "${conCatString%%"$delimiter"*}" ) conCatString=${conCatString#*"$delimiter"}
done
echo "Print out the different words separated by double pipe '||'" for word in "${splitMultiChar[@]}"; do
echo $word
done
```

CMD:

```
labh@labh-virtual-machine:~/Desktop/d2$ bash 64.sh
****Example to show use of IFS to split a string****
The string we are going to split by hyphen '-' is: Learn-Bash-From-EduCBA Print out the different words separated by hyphen '-'

Learn Bash From EduCBA
Setting IFS back to whitespace

****Example to show split a string without IFS****
The string we are going to split by comma ',' is: Learn,Bash,From,EduCBA Print out the different words separated by comma ","
Learn Bash From EduCBA


****Example to show split a string without IFS****
The string we are going to split by double pipe '||' is: Learn||Bash||From||EduCBA Print out the different
```

words separated by double pipe '||'
Learn Bash From EduCBA

Q.1 Shell Script to display size of file in bytes
TE:
read -p "Enter filename" first if [-f $first]
then
echo "File exists. Size of $first file in bytes: " ls -lh $first | cut -c 28-31
else
echo "File does not exists" fi

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 62.sh Enter filenamekc13
File exists. Size of kc13 file in bytes: 139

Q.2 Find no of characters in string using wc command
TE:
read -p "Enter a filename: " first

if [-f $first] then
echo "File $first exists. No of characters in the file:" wc -c $first
else
echo File does not exists fi

CMD:
labh@labh-virtual-machine:~/Desktop$ sh 63.sh
Enter a filename: kc13
File kc13 exists. No of characters in the file: 139
kc13

Q.3 Compare contents of 2 files and check if same or not
TE:
read -p "Enter file 1: " f read -p "Enter file 2: " s

if [-f $f] then
if [-f $s] then
echo "Both files exists" cmp $f $s
x=`echo $?` if [$x -eq 0] then
echo "Both files have same content" else
echo "Both files do not have same content" fi

```
else
echo "Enter valid filename" fi
else
echo "Enter valid filename" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 64.sh Enter file 1: kc12
Enter file 2: kc12 Both files exists
Both files have same content

labh@labh-virtual-machine:~/Desktop$ sh 64.sh Enter file 1: kc12
Enter file 2: kc13 Both files exists
kc12 kc13 differ: byte 1, line 1
Both files do not have same content
```

Q.4 Write script to read filename from user and convert it's characters to lowercase to uppercase

TE:

```
read -p "Enter filename: " first

if [ -f $first ] then
echo "File exists. Original Contents: " cat
$first
echo "Converting to uppercase..." tr
'[a-z]' '[A-Z]' < $first > second cat
second
else
echo "Invalid file" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 65.sh Enter filename: kc13
File exists. Original Contents: this is some other content Some other content!!
A new Line here! This is the 2nd file Bye!
this is some other content Some other content!!
Converting to uppercase...
THIS IS SOME OTHER CONTENT SOME OTHER CONTENT!!
A NEW LINE HERE! THIS IS THE 2ND FILE BYE!

THIS IS SOME OTHER CONTENT SOME OTHER CONTENT!!
```

Q.5 Read a c program file and compile and execute the same and display output on the screen

TE:

```
read -p "Enter filename: " first
```

```
if [ -f $first ] then
echo "File exists. Compiling and executing now          "
gcc $first
echo "Result: "
./a.out else
echo "Enter valid filename" fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 66.sh Enter filename: p1.c
File exists. Compiling and executing now....
Result:
Todays LSA Lecture
labh@labh-virtual-machine:~/Desktop$ sh 66.sh Enter filename: p2.c
File exists. Compiling and executing now....
Result:
Enter 2 numbers: 12
13
Sum of 12 and 13: 25
```

Q.6 Shell script to call and execute another shell script file into current file

TE:

```
echo "Enter shell script: " read first

if [ -f $first ] then
echo "File exists. Executing.      "
sh $first else
echo "Invalid " fi
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 67.sh Enter shell script:
66.sh
File exists. Executing....
Enter filename: p1.c
File exists. Compiling and executing now....
Result:
Todays LSA Lecture
labh@labh-virtual-machine:~/Desktop$ sh 67.sh Enter shell script:
68.sh Invalid
```

****

-nt : checks if file is newer than other. True if newer

-ot : Reverse of above

Q.7 Read 2 files and check which file is newer than other file

TE:

```
read -p "Enter file 1: " first read -p "Enter file 2: " second
if [ -f $first ] then
if [ -f $second ] then
echo "Exists"
if [ $first -nt $second ] then
echo "$first is newer than $second" else
echo "$second is newer than $first" fi
else
echo "Not available" fi
else
echo "Not available" fi
echo "checking older" if [ -f $first ]
then
if [ -f $second ]
then
echo "Exists"
if [ $second -ot $first ] then
echo "$first is newer than $second" else
echo "$second is newer than $first" fi
else
echo "Not available" fi
else
echo "Not available" fi
```

CMD:

```
Enter file 1: kc13 Enter file 2: kc12 Exists
kc12 is newer than kc13 checking older
Exists
kc12 is newer than kc13
```

Q.8 Read file check owner of file is current user or not

TE:

```
read -p "Enter file: " first x=`echo $USER`

if [ -f $first ] then
echo "Checking  "
#y=`ls -l $first | cut -c 14-19` #echo $y
if [ -G $first ] then
echo "User is group owner. Owner is $USER" else
if [ -O $first ] then
echo "Owner is current user." else
echo "Owner is not current user" fi
fi else
```

CMD:

```
labh@labh-virtual-machine:~/Desktop$ sh 69.sh Enter file: kc13
Checking ....
User is group owner. Owner is labh
```

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 9

Group and User Management

THEORY

There are various options to manage groups and users in linux. Some options that will be looked into are:

1. **groupadd :** To create new groups of users in linux. One can verify the creation by checking the /etc/group file's content

   Options:

-g	Add Group id
-o	Used when duplicate group id is to be entered
-help	Display details of the command

2. **groupmod :** To change the details of a particular group in linux. Again the verification of the /etc/group file can be done to check the changes have been made or not.

   Options:

-g	Add Group id
-o	Used when duplicate group id is to be entered
-n	Change the name of the group
-help	Display details of the command

3. gpasswd : To add users to groups. Can also be used to set the password for the group when no options and only group name is provided as inputs.

Options:

-a	Add a single user to the group at a time

-M	Add multiple users to the group at a time
-d	Remove user from group
-A	Make a user the group administrator

4. **groupdel :** To delete groups
5. **user add :** To add users. Has tons of options with regards to creating users.
   Options:

-u	Add specific user id
-m -d	Used when a different home directory is provided
-g	User with given group id
-e	Set the account expiry date
-G	To add a new user to multiple groups
-M	Creating user without home directory
-s	Creating user with given login shell
-c	Creating user with custom comments
-f	Specifies no of days after which password expires

6. **usermod :** To change details of the user. Same options will be used as with useradd.

Can be used with -L (Lock user account), or U (Unlock user account)

7. **adduser :** To create a user wherein all the data will be asked after the issuance of the command.
8. **chage** : To verify the details of the user. chage -l : Will display the long details of the specified user.
9. **passwd :** To set password for the user
10. **userdel -r :** To delete all users in a group

PRACTICAL

1. **Create a new group account with name “tycsg1”.**

   labh@labh-virtual-machine:~$ sudo groupadd tycsg1

   [sudo] password for tycs-123:

2. **Display the details of group “tycsg1”.**

   labh@labh-virtual-machine:~$ grep tycsg1 /etc/group

   tycsg1:x:1001:

3. **Create new group accounts with names “tycsg2”, “tycsg3”, “tycsg4”, “tycsg5” and “tycsg6”.**

   labh@labh-virtual-machine:~$ sudo groupadd tycsg2

   labh@labh-virtual-machine:~$ sudo groupadd tycsg3

   labh@labh-virtual-machine:~$ sudo groupadd tycsg4

   labh@labh-virtual-machine:~$ sudo groupadd tycsg5

   labh@labh-virtual-machine:~$ sudo groupadd tycsg6

4. **Display the details of all above created groups.**

   labh@labh-virtual-machine:~$ grep ^"tycsg" /etc/group

   tycsg1:x:1001:

   tycsg2:x:1002:

   tycsg3:x:1003:

   tycsg4:x:1004:

   tycsg5:x:1005:

   tycsg6:x:1006:

5. **Create a new group “tycsg8” with specific unique groupid, and then display its details.**

   labh@labh-virtual-machine:~$ sudo groupadd -g 1027 tycsg8

   labh@labh-virtual-machine:~$ grep tycsg8 /etc/group

   tycsg8:x:1027:

6. **Create a new group “tycsg7” with duplicate groupid, and then display its details.**

labh@labh-virtual-machine:~$ sudo groupadd -o -g 1027 tycsg7

labh@labh-virtual-machine:~$ grep ^"tycsg" /etc/group

tycsg1:x:1001:

tycsg2:x:1002:

tycsg3:x:1003:

tycsg4:x:1004:

tycsg5:x:1005:

tycsg6:x:1006:

tycsg8:x:1027:

tycsg7:x:1027:

7. Change the name of group "tycsg1" to "newtycsg1", and then display its details.

labh@labh-virtual-machine:~$ sudo groupmod -n newtycsg1 tycsg1

labh@labh-virtual-machine:~$ grep ^"newtycsg" /etc/group

newtycsg1:x:1001:

8. Change groupid of group “newtycsg1”, and then display its details.

labh@labh-virtual-machine:~$ sudo groupmod -g 1030 newtycsg1

labh@labh-virtual-machine:~$ grep ^"newtycsg" /etc/group newtycsg1:x:1030:

9. Use same groupid for other group “tycsg2” using –o option.

labh@labh-virtual-machine:~$ sudogroupmod -o -g 1030 tycsg2

labh@labh-virtual-machine:~$ grep ^"tycsg2" /etc/group tycsg2:x:1030:

10. Display help message for above commands and its options.

labh@labh-virtual-machine:~$ groupadd --help

Usage: groupadd [options] GROUP

Options:

-f, --force exit successfully if the group already exists,
and cancel -g if the GID is already used
-g, --gid GID use GID for the new group
-h, --help display this help message and exit
-K, --key KEY=VALUE override /etc/login.defs defaults
-o, --non-unique allow to create groups with duplicate
(non-unique) GID
-p, --password PASSWORD use this encrypted password for the new group
-r, --system create a system account
-R, --root CHROOT_DIR directory to chroot into
-P, --prefix PREFIX_DIR directory prefix
--extrausers Use the extra users database

labh@labh-virtual-machine:~$ groupmod -h
Usage: groupmod [options] GROUP

Options:
-g, --gid GID change the group ID to GID
-h, --help display this help message and exit
-n, --new-name NEW_GROUP change the name to NEW_GROUP
-o, --non-unique allow to use a duplicate (non-unique) GID
-p, --password PASSWORD change the password to this (encrypted)
PASSWORD
-R, --root CHROOT_DIR directory to chroot into
-P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files

11. Add user “ty1” in group “tycsg3”, and then display its details.

labh@labh-virtual-machine:~$ sudo useradd ty1

labh@labh-virtual-machine:~$ sudo gpasswd -a ty1 tycsg3

Adding user ty1 to group tycsg3

labh@labh-virtual-machine:~$ grep tycsg3 /etc/group

tycsg3:x:1003:ty1

12. Add multiple users “ty2”, “ty3”, “ty4”, “ty5”, and “ty6” to the group “tycsg3”, and then display its details.

labh@labh-virtual-machine:~$ sudo useradd ty2

labh@labh-virtual-machine:~$ sudo useradd ty3

labh@labh-virtual-machine:~$ sudo useradd ty4

labh@labh-virtual-machine:~$ sudo useradd ty5

labh@labh-virtual-machine:~$ sudo useradd ty6

labh@labh-virtual-machine:~$ sudo gpasswd -M ty2,ty3,ty4,ty5,ty6 tycsg3

labh@labh-virtual-machine:~$ grep tycsg3 /etc/group

tycsg3:x:1003:ty2,ty3,ty4,ty5,ty6

13. Remove user “ty2” from the group “tycsg3”, and then display its details.

labh@labh-virtual-machine:~$ sudo gpasswd -d ty2 tycsg3

Removing user ty2 from group tycsg3

labh@labh-virtual-machine:~$ grep tycsg3 /etc/group

tycsg3:x:1003:ty3,ty4,ty5,ty6

14. Remove multiple users “ty3” and “ty4” from the group “tycsg3”, and then display its details.

labh@labh-virtual-machine:~$ sudo gpasswd -d ty3 tycsg3

Removing user ty3 from group tycsg3

labh@labh-virtual-machine:~$ sudo gpasswd -d ty4 tycsg3

Removing user ty4 from group tycsg3

labh@labh-virtual-machine:~$ grep tycsg3 /etc/group

tycsg3:x:1003:ty5,ty6

15. Make user which is a group member “ty5” of “tycsg3” as a group administrator.

labh@labh-virtual-machine:~$ sudo gpasswd -A ty5 tycsg3

16. Set password for group “tycsg5”.

labh@labh-virtual-machine:~$ sudo gpasswd tycsg5

Changing the password for group tycsg5

New Password:

labh@labh-virtual-machine:~$ sudo grep tycsg5 /etc/gshadow

tycsg5:6VRKQs/AZ$8bYX6O0Kxk8q1c4GsjR4KwXD.9s2tf/rfDC.hOPs.2WNIBj1n2BF2xrmsUbdxcmT0/COLvxF7pJTTAwQ4PbjT1::

17. Display the details of group accounts and their passwords.

labh@labh-virtual-machine:~$ sudo cat /etc/gshadow

root:*::

daemon:*::

bin:*::

sys:*::

adm:*::syslog,tycs-123

tty:*::

disk:*::

lp:*::

mail:*::

news:*::

uucp:*::

man:*::

proxy:*::

kmem:*::

dialout:*::

fax:*::

voice:*::

cdrom:*::tycs-123

floppy:*::

tape:*::

sudo:*::tycs-123

audio:*::pulse

dip:*::tycs-123

www-data:*::

backup:*::

operator:*::

list:*::

irc:*::

src:*::

gnats:*::

shadow:*::

utmp:*::

video:*::

sasl:*::

plugdev:*::tycs-123

staff:*::

games:*::

users:*::

nogroup:*::

systemd-journal:!::

systemd-network:!::

systemd-resolve:!::

crontab:!::

messagebus:!::

systemd-timesync:!::

input:!::

sgx:!::

kvm:!::

render:!::

syslog:!::

tss:!::

bluetooth:!::

ssl-cert:!::

uuidd:!::

systemd-oom:!::

tcpdump:!::

_ssh:!::

avahi-autoipd:!::

netdev:!::

avahi:!::

lpadmin:!::tycs-123

rtkit:!::

whoopsie:!::

sssd:!::

nm-openvpn:!::

scanner:!::saned

saned:!::

colord:!::

geoclue:!::

pulse:!::

pulse-access:!::

gdm:!::

lxd:!::tycs-123

tycs-123:!::

sambashare:!::tycs-123

tycsg2:!::

tycsg3:!:ty5:ty5,ty6

tycsg4:!::

tycsg5:6VRKQs/AZ$8bYX6O0Kxk8q1c4GsjR4KwXD.9s2tf/rfDC.hOPs.2WNIBj1n2BF2xrmsUbdxcmT0/COLvxF7pJTTAwQ4PbjT1::

tycsg6:!::

tycsg8:!::

tycsg7:!::

newtycsg1:!::

ty1:!::

ty2:!::

ty3:!::

ty4:!::

ty5:!::

ty6:!::

18. Display help message for above command and its options.

labh@labh-virtual-machine:~$ gpasswd -h

Usage: gpasswd [option] GROUP

Options:

-a, --add USER add USER to GROUP

-d, --delete USER remove USER from GROUP

-h, --help display this help message and exit

-Q, --root CHROOT_DIR directory to chroot into

-r, --remove-password remove the GROUP's password

-R, --restrict restrict access to GROUP to its members

-M, --members USER,... set the list of members of GROUP

--extrausers use the extra users database

-A, --administrators ADMIN,...set the list of administrators for GROUP

Except for the -A and -M options, the options cannot be combined.

19. Delete the groups “newtycsg1” and “tycsg2”

labh@labh-virtual-machine:~$ sudo groupdel newtycsg1

labh@labh-virtual-machine:~$ sudo groupdel tycsg2

labh@labh-virtual-machine:~$ cat /etc/group

root:x:0:

daemon:x:1:

bin:x:2:

sys:x:3:

adm:x:4:syslog,tycs-123

tty:x:5:

disk:x:6:

lp:x:7:

mail:x:8:

news:x:9:

uucp:x:10:

man:x:12:

proxy:x:13:

kmem:x:15:

dialout:x:20:

fax:x:21:

voice:x:22:

cdrom:x:24:tycs-123

floppy:x:25:

tape:x:26:

sudo:x:27:tycs-123

audio:x:29:pulse

dip:x:30:tycs-123

www-data:x:33:

backup:x:34:

operator:x:37:

list:x:38:

irc:x:39:

src:x:40:

gnats:x:41:

shadow:x:42:

utmp:x:43:

video:x:44:

sasl:x:45:

plugdev:x:46:tycs-123

staff:x:50:

games:x:60:

users:x:100:

nogroup:x:65534:

systemd-journal:x:101:

systemd-network:x:102:

systemd-resolve:x:103:

crontab:x:104:

messagebus:x:105:

systemd-timesync:x:106:

input:x:107:

sgx:x:108:

kvm:x:109:

render:x:110:

syslog:x:111:

tss:x:112:

bluetooth:x:113:

ssl-cert:x:114:

uuidd:x:115:

systemd-oom:x:116:

tcpdump:x:117:

_ssh:x:118:

avahi-autoipd:x:119:

netdev:x:120:

avahi:x:121:

lpadmin:x:122:tycs-123

rtkit:x:123:

whoopsie:x:124:

sssd:x:125:

nm-openvpn:x:126:

scanner:x:127:saned

saned:x:128:

colord:x:129:

geoclue:x:130:

pulse:x:131:

pulse-access:x:132:

gdm:x:133:

lxd:x:134:tycs-123

tycs-123:x:1000:

sambashare:x:135:tycs-123

tycsg3:x:1003:ty5,ty6

tycsg4:x:1004:

tycsg5:x:1005:

tycsg6:x:1006:

tycsg8:x:1027:

tycsg7:x:1027:

ty1:x:1001:

ty2:x:1031:

ty3:x:1032:

ty4:x:1033:

ty5:x:1034:

ty6:x:1035:

20. Display help message for above command and its options.

labh@labh-virtual-machine:~$ groupdel -h

Usage: groupdel [options] GROUP

Options:

-h, --help display this help message and exit

-R, --root CHROOT_DIR directory to chroot into

-P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files

-f, --force delete group even if it is the primary group of a user

--extrausers Use the extra users database

21. Create a new user in “ty1a”.

labh@labh-virtual-machine:~$ sudo useradd ty1a

22. Display the details of user “ty1a”.

labh@labh-virtual-machine:~$ grep ty1a /etc/passwd

ty1a:x:1007:1007::/home/ty1a:/bin/sh

23. Create a new user “ty4a” with different home directory.

labh@labh-virtual-machine:~$ sudo useradd -m -d /home/ty2aa ty4a

24. Display the details of user “ty4a”.

labh@labh-virtual-machine:~$ ls /home

ty2aa tycs-123

25. Create a new user “ty3b” with specific user-id.

labh@labh-virtual-machine:~$ sudo useradd -u 1040 ty3b

26. Display the details of user “ty3b”.

labh@labh-virtual-machine:~$ grep ty3b /etc/passwd

ty3b:x:1040:1040::/home/ty3b:/bin/sh

27. Create a new user “ty4b” with specific group-id.

labh@labh-virtual-machine:~$ sudo useradd -g 1040 ty4b

28. Display the details of user “ty4b”.

labh@labh-virtual-machine:~$ grep ty4b /etc/passwd

ty4b:x:1041:1040::/home/ty4b:/bin/sh

29. Create a new user “ty5a” to multiple groups “tycsg3”, “tycsg4” and “tycsg5”.

labh@labh-virtual-machine:~$ sudo useradd -G tycsg3,tycsg4,tycsg5 ty5a

[sudo] password for tycs-123:

30. Verify that the multiple groups assigned to the user “ty5a”.

labh@labh-virtual-machine:~$ grep ty5a /etc/group

tycsg3:x:1003:ty5,ty6,ty5a

tycsg4:x:1004:ty5a

tycsg5:x:1005:ty5a

ty5a:x:1042:

31. Create a new user “ty6a” without home directory.

labh@labh-virtual-machine:~$ sudo useradd -M ty6a

32. Display the details of user “ty6a”.

labh@labh-virtual-machine:~$ ls /home

ty2aa tycs-123

33. Create a new user “ty7a” with account expiry date.

labh@labh-virtual-machine:~$ sudo useradd -e ty7a

34. Verify the details of user “ty7a”.

labh@labh-virtual-machine:~$ sudo chage -l ty7a

Last password change : Aug 08, 2022

Password expires : never

Password inactive : never

Account expires : Dec 25, 2022

Minimum number of days between password change : 0

Maximum number of days between password change : 99999

Number of days of warning before password expires : 7

35. Create a new user “ty8a” with password expiry date by defining the number of days after a password expires.

labh@labh-virtual-machine:~$ sudo useradd -e 2022-12-25 -f 30 ty8a

36. Verify the details of user “ty8a”.

labh@labh-virtual-machine:~$ sudo grep ty8a /etc/shadow

ty8a:!:19212:0:99999:7:30:19351:

37. Create a new user “ty9a”user with custom comments. Display its details.

labh@labh-virtual-machine:~$ sudo useradd -c "I am Labh" ty9a

labh@labh-virtual-machine:~$ grep ty9a /etc/passwd

ty9a:x:1046:1046:I am Labh:/home/ty9a:/bin/sh

38. Create a new user “ty10a” with login shell. Display its details.

labh@labh-virtual-machine:~$ sudo useradd -s /bin/sh ty10a

labh@labh-virtual-machine:~$ grep ty10a /etc/passwd

ty10a:x:1047:1047::/home/ty10a:/bin/sh

39. Create a new user “ty11a” with specific home directory, default shell and comment. Display its details.

labh@labh-virtual-machine:~$ sudo useradd -m -d /home/ty11aa -s /bin/bash -c "Ty11 user" ty11a

labh@labh-virtual-machine:~$ grep ty11a /etc/passwd

ty11a:x:1048:1048:Ty11 user:/home/ty11aa:/bin/bash

40. Create a new user “ty12” with home directory, custom shell, comment and UID/GID. Display its details.

labh@labh-virtual-machine:~$ sudo useradd -m -d /home/ty12aa -s /bin/bash -c "Ty12 user" -u 1050 -g 1048 ty12a

labh@labh-virtual-machine:~$ grep ty12a /etc/passwd

ty12a:x:1050:1048:Ty12 user:/home/ty12aa:/bin/bash

41. Create a new user “ty13” with changed shell.

labh@labh-virtual-machine:~$ sudo adduser ty13a

Adding user `ty13a' ...

Adding new group `ty13a' (1002) ...

Adding new user `ty13a' (1009) with group `ty13a' ...

The home directory `/home/ty13a' already exists.  Not copying from `/etc/skel'.

New password:

BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word

Retype new password:

Sorry, passwords do not match.

New password:

BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word

Retype new password:

passwd: password updated successfully

Changing the user information for ty13a

Enter the new value, or press ENTER for the default

Full Name []: ty13user

Room Number []: 1

Work Phone []: 123

Home Phone []: 22

Other []: 69

Is the information correct? [Y/n] Y

labh@labh-virtual-machine:~$ grep ty13a /etc/passwd

ty13a:x:1009:1002:ty13user,1,123,22,69:/home/ty13a:/bin/bash

labh@labh-virtual-machine:~$ sudo usermod -s /bin/sh ty13

labh@labh-virtual-machine:~$ grep ty13a /etc/passwd

ty13a:x:1009:1002:ty13user,1,123,22,69:/home/ty13a:/bin/sh

42. Set password for all above created user accounts.

labh@labh-virtual-machine:~$ sudo passwd ty12a

New password:

Retype new password:

passwd: password updated successfully

43. Display help for above command.

labh@labh-virtual-machine:~$ useradd -h

Usage: useradd [options] LOGIN

useradd -D

useradd -D [options]

Options:

--badnames do not check for bad names

-b, --base-dir BASE_DIR base directory for the home directory of the

new account

--btrfs-subvolume-home use BTRFS subvolume for home directory

-c, --comment COMMENT GECOS field of the new account

-d, --home-dir HOME_DIR home directory of the new account

-D, --defaults print or change default useradd configuration

-e, --expiredate EXPIRE_DATE expiration date of the new account

-f, --inactive INACTIVE password inactivity period of the new account

-g, --gid GROUP name or ID of the primary group of the new account

-G, --groups GROUPS list of supplementary groups of the new account

-h, --help display this help message and exit

-k, --skel SKEL_DIR use this alternative skeleton directory

-K, --key KEY=VALUE override /etc/login.defs defaults

-l, --no-log-init do not add the user to the lastlog and faillog databases

-m, --create-home create the user's home directory

-M, --no-create-home do not create the user's home directory

-N, --no-user-group do not create a group with the same name as the user

-o, --non-unique allow to create users with duplicate (non-unique) UID

-p, --password PASSWORD encrypted password of the new account

-r, --system create a system account

-R, --root CHROOT_DIR directory to chroot into

-P, --prefix PREFIX_DIR prefix directory where are located the /etc/* files

-s, --shell SHELL login shell of the new account

-u, --uid UID user ID of the new account

-U, --user-group create a group with the same name as the user

-Z, --selinux-user SEUSER use a specific SEUSER for the SELinux user mapping

```
        --extrausers            Use the extra users database
```

```
labh@labh-virtual-machine:~$ adduser -h
adduser [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--firstuid ID] [--lastuid ID] [--gecos GECOS] [--ingroup GROUP | --gid ID]
[--disabled-password] [--disabled-login] [--add_extra_groups]
[--encrypt-home] USER
   Add a normal user

adduser --system [--home DIR] [--shell SHELL] [--no-create-home] [--uid ID]
[--gecos GECOS] [--group | --ingroup GROUP | --gid ID] [--disabled-password]
[--disabled-login] [--add_extra_groups] USER
   Add a system user

adduser --group [--gid ID] GROUP
addgroup [--gid ID] GROUP
   Add a user group

addgroup --system [--gid ID] GROUP
   Add a system group

adduser USER GROUP
   Add an existing user to an existing group

general options:
  --quiet | -q      don't give process information to stdout
  --force-badname   allow usernames which do not match the
                    NAME_REGEX[_SYSTEM] configuration variable
```

--extrausers uses extra users as the database

--help | -h usage message

--version | -v version number and copyright

--conf | -c FILE use FILE as configuration file

labh@labh-virtual-machine:~$ passwd -h

Usage: passwd [options] [LOGIN]

Options:

-a, --all report password status on all accounts

-d, --delete delete the password for the named account

-e, --expire force expire the password for the named account

-h, --help display this help message and exit

-k, --keep-tokens change password only if expired

-i, --inactive INACTIVE set password inactive after expiration
to INACTIVE

-l, --lock lock the password of the named account

-n, --mindays MIN_DAYS set minimum number of days before password
change to MIN_DAYS

-q, --quiet quiet mode

-r, --repository REPOSITORY change password in REPOSITORY repository

-R, --root CHROOT_DIR directory to chroot into

-S, --status report password status on the named account

-u, --unlock unlock the password of the named account

-w, --warndays WARN_DAYS set expiration warning days to WARN_DAYS

-x, --maxdays MAX_DAYS set maximum number of days before password
change to MAX_DAYS

44. Change information of user account “ty9a”. Display its details.

labh@labh-virtual-machine:~$ grep ty9a /etc/passwd

ty9a:x:1046:1046:I am Labh:/home/ty9a:/bin/sh

labh@labh-virtual-machine:~$ sudo usermod -c "ty9 user" ty9a

[sudo] password for tycs-123:

labh@labh-virtual-machine:~$ grep ty9a /etc/passwd

ty9a:x:1046:1046:ty9 user:/home/ty9a:/bin/sh

45. Change home directory of user account “ty10” to “ty100” in /home. Display its details.

labh@labh-virtual-machine:~$ grep ty10a /etc/passwd

ty10a:x:1047:1047::/home/ty10a:/bin/sh

labh@labh-virtual-machine:~$ mkdir dir1

labh@labh-virtual-machine:~$ ls

Desktop dir1 Documents Downloads Music Pictures Public snap Templates Videos

labh@labh-virtual-machine:~$ cd dir1

labh@labh-virtual-machine:~/dir1$ pwd

/home/tycs-123/dir1

labh@labh-virtual-machine:~/dir1$ cd ../

labh@labh-virtual-machine:~$ sudo usermod -d /home/tycs-123/dir1 ty10a

labh@labh-virtual-machine:~$ grep ty10a /etc/passwd

ty10a:x:1047:1047::/home/tycs-123/dir1:/bin/sh

labh@labh-virtual-machine:~$ ls /home

ty11aa ty12aa ty13a ty2aa tycs-123 tycs1

labh@labh-virtual-machine:~$ ls /home/tycs-123

Desktop dir1 Documents Downloads Music Pictures Public snap Templates Videos

46. Set expiry date on user account “ty11”.

labh@labh-virtual-machine:~$ sudo chage -l ty11a

Last password change : Aug 08, 2022

Password expires : never

Password inactive : never

Account expires : never

Minimum number of days between password change : 0

Maximum number of days between password change : 99999

Number of days of warning before password expires : 7

labh@labh-virtual-machine:~$ sudo chage -E 2022-10-30 ty11a

labh@labh-virtual-machine:~$ sudo chage -l ty11a

Last password change : Aug 08, 2022

Password expires : never

Password inactive : never

Account expires : Oct 30, 2022

Minimum number of days between password change : 0

Maximum number of days between password change : 99999

Number of days of warning before password expires : 7

47. Verify the details.

labh@labh-virtual-machine:~$ sudo chage -l ty11a

Last password change : Aug 08, 2022

Password expires : never

Password inactive : never

Account expires : Oct 30, 2022

Minimum number of days between password change : 0

Maximum number of days between password change : 99999

Number of days of warning before password expires : 7

48. Change primary group of user “ty12a” to group “ty10a”.

labh@labh-virtual-machine:~$ sudo usermod -g ty10a ty12a

labh@labh-virtual-machine:~$ id ty12a

uid=1050(ty12a) gid=1047(ty10a) groups=1047(ty10a)

49. Add a new group “tycsg8” to an existing user “ty13a”.

labh@labh-virtual-machine:~$ sudo usermod -G tycsg8 ty13a

labh@labh-virtual-machine:~$ id ty13a

uid=1009(ty13a) gid=1002(ty13a) groups=1002(ty13a),1027(tycsg8)

50. Add supplementary group “tycsg7” to user “ty10a” and primary group “ty10a” remains the same.

labh@labh-virtual-machine:~$ sudo usermod -a -G tycsg7 ty12a

labh@labh-virtual-machine:~$ id ty12a

uid=1050(ty12a) gid=1047(ty10a) groups=1047(ty10a),1027(tycsg8)

labh@labh-virtual-machine:~$ grep tycsg7 /etc/group

tycsg7:x:1027:ty12a

51. Change login name of user “ty5” to “ty55”. Verify the details.

labh@labh-virtual-machine:~$ sudo usermod -l ty55a ty5a

labh@labh-virtual-machine:~$ id ty5a

id: ‘ty5a’: no such user

labh@labh-virtual-machine:~$ id ty55a

uid=1042(ty55a) gid=1042(ty5a) groups=1042(ty5a),1003(tycsg3),1004(tycsg4),1005(tycsg5)

52. Lock the user account “ty12a”. Display the details.

labh@labh-virtual-machine:~$ sudo usermod -L ty12a

labh@labh-virtual-machine:~$ sudo grep ty12a /etc/shadow

ty12a:!yj9T$hMiEEbZID2w1unVOGQ2pU.$eictzx9vvX3ruwkSWh4Fb31CHCJZNaXiNwVf/9lJog6:19213:0:99999:7:::

53. Unlock the user account “ty12a”. Display the details.

labh@labh-virtual-machine:~$ sudo usermod -U ty12a

labh@labh-virtual-machine:~$ sudo grep ty12a /etc/shadow

ty12a:yj9T$hMiEEbZID2w1unVOGQ2pU.$eictzx9vvX3ruwkSWh4Fb31CHCJZNaXiNwVf/9lJog6:19213:0:99999:7:::

54. Change user’s default home directory to a new location of user account “ty2a”.

labh@labh-virtual-machine:~$ sudo useradd ty2a

labh@labh-virtual-machine:~$ sudo usermod -d /var/ty2a -m ty2a

labh@labh-virtual-machine:~$ ls /var

backups cache crash lib local lock log mail metrics opt run snap spool tmp

labh@labh-virtual-machine:~$ grep ty2a /etc/passwd

ty4a:x:1008:1008::/home/ty2aa:/bin/sh

ty2a:x:1051:1051::/var/ty2a:/bin/sh

55. Change shell to /bin/zsh of user account “ty10a”.

labh@labh-virtual-machine:~$ sudo usermod -s /bin/zsh ty10a

labh@labh-virtual-machine:~$ grep ty10a /etc/passwd ty10a:x:1047:1047::/home/tycs-123/dir1:/bin/zsh

56. Display the id of user account “ty11a”.

labh@labh-virtual-machine:~$ id ty11a

uid=1048(ty11a) gid=1048(ty11a) groups=1048(ty11a)

57. Change user id of user account “ty11a”.

labh@labh-virtual-machine:~$ id ty11a

uid=1048(ty11a) gid=1048(ty11a) groups=1048(ty11a)

labh@labh-virtual-machine:~$ sudo usermod -u 1084 ty11a

labh@labh-virtual-machine:~$ id ty11a

uid=1084(ty11a) gid=1048(ty11a) groups=1048(ty11a)

58. Modify account details with multiple options of user account “ty12a”.Verify the details.

labh@labh-virtual-machine:~$ sudo usermod -d /var/dir1 -e 2022-10-30 -s /bin/sh -c "Ty12 user" -u 1050 -g 1047 -a -G tycsg7 ty12a

labh@labh-virtual-machine:~$ grep ty12a /etc/passwd

ty12a:x:1050:1047:Ty12 user:/var/dir1:/bin/sh

labh@labh-virtual-machine:~$ id ty12a

uid=1050(ty12a) gid=1047(ty10a) groups=1047(ty10a),1027(tycsg8)

59. Display the original user id and group id of user account “ty13a”.

labh@labh-virtual-machine:~$ id ty13a

uid=1009(ty13a) gid=1002(ty13a) groups=1002(ty13a),1027(tycsg8)

60. Change UID and GID of the user “ty13a”.

labh@labh-virtual-machine:~$ sudo usermod -u 1052 -g 1002 ty13a

61. Now, display the user id and group id of user account “ty13a”.

labh@labh-virtual-machine:~$ id ty13a

uid=1052(ty13a) gid=1002(ty13a) groups=1002(ty13a),1027(tycsg8)

62. Delete all above created user along with its related files and group accounts. STEPS:

i. Delete all users created. Check /etc/passwd to verify if all users are deleted.

ii. Delete all groups created. Check /etc/group to verify if all groups are deleted.

iii. Delete the directories created in /home.

Command for deleting users:

labh@labh-virtual-machine:~$ sudo userdel -r ty1

labh@labh-virtual-machine:~$ sudo userdel -r ty2

labh@labh-virtual-machine:~$ sudo userdel -r ty3

labh@labh-virtual-machine:~$ sudo userdel -r ty4

labh@labh-virtual-machine:~$ sudo userdel -r ty5

labh@labh-virtual-machine:~$ sudo userdel -r ty6

labh@labh-virtual-machine:~$ sudo userdel -r ty1a

labh@labh-virtual-machine:~$ sudo userdel -r ty4a

labh@labh-virtual-machine:~$ sudo userdel -r ty4b

labh@labh-virtual-machine:~$ sudo userdel -r ty6a

labh@labh-virtual-machine:~$ sudo userdel -r ty7a

labh@labh-virtual-machine:~$ sudo userdel -r ty8a

labh@labh-virtual-machine:~$ sudo userdel -r ty9a

labh@labh-virtual-machine:~$ sudo userdel -r ty10a

labh@labh-virtual-machine:~$ sudo userdel -r ty11a

labh@labh-virtual-machine:~$ sudo userdel -r ty13a

labh@labh-virtual-machine:~$ sudo userdel -r ty2a

labh@labh-virtual-machine:~$ sudo userdel -r ty55a

Command for deleting groups:

labh@labh-virtual-machine:~$ sudo groupdel tycsg3

labh@labh-virtual-machine:~$ sudo groupdel tycsg4

labh@labh-virtual-machine:~$ sudo groupdel tycsg5

labh@labh-virtual-machine:~$ sudo groupdel tycsg6

labh@labh-virtual-machine:~$ sudo groupdel tycsg7

labh@labh-virtual-machine:~$ sudo groupdel tycsg8

labh@labh-virtual-machine:~$ sudo groupdel ty3b

labh@labh-virtual-machine:~$ sudo groupdel ty5a

Command for deleting directories in /home:

labh@labh-virtual-machine:~$ sudo rm -r /home/ty11aa

[sudo] password for tycs-123:

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 10

Install apache web server

THEORY:

Apache is one of the most widely used free, open-source Web Server applications in the world, mostly used in Unix-like operating systems but can also be used in windows. As a developer or system administrator, it will be very helpful for you to know about the Apache webserver. It has many notable features among which Virtual hosting is one such notable feature that allows a single Apache Web Server to serve a different number of websites.

We will discuss some of the most useful commands to manage Apache webserver (also called httpd on some other Linux-based distros) in Linux that you should know as a developer or system administrator. The commands that are going to be discussed must be executed as a root or sudo user.

The most common protocol used to transfer Web pages is the Hyper Text Transfer Protocol (HTTP). Protocols such as Hyper Text Transfer Protocol over Secure Sockets Layer (HTTPS), and File Transfer Protocol (FTP), a protocol for uploading and downloading files, are also supported.

Apache Web Servers are often used in combination with the MySQL database engine, the HyperText Preprocessor (PHP) scripting language, and other popular scripting languages such as Python and Perl. This configuration is termed LAMP (Linux, Apache, MySQL and Perl/Python/PHP) and forms a powerful and robust platform for the development and deployment of Web-based applications.

Apache is a widely popular and open source Hypertext Transfer Protocol (HTTP) server software. It is released under the Apache license. Apache offers some of the following benefits and advantages:

- It is stable.
- It is used, backed, and supported by several major sites and organizations.
- The entire program and related components are open source.
- It works on a large number of platforms (all popular variants of Linux/UNIX, some of the not-sopopular variants of UNIX, and even Microsoft Windows).
- It is extremely flexible.
- It has proved to be secure.

PRACTICAL:

COMMANDS:

labh@labh-virtual-machine:~/Desktop$ sudo apt update

[sudo] password for labh:

labh@labh-virtual-machine:~/Desktop$ sudo apt install apache2

labh@labh-virtual-machine:~/Desktop$ sudo ufw app list

Available applications:

Apache

Apache Full

Apache Secure

CUPS

OpenSSH

labh@labh-virtual-machine:~/Desktop$ sudo ufw allow 'Apache Full'

Rule added

Rule added (v6)

labh@labh-virtual-machine:~/Desktop$ sudo systemctl status apache2

● apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese>

Active: active (running) since Wed 2022-09-28 19:45:15 IST; 2min 19s ago

Docs: https://httpd.apache.org/docs/2.4/

Main PID: 4057 (apache2)

Tasks: 55 (limit: 2247)

Memory: 5.0M

CPU: 54ms

CGroup: /system.slice/apache2.service

├─4057 /usr/sbin/apache2 -k start

├─4058 /usr/sbin/apache2 -k start

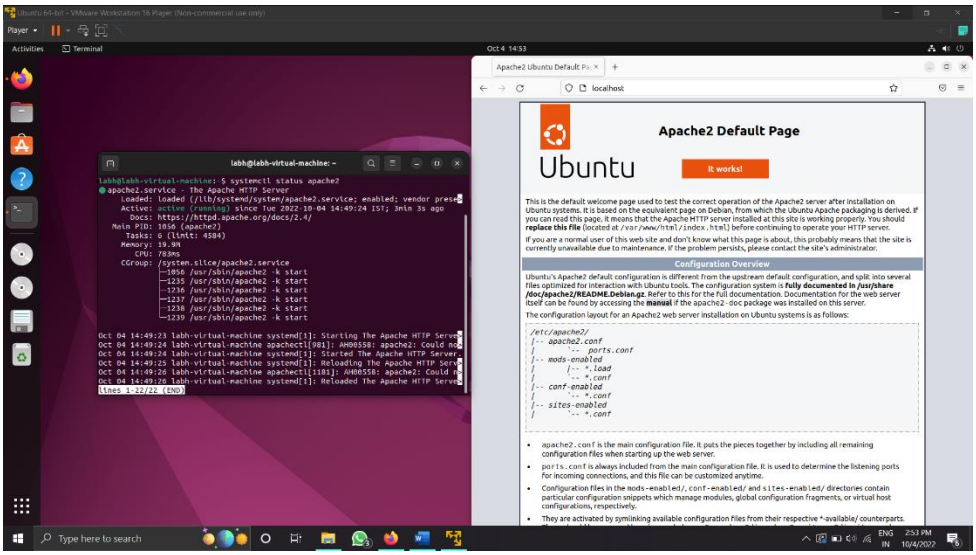
└─4059 /usr/sbin/apache2 -k start

Sep 28 19:45:15 labh-virtual-machine systemd[1]: Starting The Apache HTTP Serv>

Sep 28 19:45:15 labh-virtual-machine apachectl[4056]: AH00558: apache2: Could >

Sep 28 19:45:15 labh-virtual-machine systemd[1]: Started The Apache HTTP Serve>

lines 1-16/16 (END)



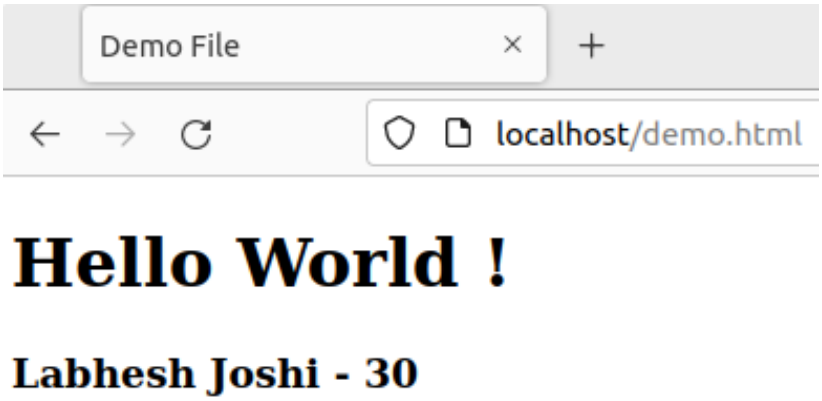
labh@labh-virtual-machine:/$ cd var/www/html

labh@labh-virtual-machine:/var/www/html$ sudo gedit demo.html

```
<!DOCTYPE HTML>
        <html>
                <head>
                        <title>Demo File</title>
                <head>
                <body>
                        <h1>Hello World !</h1>
                        <h3>Labhesh Joshi - 30</h3>
                </body>
```

```
</html>
```

Output:



Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 11

MySQL in Linux

THEORY:

MySQL is an open-source relational database management system (RDBMS). A relational database organizes data into one or more data tables in which data types may be related to each other; these relations help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation).

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for *Linux*, *Apache*, *MySQL*, *Perl*/*PHP*/*Python*. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr, MediaWiki, Twitter, and YouTube.

A.Steps to install MYSQL:

1.tycs-123@tycs123-virtual-machine:~$ sudo apt update

Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease

Hit:2 http://in.archive.ubuntu.com/ubuntu jammy InRelease

Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease

Hit:4 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease

Building dependency tree... Done

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

Do you want to continue? [Y/n] y

2.tycs-123@tycs123-virtual-machine:~$ sudo systemctl status mysql.service

● mysql.service - MySQL Community Server

Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)

Active: active (running) since Fri 2022-08-19 21:00:32 IST; 2min 29s ago

Process: 18615 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)

Main PID: 18623 (mysqld)

Status: "Server is operational"

Tasks: 38 (limit: 2247)

Memory: 353.6M

CPU: 1.968s

CGroup: /system.slice/mysql.service

└─18623 /usr/sbin/mysqld

Aug 19 21:00:31 tycs123-virtual-machine systemd[1]: Starting MySQL Community Server...

Aug 19 21:00:32 tycs123-virtual-machine systemd[1]: Started MySQL Community Server.

B.Steps to log in mysql:

1.tycs-123@tycs123-virtual-machine:~$ sudo mysql -p -u root

Enter password:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 8

Server version: 8.0.30-0ubuntu0.22.04.1 (Ubuntu)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

Create a database:

1.mysql> create database mydb;

Query OK, 1 row affected (0.03 sec)

2.mysql> use mydb;

Database changed

3.mysql> create table student(Name varchar(20),RollNo INT,Course varchar(20));

Query OK, 0 rows affected (0.05 sec)

4.mysql> desc student;

```
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| Name   | varchar(20) | YES  |     | NULL    |       |
| RollNo | int         | YES  |     | NULL    |       |
| Course | varchar(20) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

5.mysql> insert into student values('panda', 18, 'cs');

Query OK, 1 row affected (0.02 sec)

6.mysql> select * from student;

+-------+--------+--------+

| Name | RollNo | Course |

+-------+--------+--------+

| panda | 18 | cs |

+-------+--------+--------+

1 row in set (0.00 sec)

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 12

Python installation

Steps:

labh@labh-virtual-machine:~$ sudo apt update

labh@labh-virtual-machine:~$ sudo apt install software-properties-common

labh@labh-virtual-machine:~$ python3 –version

Python 3.10.6

labh@labh-virtual-machine:~$ sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev wget

labh@labh-virtual-machine:~$ cd /usr/src

labh@labh-virtual-machine:/usr/src$

labh@labh-virtual-machine:/usr/src$ sudo wget https://www.python.org/ftp/python/3.10.6/Python-3.10.6.tgz

[sudo] password for labh:
--2022-10-06 12:56:57-- https://www.python.org/ftp/python/3.10.6/Python-3.10.6.tgz
Resolving www.python.org (www.python.org)... 151.101.152.223, 2a04:4e42:24::223
Connecting to www.python.org (www.python.org)|151.101.152.223|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 25986768 (25M) [application/octet-stream]
Saving to: ‘Python-3.10.6.tgz.1’

Python-3.10.6.tg 100%[=======>] 24.78M 3.81MB/s in 7.2s

2022-10-06 12:57:05 (3.45 MB/s) - ‘Python-3.10.6.tgz.1’ saved [25986768/25986768]

labh@labh-virtual-machine:/usr/src$ sudo tar -xzf Python-3.10.6.tgz

labh@labh-virtual-machine:/usr/src$ ls

linux-headers-5.15.0-46 linux-headers-5.15.0-48-generic
linux-headers-5.15.0-46-generic Python-3.10.6
linux-headers-5.15.0-47 Python-3.10.6.tgz
linux-headers-5.15.0-47-generic Python-3.10.6.tgz.1
linux-headers-5.15.0-48
labh@labh-virtual-machine:/usr/src$ cd Python-3.10.6

labh@labh-virtual-machine:/usr/src/Python-3.10.6$

labh@labh-virtual-machine:/usr/src/Python-3.10.6$ sudo ./configure --enable-optimization

labh@labh-virtual-machine:/usr/src/Python-3.10.6$ sudo make altinstall

labh@labh-virtual-machine:/usr/src/Python-3.10.6$ python3.10 --version
Python 3.10.6

labh@labh-virtual-machine:~$ python3 hello.py

Hello World

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 13

Install phpmyAdmin server

Step 1: Install phpmyadmin

tycs-123@tycs123-virtual-machine:~$ sudo apt install phpmyadmin

Space to select apache as webserver and tab to select ok

For configuration select no

Step 2: Restart apache and mysql

tycs-123@tycs123-virtual-machine:~$ sudo systemctl restart apache2

tycs-123@tycs123-virtual-machine:~$ sudo systemctl restart mysql.service

Step 3: Check if installed

tycs-123@tycs123-virtual-machine:~$ sudo systemctl status apache2

check for active **(running)** in green

Step 4: Configure php

tycs-123@tycs123-virtual-machine:~$ sudo a2enconf phpmyadmin

[sudo] password for tycs-123:

Conf phpmyadmin already enabled

Step 5: Set a password

tycs-123@tycs123-virtual-machine:~$ sudo mysql

mysql> alter user 'root'@'localhost' identified with mysql_native_password by 'Admin@123';

Query OK, 0 rows affected (0.00 sec)

mysql> exit

tycs-123@tycs123-virtual-machine:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root: **Admin@123**

Press y|Y for Yes, any other key for No: **y**

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: **1**

Change the password for root ? ((Press y|Y for Yes, any other key for No) : n

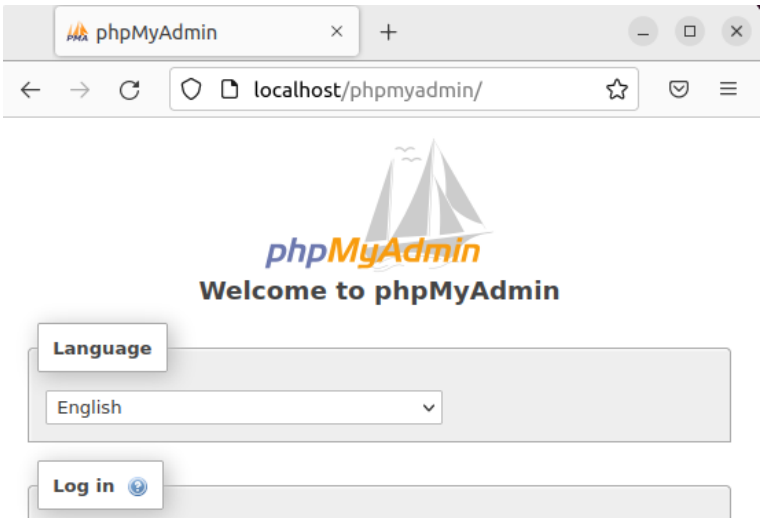
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : n

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y

Step 6: Open localhost/phpmyadmin on firefox again and the password is the one you used for alter user command.



Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 14

Install secure shell - ssh

THEORY:

SSH is a suite of network communication tools that are collectively based on an open protocol/standard that is guided by the Internet Engineering Task Force (IETF). It allows users to connect to a remote server just as they would using Telnet, rlogin, FTP, and so on, except that the session is 100- percent encrypted. Someone using a packet sniffer merely sees encrypted traffic going by. Should they capture the encrypted traffic, decrypting it could take a long time.

Secure Shell relies on a technology called public-key cryptography. It works similarly to a safe deposit box at the bank: You need two keys to open the box or at least multiple layers of security/checks have to be crossed. In the case of public-key cryptography, you need two mathematical keys: a public one and a private one. Your public key can be published on a public web page, printed on a T-shirt, or posted on a billboard in the busiest part of town. Anyone who asks for it can have a copy. Any data encrypted with the public key can be decrypted with the private key. On the other hand, your private key must be protected to the best of your ability. It is this piece of information that makes the data you want to encrypt truly secure. Any data encrypted with the private key can be decrypted with the public key. Every public key/private key combination is unique.

To make things even more interesting, SSH regularly changes its session key. (This is a randomly generated, symmetric key for encrypting the communication between the SSH client and server. It is shared by the two parties in a secure manner during SSH connection setup.) In this way, the data stream gets encrypted differently every few minutes. Thus, even if someone happened to figure out the key for a transmission, that miracle would be valid for only a few minutes until the keys changed again.

The OpenSSH project was spearheaded by the OpenBSD project. OpenBSD is a version of the Berkeley Software Distribution (BSD) operating system (another UNIX variant) that strives for the best security of any operating system available. The core of the OpenSSH package is considered part of the OpenBSD project and is thus simple and specific to the OpenBSD operating system.

Here is a quick rundown of some SSH clients and other useful SSH resources:
PuTTY: This is probably one of the oldest and most popular SSH implementations for the Win32 (Microsoft Windows) platforms. It is extremely lightweight and can either be used as a stand-alone, self-contained executable or be installed like other Windows programs. The website also hosts other tools such as pscp, which is a Windows command-line version of Secure Copy (SCP).

OpenSSH for Apple OS X: OS X is actually a UNIX-based and UNIX-compliant operating system. One of its main core components—the kernel—is based on the BSD kernel. So you shouldn't be too surprised that OpenSSH is part of the OS X system. When you open the terminal application, you can simply issue the ssh command. OS X systems also ship with an OpenSSH SSH server.

MindTerm, multiplatform: This program supports versions 1 and 2 of the SSH protocol. Written in 100-percent Java, it works on many UNIX platforms (including Linux), as well as Windows and OS X. See the web page for a complete list of tested operating systems.

Cygwin: This might be a bit of an overkill, but it is well worth the initial effort involved with getting it set up. It is a collection of tools that provides a Linux environment for Windows. It provides an environment to run numerous Linux/UNIX programs without extensive changes to their source code. Under cygwin, you can run all your favorite GNU/Linux programs, such as bash, grep, find, nmap, gcc, awk, vim, emacs, rsync, OpenSSH client, OpenSSH server, and so on, as though you were at a traditional GNU/Linux shell.

FileZilla: FileZilla client is a cross-platform FTP, FTPS, and SFTP client.

PowerShell: A native port of OpenSSH to Microsoft Windows platforms via the PowerShell environment.

OpenSSH: is only as secure as the weakest connection between the user and the server. This means that, for example, if a user uses Telnet to connect from host A to host B and then uses ssh to connect to host C, the entire connection can be monitored from the link between host A and host B. The fact that the link between host B and host C is encrypted becomes irrelevant.

Step 1: Install putty

Google → putty → putty.exe (the SSH and Telnet client itself) (64-bit x86)

Step 2: Install shell

tycs-123@tycs123-virtual-machine:~$ sudo apt install openssh-server

Step 3: Check installation

tycs-123@tycs123-virtual-machine:~$ sudo systemctl status ssh

Step 4: Install net tools

tycs-123@tycs123-virtual-machine:~$ sudo apt install net-tools

Step 5: check ip address

tycs-123@tycs123-virtual-machine:~$ ifconfig

or

tycs-123@tycs123-virtual-machine:~$ hostname -I

192.168.139.129

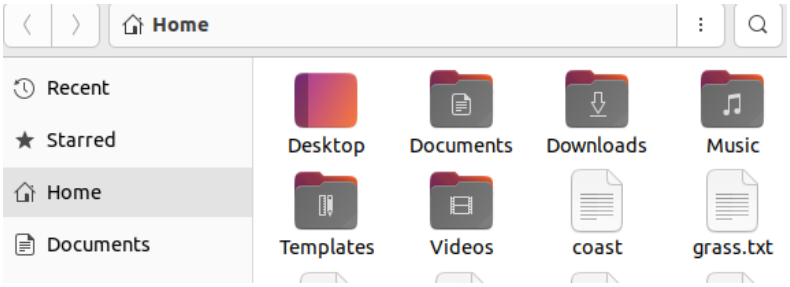
Step 6: Perform in putty

Open putty and paste the ip address

login as: tycs-123

tycs-123@192.168.139.129's password: admin123

tycs-123@tycs123-virtual-machine:~$ touch grass.txt



Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:05-08-2022**

US-TCS-501 Linux Practical 15

FTP

Install vsftpd (very secure file transfer protocol):

tycs-123@tycs123-virtual-machine:~$ sudo apt install vsftpd

Check if config file of vsftpd is created or not:

tycs-123@tycs123-virtual-machine:~$ ls /etc

Copy the vsftpd file to vsftpd1 in the same directory:

tycs-123@tycs123-virtual-machine:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd1.conf

Check the status:

tycs-123@tycs123-virtual-machine:~$ sudo ufw status

Status: inactive

Enable the status to active and check again:

tycs-123@tycs123-virtual-machine:~$ sudo ufw enable

Firewall is active and enabled on system startup

tycs-123@tycs123-virtual-machine:~$ sudo ufw status

Status: active

Add ports 21 and 20 to esblish connection for sending and receiving files and check in status if it has been added:

tycs-123@tycs123-virtual-machine:~$ sudo ufw allow 21/tcp

Rule added

Rule added (v6)

tycs-123@tycs123-virtual-machine:~$ sudo ufw allow 20/tcp

Rule added

Rule added (v6)

tycs-123@tycs123-virtual-machine:~$ sudo ufw status

Give the following command and check if it has been added in status:

tycs-123@tycs123-virtual-machine:~$ sudo ufw allow 40000:50000/tcp

Rule added

Rule added (v6)

tycs-123@tycs123-virtual-machine:~$ sudo ufw status

Give command to allow SSH:

tycs-123@tycs123-virtual-machine:~$ sudo ufw allow OpenSSH

Rule added

Rule added (v6)

tycs-123@tycs123-virtual-machine:~$ sudo ufw status

Restart vsftpd and get ip address:

tycs-123@tycs123-virtual-machine:~$ sudo systemctl restart vsftpd

tycs-123@tycs123-virtual-machine:~$ hostname -I

192.168.114.130

Open in the apache browser:

ftp:// 192.168.114.130/

ftp://192.168.114.130

Open vsftpd file:

tycs-123@tycs123-virtual-machine:~$ sudo gedit /etc/vsftpd.conf

Uncomment following lines in the file:

i. anonymous_enable=NO

ii. local_enable=YES

iii. write_enable=YES

iv. local_umask=022

v. ascii_upload_enable=YES

vi. ascii_download_enable=YES

Add following lines at the end of the file and save it:

vii. pasv_min_port=40000

viii. pasv_max_port=50000

Restart vsftpd:

tycs-123@tycs123-virtual-machine:~$ sudo systemctl restart vsftpd

Name: Labhesh Joshi **Roll no: KCTBCS030** **Date:29-07-2022**

US-TCS-501 Linux Practical 16

Install Samba Server

THEORY:

Samba is a powerful suite of applications that helps UNIX-based systems (such as Linux) interoperate with Windows-based and other operating systems. It is an open source implementation of the Server Message Block/Common Internet File System (SMB/CIFS) protocol suite.

Samba transparently provides file and print sharing services to Windows clients as well as other networked clients running other operating systems. It does this through the use of the native Microsoft networking protocols SMB/CIFS. From a system administrator's point of view, this means you can deploy a Linux/UNIX-based server and use it to provide file sharing, authentication, print, and other services to other non-native Linux clients such as Microsoft Windows systems. Using Samba means that Windows systems can use their native tongue to talk to the Linux server—which means fewer hassles for you and seamless integration for your users.

The Linux/UNIX login/password mechanism is radically different from the Windows Active Directory model, which uses domain controllers (DCs). Thus, it's important for the system administrator to maintain consistency in the logins and passwords across both platforms. Users may need to work in heterogeneous environments and may need access to the different platforms for various reasons. It is thus useful to make working in such environments as seamless as possible without having to worry about users' needing to reauthenticate separately on the different platforms, worry about cached passwords that don't match between servers, and other issues. Relative to Samba, several options are available for handling username and password issues in heterogeneous environments, including the following:

- **Linux pluggable authentication modules (PAMs):** Allow you to authenticate users against a DC. This means you still have two user lists—one local and one on the DC—but your users need to keep track of their passwords only on the Windows system.
- **Samba as a DC:** Allows you to keep all your logins and passwords on the Linux system, while all your Windows boxes authenticate with Samba. When Samba is used with a Lightweight Directory Access Protocol (LDAP) back-end for this, you will have a scalable and extensible solution.

- **Custom script:** Allows you to use your own custom script. For sites with a well-established system for maintaining logins and passwords, it isn't unreasonable to come up with a custom script. This can be done using a scripting language with good cross-platform support. Such scripts can be coaxed to allow changes to the Security Access Manager (SAM) to update the DC's password list.

Windows-based systems use encrypted passwords when communicating with the DC and any server requiring authentication (including Linux and Samba). The encryption algorithm used by Windows is different from UNIX's, however, and, therefore, is not compatible. Here are your choices for handling this conflict:

- Edit the Registry on Windows clients to disable the use of encrypted passwords. The Registry entries that need to be changed are listed in the docs directory in the Samba package. As of version 3 of Samba, this option is no longer necessary.
- Configure Samba to use Windows-style encrypted passwords.

The first solution has the benefit of not pushing you to a more complex password scheme. On the other hand, you may have to apply the Registry fix on all your clients. The second option, of course, has the opposite effect: for a little more complexity on the server side, you don't have to modify any of your clients.

The smbd daemon handles the actual sharing of file systems and printer services for clients. It is also responsible for user authentication and resource-locking issues. It starts by binding to port 139 or 445 and then listens for requests. Every time a client authenticates itself, smbd makes a copy of itself; the original goes back to listening to its primary port for new requests, and the copy handles the connection for the client. This new copy also changes its effective user ID from root to the authenticated user.

The nmbd daemon is responsible for handling NetBIOS name service requests. nmbd can also be used as a drop-in replacement for a Windows Internet Name Server (WINS). It begins by binding itself to port 137; unlike smbd, however, nmbd does not create a new instance of itself to handle every query. In addition to name service requests, nmbd handles requests from master browsers, domain browsers, and WINS servers—and as such, it participates in the browsing protocols that make up the popular Windows My Network Places of systems. The services provided by the smbd and nmbd daemons complement each other.

Finally, the service provided by winbindd can be used to query native Windows servers for user and group information, which can then be used on purely Linux/UNIX platforms. It does this by using Microsoft Remote Procedure Call (RPC) calls, PAM, and the name service switch (NSS) capabilities found in modern C libraries. Its use can be extended through the use of a PAM module (pam_winbind) to provide authentication services. This service is controlled separately from the main smb service and can run independently.

Steps:

1) **labh@labh:~$ sudo apt update**

2) **labh@labh:~$ sudo apt install samba**

3) **labh@labh:~$ ls /etc/samba**

4) **labh@labh:~$ mkdir smbdir1**

5) **allow access to others for smbdir1**

6) **labh@labh:~$ cd smbdir1**

7) **labh@labh:~/smbdir1$ cat > myfile1.txt**
hello

8) **labh@labh:~/smbdir1$ cd ..**

9) **labh@labh:~$ sudo adduser sambauser1**

10) **labh@labh:~$ sudo systemctl status smbd**

11) **Make the following changes in smb.conf file**
   i. **comment = Welcome**
   ii. **path = /home/umar/smbdir1**
   iii. **read only = NO**
   iv. **browsable = YES**

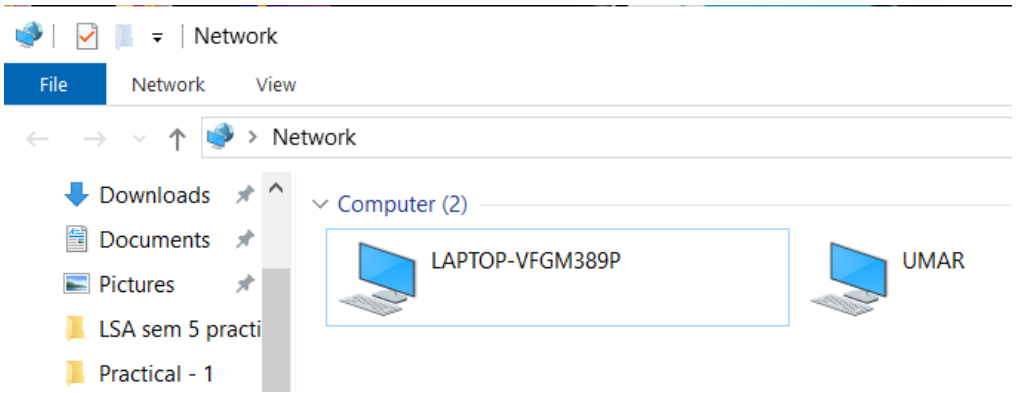
12) **labh@labh:~$ sudo ufw allow samba**

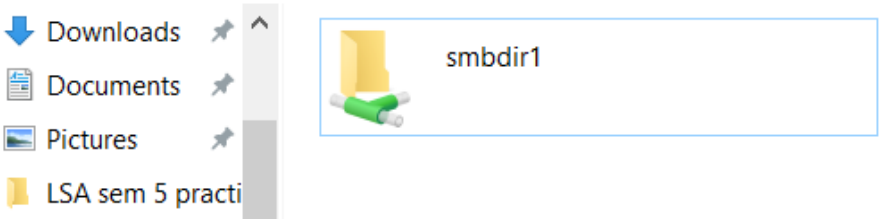
13) **labh@labh:~$ sudo systemctl restart smbd**

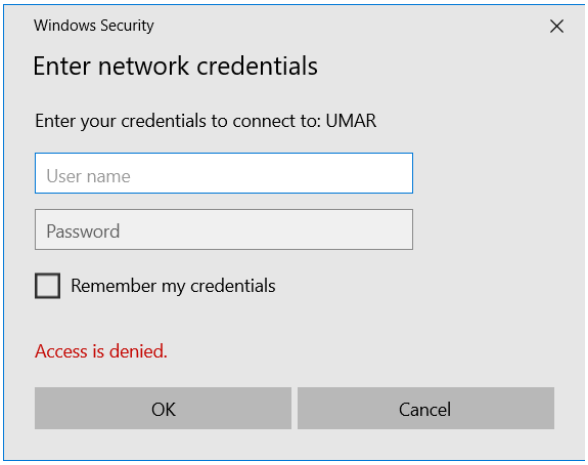
14) **labh@labh:~$ sudo systemctl status smbd**

15) **labh@labh:~$ sudo smbpasswd -a sambauser1**

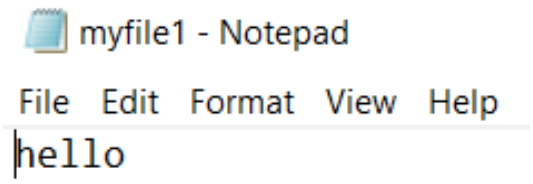
16) **Follow the steps:**
   i. **In Windows OS, click on Network icon on desktop.**
   ii. **You will get UMAR icon in window, click on that, it will display smbdir1 folder.**
   iii. **Click on the folder you created.**
   iv. **Enter the new username and password, which you created in Ubuntu for samba server. (i.e sambauser1 and its password)**
   v. **Now you can share folder, files from Windows to Linux and the other way around.**




Downloads
Documents
Pictures
LSA sem 5 practi
smbdir1


Windows Security
Enter network credentials
Enter your credentials to connect to: UMAR
User name
Password
Remember my credentials
Access is denied.
OK
Cancel


myfile1
31-08-2020 16:54
Text Document
1 KB


myfile1 - Notepad
File Edit Format View Help
hello