EXPERIMENT NO: 3

AIM: Familiarization of linux commands.

CO2: Perform system administration task.

PROCEDURE:

pwd - Print the working directory find the path of the current working directory

$pwd

```
student@t2:~$ pwd
/home/student
```

ls – To view the content of the directory

$ls

```
student@t2:~$ ls
Desktop  Documents  Downloads  inmca  Music  neha  neha1  Pictures  Public  PycharmProjects  snap  Templates  Videos
```

ls -R – To list the contents of sub directory

$ls -R

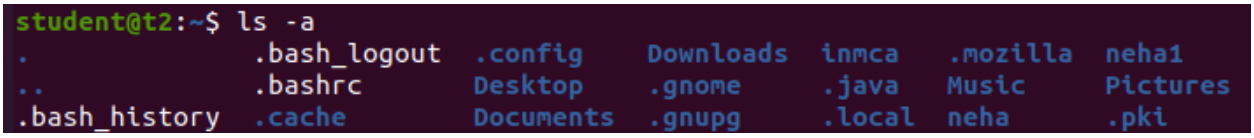
```
student@t2:~$ ls -R
.:
Desktop  Documents  Downloads  inmca  Music  neha  neha1  Pictures  Public  PycharmProjects  snap  Templates  Videos

./Desktop:

./Documents:

./Downloads:

./inmca:
f1  f2
```

ls -l – Long listing of the contents

```
student@t2:~$ ls -l
total 52
drwxr-xr-x 2 student student 4096 Jun 17  2022 Desktop
drwxr-xr-x 2 student student 4096 Jun 17  2022 Documents
drwxr-xr-x 2 student student 4096 Jun 17  2022 Downloads
drwxrwxr-x 4 student student 4096 Jan 27 16:13 inmca
```

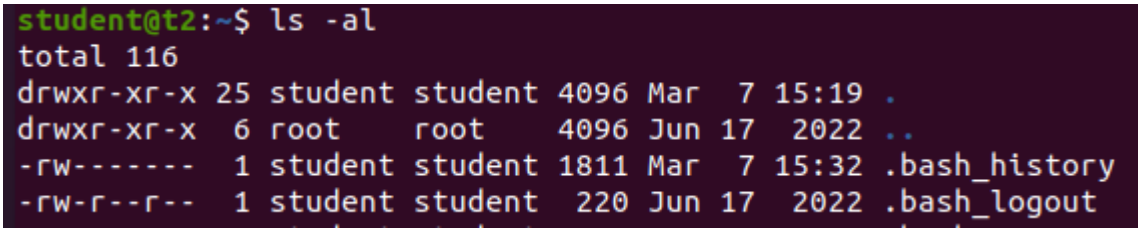
ls -a – To list the all hidden files

$ls -a



ls -al – List the files and directories with detailed information.

$ls -al

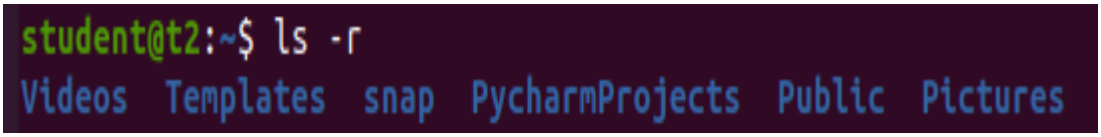


ls -t – List the files sorted in the order of last modified.

$ls -t

ls -r – To reverse the natural sorting order

$ls -r



history – To review the command that have been previously executed for a certain period of time.

$history

```
student@t2:~$ history
    1  ./studio.sh
    2  ./studio.sh
    3  su mca
    4  man
    5  man echo
    6  man ls
    7  read
    8  cat >file1.txt<<EOF
    9  pwd
   10  ls
```

man – learn and understand about different command right from the shell using man command

$man

```
LS(1)                                   User Commands                                   LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).  Sort entries alphabetically if none of -cftu-
       vSUX nor --sort is specified.

       Mandatory arguments to long options are mandatory for short options too.

       -a, --all
              do not ignore entries starting with .

       -A, --almost-all
              do not list implied . and ..

       --author
              with -l, print the author of each file
```

mkdir – To create a new directory

$mkdir

```
student@t2:~$ mkdir sampledire
student@t2:~$ cd sampledire
student@t2:~/sampledire$ ls
student@t2:~/sampledire$
```

rmdir – To remove a directory

$rmdir

```
student@t2:~$ rmdir sampledire
student@t2:~$ cd sampledire
bash: cd: sampledire: No such file or directory
```

touch – To create new empty file

$touch

```
student@t2:~$ touch file1
student@t2:~$
```

cat – Concatenate the files and print on the standard output

$cat

cat > filename.txt – To create a file with inserting contents

$cat > file.txt

```
student@t2:~$ cat > file.txt
Green
red
^Z
[2]+  Stopped                 cat > file.txt
```

cat filename.txt – To view the content of the file

$cat file.txt

```
student@t2:~$ cat file.txt
Green
red
```

cat >> filename.txt – To append new contents to an existing file

$cat >> file.txt

```
student@t2:~$ cat >> file.txt
Blue
Yellow
^Z
[3]+  Stopped                 cat >> file.txt
student@t2:~$ cat file.txt
Green
red
Blue
Yellow
```

cat -n filename.txt – Number all output lines

$cat -n file.txt

```
student@t2:~$ cat -n file.txt
     1  Green
     2  red
     3  Blue
     4  Yellow
```

cat -b filename.txt – To remove the empty lines

$cat -b file.txt

```
student@t2:~$ cat -n file.txt
     1  Green
     2  red
     3  Blue
     4  Yellow
     5
     6
     7  Pink
student@t2:~$ cat -b file.txt
     1  Green
     2  red
     3  Blue
     4  Yellow


     5  Pink
```

cat -E filename.txt – Display $ at end of each line

$cat -E file.txt

```
student@t2:~$ cat -E file.txt
Green$
red$
Blue$
Yellow$
$
$
Pink$
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

EXPERIMENT NO :4

AIM : Familiarization of linux commands

CO2: Perform system administration task.

PROCEDURE:

cut - For cutting out the sections from each line of files and writing the result to standard output.

- $cut -b1 names – cut by bytes

```
student@t2:~/student$ cut -b1 names
N
A
j
n
n
A
a
a
```

- $cut -c3 names – cut by character position

```
student@t2:~/student$ cut -c3 names
h
y
m
n
t
d
h
```

- $cut -d - - f1 file – to show first column

- $cut -c1 1,2 names – To specify the columns

```
student@t2:~/student$ cut -c 1,2 names
Ne
Ar
jo
na
ni
Ar
at
an
```

- $cut -d ‘’ -f1 file5 – To cut the space

paste - To join files horizontally (each files consisting of different lines)

- $paste names file>file3

```
student@t2:~/student$ paste names file >file3
student@t2:~/student$ cat file3
Neha    Neha 45
Arya    arya 54
jomol   ardra 76
nandana
nithasha
Ardra
athira
anagha
```

- $paste -d '%' names file

```
student@t2:~/student$ paste -d '%' names file
Neha%Neha 45
Arya%arya 54
jomol%ardra 76
nandana%
nithasha%
Ardra%
athira%
anagha%
```

- $paste -s names – To print in single line

```
student@t2:~/student$ paste -s names
Neha    Arya    jomol   nandana nithasha        Ardra   athira  anagha
```

copy - To copy the contents of the file

- $cp -r neha student

```
student@t2:~/student$ cp -r neha student
cp: cannot stat 'neha': No such file or directory
student@t2:~/student$ cd ..
student@t2:~$ cp -r neha student
student@t2:~$ cd neha
student@t2:~/neha$ ls neha
ls: cannot access 'neha': No such file or directory
student@t2:~/neha$ ls
EOF  file1.txt  file2.txt  file3.txt  file4.txt  file5.txt  mark  mark1  output.txt
student@t2:~/neha$ cd ..
student@t2:~$ cd student
student@t2:~/student$ ls
file  file1  file3  filee  haai  hai  mark5  markk  marks  names  neha  outputs  Overoll
student@t2:~/student$
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

EXPERIMENT NO : 5

AIM : Familiarization of linux commands

CO2: Perform system administration task.

Procedure:

read command – To read the contents of a line we use 'read' command

- $read

```
student@t2:~$ cd neha
student@t2:~/neha$ read
my name is neha
```

REPLY - to print the read line

- $echo $REPLY

```
student@t2:~/neha$ echo $REPLY
my name is neha
```

- $read - to store the read content in different variables

```
student@t2:~/neha$ read var1 var2 var3
my name is neha
```

- $echo "[$var1] [var2] [var3]"-to print the variable stored contents

```
student@t2:~/neha$ echo "[$var1][$var2][$var3]"
[my][name][is neha]
```

- $read my\ name is\ neha -to read multiple lines using \

```
student@t2:~/neha$ read
my \
> name is \
> Neha
```

- $read -p -to promt the text

```
student@t2:~/neha$ read -p "Enter your name"
Enter your name Neha
student@t2:~/neha$ echo "My name is $REPLY"
My name is  Neha
```

- $read -n 6 -p -to limit the specified text

```
student@t2:~/neha$ read -n 6 -p "enter 6 charachtes only"
enter 6 charachtes only nehaastudent@t2:~/neha$ read -s -p "enter the password"
```

- $read -s -p -To secure the password

```
enter 6 charachtes only nehaastudent@t2:~/neha$ read -s -p "enter the password"
enter the passwordstudent@t2:~/neha$ echo "password is $REPLY"
password is 6656666666
```

- $wc profile -To display the number of lines,bits,word

```
student@t2:~/neha$ cat > profile
My name is Neha
Student of AJCE
Coming from Kannur
^Z
[1]+  Stopped                 cat > profile
student@t2:~/neha$ wc profile
 3 10 51 profile
```

- $wc -l -number of lines

```
student@t2:~/neha$ wc -l profile
3 profile
```

- $wc -m -number of bit

```
student@t2:~/neha$ wc -m profile
51 profile
student@t2:~/neha$ wc -c profile
51 profile
```

- $wc -w -number of word

```
student@t2:~/neha$ wc -w profile
10 profile
```

- $wc -L -To display the length of longest line

```
student@t2:~/neha$ wc -L profile
18 profile
```

more –‘more’ command is similar to ‘cat’ command to display the contents .The only difference is that in case of larger files ‘cat’ command output will scroll off your screen while ‘more’ command display output once screenful at atime.

more filename-

- $more contents

```
student@t2:~/neha$ more contents
The Muzhapiilangad beach is located parallel to National Highway 66 (formerly National Highway 17) between Kannur and Thalassery.[5]

The beach festival is celebrated in the month of April and it is one of the important tourist attraction in the district of Kannur in Kerala.
The youth also try many driving stunts in cars like drifting and wheeling in bikes as this is a paradise for driving along the shore.
```

```
student@t2:~/neha$ more +10 contents
The beach festival is celebrated in the month of April and it is one of the important
The youth also try many driving stunts in cars like drifting and wheeling in bikes as
```

- $more +/through contents – The string contained paragraph will show.

```
Green Island in English). It is possible to walk to the island during low tide from the nearby Dharmadam bea
student@t2:~/neha$ more +/through contents

...skipping
The beach festival is celebrated in the month of April and it is one of the important tourist attraction in t
The youth also try many driving stunts in cars like drifting and wheeling in bikes as this is a paradise for

There is an unpaved road winding through coconut groves leading to the beach. To get to this road, if you are
```

- $read -d contents – It helps the user to navigate according to the instruction

```
student@t2:~/neha$ more -d contents
The Muzhapiilangad beach is located parallel to National Highway 66 (formerly National Highway 17) between Kannur and Thalassery.[5]

The beach festival is celebrated in the month of April and it is one of the important tourist attraction in the district of Kannur in Kerala.
The youth also try many driving stunts in cars like drifting and wheeling in bikes as this is a paradise for driving along the shore.
```

```
There is an unpaved road winding through coconut groves leading to the beach. To get to this road, if you are driving from Tellicherry towards
 Kannur, take the left turn just before the railway over bridge (first railway crossing) you encounter after crossing the Moidu bridge. The be
ach is about 3.8 kms long and curves in a wide area providing a good view of Kannur on the north. Local laws allow beachgoers to drive for a f
ull 3.4 kms directly on the sands of the beach. The beach is bordered by black rocks, which also protect it from the stronger currents of the
ocean. These rocks provide habitat for Blue mussel, a delicious seafood. Beach attracts bird-watchers from far off places as hundreds of birds
 flock here during various seasons.
--More--(95%)
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

EXPERIMENT NO:6

AIM: Familiarization of linux commands

CO2: Perform system administration task

PROCEDURE:

grep – Filter the content which makes our search easy

- $ grep 43 marks

OUTPUT:

```
student@t2:~$ grep 43 marks
social-43
```

- $ grep -i maths marks – case insensitive search

OUTPUT:

```
student@t2:~$ grep -i maths marks
maths-23
```

- $ grep -v 66 marks – Display all the contents except the searched contents.

```
student@t2:~$ grep -v 66 marks
maths-23
englisg-45
social-43
science-33
```

- $ grep -A1 maths marks -Display the contents along with one line after that

```
student@t2:~$ grep -A1 maths marks
maths-23
englisg-45
```

- $ grep -B1 malayalam marks – Display the contents along with one line before that

```
student@t2:~$ grep -B1 malayalam marks
science-33
malayalam-66
```

- $ grep -C1 social marks – Display the contents along with one line before and after .

```
student@t2:~$ grep -C1 social marks
englisg-45
social-43
science-33
```

head – Display the first ten lines of the contents

- $ head demo1.txt

```
student@t2:~$ head demo1.txt
1
2
3
4
5
6
7
8
9
10
```

head-5 – Display the firat five contents

- $ head -5 demo1.txt

```
student@t2:~$ head -5 demo1.txt
1
2
3
4
5
```

t ail – Display the last ten lines of the content

- $ tail demo1.txt

```
student@t2:~$ tail demo1.txt
9
10
11
12
1
13
14
15
16
17
```

- $ tail -5 demo1.txt

```
student@t2:~$ tail -5 demo1.txt
13
14
15
16
17
```

mv – To move the file or directory

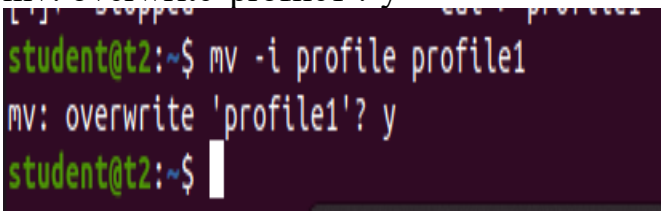
- $ mv demo1.txt marks
- $ cat marks

```
student@t2:~$ cat marks
1
2
3
4
5
6
7
8
9
10
11
12
1
13
14
15
16
17
```

- $ mv -b marks profile – To backup the files.
- $ ls

- $ mv -i profile profile1 – Ask for overwrite the file or what

mv: overwrite 'profile1'? y



Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

EXPERIMENT NO :7

AIM: Familiarization of linux commands

CO2: Perform system administration task

PROCEDURE:

expr -Evaluate the given expression and display the output .

- $ expr 12 + 8

```
student@t2:~/neha$ expr 12 + 8
20
```

- $ expr 12 – 8

```
student@t2:~/neha$ expr 12 - 8
4
```

- $ expr 12 * 3

```
student@t2:~/neha$ expr 12 \* 3
36
```

- $ expr 12 / 4

```
student@t2:~/neha$ expr 12 / 4
3
```

- $ read x

```
student@t2:~/neha$ read x
3
```

- $ read y

```
student@t2:~/neha$ read y
20
```

- $ expr $x + $y

```
student@t2:~/neha$ expr $x + $y
23
```

- $ df – Get a report on system disk space usage

```
student@t2:~/neha$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             3950476        0   3950476   0% /dev
tmpfs             797752     1820    795932   1% /run
/dev/sda6      143074460 25784920 109948948  19% /
tmpfs            3988752    31064   3957688   1% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
tmpfs            3988752        0   3988752   0% /sys/fs/cgroup
/dev/loop0           128      128         0 100% /snap/bare/5
```

du – To check how much space of a file or directory takes in current directory

- $ du file1.txt

```
student@t2:~/neha$ du file1.txt
4       file1.txt
```

sudo - allows you to run a command as root

- $sudo useradd neha

```
mca@t2:~$ sudo useradd neha
[sudo] password for mca:
```

- $sudo passwd neha

```
mca@t2:~$ sudo passwd neha
New password:
Retype new password:
passwd: password updated successfully
```

- sudo groupadd -g 333 mcastaff

```
mca@t2:~$ sudo groupadd -g 333 mcastaff
```

- $sudo usermod -G mcastaff neha

```
mca@t2:~$ sudo usermod -G mcastaff neha
```

id – Used to find group name and numeric id

- $id neha

```
mca@t2:~$ id neha
uid=1004(neha) gid=1005(neha) groups=1005(neha),333(mcastaff)
```

compgen - It is bash built-in command and it will show all available commands, aliases, and functions for you.

- $compgen -g - Used to display all the groups

```
mca@t2:~$ compgen -g
root
daemon
bin
sys
adm
tty
disk
lp
```

chmod – Used to change access permissions of files and directories .It stands change mode (read (r), write (w), execute (x)).

- $chmod -wx t.txt

```
mca@t2:~$ chmod -wx t.txt
mca@t2:~$ cat >> t.txt
bash: t.txt: Permission denied
```

- $chmod rwx t.txt

```
mca@t2:~$ chmod +rwx t.txt
mca@t2:~$ cat >> t.txt
added content
^Z
[1]+  Stopped                 cat >> t.txt
mca@t2:~$ cat t.txt
hello
world
let
start
network
added content
```

sudo chown – Used to change a file ownership or directory ownership for a user or a group down stands for change owner.

- $sudo chown neha file1.txt

```
mca@t2:~$ sudo chown neha t.txt
[sudo] password for mca:
Sorry, try again.
[sudo] password for mca:
```

- $sudo userdel

```
mca@t2:~$ sudo uderdel neha
[sudo] password for mca:
sudo: uderdel: command not found
mca@t2:~$ sudo userdel neha
mca@t2:~$ sudo userdel neha
userdel: user 'neha' does not exist
mca@t2:~$ sudo groupdel mcastaff
mca@t2:~$ sudo groupdel mcastaff
groupdel: group 'mcastaff' does not exist
mca@t2:~$
```

Result

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

EXPERIMENT NO: 8

AIM: Familiarization of linux commands.

CO2: Perform system administration task.

PROCEDURE

1. Ip addr :

```
mca@t2:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp3s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 78:24:af:ba:c3:58 brd ff:ff:ff:ff:ff:ff
    inet 192.168.6.17/24 brd 192.168.6.255 scope global noprefixroute enp3s0
       valid_lft forever preferred_lft forever
    inet6 fe80::72e3:812b:24c7:3a22/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
mca@t2:~$ ssh mca@192.168.6.18/24
ssh: Could not resolve hostname 192.168.6.18/24: Name or service not known
mca@t2:~$ ssh mca@192.168.6.18
ssh: connect to host 192.168.6.18 port 22: Connection refused
mca@t2:~$ sudo apt-get update
[sudo] password for mca:
Get:1 https://dl.google.com/linux/chrome/deb stable InRelease [1,811 B]
Hit:2 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:4 http://ppa.launchpad.net/maarten-fonville/android-studio/ubuntu focal InRelease [17.6 kB]
Get:5 https://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1,079 B]
Get:6 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
```

```
mca@t2:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-client openssh-sftp-server ssh-import-id
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
The following packages will be upgraded:
  openssh-client
1 upgraded, 4 newly installed, 0 to remove and 480 not upgraded.
Need to get 1,359 kB of archives.
After this operation, 6,010 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

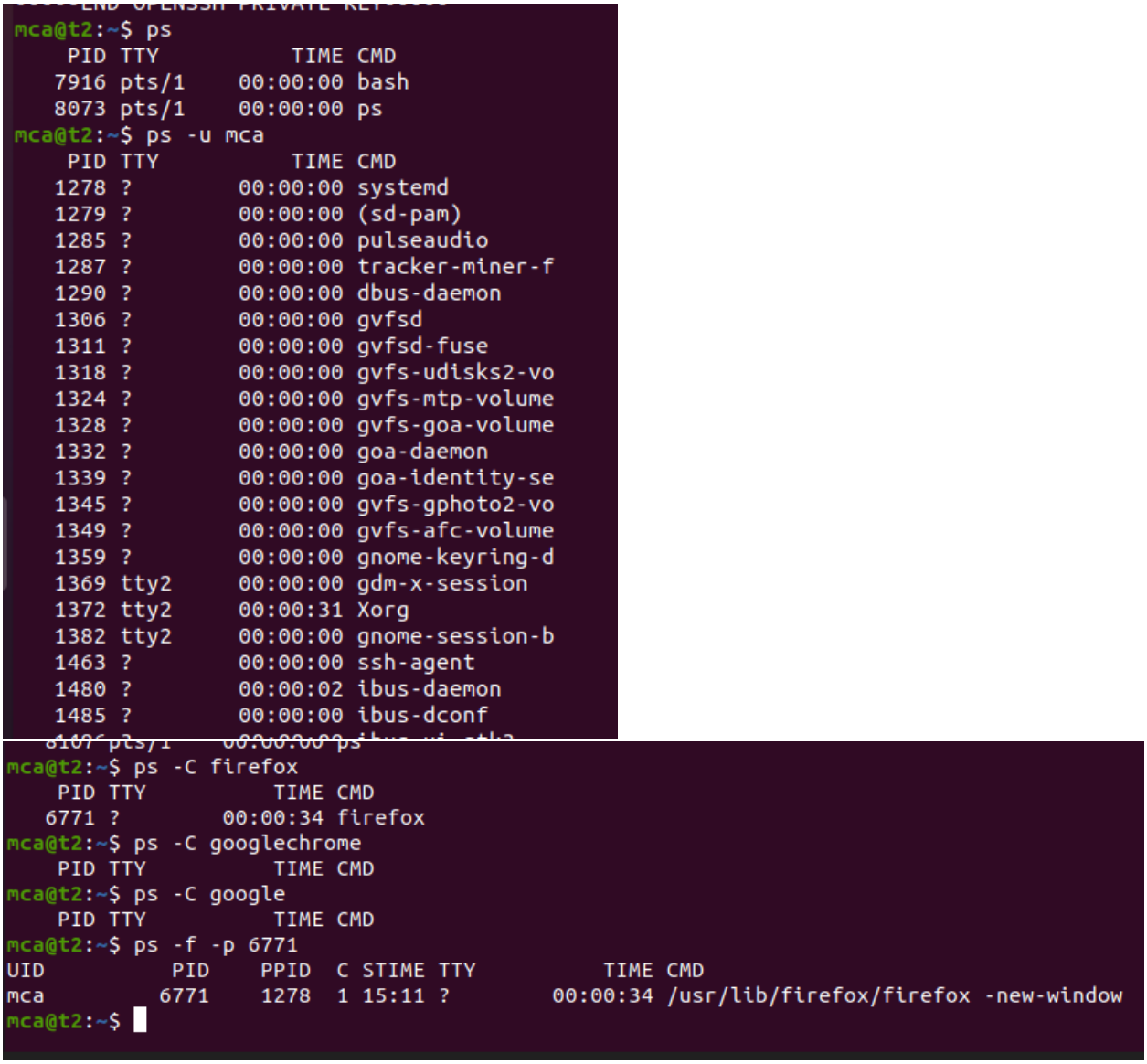
```
Rules updated (v6)
mca@t2:~$ ssh mca@192.168.6.18
The authenticity of host '192.168.6.18 (192.168.6.18)' can't be established.
ECDSA key fingerprint is SHA256:yaXBDEDgESH2nkHaWrp6Of4c088CkWQ1ApKTNcRzsVI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.6.18' (ECDSA) to the list of known hosts.
mca@192.168.6.18's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-26-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


698 updates can be installed immediately.
459 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Your Hardware Enablement Stack (HWE) is supported until April 2025.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

mca@t2:~$ cat > neha.txt
Neha a
18/2001
kannur
^Z
[1]+  Stopped                 cat > neha.txt
mca@t2:~$ ls
Desktop  Documents  Downloads  file  Music  neha.txt  Pictures  Public  Templates  Videos
mca@t2:~$ ls
Desktop  Documents  Downloads  file  Music  neha.txt  nithasha  Pictures  Public  Templates  Videos
```

ssh stands for secure shell

$ssh-keygen : generate a key for ssh

```
mca@t2:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/mca/.ssh/id_rsa): abcd.txt
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in abcd.txt
Your public key has been saved in abcd.txt.pub
The key fingerprint is:
SHA256:EkU5xAxbft3bBtSj1YtbjAkG+Noo8UhSSdVmnCDnnlI mca@t2
The key's randomart image is:
+---[RSA 3072]----+
|     .o+XXoo   ...|
|      o+*=* + o oo|
|     . oE=.o o X o|
|    . oo..o   = B |
|     o.=oS     + o|
|      o.= .    . . |
|       .           |
|                   |
|                   |
+----[SHA256]-----+
```

Ps : currently running program



Result :

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

SHELL SCRIPTING

EXPERIMENT NO : 1

AIM : Shell script to display current time

CO4 : Write shell scripts required for system administration

PROCEDURE

#!/bin/bash

Date

OUTPUT

```
neha@hpi5-22:~$ vi first.sh
neha@hpi5-22:~$ chmod +x first.sh
neha@hpi5-22:~$ ./first.sh
Sun Apr 16 21:11:30 IST 2023
neha@hpi5-22:~$
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 2

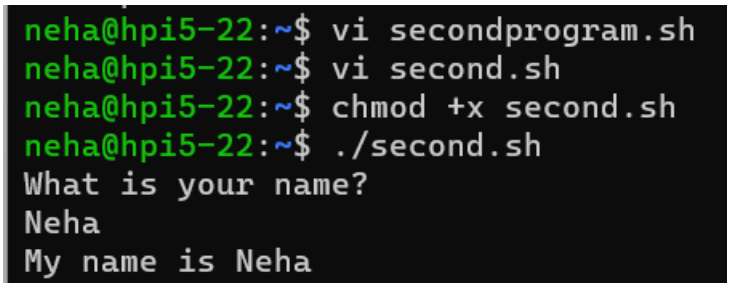
AIM : Shell script to display your name

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
echo "What is your name?"
read name
echo "My name is $name"
```

OUTPUT



RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 3

AIM : Shell script to commands

CO4 : Write shell scripts required for system administration

PROCEDURE

#!/bin/bash

date

ls

pwd

mkdir file1

OUTPUT

```
neha@hpi5-22:~$ vi third.sh
neha@hpi5-22:~$ chmod +x third.sh
neha@hpi5-22:~$ ./third.sh
Sun Apr 16 21:16:23 IST 2023
0         add3.sh     file.sh    firstp.sh         gttwo.sh    mod3.sh     rel.sh
add.sh    div3.sh     file1      firstprogram.sh   hh.sh       mul3.sh     rel1.sh
add1.sh   evenodd.sh  file1.sh   five.sh           inc3.sh     op.sh       second.sh
add2.sh   file        first.sh   four.sh           logical.sh  postneg.sh  secondprogram.sh
/home/neha
mkdir: cannot create directory ‘file1’: File exists
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 4

AIM : Shell script to demonstrate variables

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
echo "File name:$0"
echo "first prameter:$1"
echo "second parameter:$2"
echo "quoted values:$@"
echo "quoted values:$*"
echo "Total no of parameters:$#"
```

OUTPUT

```
neha@hpi5-22:~$ vi four.sh
neha@hpi5-22:~$ chmod +x four.sh
neha@hpi5-22:~$ ./four.sh
File name:./four.sh
first prameter:
second parameter:
quoted values:
quoted values:
Total no of parameters:0
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 5

AIM : Shell script to display an array

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
name[0]="neha"
name[1]="arya"
name[2]="ardra"
name[3]="anju"
echo "first index : ${name[0]}"
echo "second index : ${name[1]}"
```

OUTPUT

```
neha@hpi5-22:~$ vi six.sh
neha@hpi5-22:~$ chmod +x six.sh
neha@hpi5-22:~$ ./six.sh
first index : neha
second index : arya
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 6

AIM : Shell script to demonstrate arithematic operators

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
read -p "Enter a: " a
read -p "Enter b: " b
add=$(( a + b ))
echo "Sum is: $add"
sub=$(( a - b ))
echo "sub is : $sub"
mul=$(( a * b ))
echo "mul is : $mul"
div=$(( a / b ))
echo "div is : $div"
mod=$(( a % b ))
echo "mod is : $mod"
if [ $a == $b ]
then
        echo "a is qual to b"
fi
if [ $a != $b ]
then
        echo "a is not equal to b"
fi
(( ++a ))
```

```
echo "increment operator on a $a"
(( --b ))
echo "Decrement operator on b $b"
```

```
neha@hpi5-22:~$ vi op.sh
neha@hpi5-22:~$ chmod +x op.sh
neha@hpi5-22:~$ ./op.sh
Enter a: 5
Enter b: 3
Sum is: 8
sub is : 2
mul is : 15
div is : 1
mod is : 2
a is not equal to b
increment operator on a 6
Decrement operator on b 2
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 7

AIM : Shell script to demonstrate relational operators

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
read -p "Enter a:" a
read -p "Enter b:" b
if(( $a == $b ))
then
        echo " a is equal to b"
else
        echo "a is not equal to b"
fi
if(( $a != $b ))
then
        echo " a is not equal to b"
else
        echo "a is equal to b"
fi
if(( $a < $b ))
then
        echo " a is less than b"
else
        echo "a is not less than b"
fi
if(( $a <= $b ))
```

```
then
    echo " a is less than or equal to  b"
else
    echo "a is not less than or equal to  b"
fi
if(( $a > $b ))
then
    echo " a is greater than b"
else
    echo "a is not greater than b"
fi
if(( $a >= $b ))
then
    echo " a is greater than or equal to  b"
else
    echo "a is not greater than or equal to  b"
fi
```

```
neha@hpi5-22:~$ vi rel.sh
neha@hpi5-22:~$ chmod +x rel.sh
neha@hpi5-22:~$ ./rel.sh
Enter a:3
Enter b:2
a is not equal to b
 a is not equal to b
a is not less than b
a is not less than or equal to  b
 a is greater than b
 a is greater than or equal to  b
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 8

AIM : Shell script to demonstrate logical operators

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
read -p "Enter a:" a
read -p "Enter b:" b
if(($a == "true" & $b == "true" ))
then
	echo Both are true
else
	echo Both are not true
fi
if(($a == "true" || $b == "true" ))
then
	echo Atleast one of them is true
else
	echo None of them is true
fi
if(( ! $a == "true" ))
then
	echo a was initially false
else
	echo a was initially true
fi
```

```
neha@hpi5-22:~$ vi logical.sh
neha@hpi5-22:~$ chmod +x logical.sh
neha@hpi5-22:~$ ./logical.sh
Enter a:true
Enter b:true
Both are true
Atleast one of them is true
a was initially true
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 9

AIM : Shell script to demonstrate even and odd numbers

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
read -p "Enter a number :" num
if(( num % 2 == 0 ))
then
    echo "Number is even"
else
    echo "Number is odd"
fi
```

```
neha@hpi5-22:~$ vi evenodd.sh
neha@hpi5-22:~$ chmod +x evenodd.sh
neha@hpi5-22:~$ ./evenodd.sh
Enter a number :2
Number is even
```

```
neha@hpi5-22:~$ ./evenodd.sh
Enter a number :3
Number is odd
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 10

AIM : Shell script to demonstrate simple if loop

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
read -p "Enter first number :" $a
read -p "Enter second number :" $b
if [ $a == $b ]
then
   echo "a is equal to b"
fi
if [ $a != $b ]
then
   echo "a is not equal to b"
fi
```

OUTPUT

```
neha@hpi5-22:~$ vi if.sh
neha@hpi5-22:~$ chmod +x if.sh
neha@hpi5-22:~$ ./if.sh
Enter first number :2
Enter second number :3
a is equal to b
a is not equal to b
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

EXPERIMENT NO : 11

AIM : Shell script to demonstrate if else loop

CO4 : Write shell scripts required for system administration

PROCEDURE

```
#!/bin/bash
read -p "Enter number :" num
if (( $num>=0 && $num<=10 ))
then
    echo " Number is between 0 and 10"
elif (( $num >= 11 && $num <= 20 ))
then
    echo "Number is betwwe 11 and 20"
elif (( $num >= 21 && $num <= 30 ))
then
    echo "Number is between 21 and 30"
else
    echo "Number is greater than 30"
fi
```

OUTPUT

```
neha@hpi5-22:~$ vi ifelse.sh
neha@hpi5-22:~$ chmod +x ifelse.sh
neha@hpi5-22:~$ ./ifelse.sh
Enter number :3
 Number is between 0 and 10
```

RESULT

The program was executed and the result was successfully obtained. Thus CO4 was obtained.