**Experiment No.: 3**

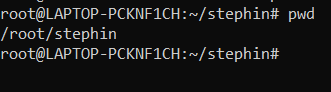
**Aim:** Familiarization of linux commands.

**CO2:** Perform system administration task.

**Procedure:**

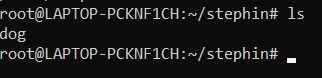
1.pwd - Print the working directory find the path of the current working directory

$pwd



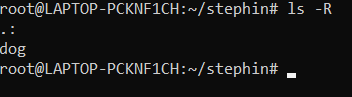
2.ls – To view the content of the directory

$ls

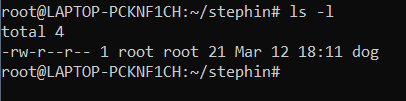


1.ls -R – To list the contents of sub directory

$ls -R

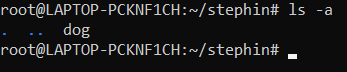


2.ls -l – Long listing of the contents



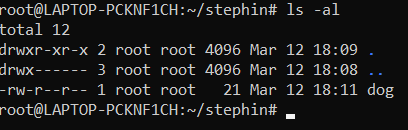
3.ls -a – To list the all hidden files

$ls -a



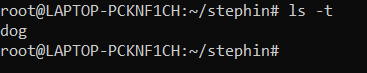
4.ls -al – List the files and directorys with detaild information.

$ls -al



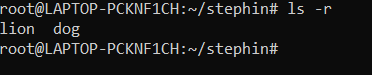
5.ls -t – List the files sorted in the order of last modified.

$ls -t



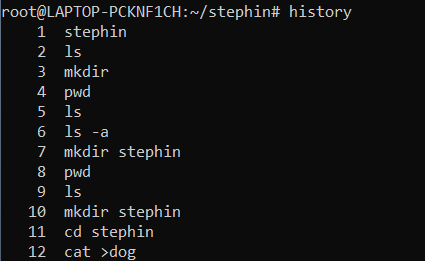
6.ls -r – To reverse the natural sorting order

$ls -r



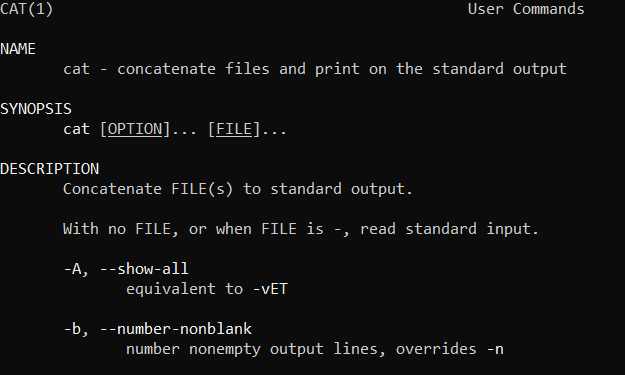
3.history – To review the command that have been previously executed for a certain period of time.

$history



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

$man



5.mkdir – To create a new directory

$mkdir



6.rmdir – To remove a directory

$rmdir



7.touch – To create new empty file

$touch

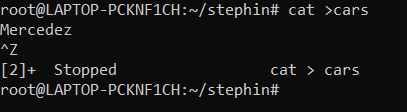


8.cat – Concatenate the files and print on the standard output

$cat

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

$cat > file.txt



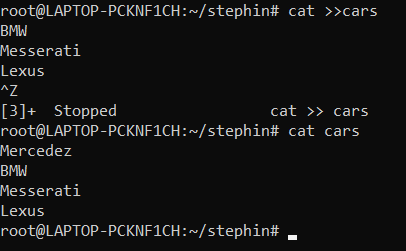
b. cat filename.txt – To view the content of the file

$cat file.txt



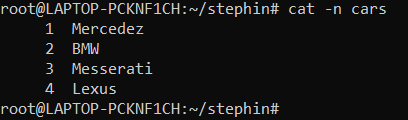
c. cat >> filename.txt – To append new contents to an existing file

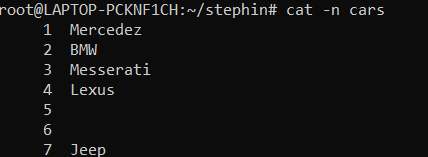
$cat >> file.txt



d. cat -n filename.txt – Number all output lines

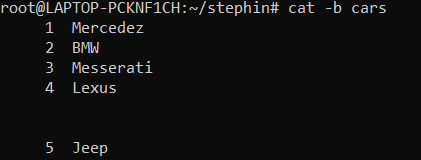
$cat -n file.txt

****

****

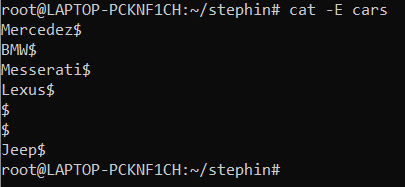
e. cat -b filename.txt – To remove the empty lines

$cat -b file.txt

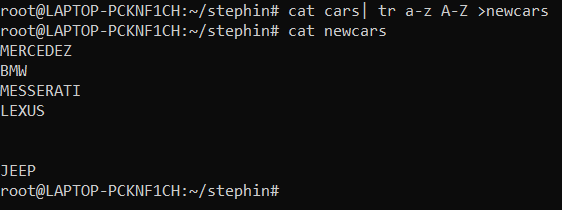


f. cat -E filename.txt – Display $ at end of each line

$cat -E file.txt



g. cat filename.txt| tr a-z A –Z > output.txt

****

**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No.: 4**

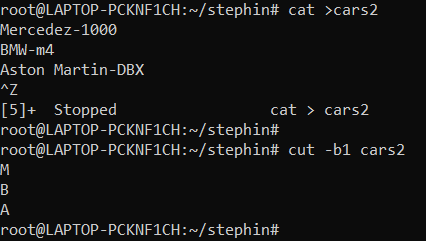
**Aim:** Familiarization of linux commands.

**CO2:** Perform system administration task.

**Procedure:**

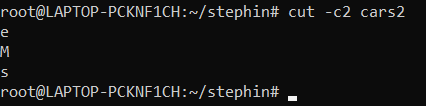
1.Cut by Byte position

$cut –b1 filename.txt



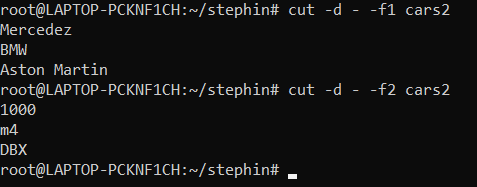
2.Cut by Character

$cut -c1 filename.txt



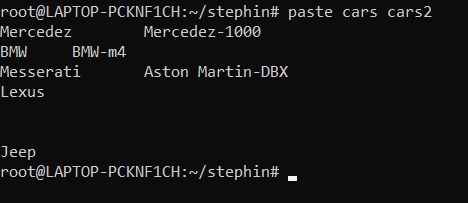
3.Cut 1st and 2nd column

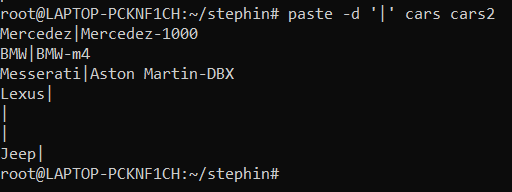
$cut –d - -f1 filename



4.Paste

$paste file1 file2





**Experiment-5**

Aim:

Familiarization of Linux Commands

Course Outcome(CO2):

Perform system administration task

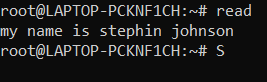
Procedure:

1. read :- Read content of one line of input into a variable

$read

echo $REPLY :- To print the input from the default variable

Output:



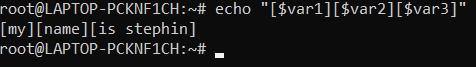
echo $REPLY :- To print the input from the default variable

* 1. read var1 var2 var3 :- To read into specific variables

$read var1 var2 var3

$echo “[$var1][$var2][$var3]”

Output:



* 1. read input \ :- To read multiple lines

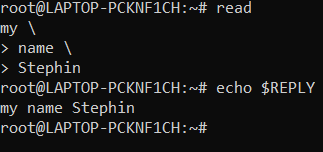
$read

My \

name is \

Stephin

Output:

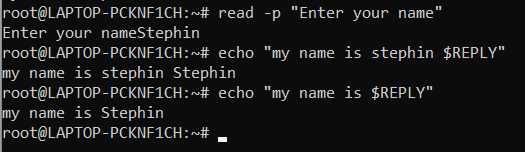


* 1. read -p :- Prompt text from user

$read -p “Enter your name”

$echo “my name is $REPLY”

Output:



* 1. read -n :- Specify limit

$read -n 6 -p “Enter six characters only”

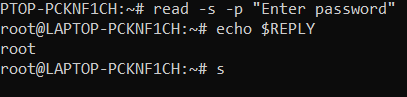
Output:



* 1. read -s :-For security. Hides input

$read -s -p “Enter the password: “

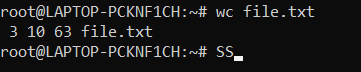
Output:



1. wc :- Word count display number of lines, number of words, number of bytes and file name

$wc file.txt

Output:



* 1. wc -l :- Display number of lines

$wc -l file.txt

Output:



* 1. wc -m :-Display number of bytes

$wc -m file.txt

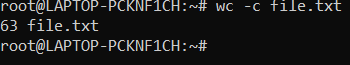
Output:



* 1. wc -c :- Display number of characters

$wc -c file.txt

Output:



* 1. wc -w :-Display number of words

$wc -w file.txt

Output:



* 1. wc -L :- Displays length of longest line

$wc -L file.txt

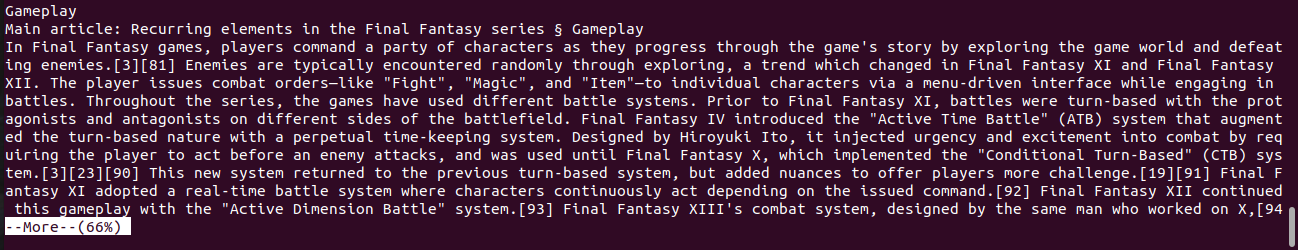
Output:



1. more :- It is similar to cat to display the content. The difference is that in case of larger files, cat command output will scroll off your screen while more command display output one screenful at a time.

$more finalfantasy.txt

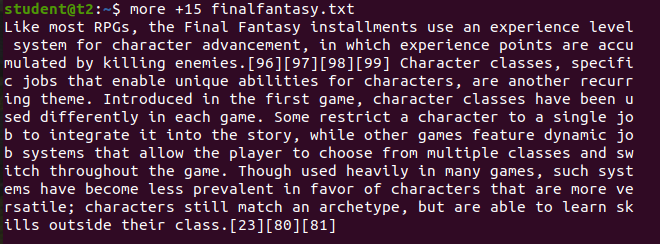
Output:



* 1. more +15 file.txt :- Will display content after the specified number of lines

$more +15 finalfantasy.txt

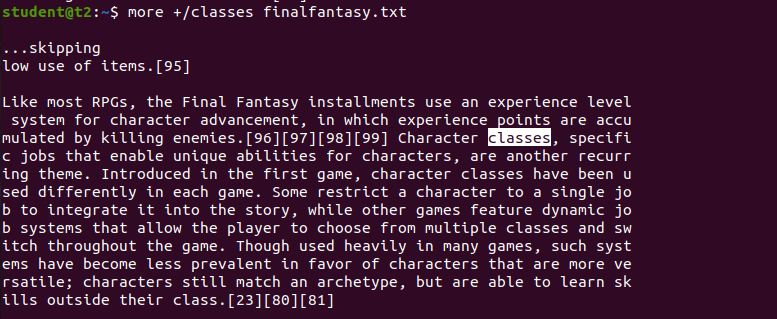
Output:

￼

* 1. more +/pattern file.txt :- Search and navigate towards a particular string and view all the instances.

$more +/classes finalfantasy.txt

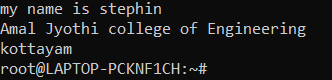
Output:



* 1. more -p file.txt :- Clear the whole screen and then display the text.

$more -p file.txt

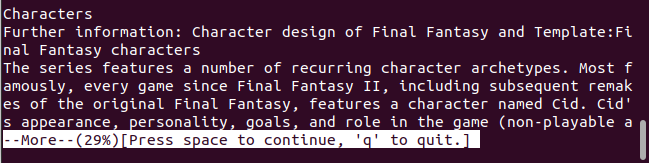
Output:



* 1. more -d file.txt :- Helps the user to navigate according to instructions, [space to continue and ‘q’ to quit]

$more -d finalfantasy.txt

Output:



**Experiment 6: Date:** 14/03/2023

Aim:

Familiarization of Linux Commands

Course Outcome(CO2):

Perform system administration task

Procedure:

1. grep :- Filtering and Searching content easily

$gerp 34 Mark

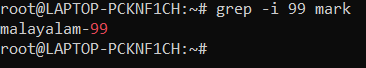
Output:



* 1. grep -i :- Case insensitive search of a particular content￼

$grep -i 42 Mark

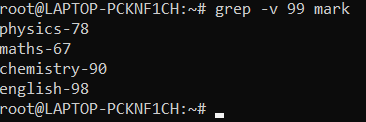
Output:



* 1. grep -v :- Exclude that content during search

$grep -v 42 Mark

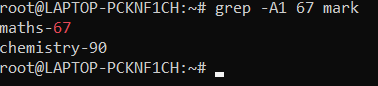
Output:



* 1. grep -A1 :- Specific content and one line after the content

$grep -A1 Maths Mark

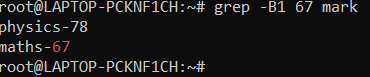
Output:



* 1. grep -B1 :- Specific content and one line before the content

$grep -B1 Maths Mark

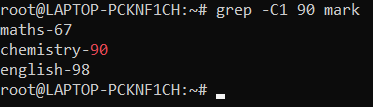
Output:



* 1. grep -C1 :- Specific content and one lone before and after the content.

$gerp -C1 Maths Mark

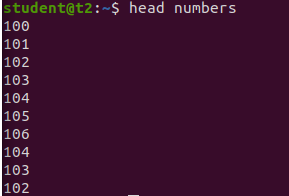
Output:



1. head :- Used to display the first content of the file(Top 10 lines by default)

$head numbers

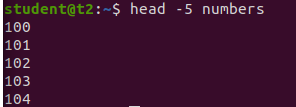
Output:-



* 1. head -number filename:- Specific number of lines

$head -5 numbers

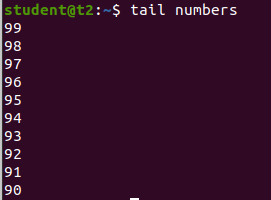
Output:



1. tail:- Used to display last contents of a file( last 10 by default)

$tail numbers

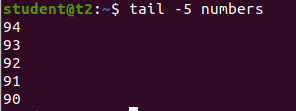
Output:



* 1. tail -number filename :- Specific number of content from last

$tail -5 numbers

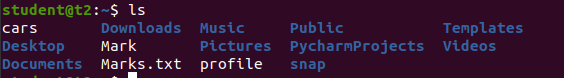
Output:



1. mv :- move from one location to another or it can be used to rename a file. Content will be overwritten.

$mv numbers Mark

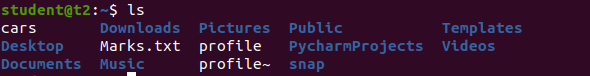
Output:-



* 1. mv -b :- To take backup of a file while moving.

$mv -b Mark profile

Output:



* 1. mv -i :- Prompt confirmation from user before overwriting.

$mv -i Marks.txt profile

Output:



# Experiment No: 7

**Aim:**

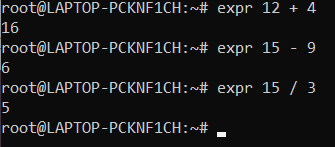
Familiarisation of Linux Commands.

# CO2:

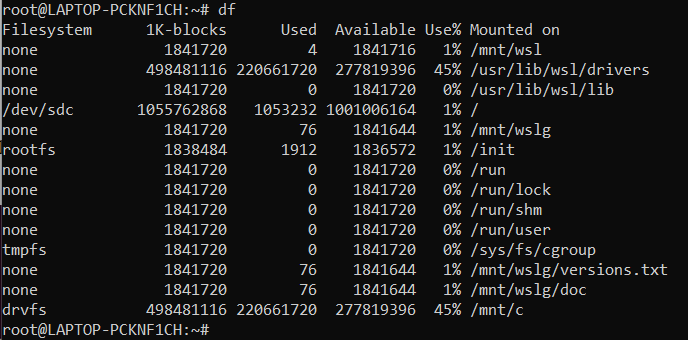
Perform System Administration tasks.

# Procedure:

**1)**Expr : To calculate the expression and print the output.



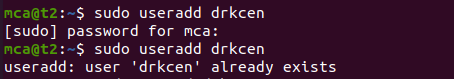
**2)**df - is used to get a report in system space usage ie.how space is taken by the files in the system



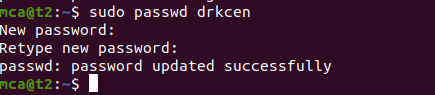
**3)**du- is used to check how much space a file/directory takes in the current directory



**4)**sudo useradd[username]



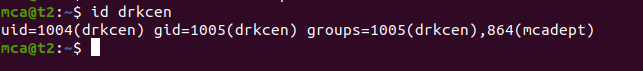
**5)**sudo passwd[username]-To update password



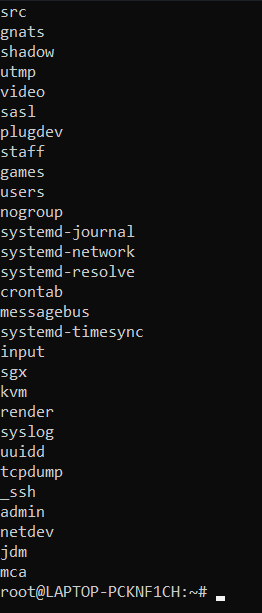
**6)**sudo groupadd -g (group id][groupname]- To create new group

sudo usermod -G[group name][username]

**7)**id [username]-used to find out user id, group id, group

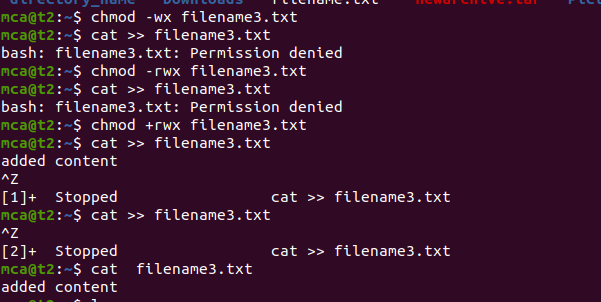


**8)**compgen -g -To Display all groups in the user

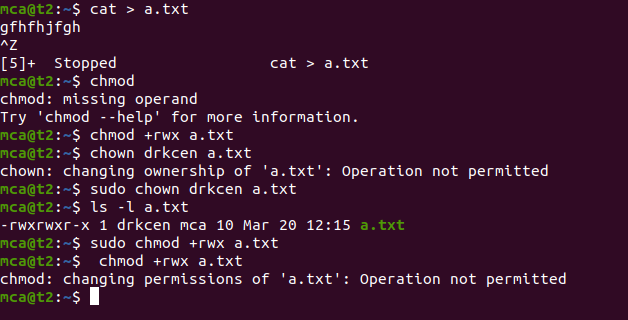


**9)**chmod- it is used to change the access permission of files and directory.Its stands for change mode

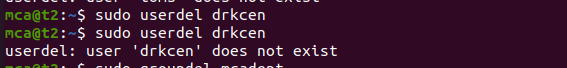
read(r),write(w),execute(x)



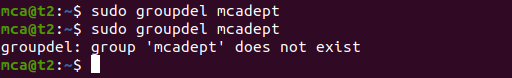
**10)**sudo chown [username][filename] -used to change a file ownership or directory ownership for a user or a group.chown stands for change owner



**11)**

****

**12)**



# Result:

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

# Experiment No: 8

**Aim:**

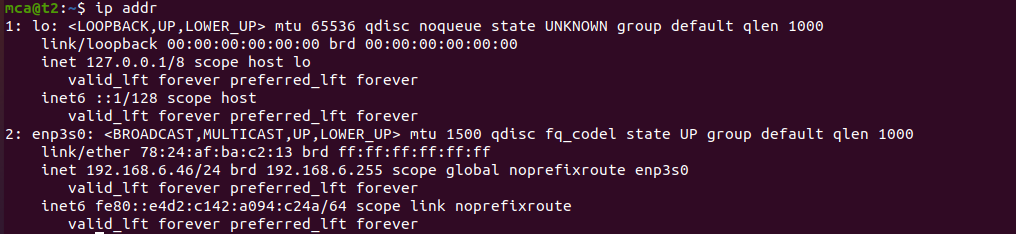
Familiarisation of Linux Commands.

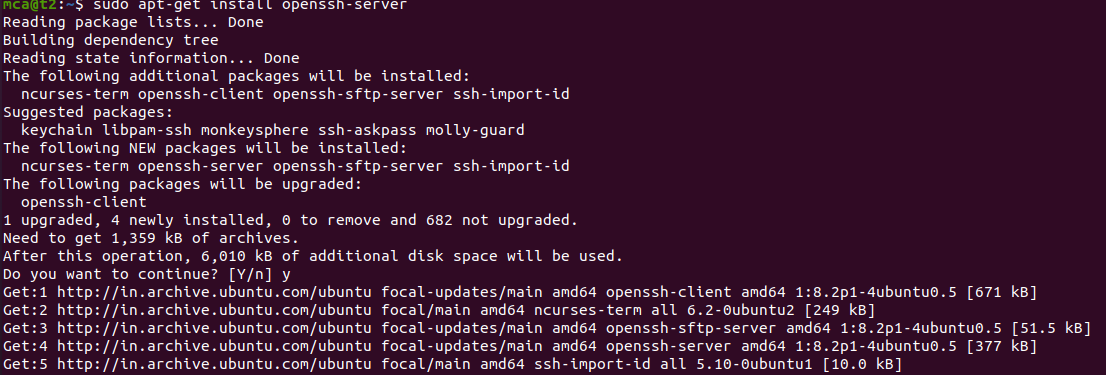
# CO2:

Perform System Administration tasks.

# Procedure:

**1)**ip addr-To get the ip address

****

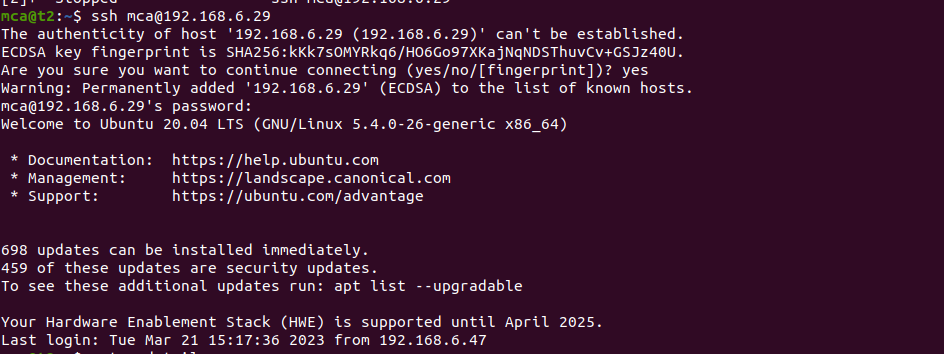
**2)**sudo apt-get install openssh-server****

**3)**sudo ufw allow [port no] - Allow access to the port

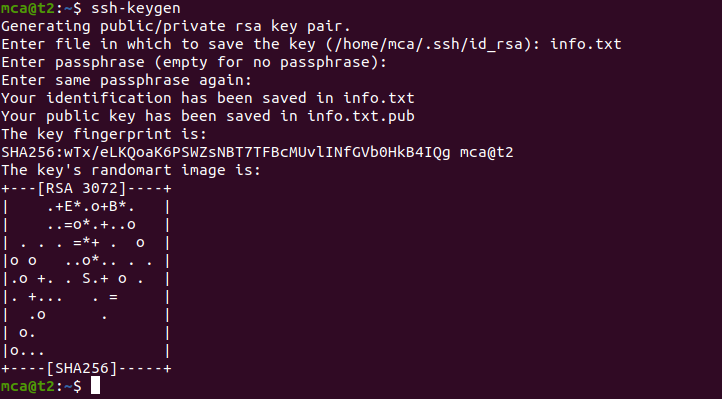
****

**4)** ssh mca@192.168.6.40 ie. ssh [username]@[neighbour ip address]

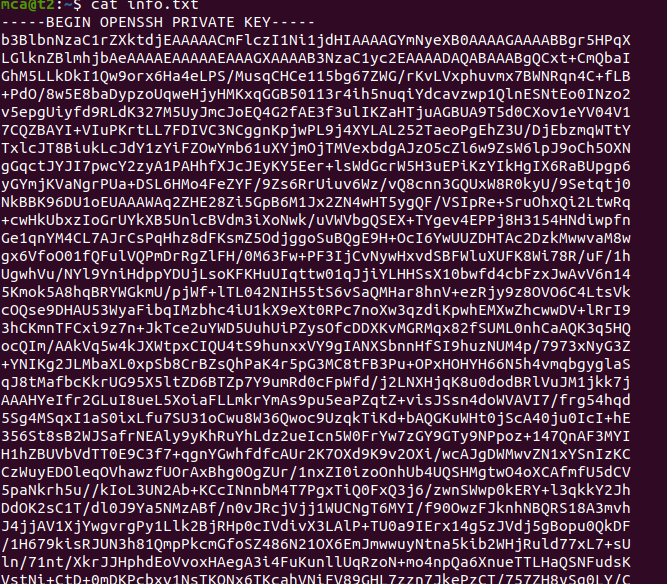
sshh stands for secure shell protocol used tyo securely connnect your or system .ssh is secures in sense it transfers data in encrypted form between host and client.

****

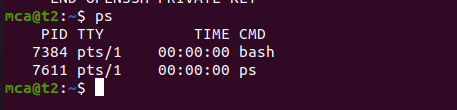
**5**)ssh-keygen -



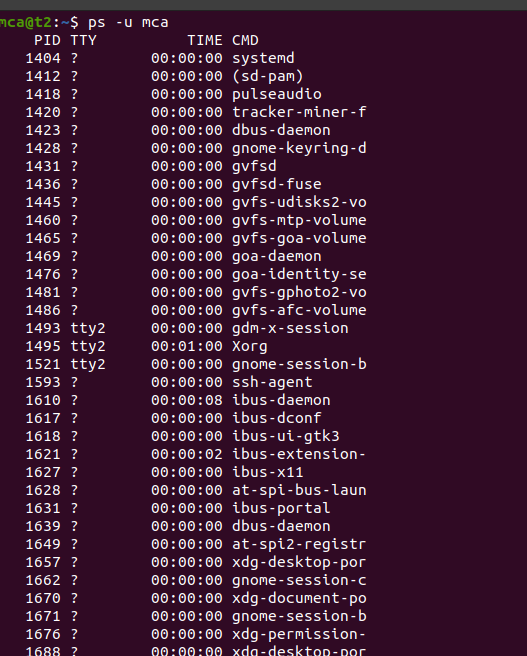
**6)**

****

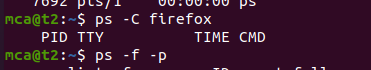
**7)** ps

****

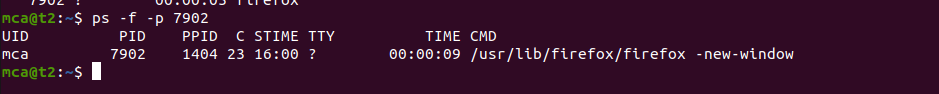
**8)**ps -u mca- Display the users currently running process information



**9**)ps -C firefox-

****

**10**)ps -f -p [process id number]

****

# Result:

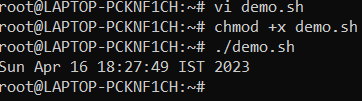
The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment 9 28-03-2023**

Aim : Shell scripting

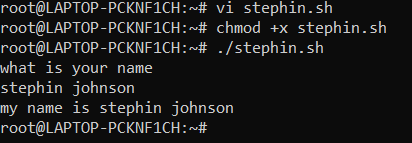
Procedure

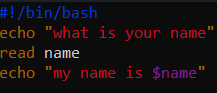
1.Shell Script to display the date



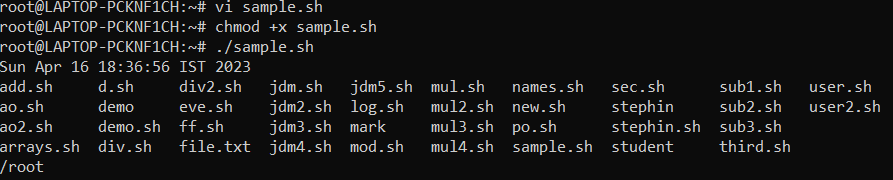


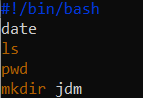
2.Shell Script to display your name



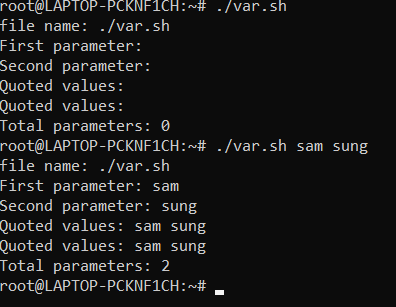


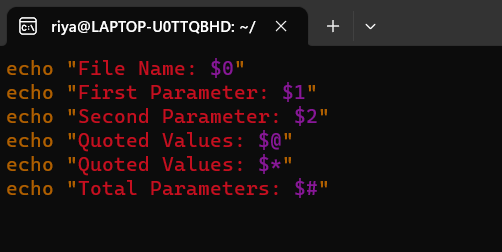
3.Shell Script to display date , pwd , ls(multiple commands)





**4.**Shell Script to demonstrate special variables

****

****

**5.Shell Script to count lines and words of the file**

**Experiment 10 Date:** 03/04/2023

**Aim:**

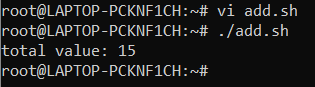
Familiarization of Linux Commands

**Course Outcome(CO4):**

Write shell scripts required for system administration

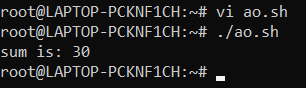
**Procedure:**

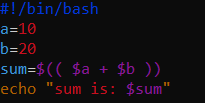
1. Shell script to add two number:

****

****

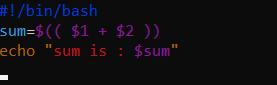
**2.** Write a shell script to initialize two numeric variables. Then perform addition operation on both values and store the result in the third variable.

****

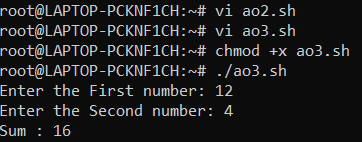
****

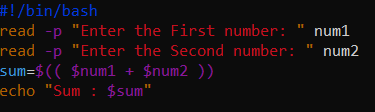
**3.** Shell script to read two numbers as command line parameters and perform the addition operation





4.Shell script which takes input from the user at run time and then calculate the sum of given number and store to a variable and show the result





**Result:**

Output displayed successfully and CO4 was obtained.

**Experiment 11 Date:** 04/04/2023

**Aim:**

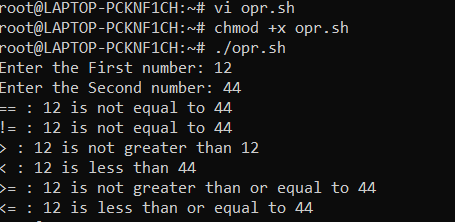
Familiarization of Linux Commands

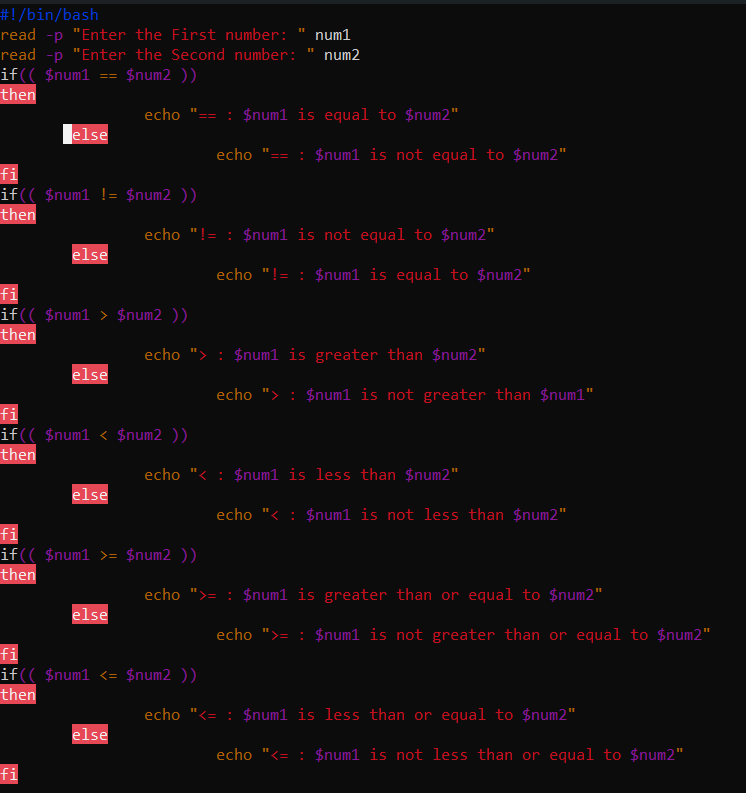
**Course Outcome(CO4):**

Write shell scripts required for system administration

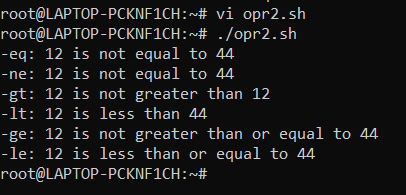
**Procedure:**

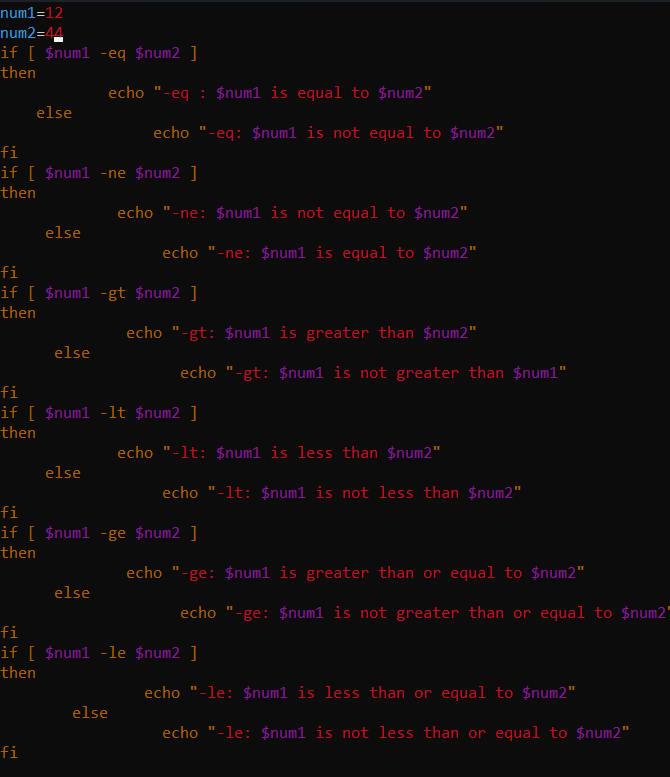
1. Shell script to demonstrate Relational operators (equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to) by taking user input



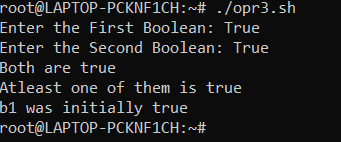


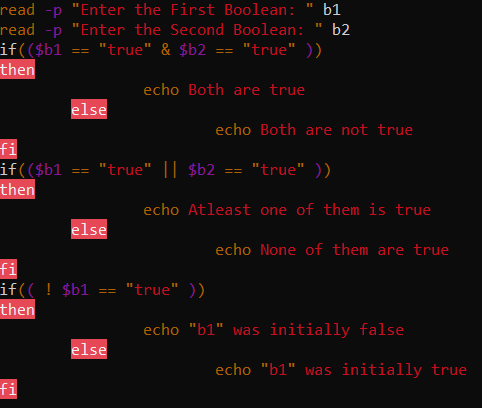
2. Shell script to demonstrate Relational operators (equal to, not equal to, greater than, less than, greater than or equal to, less than or equal to)



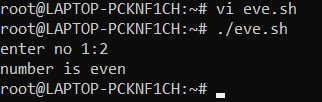


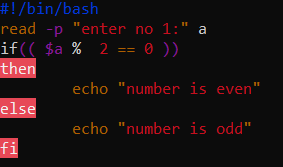
3.Shell script to demonstrate Logical operators (AND, OR, NOT) by taking user input.



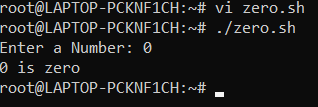


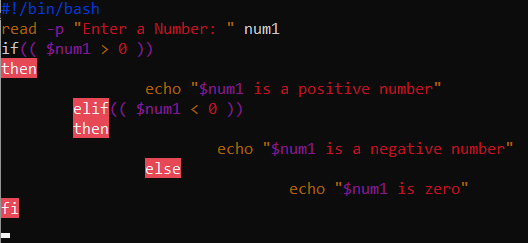
4.Write a shell script to check if a number is even or odd.



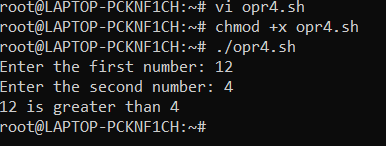


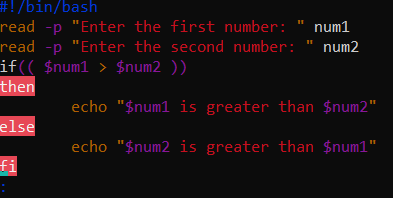
5.Write a shell script to check whether a number is positive or negative



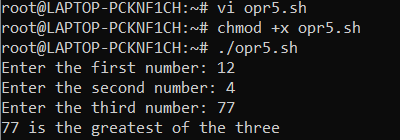


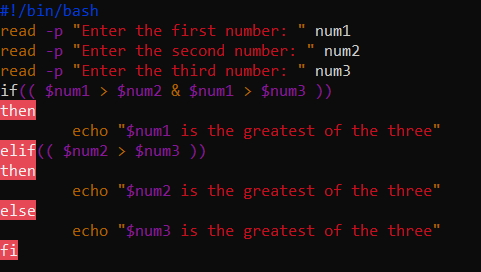
6. Write a shell script to find the greatest of two numbers





7. Write a shell script to find the greatest of three numbers





**Result:**

Output displayed successfully and CO4 was obtained.

**Experiment 12 Date:** 11/04/2023

**Aim:**

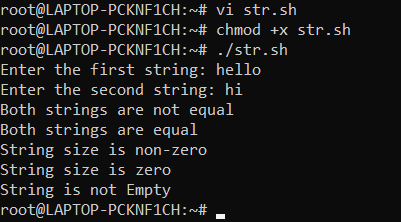
Familiarization of Linux Commands

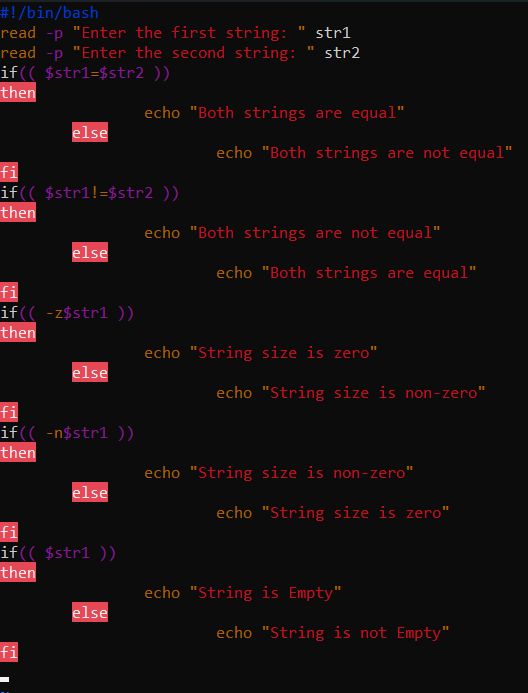
**Course Outcome(CO4):**

Write shell scripts required for system administration

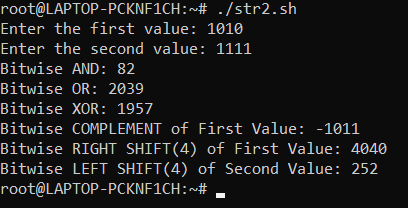
**Procedure:**

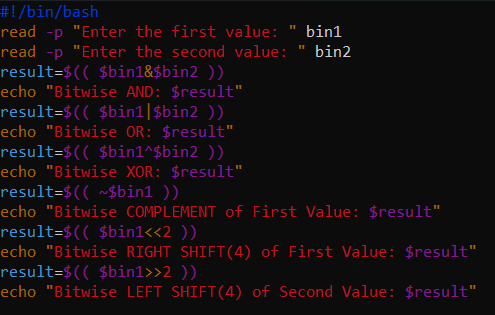
1. Shell script to demonstrate String operators (Equal, Not Equals, Size is zero, Size is non-zero, Empty string) by taking user input.



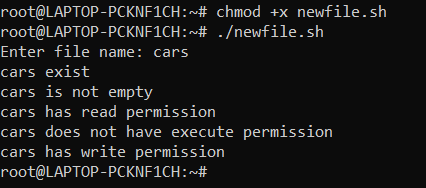


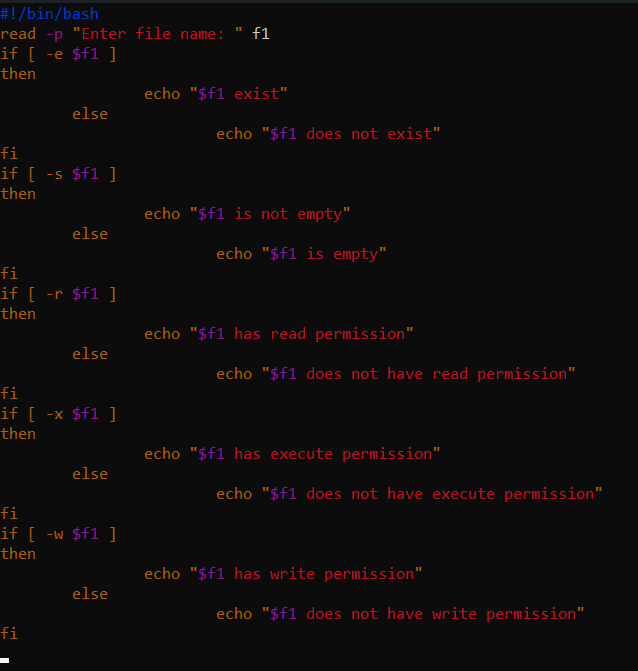
2. Shell script to demonstrate Bitwise operators (AND, OR, XOR, Complement, Right Shift, Left Shift) by taking user input



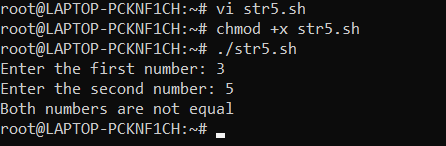


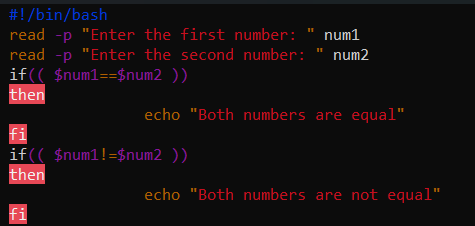
1. Shell script to demonstrate File Test operators (Exist(e), Size(s), Read Permission(r), Execute Permission(x), Write Permission(w)) by taking user input.





4. Shell Script to check if two numbers are equal using if statement





5. Shell Script to check the range of a number if numbers using else if ladder

