**Experiment No.: 3**

**Aim: Familarization of linux commands.**

**CO2: Perform System Administration Tasks.**

**Procedure**

1. Pwd-Print working directory

$ pwd

**Output Screenshot**

****

1. ls- to display a list of contents of a directory.

$ ls

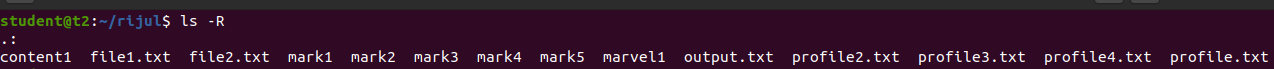
**Output Screenshot**

****

2.1 ls -R -show contents in the subdirectory.

$ls -R

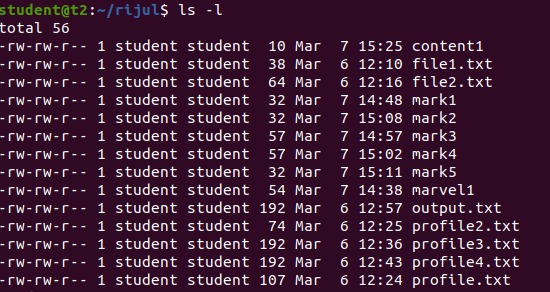
**Output Screenshot**

****

2.2 ls -l - to longlist a list.

$ls -l

**Output Screenshot**

****

2.3 ls-a – to list all hidden files.

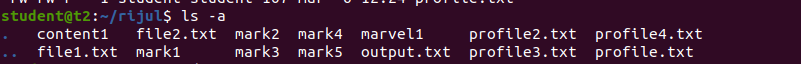
$ls -a

**Output Screenshot**

2.4 ls -al – to list the files and directories with detailed information.

$ls -al

**Output Screenshot**



2.5 ls -t – to list the files as in last modified order.

$ls -t

**Output Screenshot**

****

2.6 ls -r - to remove the natural sorting order.

$ls -r

**Output Screenshot**

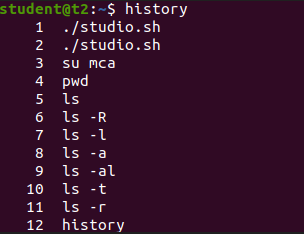
****

3. history- to review the commands that has been previously executed for certain period of

Time.

$history

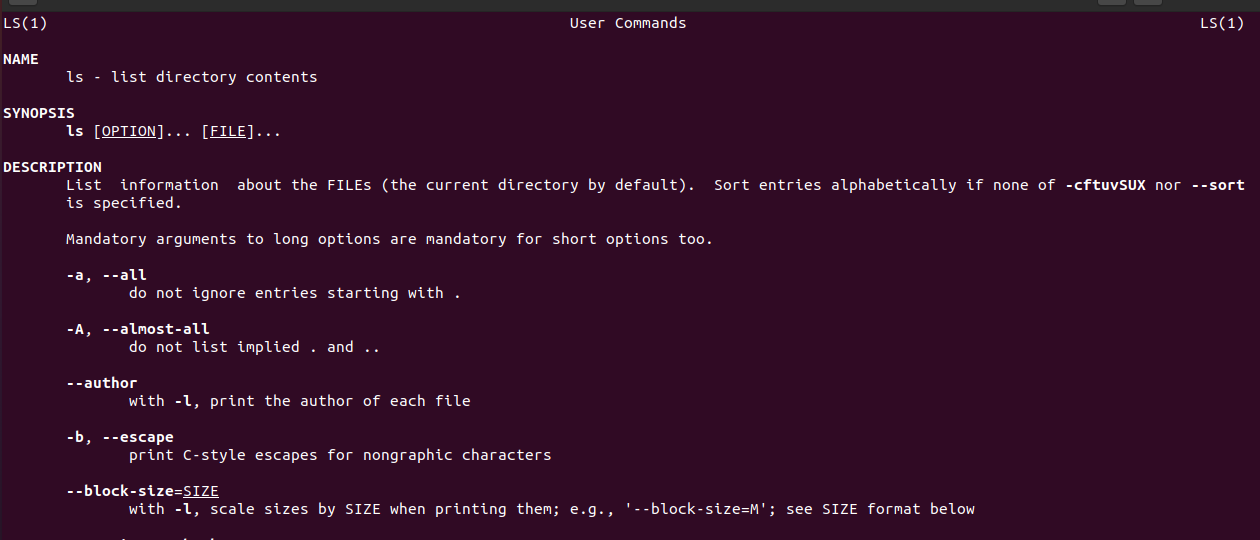
**Output Screenshot**

****

1. man – list options of various commands right from the shell.

$man ls

**Output Screenshot**

****

5. mkdir – make a new directory

$mkdir rijul2

6. cd – change directory

$cd rijul2

7.cd.. – move to previous directory.

$cd ..

8.rmdir- remove an existing directory

$rmdir

9.touch- create a new file

$touch file.txt

10. cat > filename: create a new file and append the content.

$cat > fruit.txt

Apple

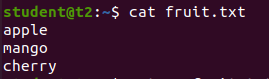
Mango

Cherry

11. cat filename – to display the file content

$cat fruit.txt

**Output Screenshot**



12. cat >> filename- add more content to the same existing file.

$cat >> fruit.txt

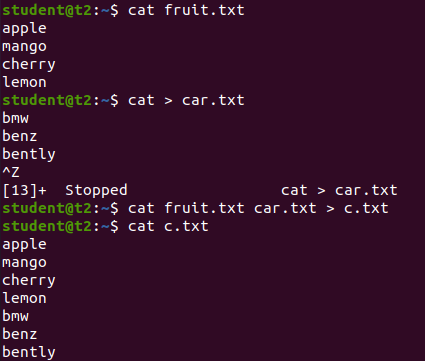
Lemon

13.cat filename filename > filename – combine the contents of the two file and store it in another file.

$cat fruit.txt car.txt > c.txt

$cat c.txt

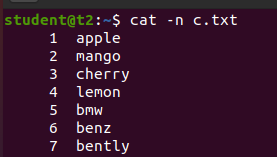
**Output Screenshot**



14. cat -n filename – display the content with line numbers

$cat -n c.txt

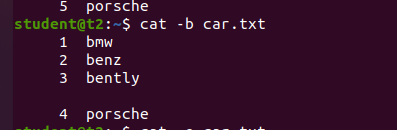
**Output Screenshot**



14.1 cat -b filename – remove empty line numbers

$cat -b car.txt

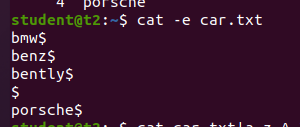
**Output Screenshot**



14. cat -e filename – append $ at end of each line.

$cat -e car.txt

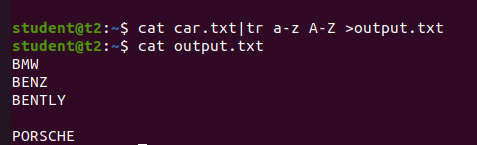
**Output Screenshot**



15. cat filename| tr a-z A-Z > filename – change context of file to uppercase.

$cat car.txt| tr a-z A-Z > output.txt

**Output Screenshot**



**Result**

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

**Experiment No.: 4**

**Aim: Familarization of linux commands.**

**CO2: Perform System Administration Tasks.**

**Procedure**

1. cut – for cutting out the sections from each lines of files and writing the result to standard output.

1.1 cut -b filename – cut the letters of words from each lines

$cat >marvel1

Ironman

Captain America

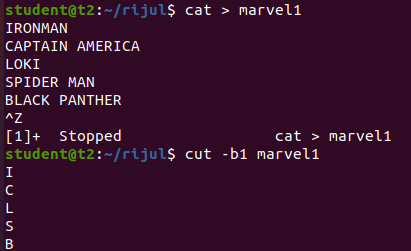
Loki

Spiderman

Blackpanther

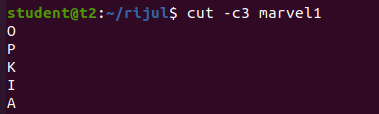
$cat -b1 marvel1

**Output Screenshot**



1.2 $cut -c3 marvel1 – cutting the third letter of words from each line.

**Output Screenshot**

****

1.3 cut -d - -f1 filename – cutting the highfen(delimiter) from the file.

$cat > mark1

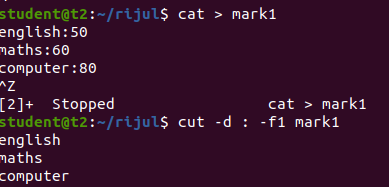
English-50

Maths-60

Computer-80

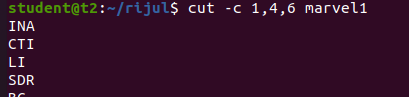
$cut -d - -f1 mark1

**Output Screenshot**



1.4 $cut -c 1,4,6 marvel1 – cutting 1st,4th and 6th letters.

**Output Screenshot**

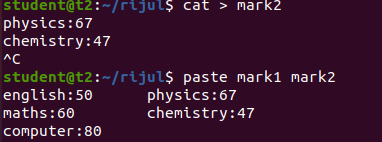


2. paste- join files horizontally[each file consisting of different lines]

2.1 paste filename1 filename2

$paste mark1 mark2

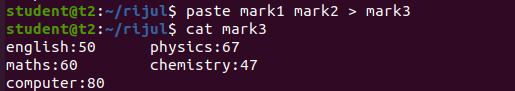
**Output Screenshot**



2.2 $paste mark1 mark2 > mark3

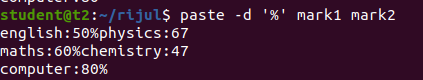
$cat mark3

**Output Screenshot**



2.3 $paste -d ‘%’ mark1 mark2 – appending % at list.

**Output Screenshot**



2.4 $paste -d ‘%’ mark1 mark2 > mark4

$cat mark4

English-50%physics-67

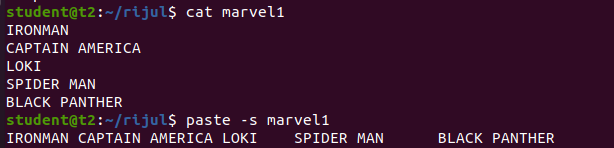
Maths-60%chemistry-47

Computer-80%

2.5 paste -s filename – display the content in the same line.

$paste -s marvel1

**Output Screenshot**



3. cp file1 file2- to copy the content to an existing file.

$cat mark1

English-50

Maths-60

Computer-80

$cat mark2

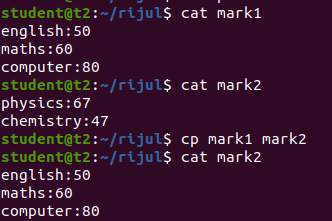
Phy-67

Chem-47

$cp mark1 mark2

$cat mark2

**Output Screenshot**

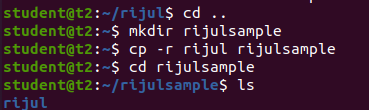


3.1 $cp -r rijul rijulsample – copy a directory along with subdirectory to another directory.

$cd rijulsample

Ls

**Output Screenshot**

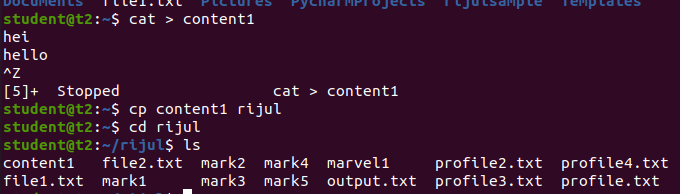


3.2 $cp content rijul – copy an content from one directory to another.

$cd rijul

$ls

**Output Screenshot**



**Result**

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

**Experiment No.: 5**

**Aim: Familarization of linux commands.**

**CO2: Perform System Administration Tasks.**

**Procedure**

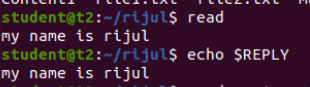
1. read- to read the content of a file

$ read

My name is rijul

$echo $REPLY

**Output Screenshot**

****

1.1 $read var1 var2 var3

My name is rijul

$ echo “[$var1][$var2][$var3]”

**Output Screenshot**

****

1.2 $ read

My \

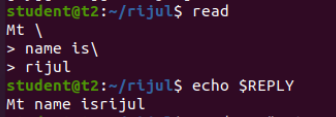
\-> to write on a different line.

>name is \

>rijul

$echo $REPLY

**Output Screenshot**

****

1.3 read -p -> prompt an input from the user.

$ read -p “Enter your name:”

**Output Screenshot**

****

$ echo “my name is $REPLY”

**Output Screenshot**

****

1.4 $read -n 6 -p “ “ -> able to enter 6 characters only.

$ read -n 6 -p “Enter 6 characters only.

**Output Screenshot**

****

1.5 $ read -s -p “enter the password:”

**Output Screenshot**

****

$ echo “password is $REPLY”

**Output Screenshot**

****

$ cat > profile

**Output Screenshot**

****

2. wc – to display the word count in an format.

$ wc profile

**Output Screenshot**

****

2.1 $wc -l profile : to display the no.of lines.

**Output Screenshot**

****

2.2 $wc -m profile : display the no.of characters.

**Output Screenshot**

****

2.3 $ wc -c profile : display no.of bytes.

**Output Screenshot**

****

2.4 $ wc – w profile : display the no.of words

**Output Screenshot**

****

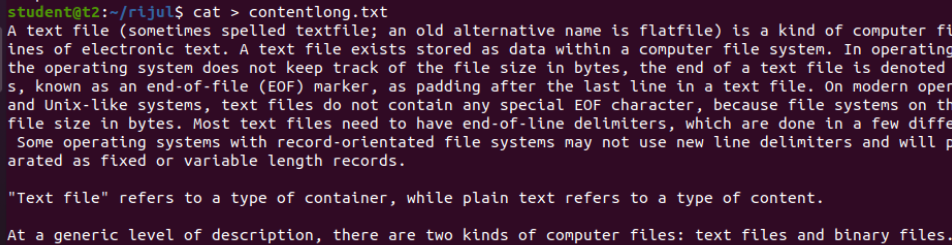
2.5 $wc -L profile : print length of the longest line.

**Output Screenshot**

****

$ cat contentlong.txt

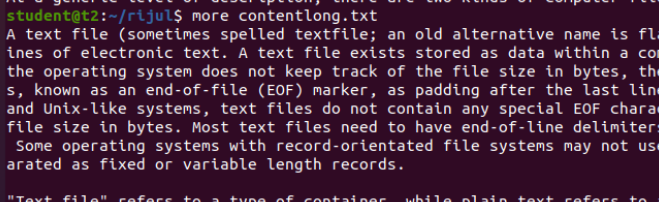
**Output Screenshot**

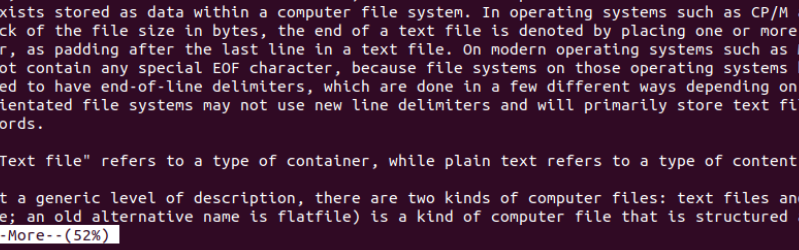
****

3. more – similar to cat to display the content, the only difference is that in case of large file cat command output will scroll off your screen while more command display output one screenful at a time.

$ more contentlong.txt

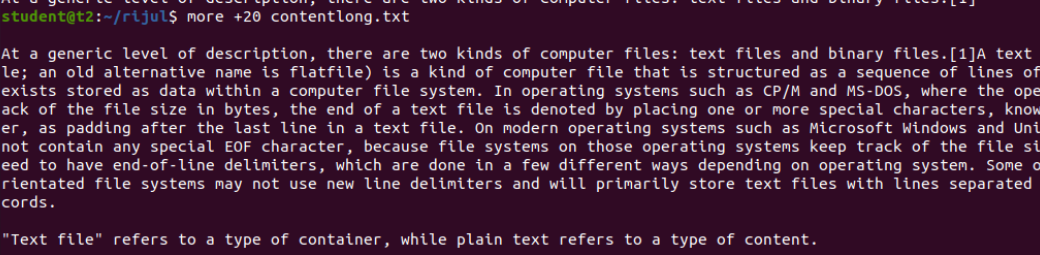
**Output Screenshot**

****



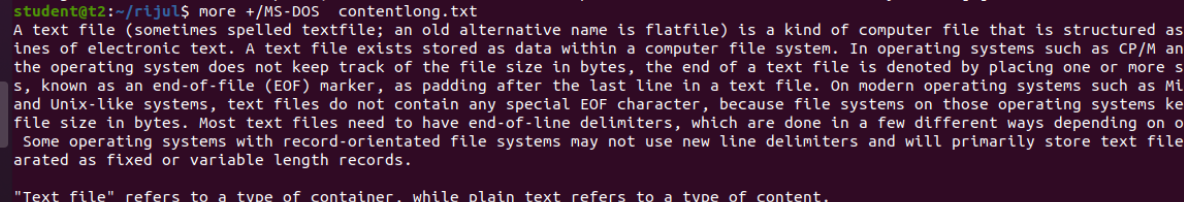
3.1 $ more +20 contentlong.txt -> display contents after 20 lines.

**Output Screenshot**

****

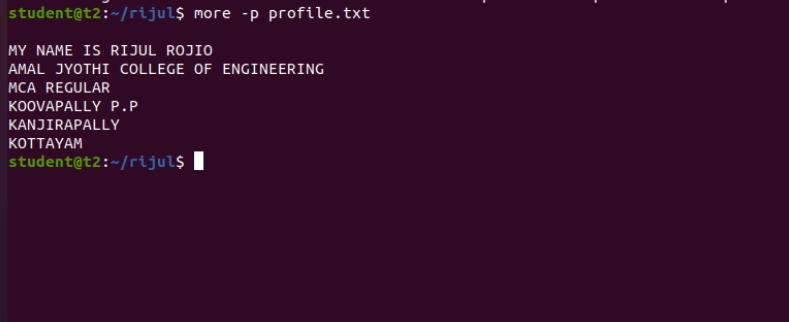
3.2 $ more +/ms-dos{pattern} contentlong.txt -> search the pattern string from the content and view all the instances,by navigating through the result.

**Output Screenshot**

****

3.3 $more -p contentlong.txt -> clear the screen and show the output.

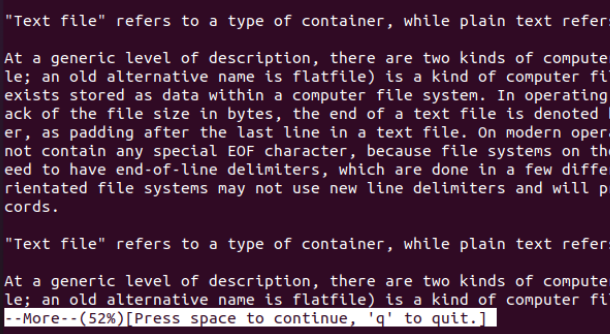
**Output Screenshot**

****

3.4 $ more -d contentlong.txt -> help the users to navigate,press space to continue,q to quit

**Output Screenshot**

****



**Result**

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

**Experiment No.: 6**

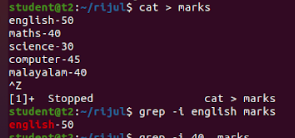
**Aim: Familarization of linux commands.**

**CO2: Perform System Administration Tasks.**

**Procedure**

1. grep-  is used to filter the contents,which makes our search easy.

**Output Screenshot**



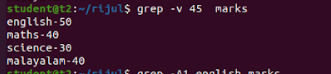
1.1  $grep -i content file\_name   -> case sensitive search

**Output Screenshot**



1.2 $grep -v content filename   ->inverted search(all content except the search content displays)

**Output Screenshot**



1.3 $grep -A1 content file\_name   ->view the content along with the  line after.

**Output Screenshot**



1.4 $ grep -B1 content file\_name     ->display content before also.

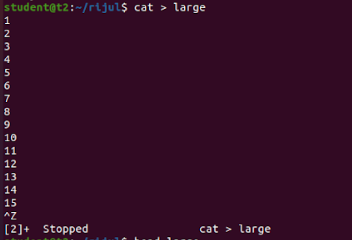
**Output Screenshot**



1.5 $ grep -C1 content file\_name    ->before and after the content displays.

**Output Screenshot**

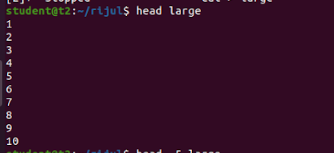




1. head - display the top content of the file by default.

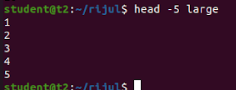
$ head file\_name

**Output Screenshot**



2.1 $ head -5 file\_name  ->displays first 5 lines.

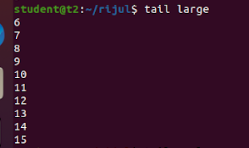
**Output Screenshot**

****

1. tail- display the content of the file by default it displays last ten lines.

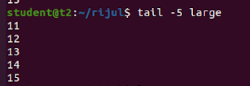
$tail file\_name

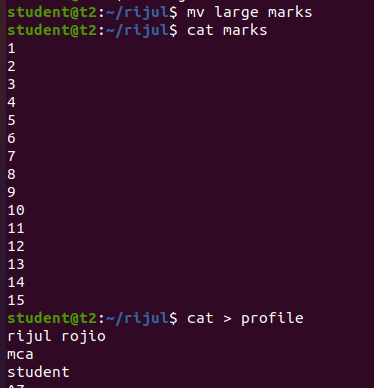
**Output Screenshot**



3.2 $tail -5 file\_name

**Output Screenshot**

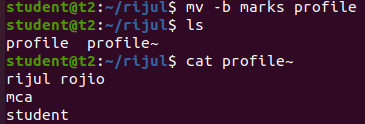


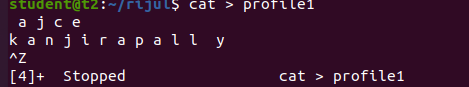


4.. mv - move

$mv -b file1 flie2   ->backups

**Output Screenshot**





4.1 $mv -i file1 file2 ->displays prompt message.

**Output Screenshot**



**Result**

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

**Experiment No.: 7**

**Aim: Familarization of linux commands.**

**CO2: Perform System Administration Tasks.**

**Procedure**

1. expr- evaluate the given expression and display the output.

1.1 $ expr 12 + 8

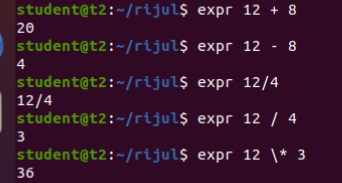
1.2 $ expr 12 - 8

1.3 $ expr 12 / 4

1.4 $ expr 12 /\* 3

1.5 $ expr $x + $y

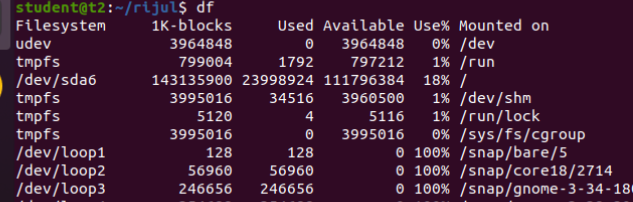
**Output Screenshot**

****

1. df -  disutilised information (represent system disk space usage)

$ df

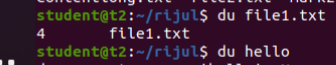
**Output Screenshot**



1. du - how much space a file or directory takes in the current directory

$ du file1.txt

**Output Screenshot**



1. $ sudo add user\_name  -> add a new user to the system.

$ sudo add rijul

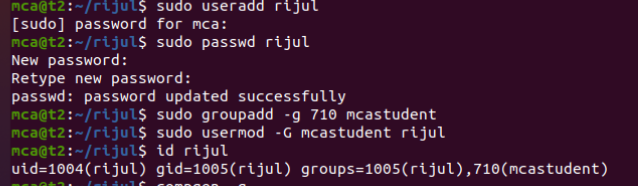
4.1 $ sudo psswd rijul

4.2 $ sudo groupadd -g 710 mca student -> adding a new group mca student with unique identity 710.

4.3 $ sudo usermod -G mcastudent rijul

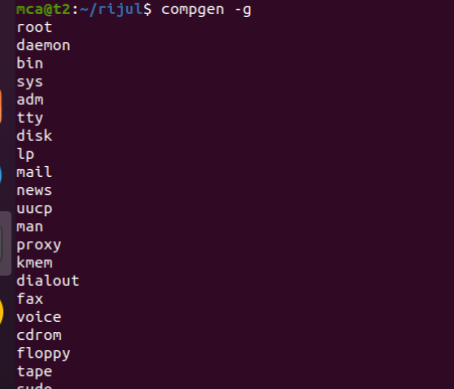
4.4 $ id rijul

**Output Screenshot**



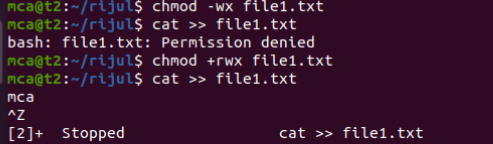
1. $ compgen -g

**Output Screenshot**



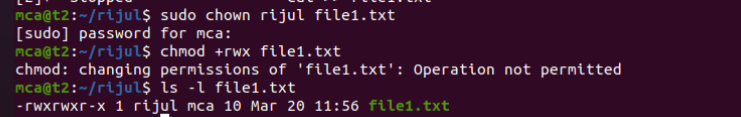
1. $ chmod -wx file1.txt  ->deny permission for write and execute.
2. $ chmod +rwx file1.txt   ->give permission to read,write,execute

**Output Screenshot**



8. $ sudo chown user\_name file1.txt ->change owner permission

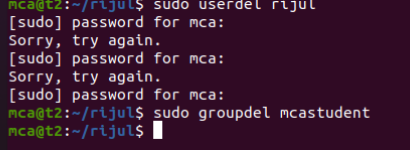
1. **Output Screenshot**



10. $ sudo userdel username

1. $ sudo groupdel groupname

**Output Screenshot**



**Result**

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

**Experiment No.: 8**

**Aim: Familarization of linux commands.**

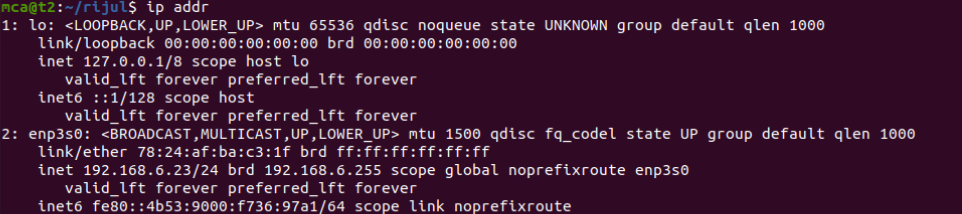
**CO2: Perform System Administration Tasks.**

**Procedure**

1. ip addr -> ip address

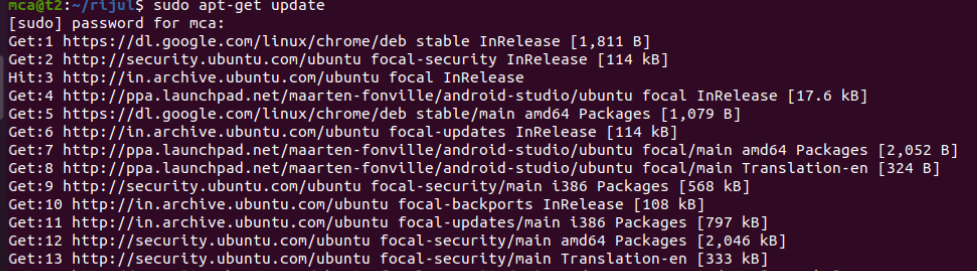
$ ip addr

**Output Screenshot**



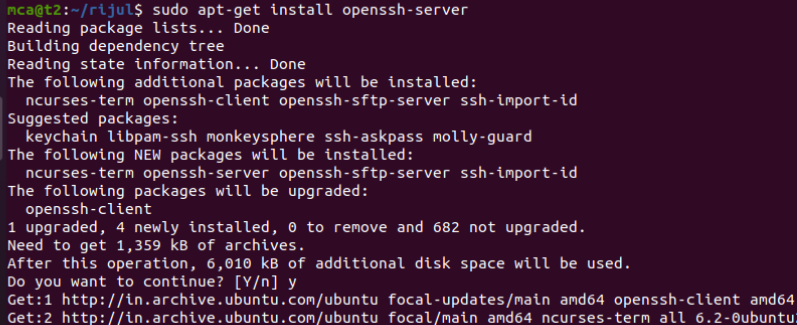
2. $ sudo apt-get update

**Output Screenshot**



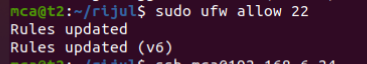
3. $ sudo apt-get install openshh-server

**Output Screenshot**



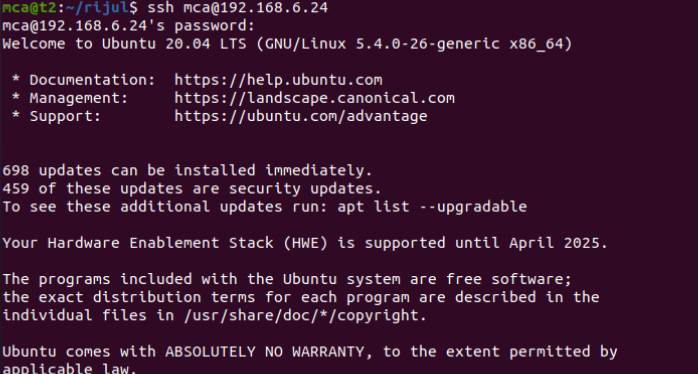
4. $ sudo ufw allow 22 -> 22-port number

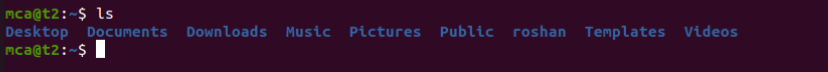
**Output Screenshot**



5. ssh user@portnumber -> ssh-secure shell protocol : used to securely connect to a remote server or a system.ssh is secured in reuse,it transfer data in encrypted form between host and clients.

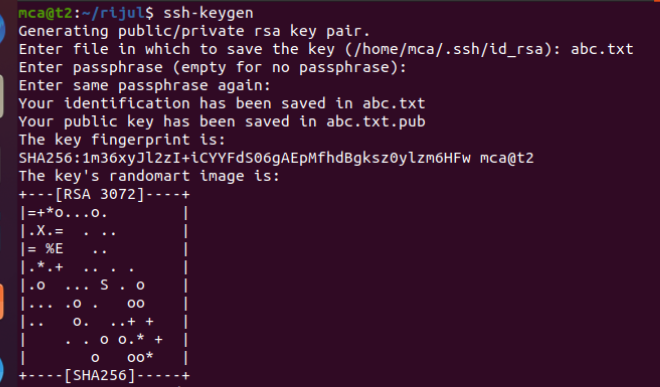
**Output Screenshot**

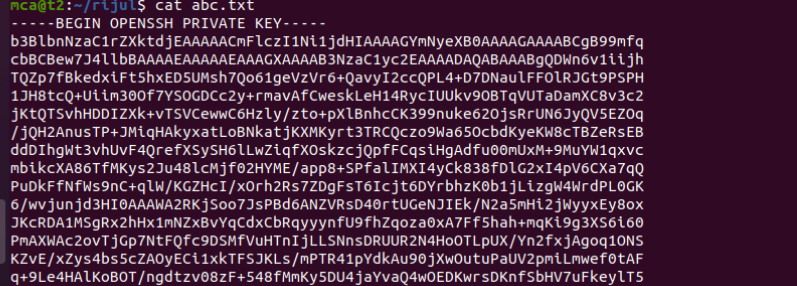




6. $ shh-keygen

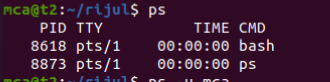
**Output Screenshot**





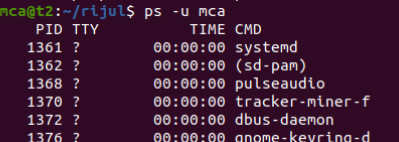
7. $ ps

**Output Screenshot**



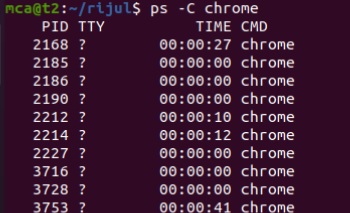
7.1 $ ps -u user\_name

**Output Screenshot**



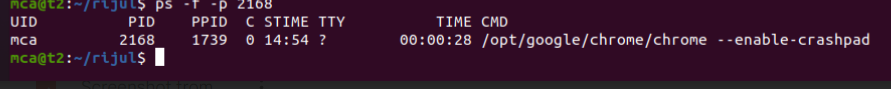
7.2 $ ps -c chrome

**Output Screenshot**



7.3 $ ps -f -p 2168

**Output Screenshot**



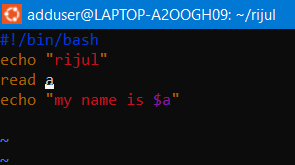
**RESULT**

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

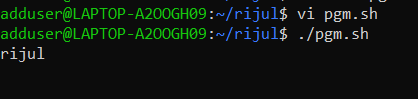
**Experiment No.: 9**

**Aim: Shell script to display your name.**

**Procedure**



**Output Screenshot**

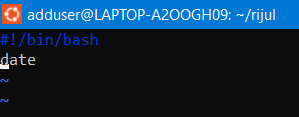


**RESULT**

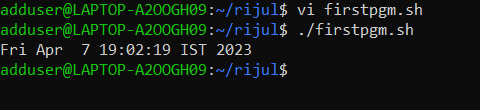
**Experiment No.: 10**

**Aim: Shell script to display date.**

**Procedure**



**Output Screenshot**

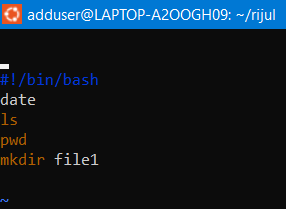


**RESULT**

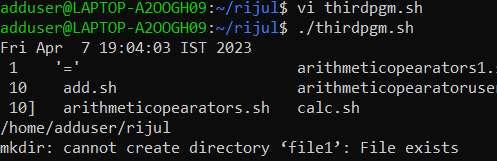
**Experiment No.:11**

**Aim: Shell script to display pwd,ls,date commands .**

**Procedure**



**Output Screenshot**

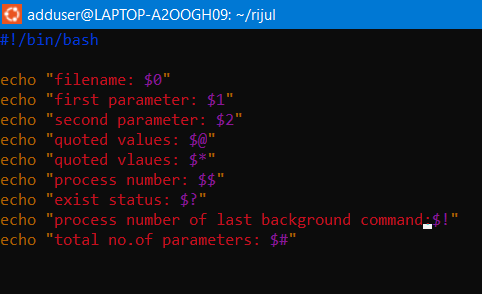


**RESULT**

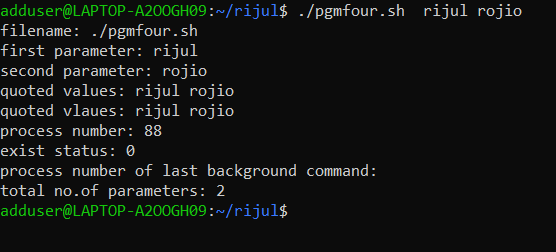
**Experiment No.: 12**

**Aim: Shell script to demonstrate variables.**

**Procedure**



**Output Screenshot**

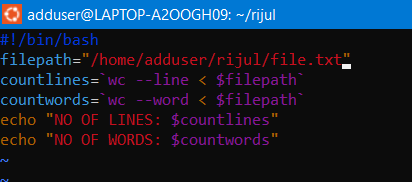


**RESULT**

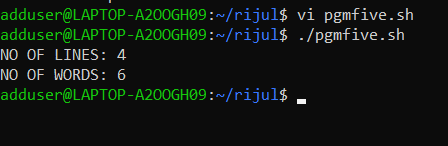
**Experiment No.: 14**

**Aim: Shell script to count words and lines in a file.**

**Procedure**



**Output Screenshot**

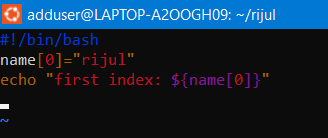


**RESULT**

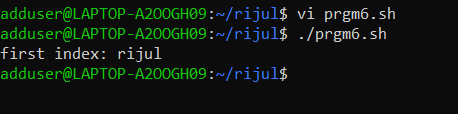
**Experiment No.: 15**

**Aim: Shell script to display array index.**

**Procedure**



**Output Screenshot**

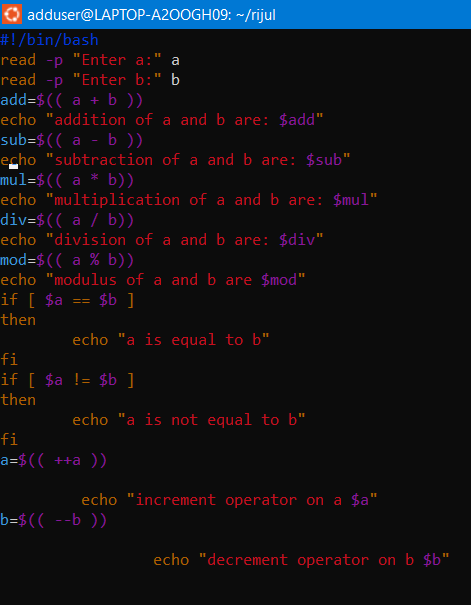


**RESULT**

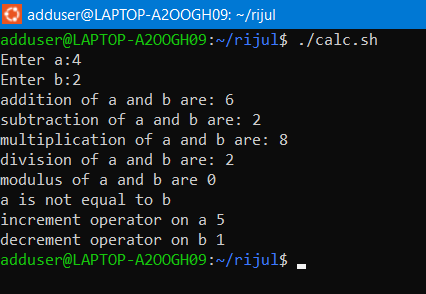
**Experiment No.: 16**

**Aim: Shell script to demonstrate Arithmetic operations.**

**Procedure**



**Output Screenshot**

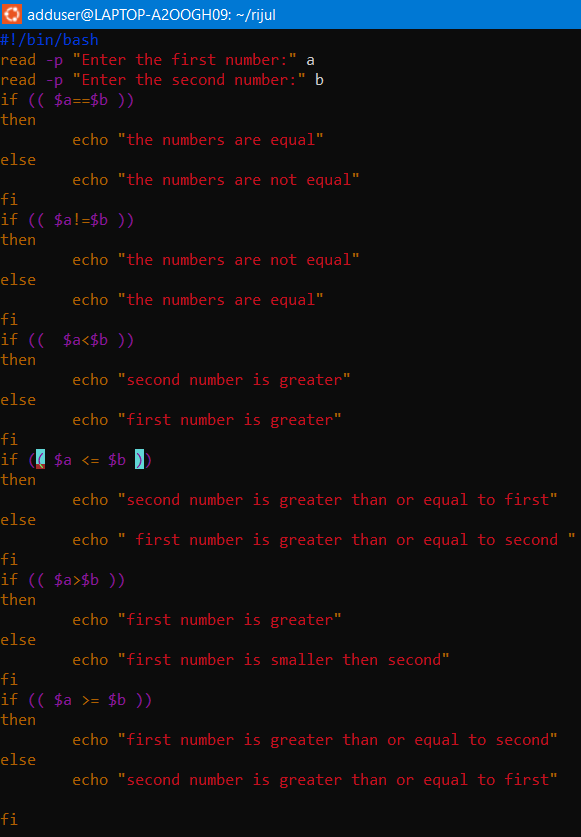


**RESULT**

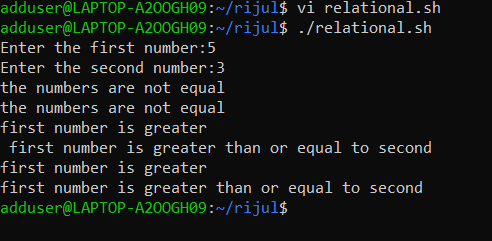
**Experiment No.: 17**

**Aim: Shell script to demonstrate Relational operations.**

**Procedure**



**Output Screenshot**

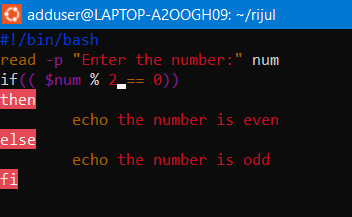


**RESULT**

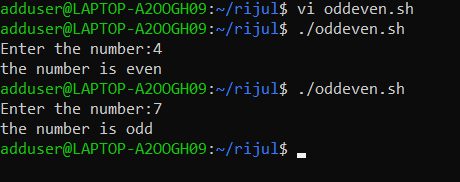
**Experiment No.: 18**

**Aim: Shell script to check whether a number is odd or even.**

**Procedure**



**Output Screenshot**

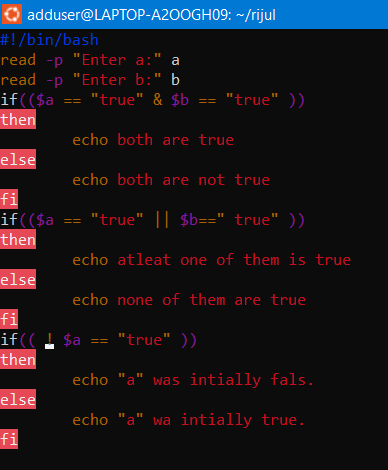


**RESULT**

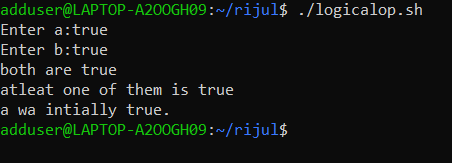
**Experiment No.: 19**

**Aim: Shell script to demonstrate Boolean operations.**

**Procedure**



**Output Screenshot**

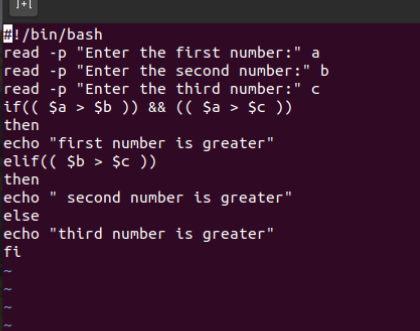


**RESULT**

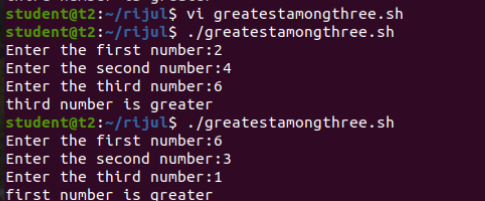
**Experiment No.: 20**

**Aim: Shell script to find the greatest of three numbers .**

**Procedure**



**Output Screenshot**



**RESULT**

**Experiment No.: 21**

**Aim: Shell script to demonstrate String operations.**

**Procedure**



**Output Screenshot**

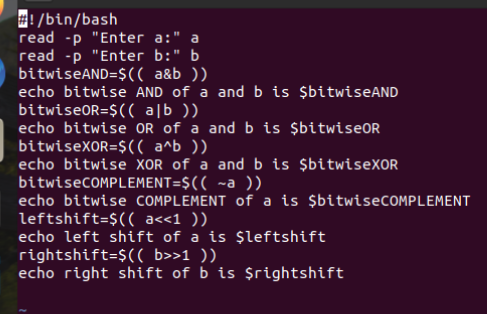


**RESULT**

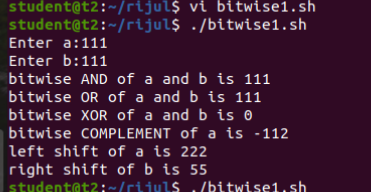
**Experiment No.: 22**

**Aim: Shell script to demonstrate bitwise operations.**

**Procedure**



**Output Screenshot**

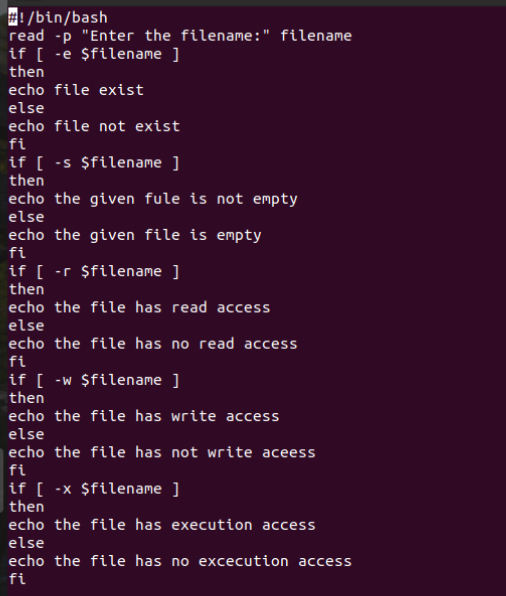


**RESULT**

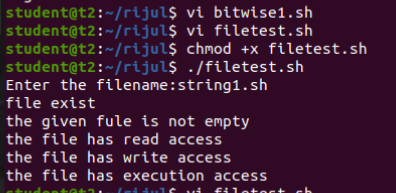
**Experiment No.: 23**

**Aim: Shell script to demonstrate file test methods.**

**Procedure**



**Output Screenshot**



**RESULT**

**Experiment No.: 24**

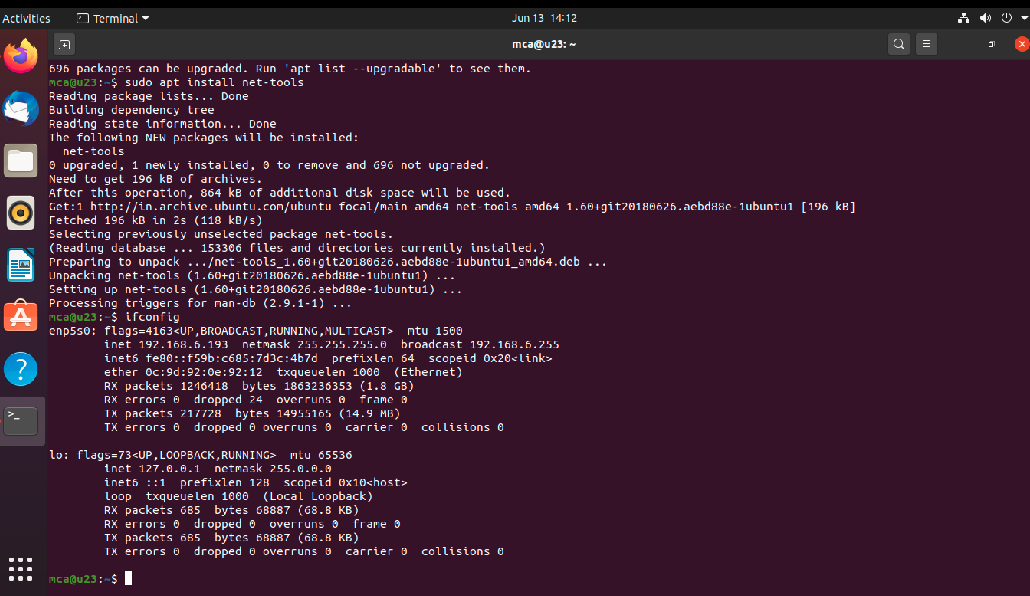
**Aim: Network  commands.**

**CO2: Perform System Administration Tasks.**

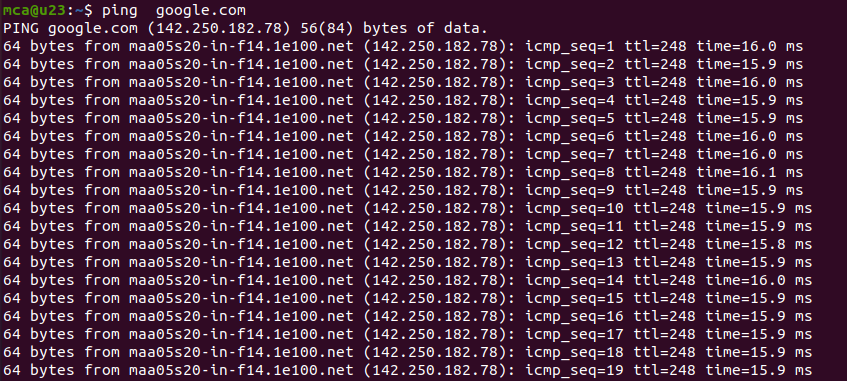
**Procedure**

1. ifconfig : fconfig command, it provides information about the network interfaces on your system, including their IP addresses, MAC addresses, network masks, and other relevant details.

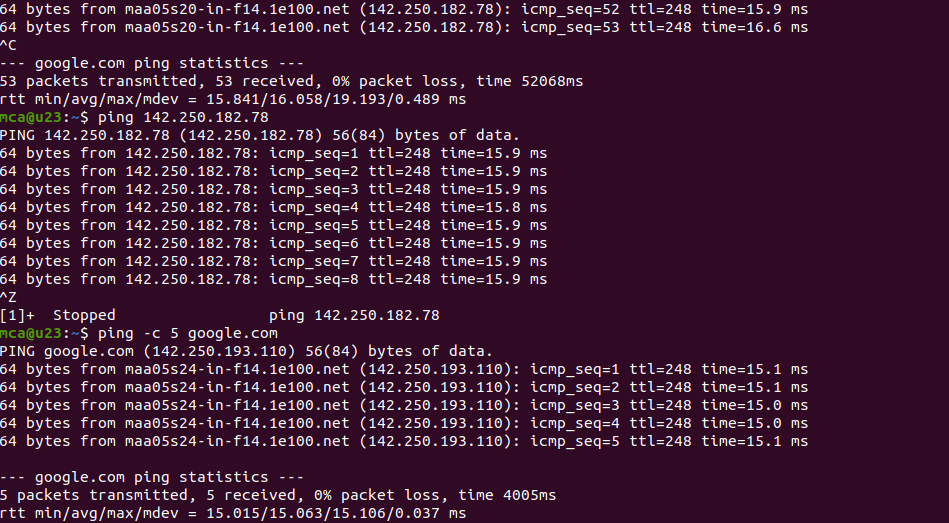
**Output Screenshot**



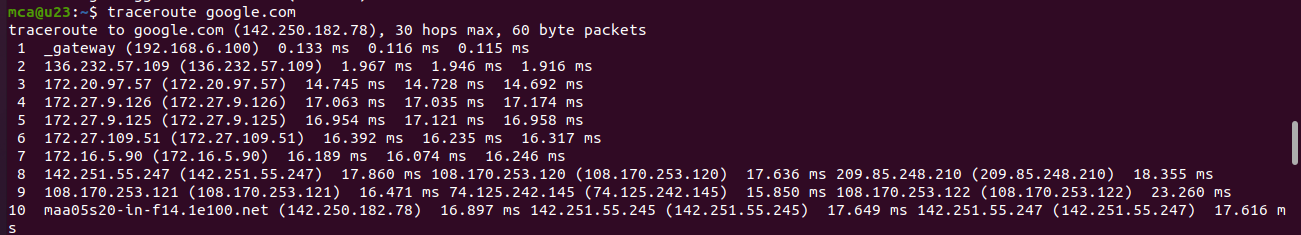
1. ping google.com : When you execute the ping command followed by a specific host or IP address, your computer sends ICMP (Internet Control Message Protocol) echo request packets to that destination. The ping utility is commonly used to troubleshoot network connectivity issues, determine network latency, and assess the overall health and responsiveness of a network.



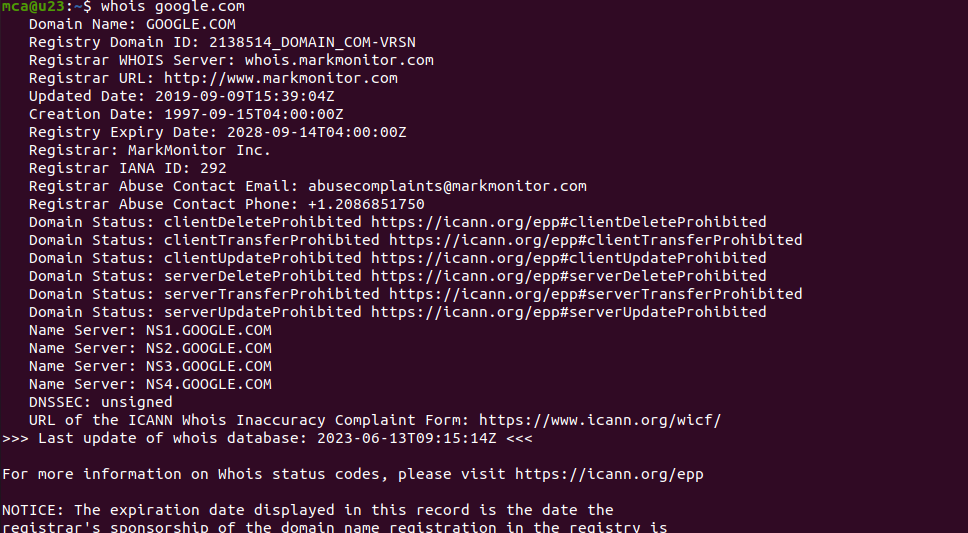
1. ping -c 5 google.com   When you execute ping -c 5, it sends a total of 5 ICMP echo request packets to the specified host or IP address. Once the 5 packets are sent, ping will stop automatically and provide you with a summary of the results, including statistics such as packet loss, round-trip time (RTT), and other relevant information



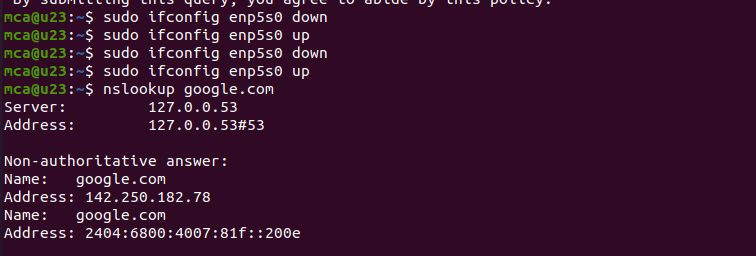
1. treceroute google.com  traceroute is a command-line network diagnostic tool used to trace the route that packets take from your computer to a destination host or IP address. It helps identify the path and measure the network latency between your computer and the destination.



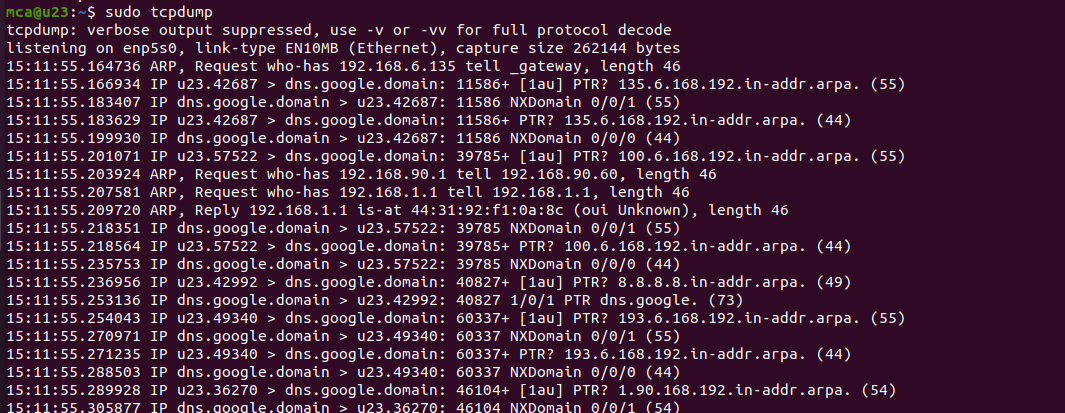
1. whois google.com   whois is a command-line utility and protocol used to retrieve information about registered domain names or IP addresses. It allows you to query a WHOIS database to obtain details about the ownership, registration, and administrative contacts associated with a particular domain name or IP address.



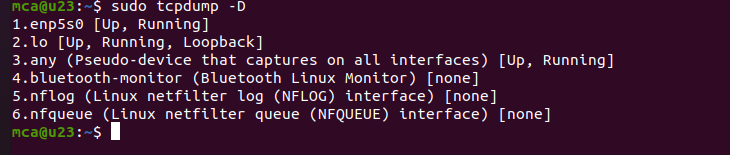
1. ifcongig enp5s0 down/up : ifconfig enp5s0 down: This command disables or "brings down" the network interface named "enp5s0".  ifconfig enp5s0 up: This command enables or "brings up" the network interface named "enp5s0".



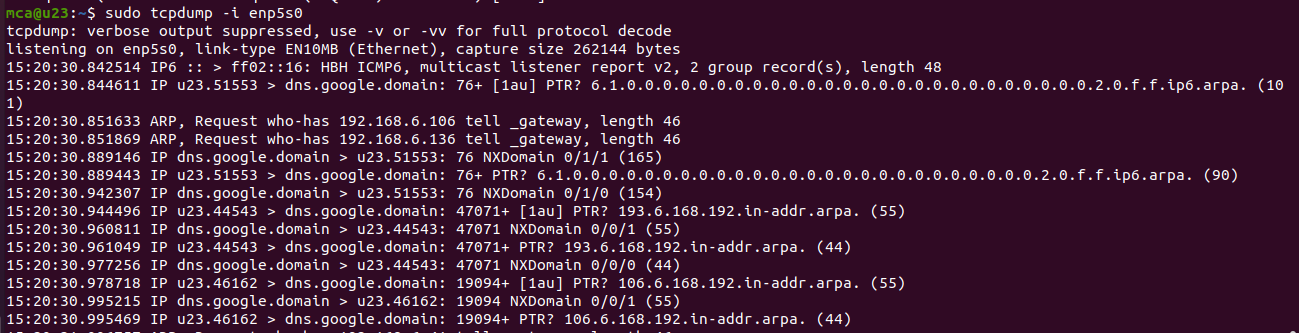
7. tcpdump :  When you run the tcpdump command, it starts capturing network packets on a specified network interface or on all interfaces if none is specified



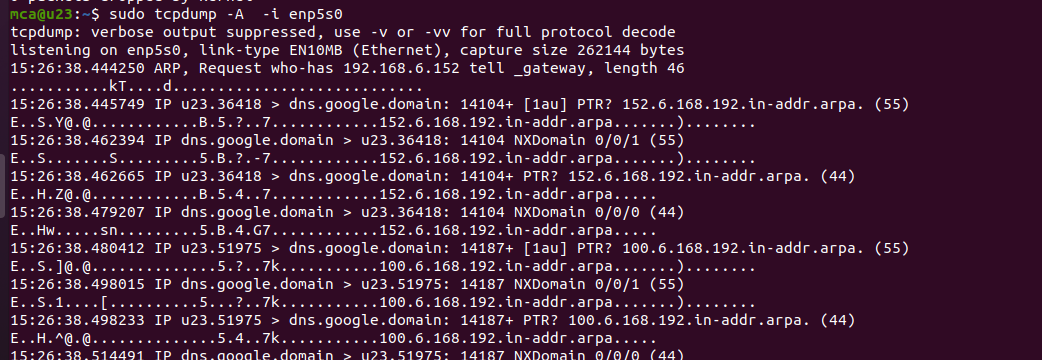
8. tcpdump -D  :  When you execute tcpdump -D, it displays a numbered list of the network interfaces detected by tcpdump on your system.



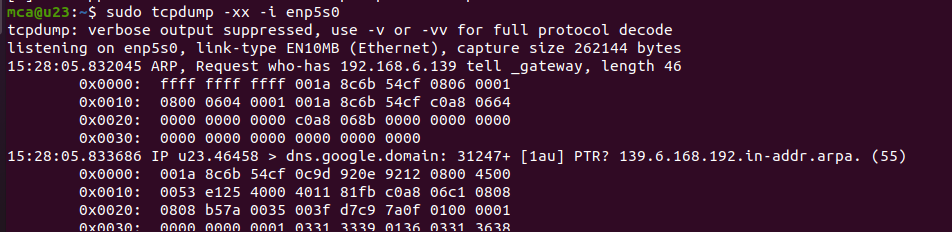
9. tcpdump -i enp5s0 :  The command tcpdump -i enp5s0 is used to capture network packets on a specific network interface, in this case, "enp5s0". When you execute this command, tcpdump will start capturing packets flowing through the specified interface.



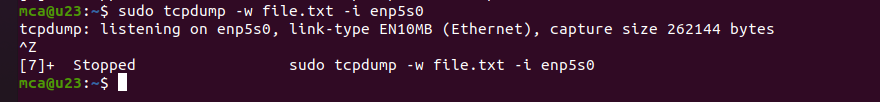
10. tcpdump -A -i enp5s0 :  The command tcpdump -A -i enp5s0 is similar to the previous command, but with an additional flag -A. This flag tells tcpdump to display the packet contents or payload in ASCII format.

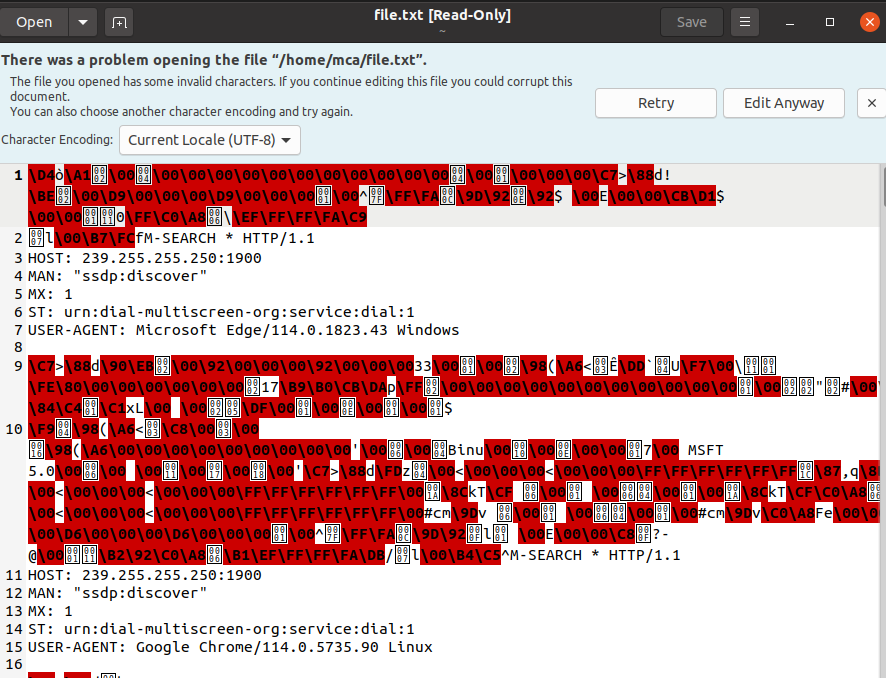


11.   tcpdump -xx -i enp5s0 :  The command tcpdump -xx -i enp5s0 is another variation of the tcpdump command with additional flags -xx. These flags modify the output format of tcpdump to display the packet contents in both hexadecimal and ASCII formats.

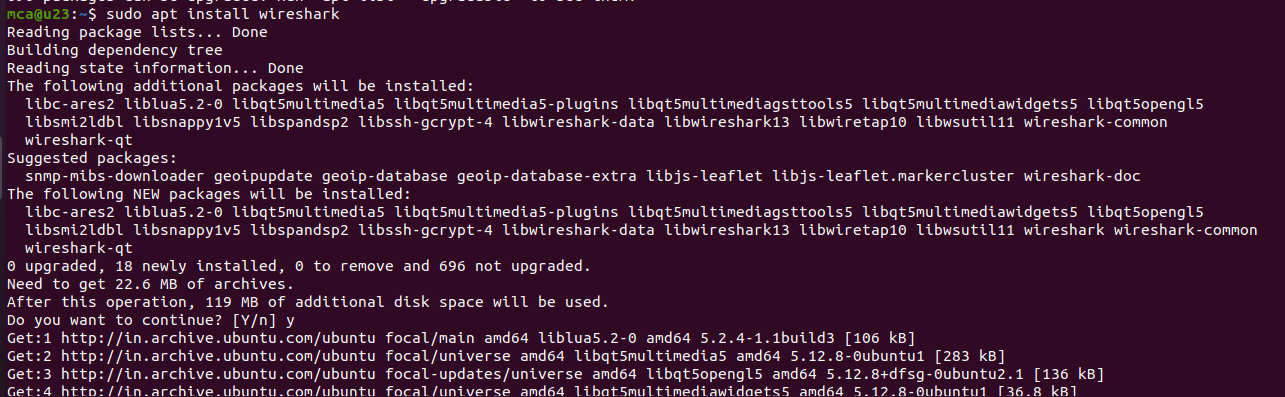


12. tcpdump -w file.txt -i enp5s0  : The command tcpdump -w file.txt -i enp5s0 is used to capture network packets on the specified network interface (enp5s0) and save the captured packets to a file named file.txt.

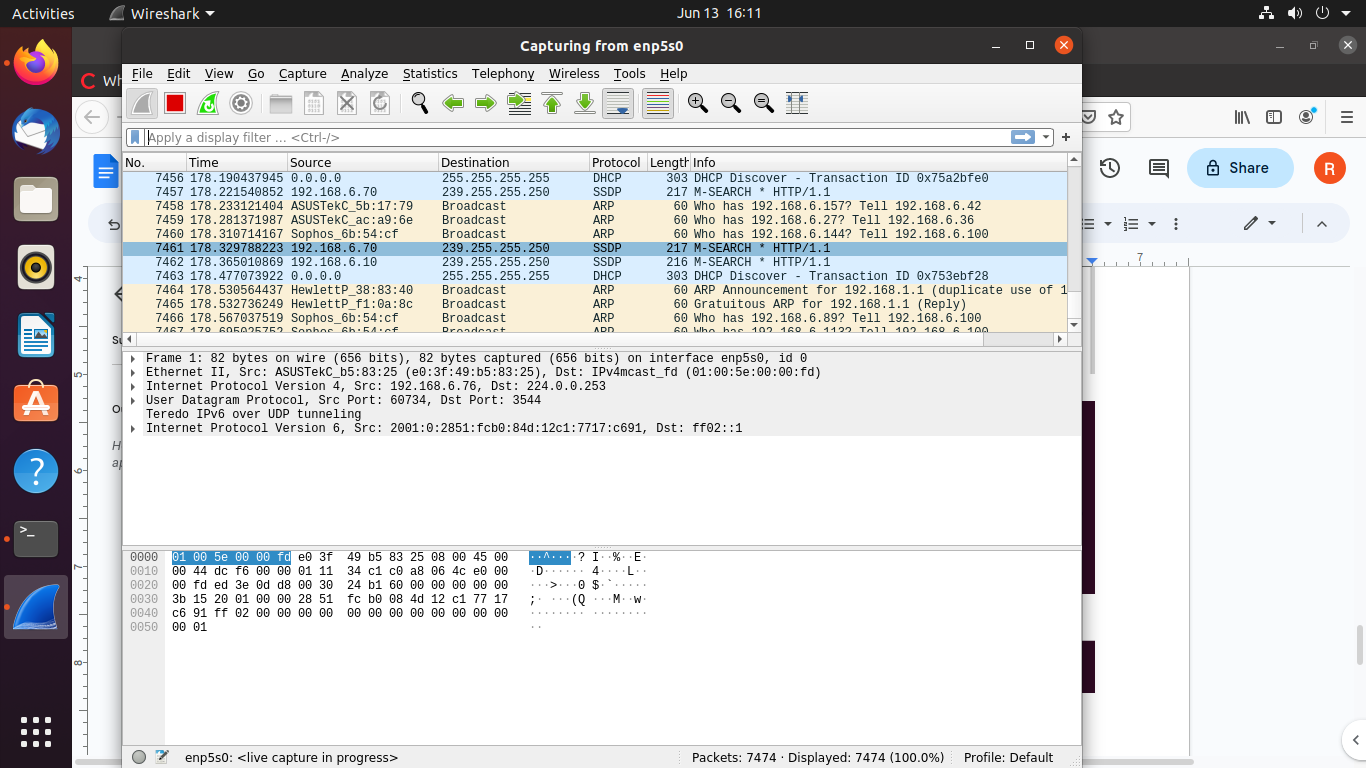




13.  wireshark :   Wireshark is a popular and powerful network protocol analyzer. It provides a graphical user interface (GUI) that allows you to capture, analyze, and inspect network packets in real-time. Wireshark is available for multiple platforms, including Windows, macOS, and Linux.







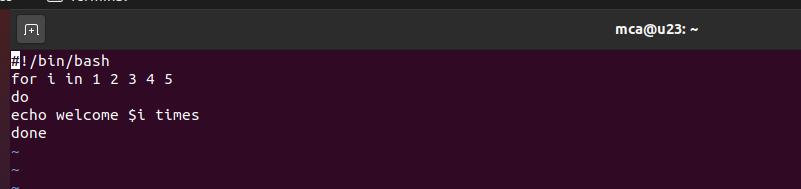
**RESULT**

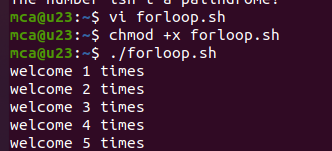
**Experiment No.: 25**

**Aim: For Loop .**

**CO**

**Procedure**

****

****

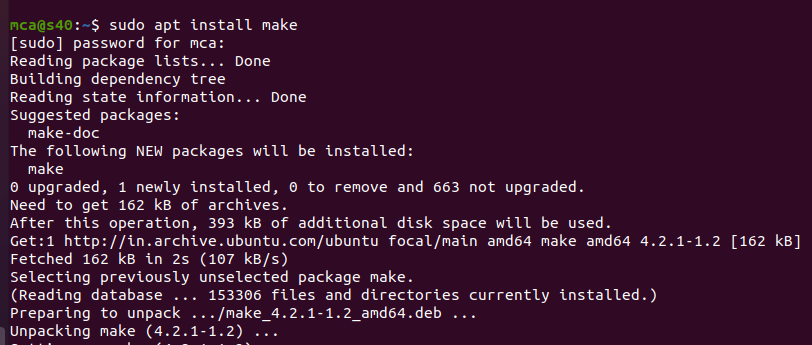
**RESULT**

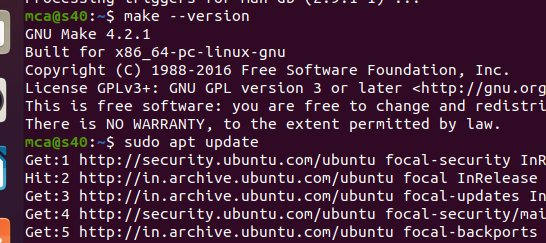
**Experiment No.: 26**

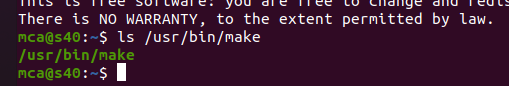
**Aim: make and cmake**

**CO**

**Procedure**

****

****

****

**main.cpp**

**#include<iostream>**

**#include functions.h**

**int main()**

**print\_hello();**

**std::cout << std::endl;**

**std::cout << the factorial of 5 is: <<factorial(5)<<std::endl;**

**return 0;**

**function1.cpp**

**#include functions.h**

**int factorial(int n){**

**if(n!=1){**

**return(n\*factorial(n-1));**

**}**

**else return 1;**

**}**

**function2.cpp**

**#include <iostream>**

**#include functions.h**

**void print\_hello()**

**{**

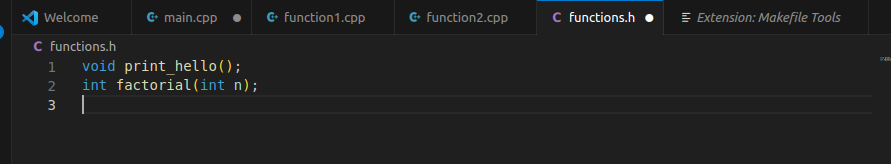
**std::cout << "Hello World" ;**

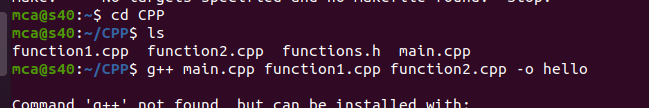
**}**

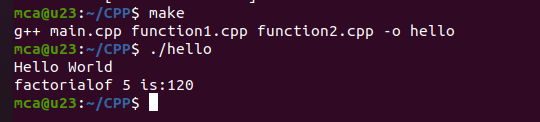
**functions.h**

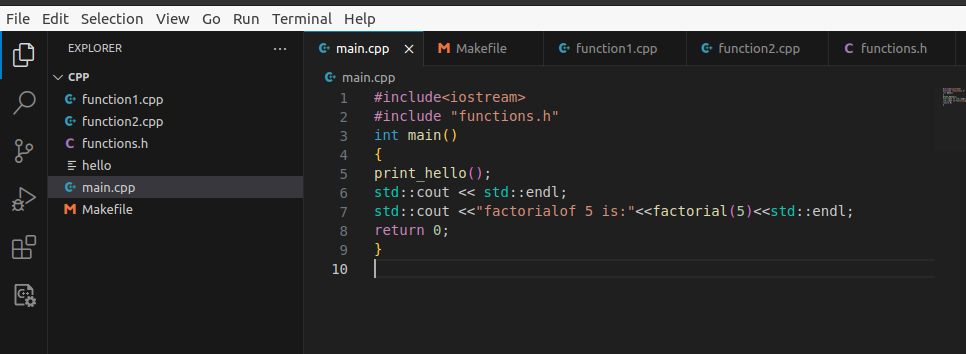
**void print\_hello();**

**int factorial(int n);**

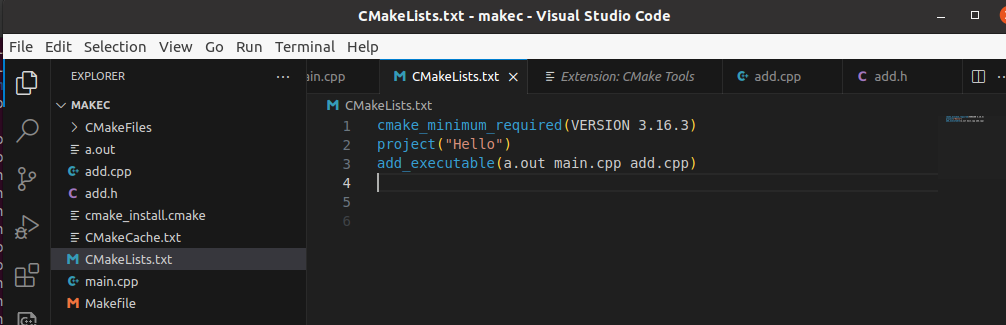
****

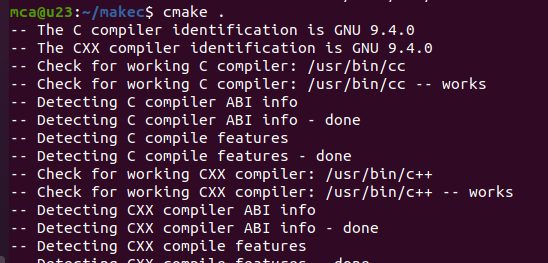
****

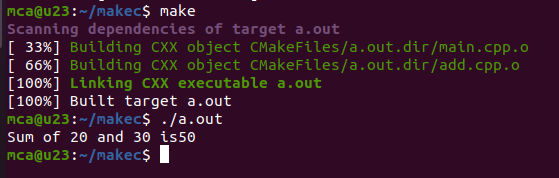
****

****

**cmake**

****

****

****

**add.cpp**

**#include "add.h"**

**int add(int a,int b)**

**{**

**return a + b;**

**}**

**main.cpp**

**#include<iostream>**

**#include "add.h"**

**int main()**

**{**

**std::cout << "Sum of 20 and 30 is" << add(20,30)<<"\n";**

**return 0;**

**}**

**add.h**

**#pragma once**

**int add(int a,int b);**

**CMakeLists.txt**

**cmake\_minimum\_required(VERSION 3.16.3)**

**project("Hello")**

**add\_executable(a.out main.cpp add.cpp)**

**RESULT**

**Experiment No.: 27**

**Aim: File System Hierarchy**

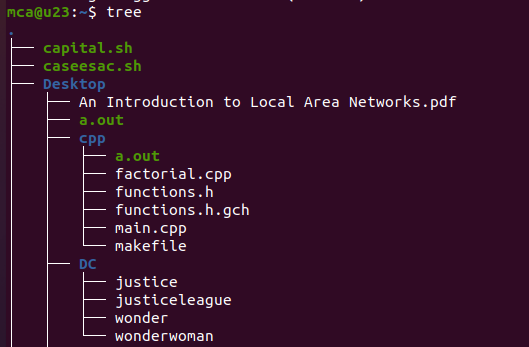
**CO**

**Procedure**

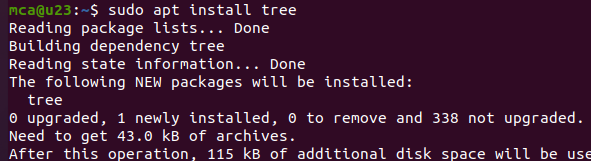
**sudo apt update**

**sudo apt install tree**

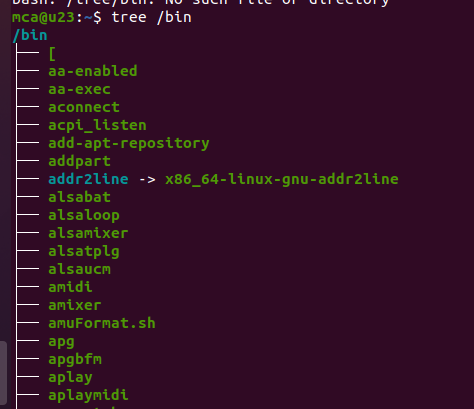
**tree**

****

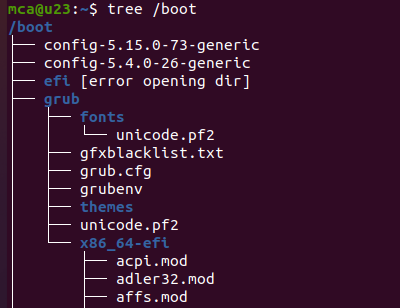
**#scrutanise each directory**

****

**tree /bin**

****

**tree /boot**

****

**Directories Explanation**

**/root ->** Contains all other directories and files

**/bin ->** Essential binary files which are accessible to all users.This files are prepared for basic functioning and for various system operations & user interactions.

**/boot** -> Files prepared for boot process.(including kernel,boot loader & initial ramdisk)

We want to load the os & prepare the system for use. All the files supporting the booting are stored here.

**/dev ->** Device files representing the physical & virtual devices. Such as hardwares,printers,cpu etc.

**/etc ->** The system config.files for various applications & services .Config files determines the behaviour,functionality and appearance of the system.

**/home ->** Directory for regular users.

**/mnt ->** Temporary mounted files .Temporarly attaching file system for eg: external devices,network shares .

**/media ->** Temporary mounted syste, for removable media

eg: usb,optical disk or ext hard drives.

**/sys ->** Expose information about system hardware and devices .

**/temp ->** Temp.files created by users and applications

**/usr ->** User programs and libraries

**/var ->** The variable data that changes frequently such as log & cache files.

**RESULT**