**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

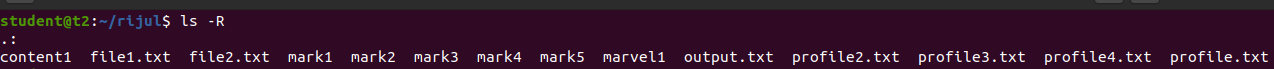
**Output Screenshot**

****

2.1 ls -R -show contents in the subdirectory.

$ls -R

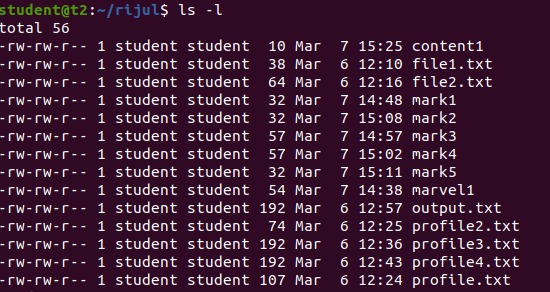
**Output Screenshot**

****

2.2 ls -l - to longlist a list.

$ls -l

**Output Screenshot**

****

2.3 ls-a – to list all hidden files.

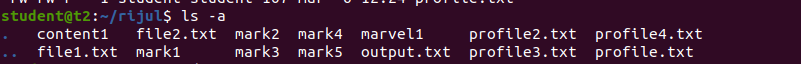
$ls -a

**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**

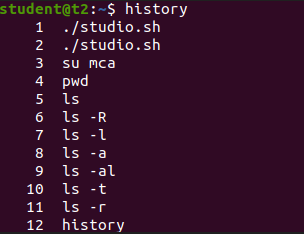
****

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

Time.

$history

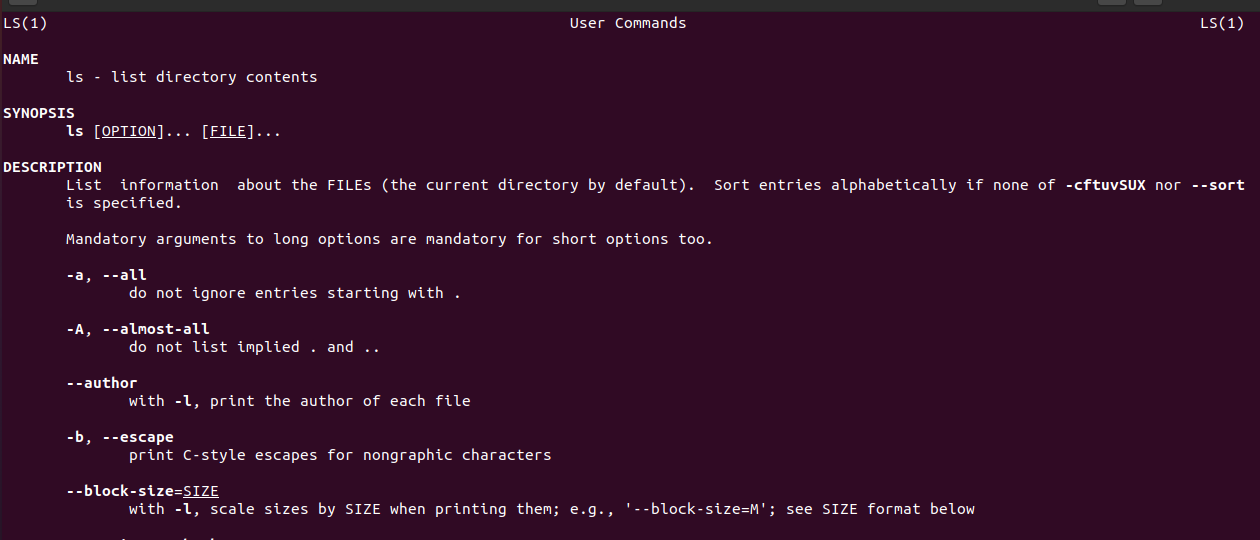
**Output Screenshot**

****

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

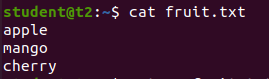
Mango

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

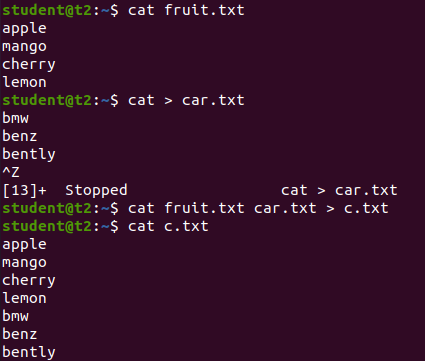
Lemon

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

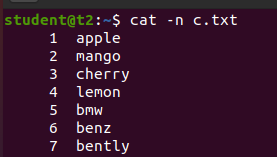
**Output Screenshot**



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

$cat -n c.txt

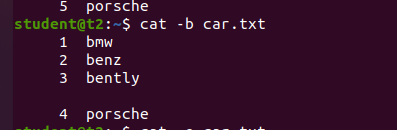
**Output Screenshot**



14.1 cat -b filename – remove empty line numbers

$cat -b car.txt

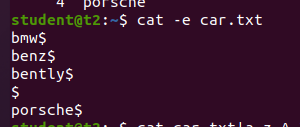
**Output Screenshot**



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

$cat -e car.txt

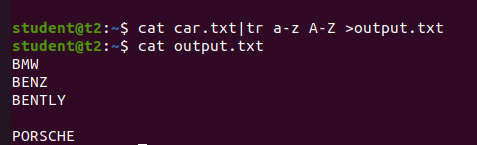
**Output Screenshot**



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

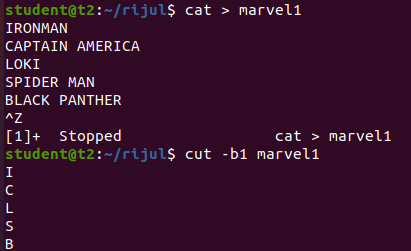
Loki

Spiderman

Blackpanther

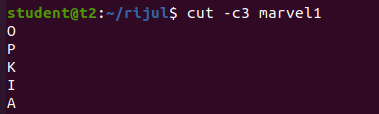
$cat -b1 marvel1

**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

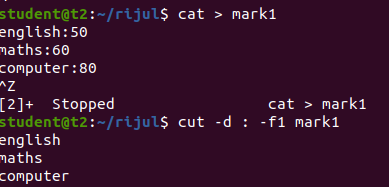
English-50

Maths-60

Computer-80

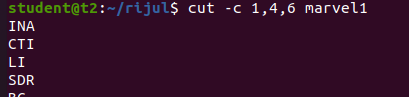
$cut -d - -f1 mark1

**Output Screenshot**



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

**Output Screenshot**

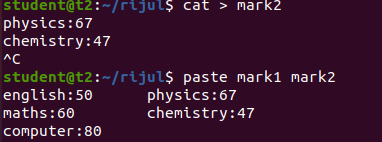


2. paste- join files horizontally[each file consisting of different lines]

2.1 paste filename1 filename2

$paste mark1 mark2

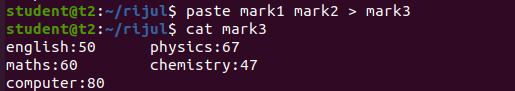
**Output Screenshot**



2.2 $paste mark1 mark2 > mark3

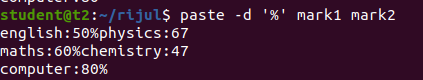
$cat mark3

**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

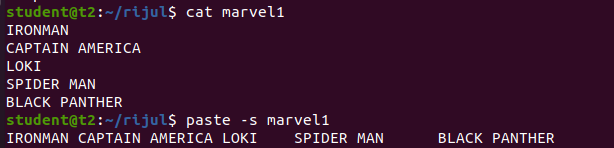
Maths-60%chemistry-47

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

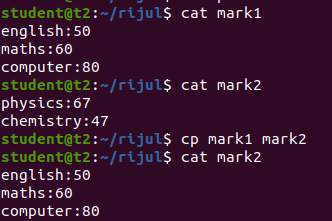
Phy-67

Chem-47

$cp mark1 mark2

$cat mark2

**Output Screenshot**

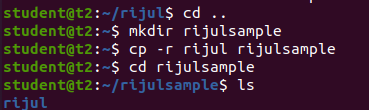


3.1 $cp -r rijul rijulsample – copy a directory along with subdirectory to another directory.

$cd rijulsample

Ls

**Output Screenshot**

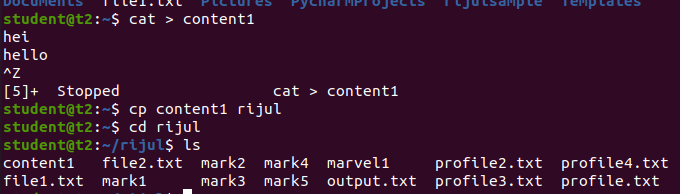


3.2 $cp content rijul – copy an content from one directory to another.

$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**

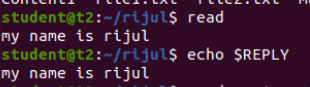
1. read- to read the content of a file

$ 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 \

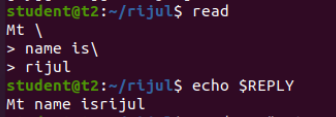
\-> to write on a different line.

>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**

****

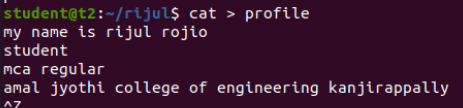
$ echo “password is $REPLY”

**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**

****

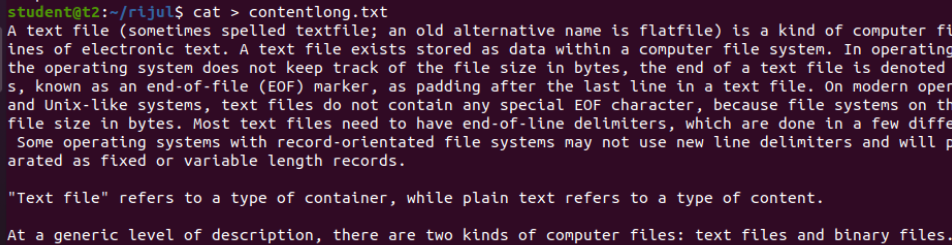
2.5 $wc -L profile : print length of the longest line.

**Output Screenshot**

****

$ cat contentlong.txt

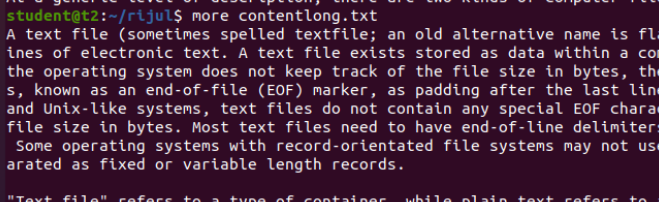
**Output Screenshot**

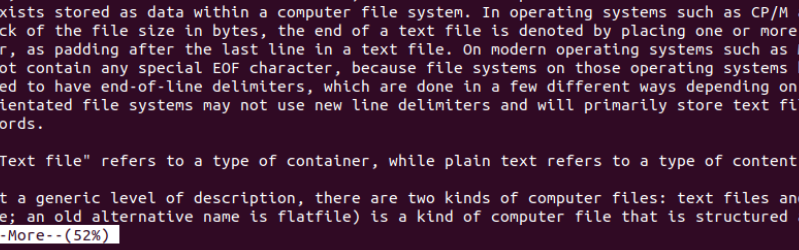
****

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

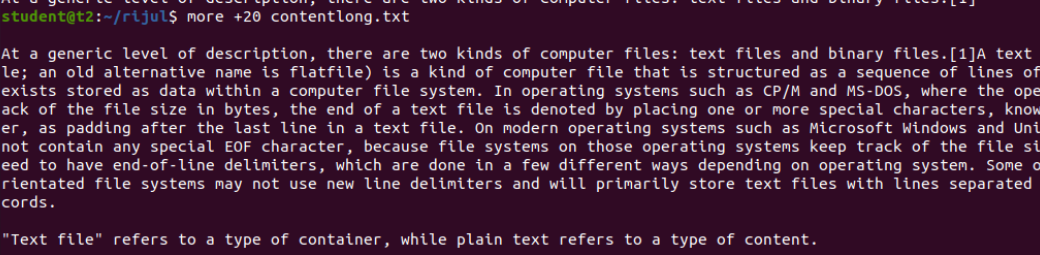
**Output Screenshot**

****



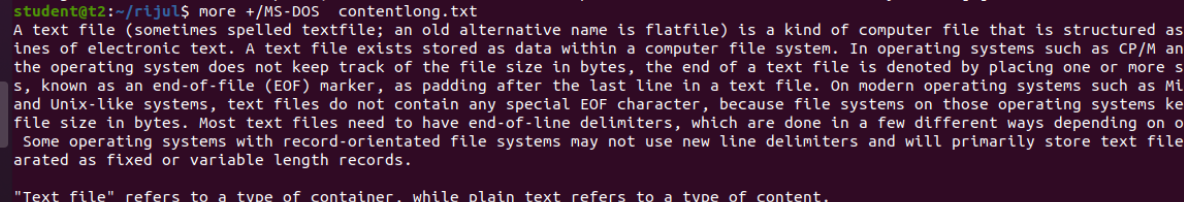
3.1 $ more +20 contentlong.txt -> display contents after 20 lines.

**Output Screenshot**

****

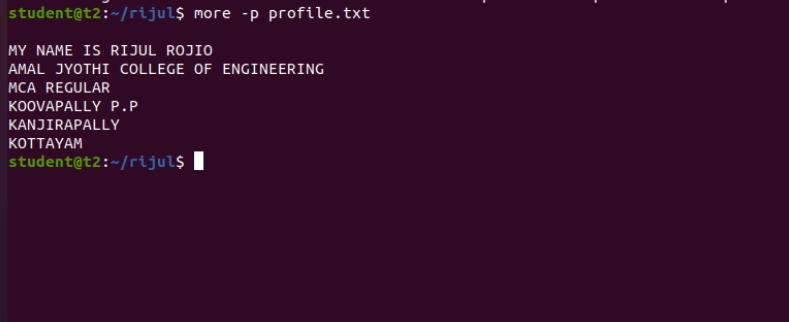
3.2 $ more +/ms-dos{pattern} contentlong.txt -> search the pattern string from the content and view all the instances,by navigating through the result.

**Output Screenshot**

****

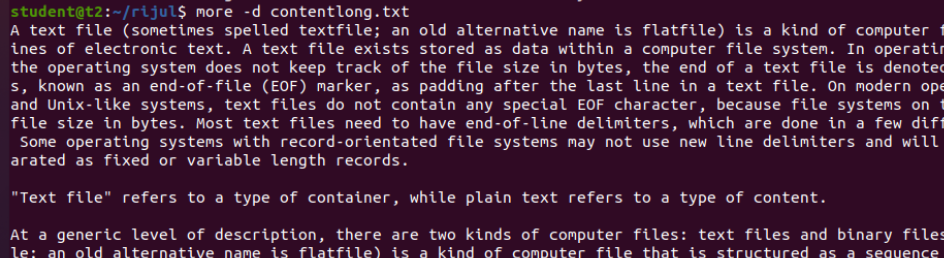
3.3 $more -p contentlong.txt -> clear the screen and show the output.

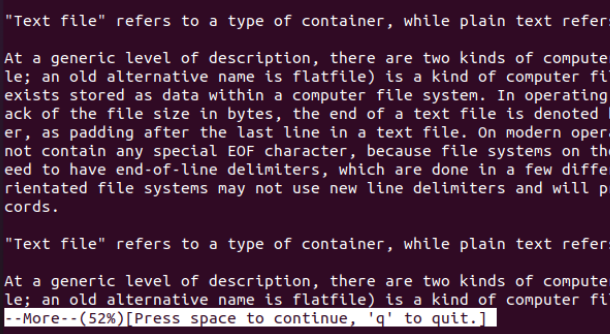
**Output Screenshot**

****

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**

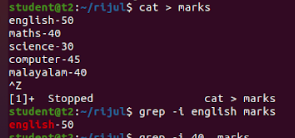
**Aim: Familarization of linux commands.**

**CO2: Perform System Administration Tasks.**

**Procedure**

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

**Output Screenshot**



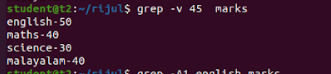
1.1  $grep -i content file\_name   -> case sensitive search

**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**

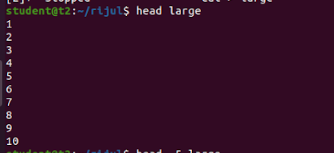




1. head - display the top content of the file by default.

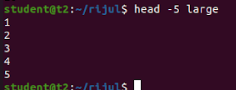
$ head file\_name

**Output Screenshot**



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

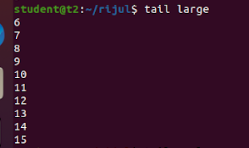
**Output Screenshot**

****

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

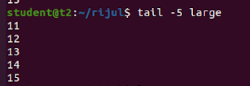
$tail file\_name

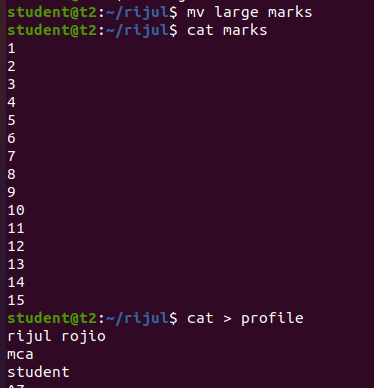
**Output Screenshot**



3.2 $tail -5 file\_name

**Output Screenshot**

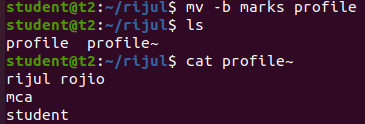


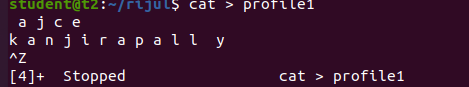


4.. mv - move

$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.