JGi JAIN DEEMED-TO-BE UNIVERSITY | FACULTY OF ENGINEERING AND TECHNOLOGY

PRINCIPLES OF OPERATING SYSTEMS LABORATORY (18CS4SP05L)

ACADEMIC SESSION: JAN 2022 – JULY 2022

Submitted to:
Mr.Anantha Babu

Student Name: Tarun S.K
USN: 20BTRCA030
Semester: 4TH SEM
Branch: CSE-AI

JGi JAIN DEEMED-TO-BE UNIVERSITY | FACULTY OF ENGINEERING AND TECHNOLOGY

LABORATORY CERTIFICATE

This is to certify that Mr./Ms. ……**Tarun S K**……bearing USN……**20BTRCA030**……..has satisfactorily completed the course of experiments in practical ……**Principle of Operating System**…….prescribed by JAIN UNIVERSITY ……Semester Course in the Laboratory of this college in the academic session Jan 2022-July 2022.

Date:

Name of the Student : Tarun S K
USN: 20BTRCA030

Date of Practical Examination: 23.06.2022

REMARKS

Signature of the Faculty
Incharge of the Batch

VALUED
Examiner 1…….
Examiner 2……

Head of the Department

CONTENTS

Experiment No.	Date	EXPERIMENT TITLE	Page No.	Remarks
1.		Execute 15 basic commands of UNIX.	1	
2.		Basics of functionality and modes of VI Editor	**7**	
3.		Write a program that accepts user name and reports if user is logged in.	13	
4.		Write a program which displays the following menu and executes the optionselected by user: 1. ls 2. pwd 3. ls –l 4. ps –fe	15	
5.		Write a program to print 10 9 8 7 6 5 4 3 2 1 .	17	
6.		Write a program that replaces all "*.txt" file names with "*.txt.old" in thecurrent.	19	
7.		Write a program that echoes itself to stdout, but backwards.	21	
8.		Write a program that takes a filename as input and checks if it is executable,if not make it executable.	22	
9.		Write a program to take string as command line argument and reverse it.	24	
10.		Create a data file called employee in the format given below: a. EmpCode Character b. EmpName Character c. Grade Character d. Years of experience Numeric e. Basic Pay Numeric $vi employee A001 ARJUN E1 01 12000.00 A006 Anand E1 01 12450.00 A010 Rajesh E2 03 14500.00 A002 Mohan E2 02 13000.00 A005 John E2 01 14500.00 A009 Denial Smith E2 04 17500.00 A004 Williams E1 01 12000.00 Perform the following functions on the file:	26	

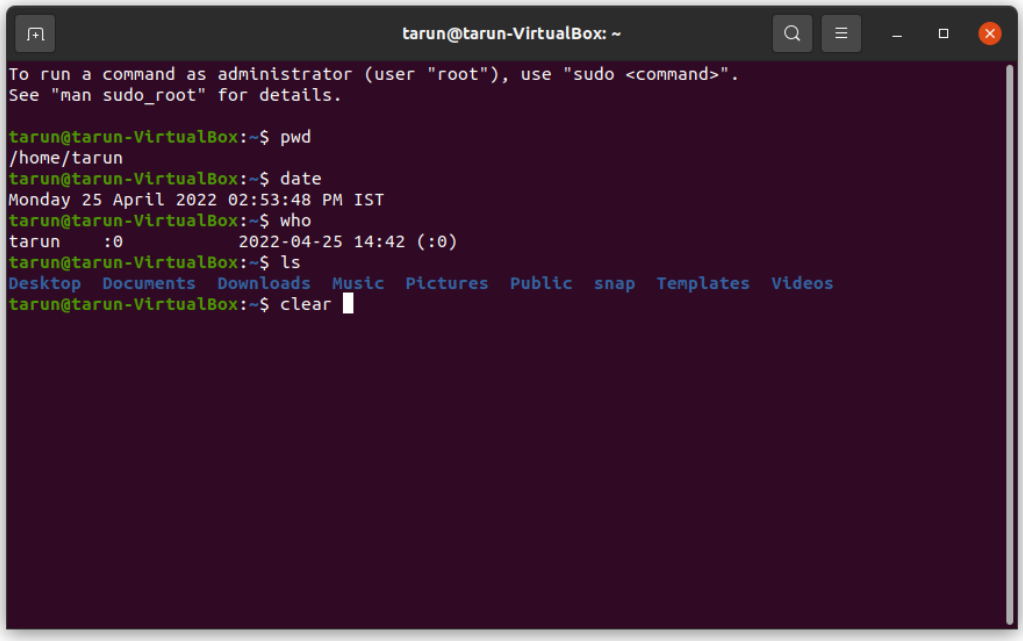
Experiment No: 1

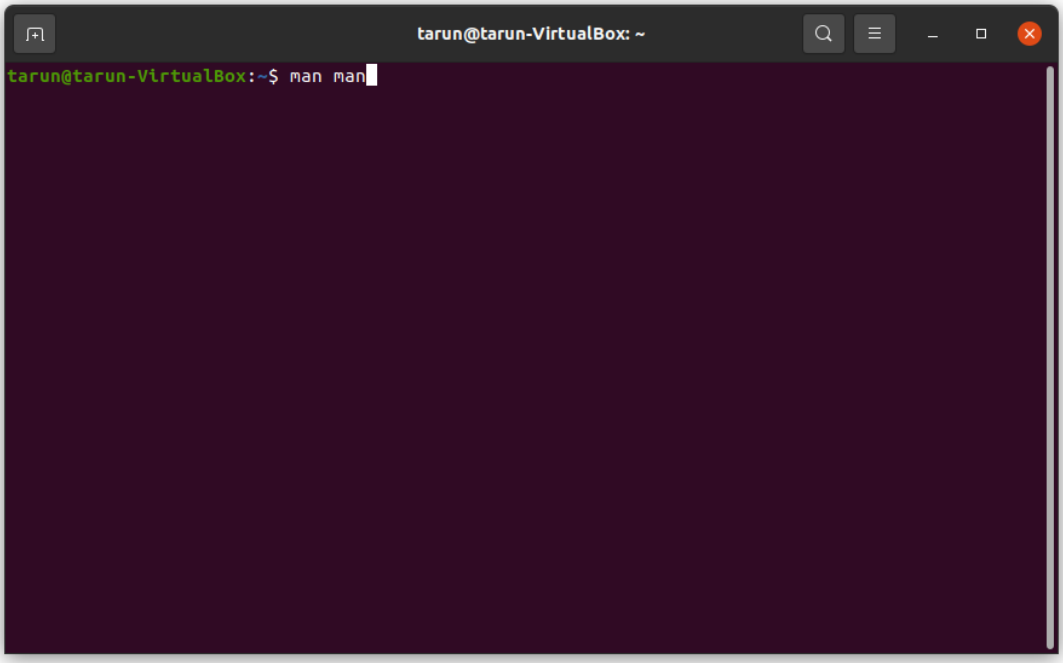
1.Execute 15 basic commands of UNIX.

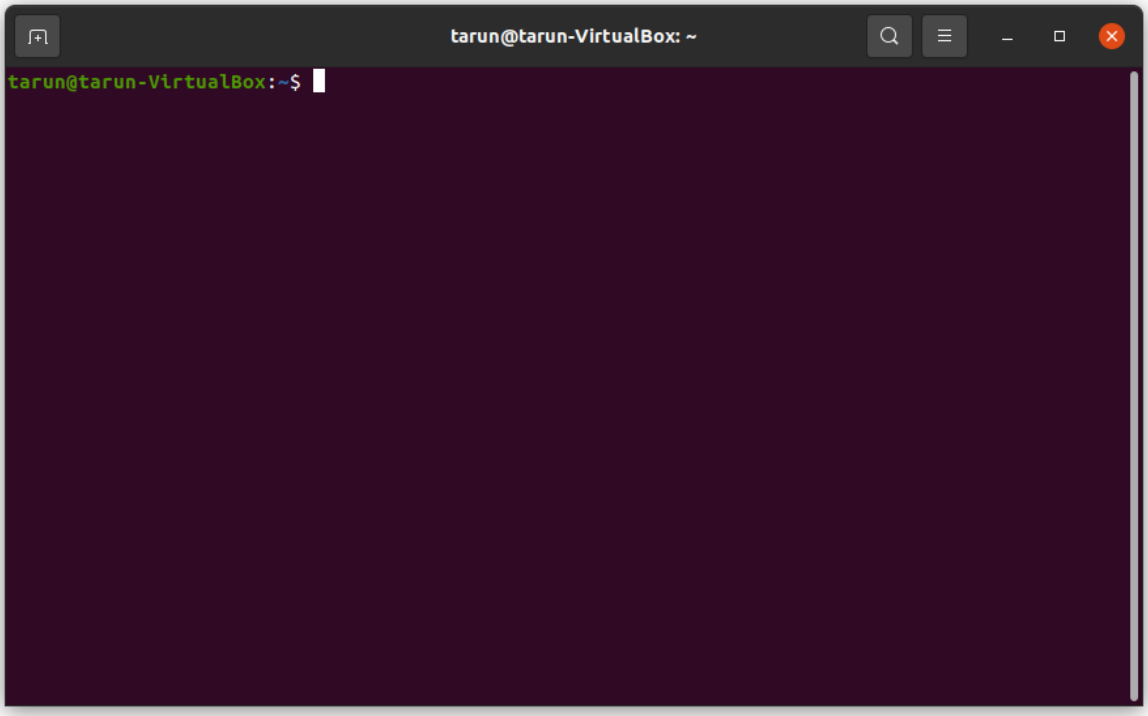
S.NO	COMMAND	EXAMPLE	DESCRIPTION
1.	pwd	pwd	It will print the current working directory
2.	date	date	it will print the current date with time
3.	who	who	print the current user
4.	ls	ls -alf	list the available files in the directory
5.	man	man ls	usually called man pages, online to explain the usage of the unix system and commands
6.	clear	clear	used to clear the terminal screen
7.	chmod	chmod	set the file permission
8.	mkdir	mkdir hello	used to create a folder
9.	cd	cd ..	is used to change the directory
10.	touch	touch hello.txt	used to create a text file in the current folder.
11.	printf	printf "hello world"	used to write a text in a file
12.	cat	cat hello.txt	used to display the item in the file
13.	cp	cp hello.txt example.txt	used to copy a file
14.	mv	mv example.txt ~/Downloads/example.txt	used to move a file

15.	rm	rm example.txt	used to remove a file

Screenshot of Output:




tarun@tarun-VirtualBox: ~
tarun@tarun-VirtualBox:~$ man man

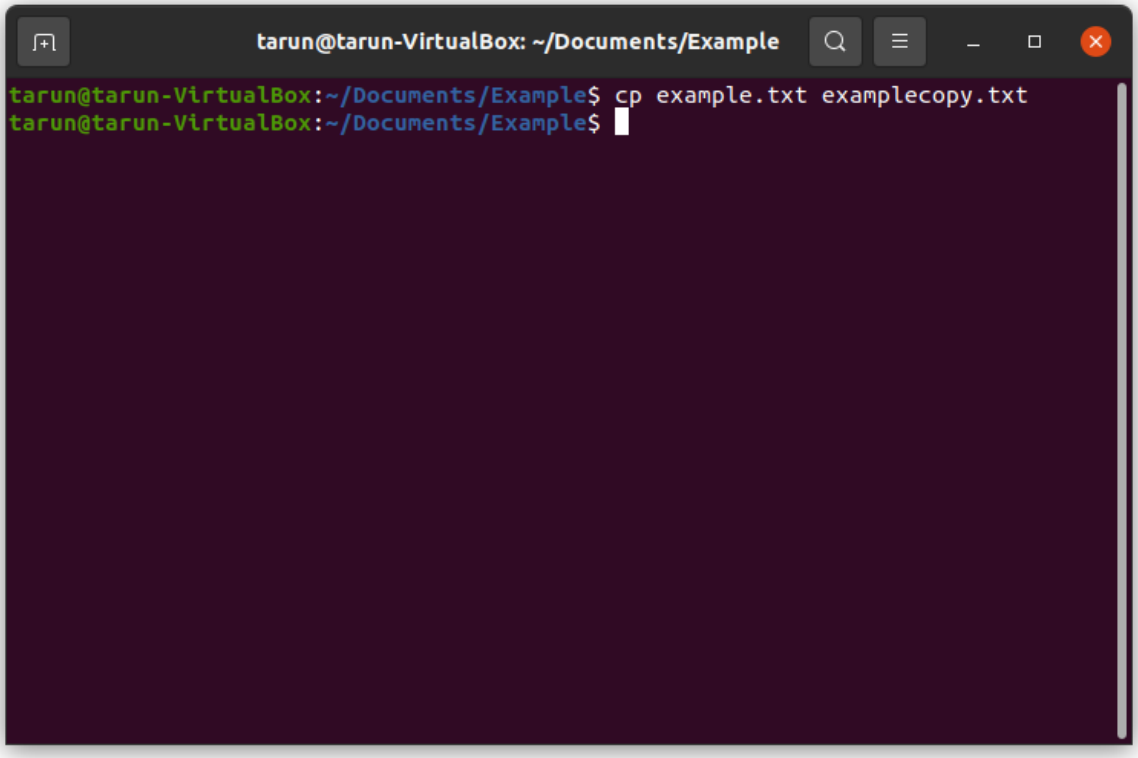

tarun@tarun-VirtualBox: ~
tarun@tarun-VirtualBox:~$

```
tarun@tarun-VirtualBox: ~/Documents/Example
tarun@tarun-VirtualBox:~$ pwd
/home/tarun
tarun@tarun-VirtualBox:~$ mkdir Documents/Example
tarun@tarun-VirtualBox:~$ cd Documents/Example
tarun@tarun-VirtualBox:~/Documents/Example$ touch example.txt
tarun@tarun-VirtualBox:~/Documents/Example$ printf "20BTRCA030-TARUN S.K"
```

```
tarun@tarun-VirtualBox: ~
MAN(1)                          Manual pager utils                          MAN(1)

NAME
       man - an interface to the system reference manuals

SYNOPSIS
       man [man options] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [man options] [section] term ...
       man -f [whatis options] page ...
       man -l [man options] file ...
       man -w|-W [man options] page ...

DESCRIPTION
       man  is  the  system's manual pager.  Each page argument given to man is normally the
       name of a program, utility or function.  The manual  page  associated  with  each  of
       these arguments is then found and displayed.  A section, if provided, will direct man
       to look only in that section of the manual.  The default action is to search  in  all
       of  the  available sections following a pre-defined order (see DEFAULTS), and to show
       only the first page found, even if page exists in several sections.

       The table below shows the section numbers of the manual  followed  by  the  types  of
       pages they contain.

       1   Executable programs or shell commands
       2   System calls (functions provided by the kernel)
 Manual page man(1) line 1 (press h for help or q to quit)
```

```
tarun@tarun-VirtualBox: ~/Documents/Example
tarun@tarun-VirtualBox:~$ pwd
/home/tarun
tarun@tarun-VirtualBox:~$ mkdir Documents/Example
tarun@tarun-VirtualBox:~$ cd Documents/Example
tarun@tarun-VirtualBox:~/Documents/Example$ touch example.txt
tarun@tarun-VirtualBox:~/Documents/Example$ printf "20BTRCA030-TARUN S.K"
20BTRCA030-TARUN S.Ktarun@tarun-VirtualBox:~/Documents/Example$
tarun@tarun-VirtualBox:~/Documents/Example$ cat example.txt
tarun@tarun-VirtualBox:~/Documents/Example$ printf "20BTRCA030-TARUN S.K" > ~/Do
cuments/Example/example.txt
tarun@tarun-VirtualBox:~/Documents/Example$ cat example.txt
20BTRCA030-TARUN S.Ktarun@tarun-VirtualBox:~/Documents/Example$
```

```
tarun@tarun-VirtualBox: ~/Documents/Example
tarun@tarun-VirtualBox:~/Documents/Example$ cp example.txt examplecopy.txt
tarun@tarun-VirtualBox:~/Documents/Example$ mv examplecopy.txt ~/Downloads/examp
le.txt
tarun@tarun-VirtualBox:~/Documents/Example$ rm example.txt
tarun@tarun-VirtualBox:~/Documents/Example$
```


tarun@tarun-VirtualBox: ~/Documents/Example
tarun@tarun-VirtualBox:~/Documents/Example$ cp example.txt examplecopy.txt
tarun@tarun-VirtualBox:~/Documents/Example$

Experiment – 2

2. Basic functionality and modes of VI editor.

One of the most important aspects to remember about vi is that most of the commands fall into one of three modes

1. vi mode: in this mode, most keys on the keyboard are defined to be a specific command. As the key or key sequence is issued, that command is executed. This is the mode vi starts in. At any time, pressing the key returns the user to vi mode.

2. command mode: to reach that mode, one must first be in vi mode, then issue a colon (“:”). That same colon will appear at the bottom left corner of the screen. Then the command may be issued following the colon. One exception to this rule is the search command; a forward slash is issued instead of the colon.

3. input mode: this is where most users expect an editor to start. This “mode” actually refers to commands issued from vi mode but that allows the user to start inputting data into the file.

Invoking vi:

Type in the command prompt: vi filename

which will put filename into a buffer, and display the file on the screen. If the file is larger than the screen can display, the screen will act as a window into the file. At the beginning of a session, the screen will display the first part of the file. If filename does not exist, vi will create it. Upon entry to vi, the bottom of the screen will print the name of the file being edited, the number of lines in the file, and the size of the file (in characters).

Write two paragraphs (whatever you want) in the file (for testing the various commands).

Exiting vi:

Usually, the new or modified file is saved when you leave vi. However, it is also possible to quit vi without saving the file.

Note: The cursor moves to the bottom of the screen whenever a colon (:) is typed. This type of command is completed by hitting the <Return> (or <Enter>) key.

:x <Enter> quit vi, writing out the modified file to file named in the original invocation

:wq<Enter> quit vi, writing out the modified file to file named in the original invocation

:q! <Enter> quit vi even though latest changes have not been saved for this vi call

Moving the Cursor:

Unlike many of the PC and Macintosh editors, the mouse does not move the cursor within the vi editor screen (older versions). You must use the key commands listed below. On some UNIX platforms, the arrow keys may be used as well; however, since vi was designed with the Qwerty keyboard (containing no arrow keys) in mind, the arrow keys sometimes produce strange effects in viand should be avoided.

If you go back and forth between a PC environment and a UNIX environment, you may find that this dissimilarity in methods for cursor movement is the most frustrating difference between the two.

In the table below, the symbol ^ before a letter means that the <Ctrl> key should be held down while the letter key is pressed.

Note: Since following are the commands they will not work in the INSERT mode. Just open the file by writing

vi filename

on the command prompt and execute the commands without pressing 'I'. Before that, make sure that something is written in the file (refer invoking vi).

j or <Enter>

[or down-arrow]

move cursor down one line

k [or up-arrow] move cursor up one line

h or <Backspace> [or left-arrow]

l or <Space>
[or right-arrow]

move cursor left one character

move cursor right one character

0 (zero) move cursor to start of the current line (the one with the cursor)

$ move cursor to the end of the current line

w move cursor to the beginning of next word

b move the cursor back to the beginning of preceding word

:0<Enter> or 1G move the cursor to the first line in the file

:$<Enter> or G move the cursor to the last line in the file

the commands they will

not work in the INSERT

mode. Just open the file by

writing

Screen Manipulation:

The following commands allow the vi editor screen (or window) to move up or down several lines and to be refreshed.

Note: Since following are the commands they will not work in the INSERT mode. Just open the file by writing

vi filename

on the command prompt and execute the commands without pressing 'I'. Before that make sure that something is written in the file (refer invoking vi).

^f move forward one screen

^b move backward one screen

Adding and Deleting Text

Unlike PC editors, you cannot replace or delete text by highlighting it with the mouse. Instead, use the commands in the following tables.

Perhaps the most important command is the one that allows you to back up and undo your last action. Unfortunately, this command acts like a toggle, undoing and redoing your most recent action. You cannot go back more than one step.

u undo whatever you just did; a simple toggle

The main purpose of an editor is to create, add, or modify text for a file.

Inserting or Adding Text:

The following commands allow you to insert and add text. Each of these commands puts the vi editor into insert mode; thus, the <Esc> key must be pressed to terminate the entry of text and to put the vi editor back into command mode.

Note 1: Since following are the commands they will not work in the INSERT mode. Just open the file by writing vi filename

on the command prompt and execute the commands without pressing 'i'. Before that make sure that something is written in the file (refer invoking vi).

Note 2: Each of these commands puts the vi editor into insert mode; thus, the <Esc> key must be pressed to terminate the entry of text and to put the vi editor back into command mode.

i insert text before the cursor, until <Esc> hit

I insert text at beginning of current line, until <Esc> hit

a append text after the cursor, until <Esc> hit

A append text to the end of current line, until <Esc> hit

Deleting Text:

The following commands allow you to delete text.

X delete single character under the cursor

Nx delete N characters, starting with a character under the cursor

Dw delete the single word beginning with a character under the cursor

dNw delete N words beginning with a character under cursor; e.g., d5w deletes 5 words

D delete the remainder of the line, starting with the current cursor position

Dd delete entire current line

Ndd or dNd delete N lines, beginning with the current line; e.g., 5dd deletes 5 lines

Cutting and Pasting Text:

The following commands allow you to copy and paste text.

Yy copy (yank, cut) the current line into the buffer

Nyy or yNy copy (yank, cut) the next N lines, including the current line, into the buffer

P put (paste) the line(s) in the buffer into the text after the current line

Searching Text:

A common occurrence in text editing is to replace one word or phrase by another. To locate instances of particular sets of characters (or strings), use the following commands.

/string search forward for the occurrence of a string in the text

?string search backward for the occurrence of a string in the text

n move to next occurrence of the search string

N move to next occurrence of the search string in opposite direction

The rest of the experiments in the list involve shell programming.

Follow the following steps in each case to execute the programs:

1. To write the programs, create new files for each program by writing in the command prompt: vi filename
2. Write the program as plain text (in Insert mode)
3. Save the file and exit
4. Run the file by giving following command in the command prompt: sh filename

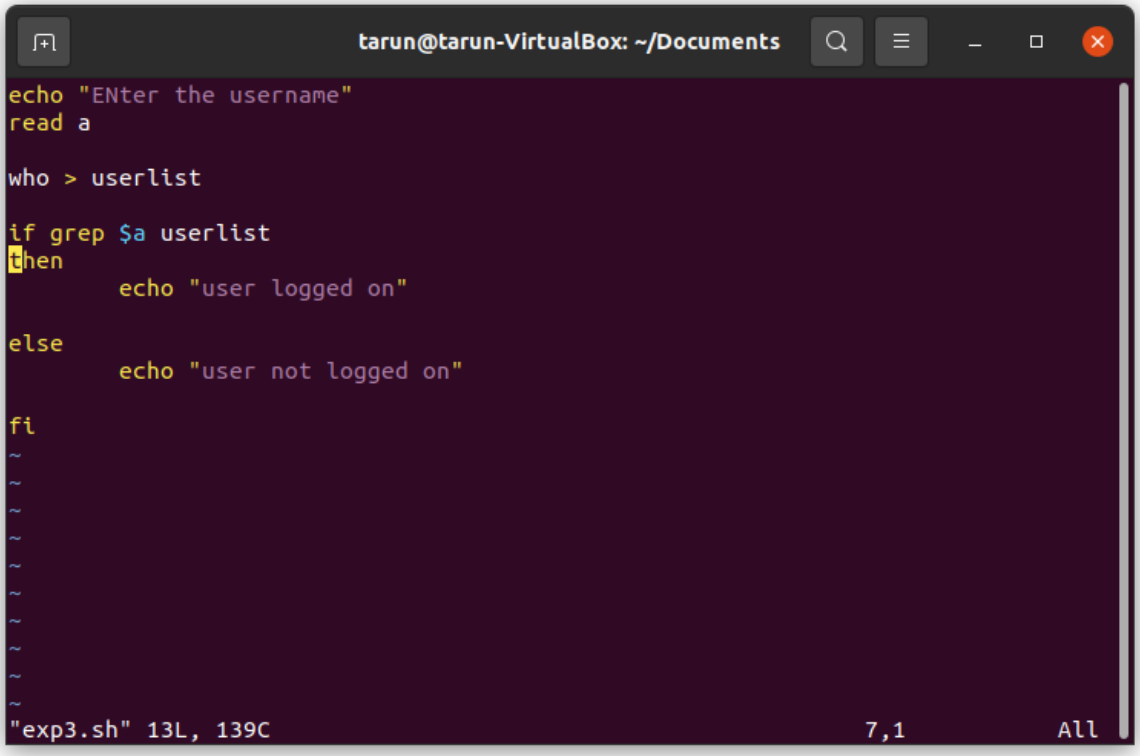
Experiment – 3

3.Write a program that accepts username and reports if the user is logged in.

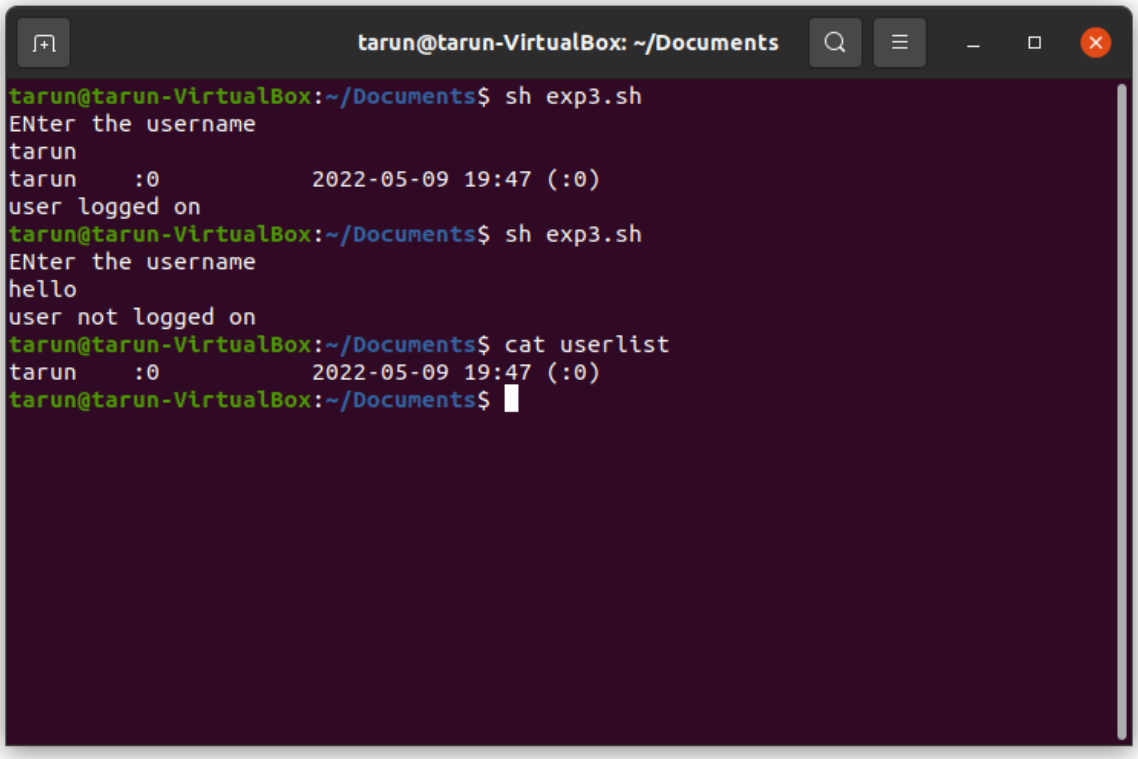
CODES:

```
echo “Enter the username” read a
who >userlist

if grep $a userlist then
echo “user logged on” else
echo “user not logged on”
```

SCREENSHOT:



Output:


tarun@tarun-VirtualBox: ~/Documents
tarun@tarun-VirtualBox:~/Documents$ sh exp3.sh
ENter the username
tarun
tarun :0 2022-05-09 19:47 (:0)
user logged on
tarun@tarun-VirtualBox:~/Documents$ sh exp3.sh
ENter the username
hello
user not logged on
tarun@tarun-VirtualBox:~/Documents$ cat userlist
tarun :0 2022-05-09 19:47 (:0)
tarun@tarun-VirtualBox:~/Documents$

Experiment – 4

4. Write a program which displays the following menu and executes the option selected by the user:

i) ls
ii) pwd
iii) who
iv) ls -l
v) ps -fe

CODE:

```
echo "Enter your choice:"

echo "1 for listing directory content"

echo "2 for print name of the current directory"
echo "3 for show who is logged on"
echo "4 for show directory content listing format"
echo "5 for listing current processes"

read ch
case $ch in
1) ls;; # lists directory content
2) pwd;; # prints name of the current directory
3) who;; # shows who is logged on
4) ls -l;; # shows directory content listing format
5) ps – fe;; # The -e option generates a list of information about every process currently
running. The -f option generates a listing that contains fewer items of information for each
process than the -l option.

*) echo "Invalid choice. Try again."
esac
```

Screenshot:

```
echo "enter your choice"
echo "1 for listing directory content"
echo "2 for print name of the current directory"
echo "3 for show who is logged on"
echo "4 for show directory content listing formart"
echo "5 for listing current processes"
read ch
case $ch in

        1)ls;;
        2)pwd;;
        3)who;;
        4)ls -l;;
        5)ps -fe;;
        *)
echo "invalid choise.try again"

esac
~
~
~
~
~
"~/Documents/Example/exp4.sh" 18L, 359C                    9,7           All
```

Output:

```
tarun@tarun-VirtualBox: ~/Documents/Example
tarun@tarun-VirtualBox:~/Documents/Example$ sh exp4.sh
enter your choice
1 for listing directory content
2 for print name of the current directory
3 for show who is logged on
4 for show directory content listing formart
5 for listing current processes
1
exp4.sh  hello.sh  userlist
tarun@tarun-VirtualBox:~/Documents/Example$ sh exp4.sh
enter your choice
1 for listing directory content
2 for print name of the current directory
3 for show who is logged on
4 for show directory content listing formart
5 for listing current processes
2
/home/tarun/Documents/Example
tarun@tarun-VirtualBox:~/Documents/Example$ sh exp4.sh
enter your choice
1 for listing directory content
2 for print name of the current directory
3 for show who is logged on
4 for show directory content listing formart
5 for listing current processes
3
tarun    :0           2022-04-25 16:06 (:0)
tarun@tarun-VirtualBox:~/Documents/Example$ sh exp4.sh
enter your choice
1 for listing directory content
2 for print name of the current directory
3 for show who is logged on
4 for show directory content listing formart
5 for listing current processes
4
total 12
-rw-rw-r-- 1 tarun tarun 359 Apr 25 17:27 exp4.sh
-rw-rw-r-- 1 tarun tarun 130 Apr 25 17:10 hello.sh
-rw-rw-r-- 1 tarun tarun  44 Apr 25 17:11 userlist
tarun@tarun-VirtualBox:~/Documents/Example$ sh exp4.sh
enter your choice
1 for listing directory content
2 for print name of the current directory
3 for show who is logged on
4 for show directory content listing formart
5 for listing current processes
5
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 16:06 ?        00:00:01 /sbin/init splash
root           2       0  0 16:06 ?        00:00:00 [kthreadd]
root           3       2  0 16:06 ?        00:00:00 [rcu_gp]
root           4       2  0 16:06 ?        00:00:00 [rcu_par_gp]
root           6       2  0 16:06 ?        00:00:00 [kworker/0:0H-events_highpri
root           7       2  0 16:06 ?        00:00:00 [kworker/0:1-events]
root           9       2  0 16:06 ?        00:00:00 [mm_percpu_wq]
root          10       2  0 16:06 ?        00:00:00 [rcu_tasks_rude_]
```

Experiment -5:

5. Write a program to print 10 9 8 7 6 5 4 3 2 1

using for loop

using while loop

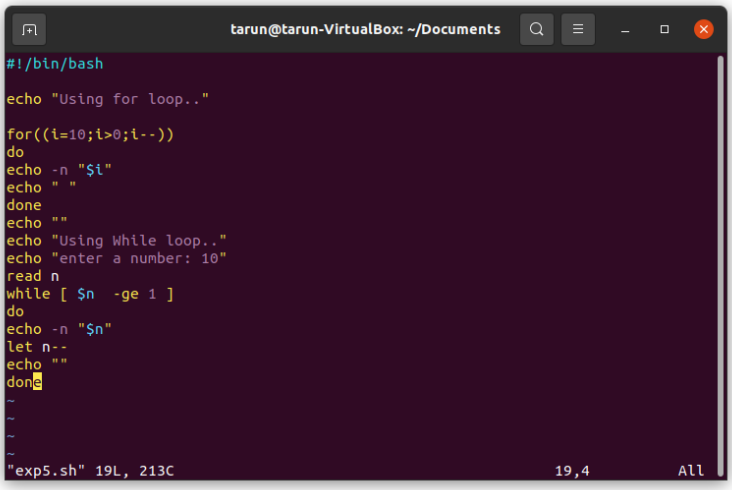
echo "Using for loop... " for ((i=10; i>0;
i--))do

echo -n "$i "
doneecho ""

echo "Using while loop..."
j=10while [$j -ge 1] do
echo -n "$j "

j=$((j - 1)) # decrease number by 1
doneecho ""

Screenshot:



Output:

```
tarun@tarun-VirtualBox: ~/Documents
tarun@tarun-VirtualBox:~/Documents$ bash ./exp5.sh
Using for loop..
10
9
8
7
6
5
4
3
2
1

Using While loop..
enter a number: 10
10
10
9
8
7
6
5
4
3
2
1
tarun@tarun-VirtualBox:~/Documents$
```

Experiment – 6

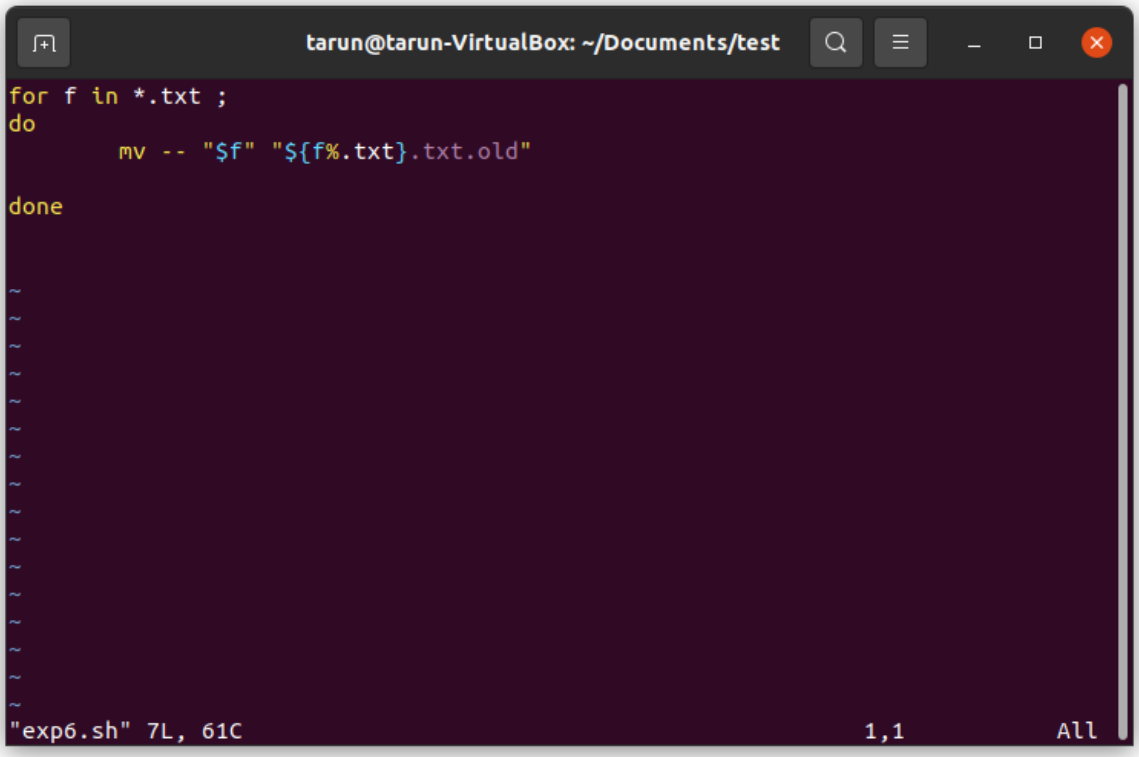
6.Write a program to print that replaces all “*.txt” file names with “*.txt.old” in the current

for f in *.txt ; do

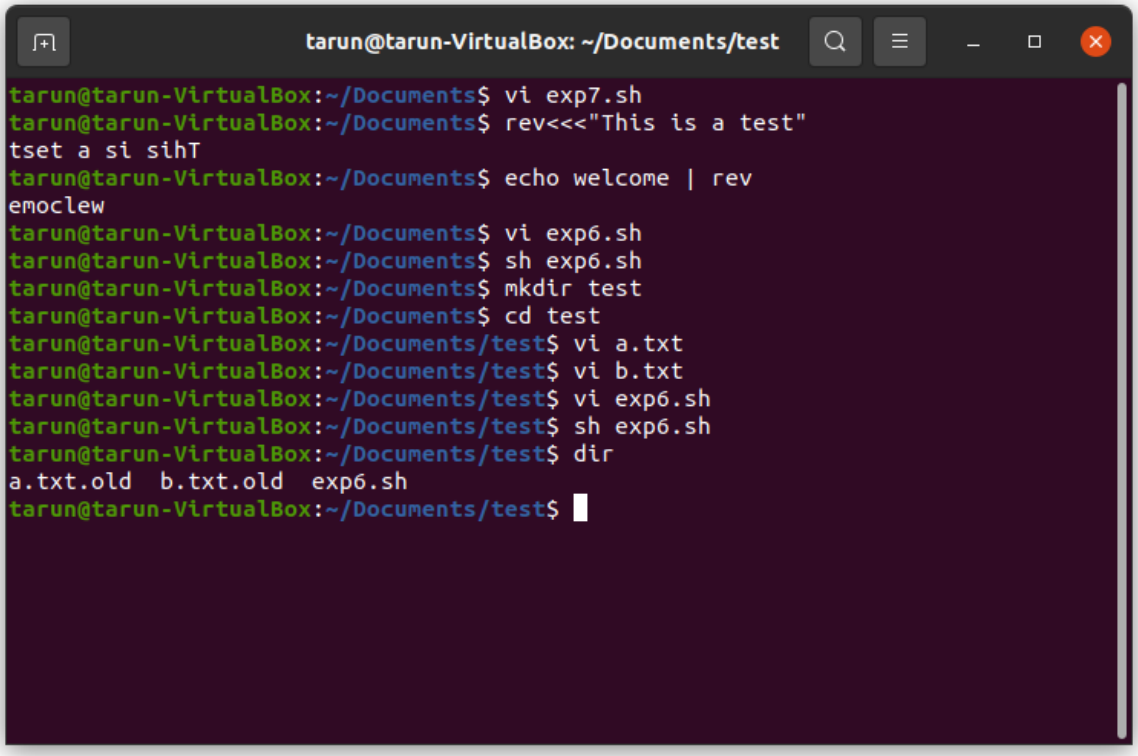
mv -- "$f" "${f%.txt}.txt.old"# -- is used to signify the end of command options, after which onlypositional parameters are accepted.

Done

Screenshot:



OUTPUT:


tarun@tarun-VirtualBox: ~/Documents/test
tarun@tarun-VirtualBox:~/Documents$ vi exp7.sh
tarun@tarun-VirtualBox:~/Documents$ rev<<<"This is a test"
tset a si sihT
tarun@tarun-VirtualBox:~/Documents$ echo welcome | rev
emoclew
tarun@tarun-VirtualBox:~/Documents$ vi exp6.sh
tarun@tarun-VirtualBox:~/Documents$ sh exp6.sh
tarun@tarun-VirtualBox:~/Documents$ mkdir test
tarun@tarun-VirtualBox:~/Documents$ cd test
tarun@tarun-VirtualBox:~/Documents/test$ vi a.txt
tarun@tarun-VirtualBox:~/Documents/test$ vi b.txt
tarun@tarun-VirtualBox:~/Documents/test$ vi exp6.sh
tarun@tarun-VirtualBox:~/Documents/test$ sh exp6.sh
tarun@tarun-VirtualBox:~/Documents/test$ dir
a.txt.old b.txt.old exp6.sh
tarun@tarun-VirtualBox:~/Documents/test$

Experiment – 7

7.Write a program that echoes itself to stdout, but backwards.
Type the following codes in command prompt:

 # Method 1

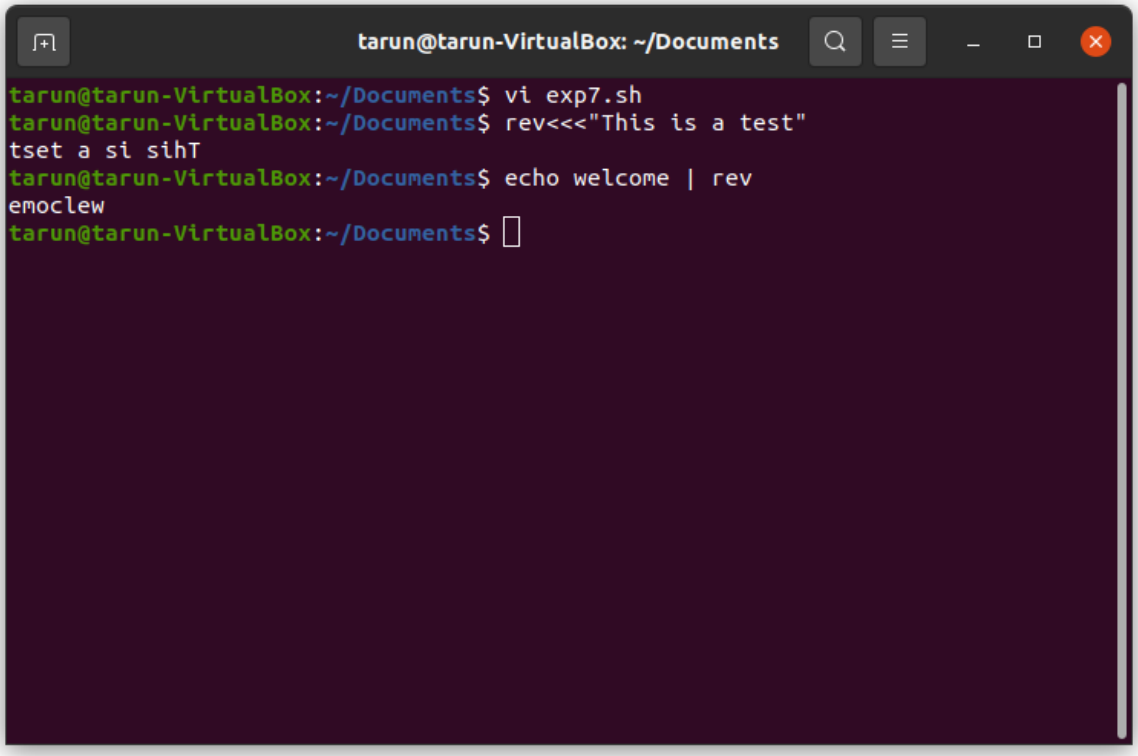
[students@localhost ~]$ rev<<<"This is a test"

#Method 2

[students@localhost~]$echo welcome | rev

Output: tset a si sihT
Output: emoclew

Screenshot:

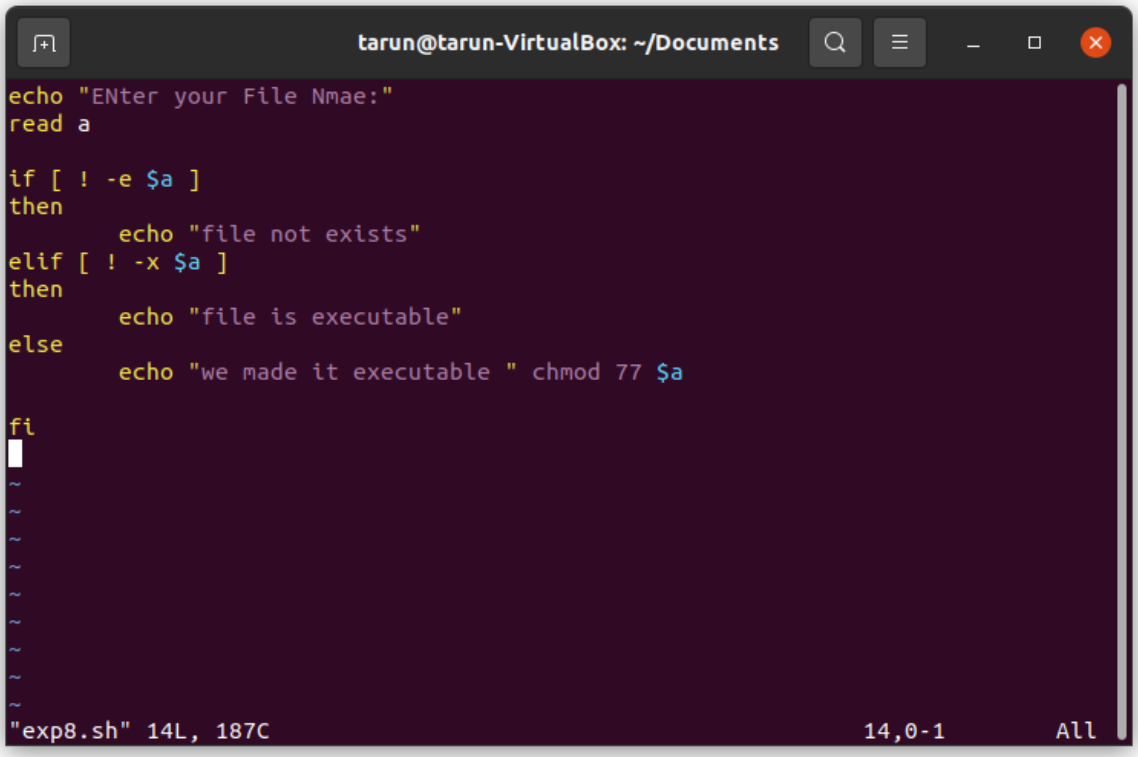


Experiment – 8

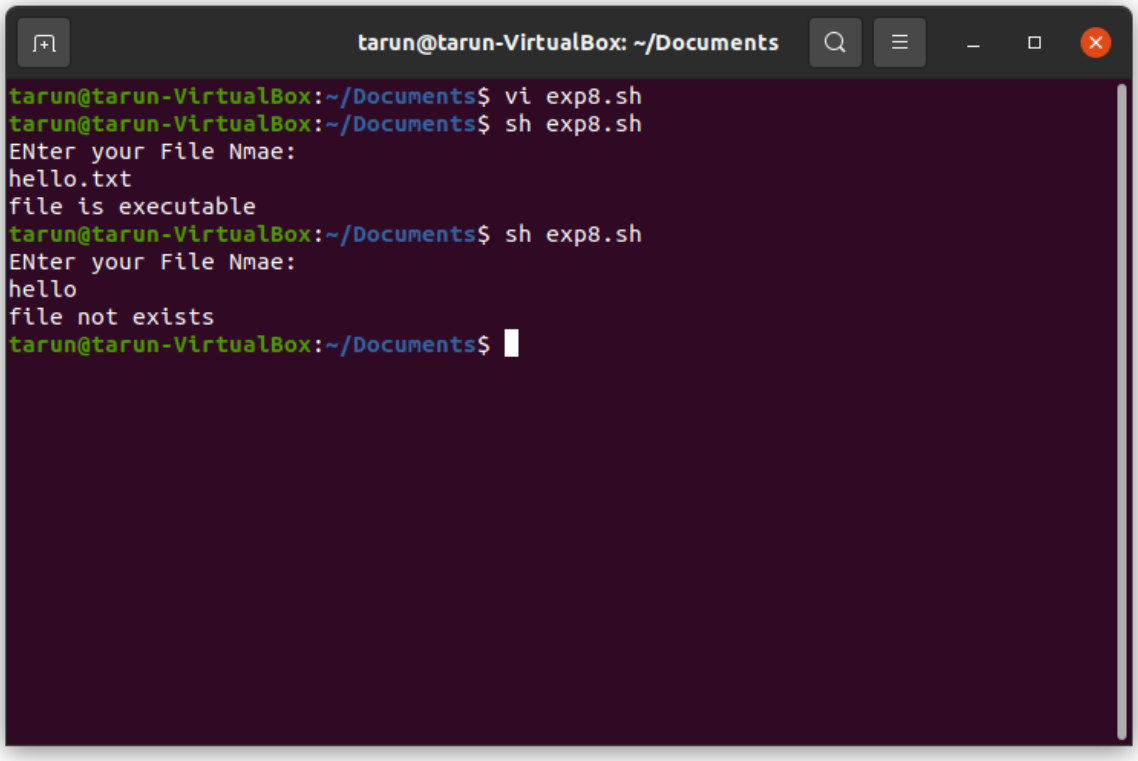
8. Write a program that takes a filename as input and checks if it is executable, if not make itexecutable.

echo “Enter your file name” read a
if [! –e $a] then
echo “file not exist” elif[! –x $a]
then
echo “file is executable” else
echo “we made it executable” chmod 777 $a
fi

SCREENSHOT:



OUTPUT:

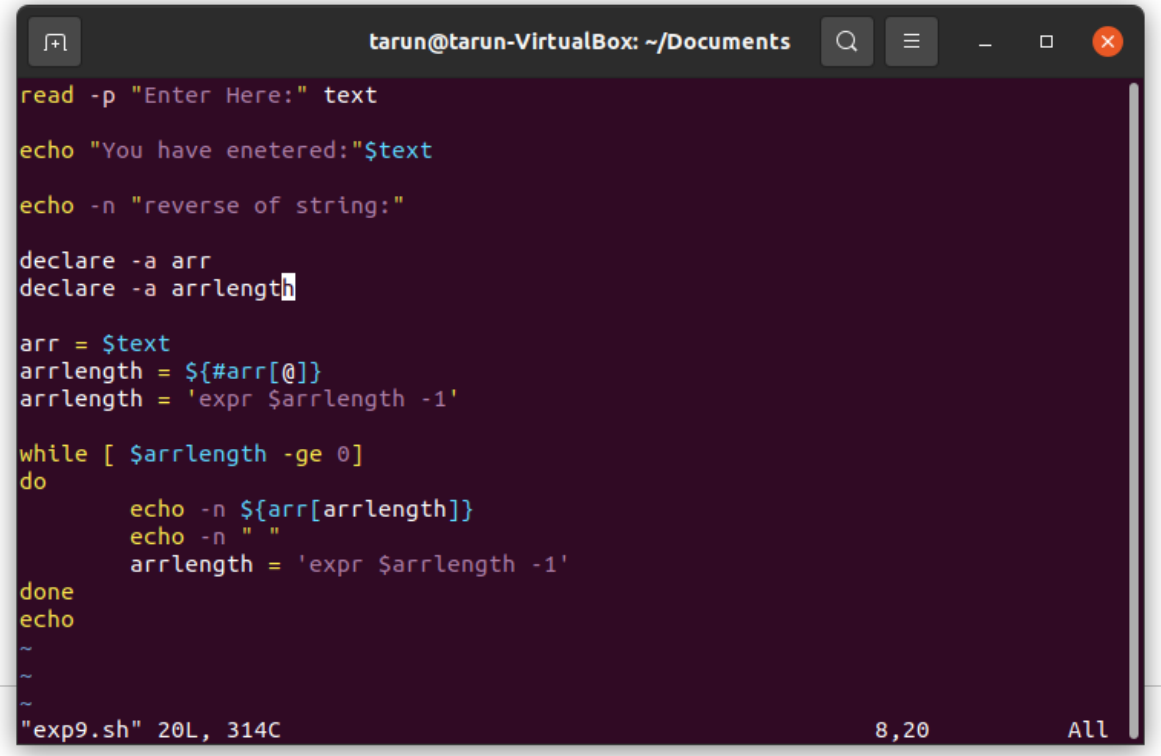

tarun@tarun-VirtualBox: ~/Documents
tarun@tarun-VirtualBox:~/Documents$ vi exp8.sh
tarun@tarun-VirtualBox:~/Documents$ sh exp8.sh
ENter your File Nmae:
hello.txt
file is executable
tarun@tarun-VirtualBox:~/Documents$ sh exp8.sh
ENter your File Nmae:
hello
file not exists
tarun@tarun-VirtualBox:~/Documents$

EXPERIMENT – 9

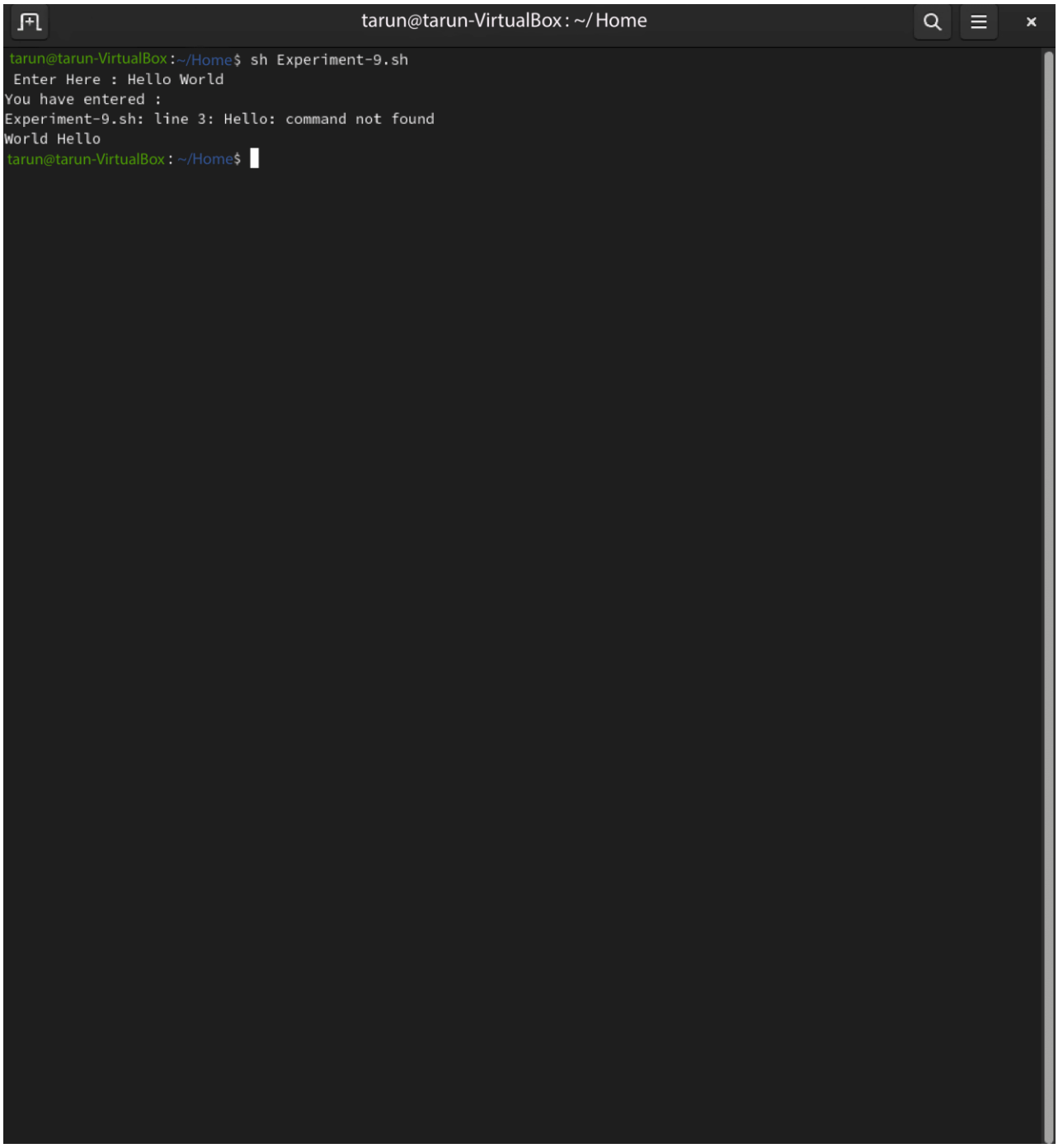
9.WAP to take a string as command line argument and reverse it.

```
read -p " Enter Here : " text
echo "You have entered : " $text
echo -n "Reverse of String : "
arr=($text)

arrlength=${#arr[@]}
arrlength=`expr $arrlength - 1`
while [ $arrlength -ge 0 ]
do
echo -n ${arr[arrlength]} echo -n " "
arrlength=`expr $arrlength - 1`
done
echo
```

SCREENSHOT:



OUTPUT:


tarun@tarun-VirtualBox : ~/ Home
tarun@tarun-VirtualBox : ~/Home$ sh Experiment-9.sh
Enter Here : Hello World
You have entered :
Experiment-9.sh: line 3: Hello: command not found
World Hello
tarun@tarun-VirtualBox : ~/Home$

Experiment – 10

10 **:** Create a data file called employee in the format given below:

a. EmpCode Character

b. EmpName Character

c. Grade Character

d. Years of experience Numeric

e. Basic Pay Numeric

Create a file named employee: vi employee
Type the following in employee. Use tabs to separate the fields.

A001	Arjun	E1	1	12000.00
A006	Anand	E1	1	12450.00
A010	Rajesh	E2	3	14500.00
A002	Mohan	E2	2	13000.00
A005	John	E2	1	14500.00
A009	Denial	E2	4	17500.00
A004	Wills	E1	1	12000.00

Save and exit.

Perform the following functions on the file. Type the following commands in command prompt:

OUTPUT CODE:

a) Sort the file on EmpCode. $cut -f1 employee | sort

b) Sort the file on EmpName. $cut –f2 employee | sort

c) Sort the file on
i) Decreasing order of basic pay. $cut –f5 employee | sort –r
ii) Increasing order of years of experience. $cut –f4 employee | sort

d) Display the number of employees whose details are included in the file. wc -l employee

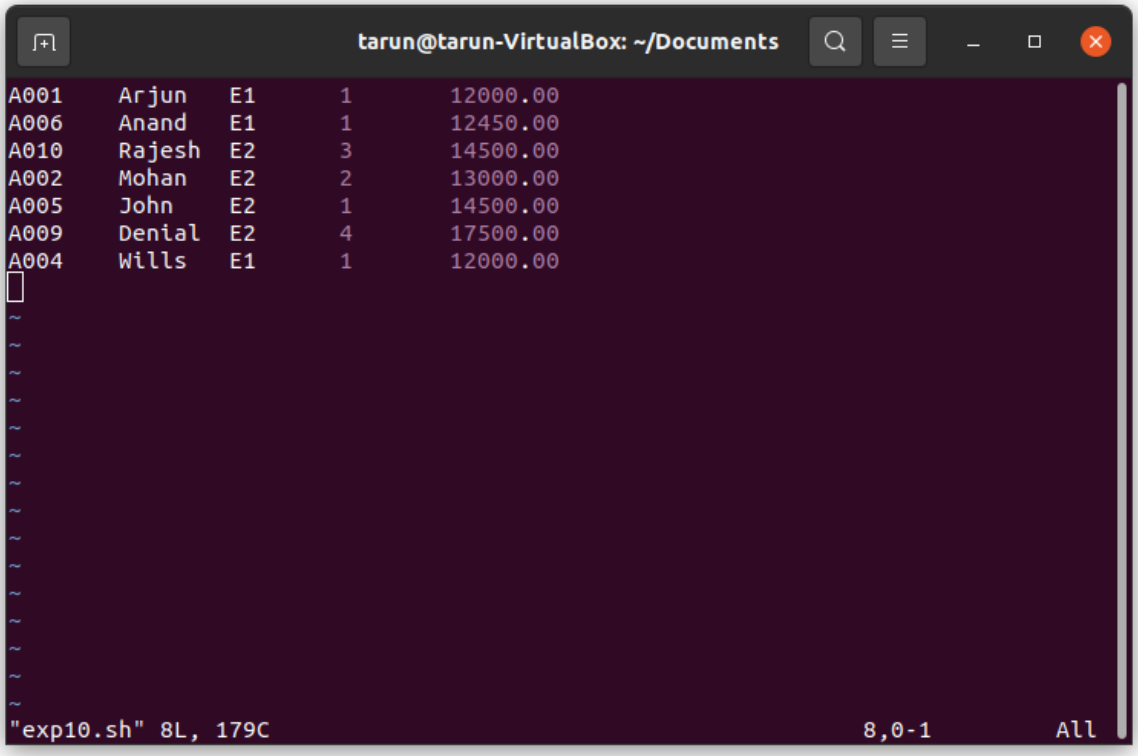
e) Display all records with ‘Wills’ a part of employee name. $cut -f2 -d “ ” employee | grep ‘^[Smith]’ | wc –l

f) Display all records with EmpName starting with ‘A’. $cut -f2 employee | grep ‘^[A]’ | wc –l

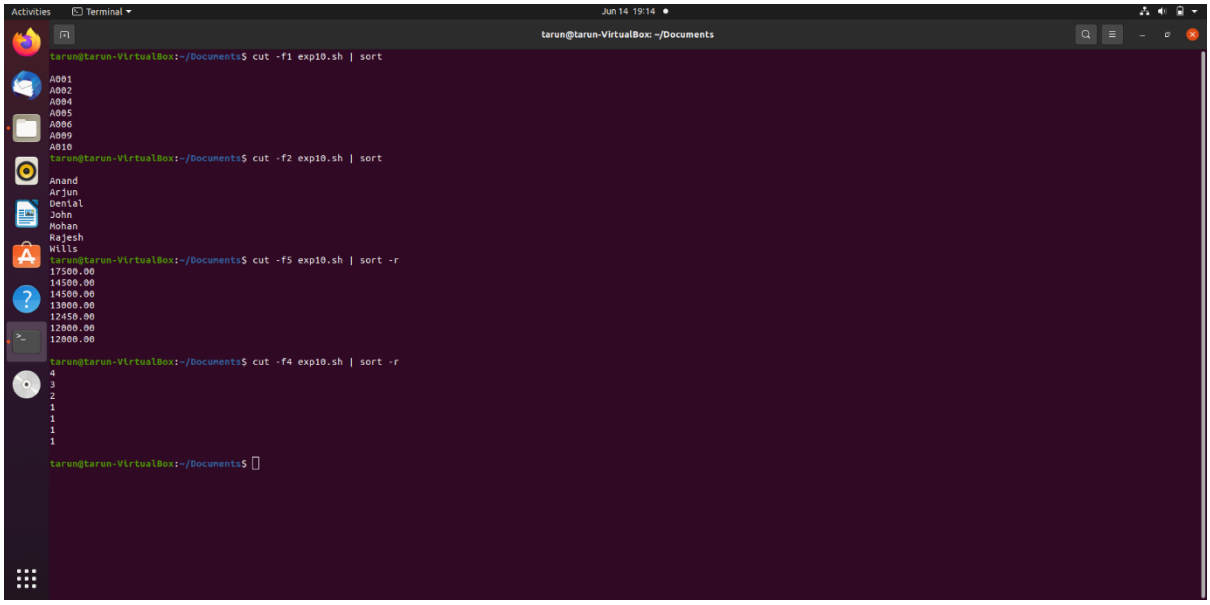
g) Display the records of Employees whose grade is E2 and have work experience of 2 to 5 years. $cut -f3 employee | grep ‘[*2]’ | cut -f4 employee | grep ‘[2-5]’

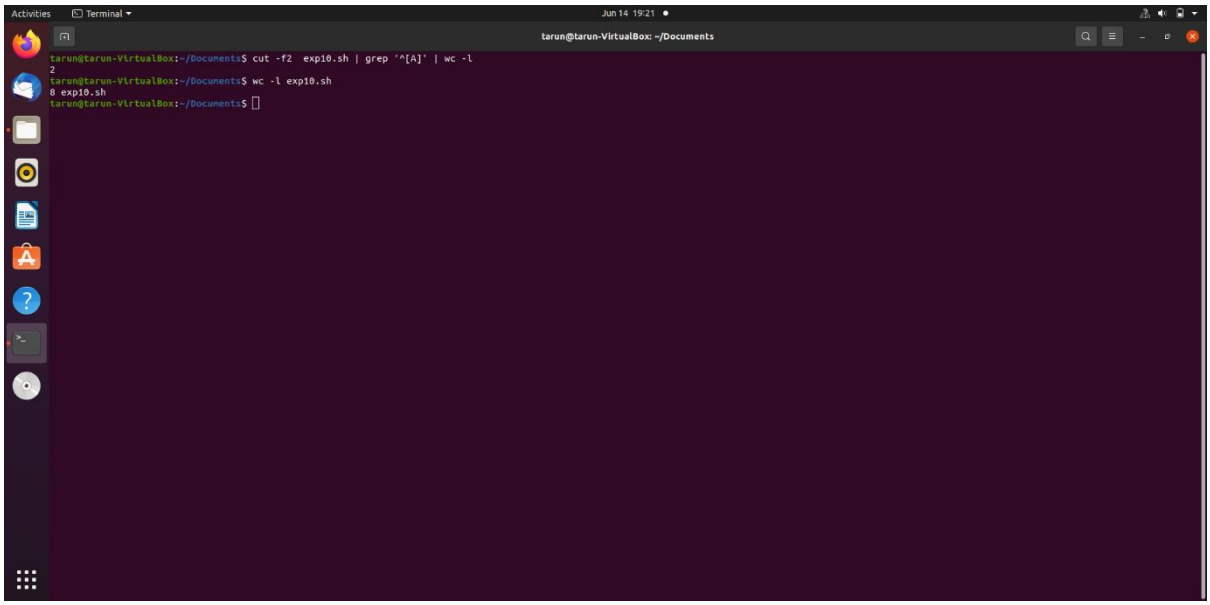
h) Display records of all employees who are not in grade E2. $cut –f3 employee | grep ‘[*1]’

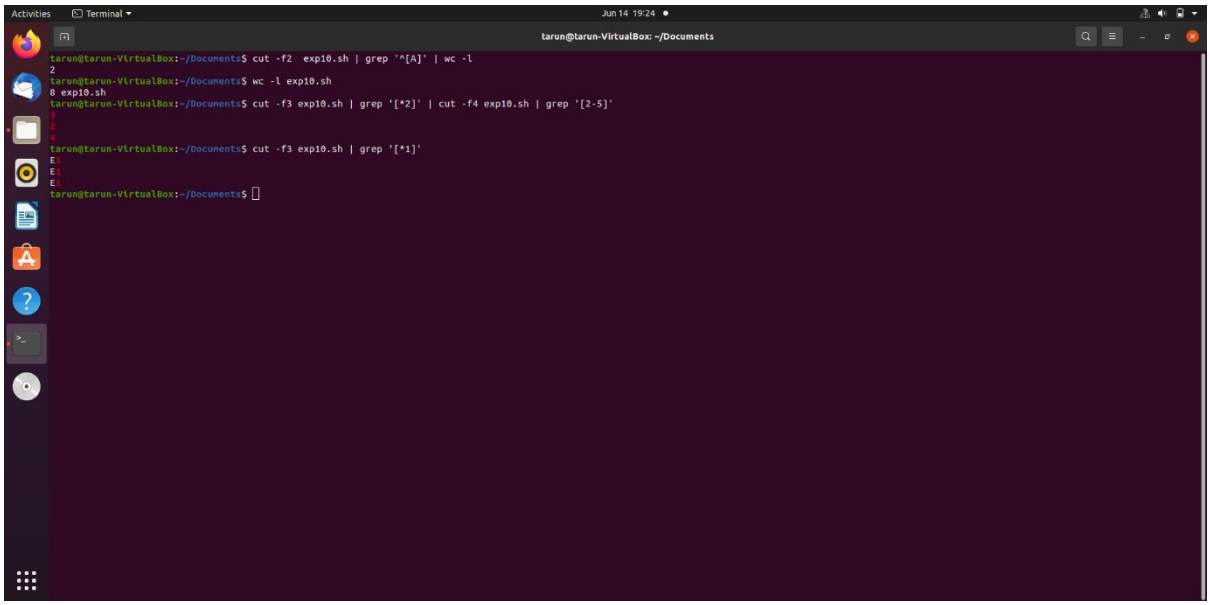
SCREENSHOT:



OUTPUT:


Activities
Terminal
Jun 14 19:14
tarun@tarun-VirtualBox: ~/Documents
tarun@tarun-VirtualBox:~/Documents$ cut -f1 exp10.sh | sort
A001
A002
A004
A005
A006
A009
A010
tarun@tarun-VirtualBox:~/Documents$ cut -f2 exp10.sh | sort
Anand
Arjun
Dental
John
Mohan
Rajesh
Wills
tarun@tarun-VirtualBox:~/Documents$ cut -f5 exp10.sh | sort -r
17500.00
14500.00
14500.00
13000.00
12450.00
12000.00
12000.00
tarun@tarun-VirtualBox:~/Documents$ cut -f4 exp10.sh | sort -r
4
3
2
1
1
1
1
tarun@tarun-VirtualBox:~/Documents$


Activities
Terminal
Jun 14 19:21
tarun@tarun-VirtualBox: ~/Documents
tarun@tarun-VirtualBox:~/Documents$ cut -f2 exp10.sh | grep '^[A]' | wc -l
2
tarun@tarun-VirtualBox:~/Documents$ wc -l exp10.sh
8 exp10.sh
tarun@tarun-VirtualBox:~/Documents$


Activities
Terminal
Jun 14 19:24
tarun@tarun-VirtualBox: ~/Documents
tarun@tarun-VirtualBox:~/Documents$ cut -f2 exp10.sh | grep '^[A]' | wc -l
2
tarun@tarun-VirtualBox:~/Documents$ wc -l exp10.sh
8 exp10.sh
tarun@tarun-VirtualBox:~/Documents$ cut -f3 exp10.sh | grep '[*2]' | cut -f4 exp10.sh | grep '[2-5]'
3
3
4
tarun@tarun-VirtualBox:~/Documents$ cut -f3 exp10.sh | grep '[*1]'
E1
E1
E1
tarun@tarun-VirtualBox:~/Documents$