Date: 13/07/2020 **Reg. No.: 19BCE0977**
Slot: L3 + L4 **Name: Swaranjana Nayak**

EXPERIMENT 1

BASICS OF UNIX COMMANDS

AIM:

To write the basics of UNIX commands for file processing and working with directories.

WORKING WITH FILES

1. **CREATING A FILE**

DESCRIPTION:
This command is used to create a file
SYNTAX:
```
$cat>filename
```
EXAMPLE:
```
$cat>Demo
```

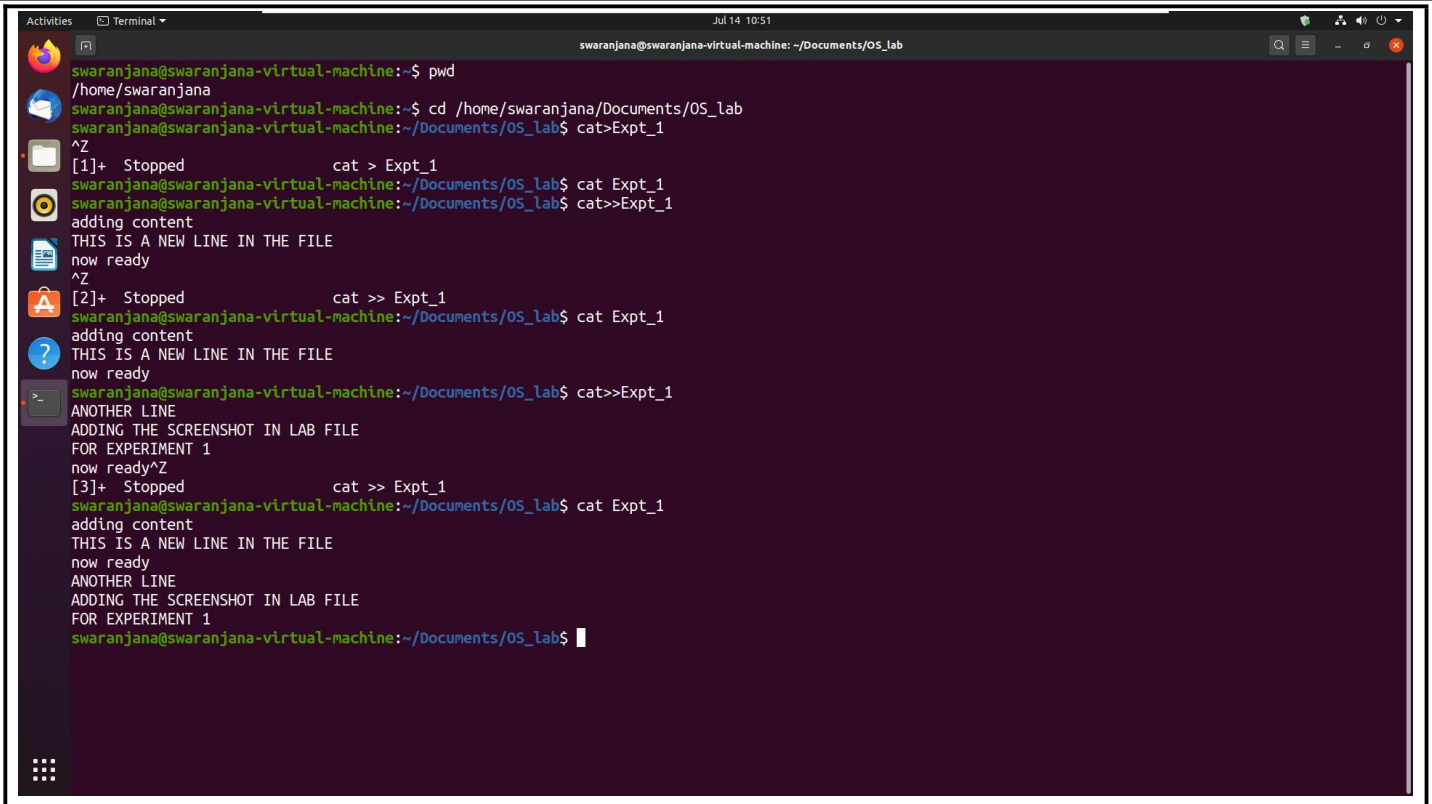
2. **ADD COMMAND**

DESCRIPTION:
This command is used to create a file
SYNTAX:
```
$cat>>filename
```
EXAMPLE:
```
$cat>>Demo
Adding contents
Now ready
^Z
[2] +  Stopped                 cat >> Expt_1
```

(Note: Ctrl + Z suspends the most recent foreground process)

3. **VIEW COMMAND**

DESCRIPTION:
This command is used to display contents a file
SYNTAX:
```
$cat filename
```
EXAMPLE:
```
$cat Demo
Adding contents
Now ready
```



4. **COPY COMMAND**

DESCRIPTION:

This command is used to copy one file to another file

SYNTAX:

$cp <oldfilename> <newfilename>

EXAMPLE:

$cp aaa bbb

5. **MOVE COMMAND**

DESCRIPTION:

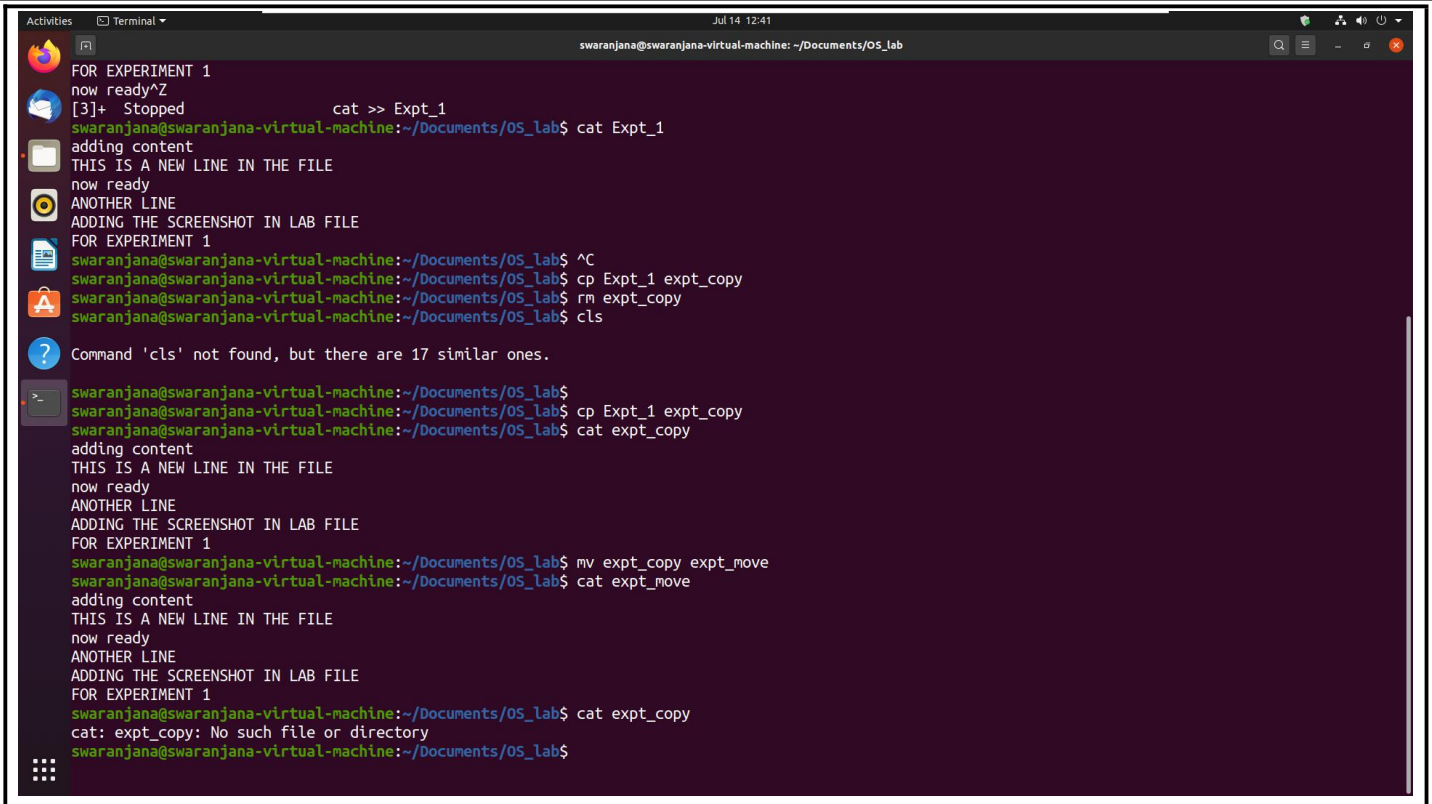
This command is used move the file from one directory to another directory

SYNTAX:

$mv <filename> <filename / directoryname>

EXAMPLE:

$mv aaa bbb



6. LIST COMMAND

DESCRIPTION:

It is used to **print** contents of a directory, by default it lists contents of current working directory(**pwd**)

SYNTAX:

```
$ ls
```

EXAMPLE:

```
$ls
Demo aaa bbb
```

7. REMOVE COMMAND

DESCRIPTION:

This command is used to remove a file.

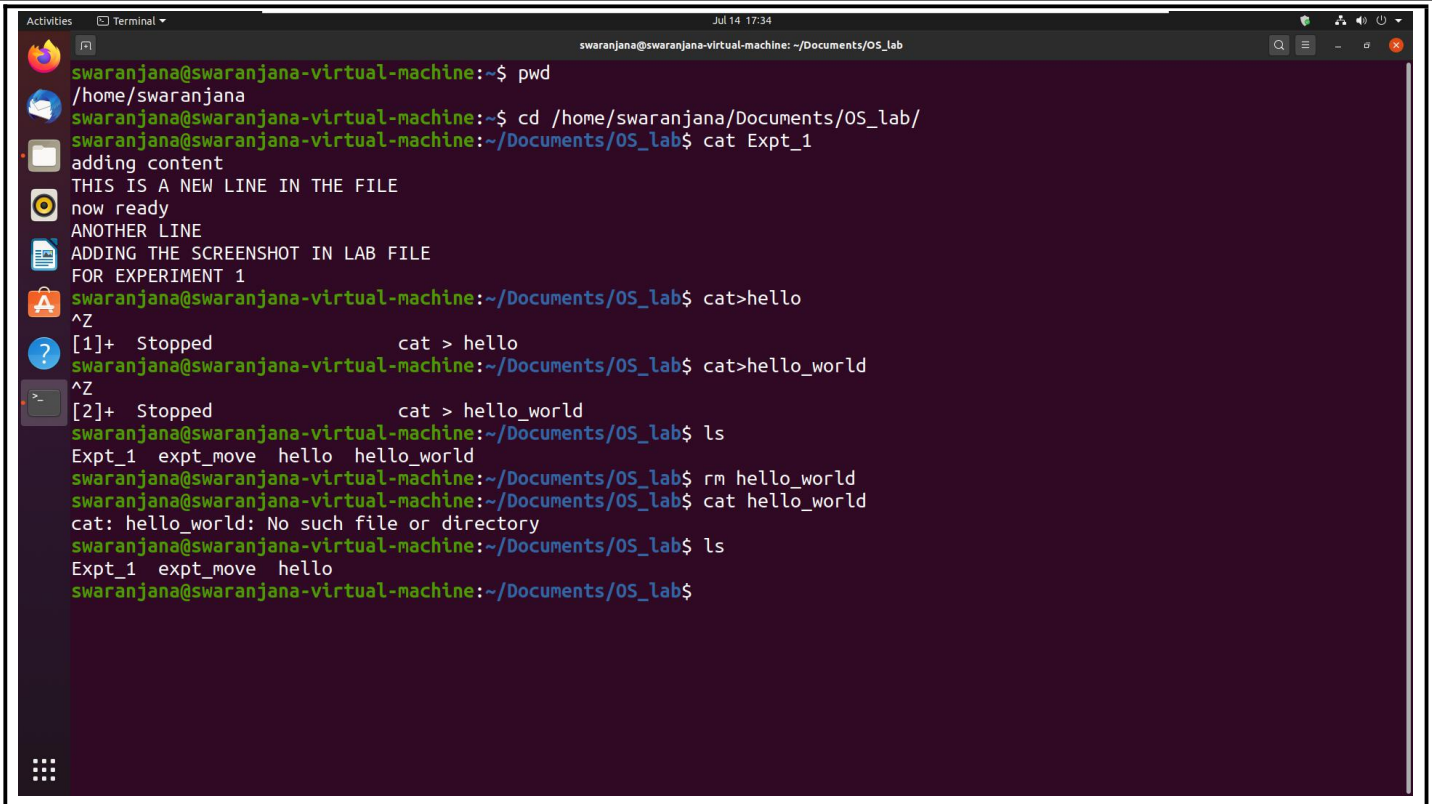
SYNTAX:

```
$rm <filename>
```

EXAMPLE:

```
$rm abc
```

(Note: ‘clear’ command is used to clear terminal screen)



8. MAN COMMAND

DESCRIPTION:

It is used to display the manual pages. The command is used to find details about other commands. Almost every command has their respective man pages, useful to get a quick overview of unknown command.

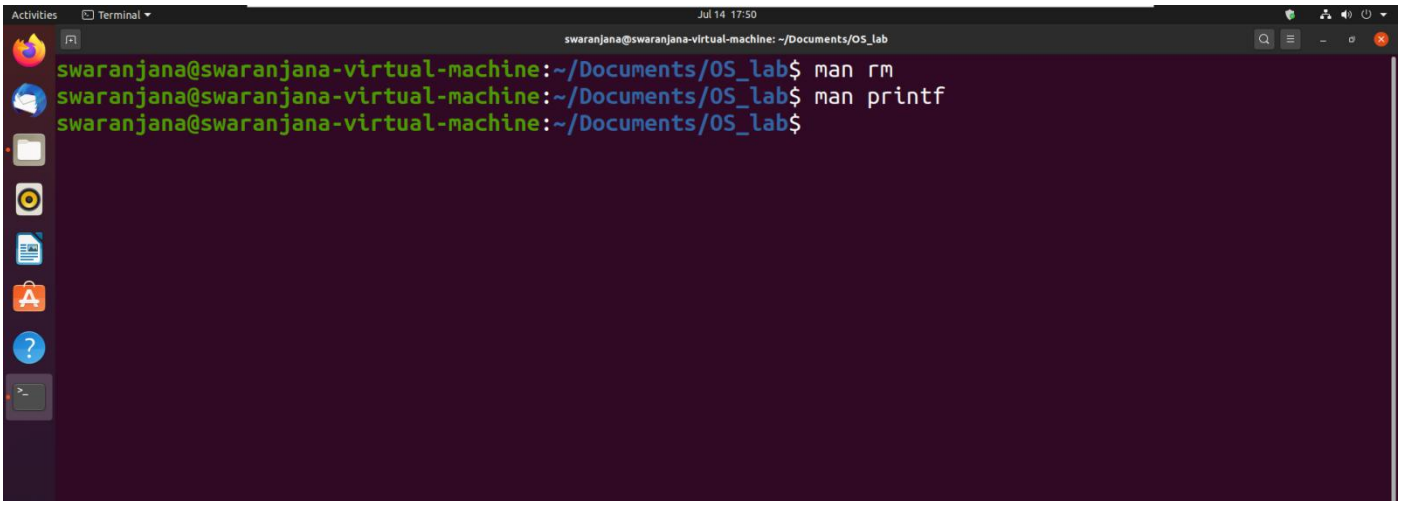
SYNTAX:

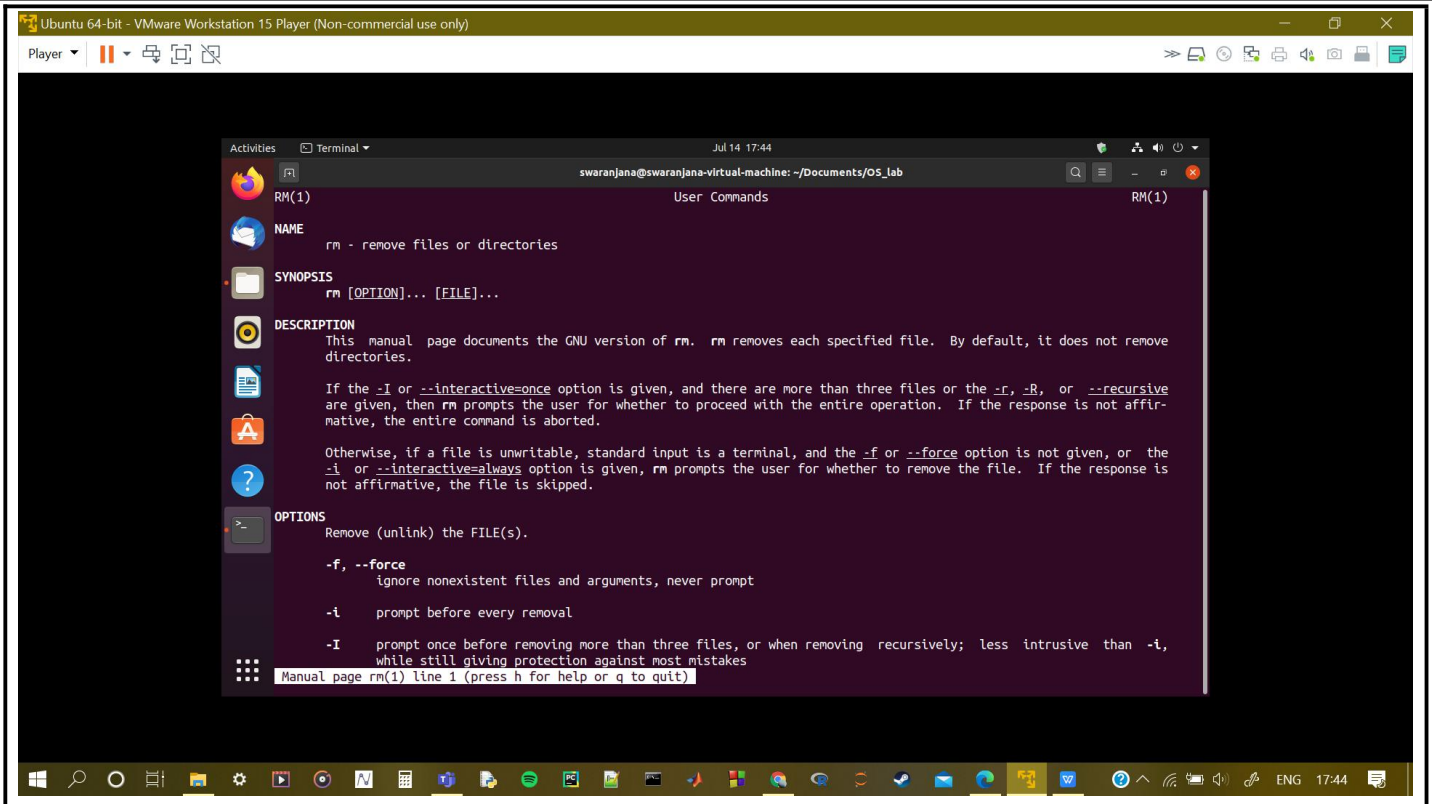
`$man commandName/ functionName`

EXAMPLE:

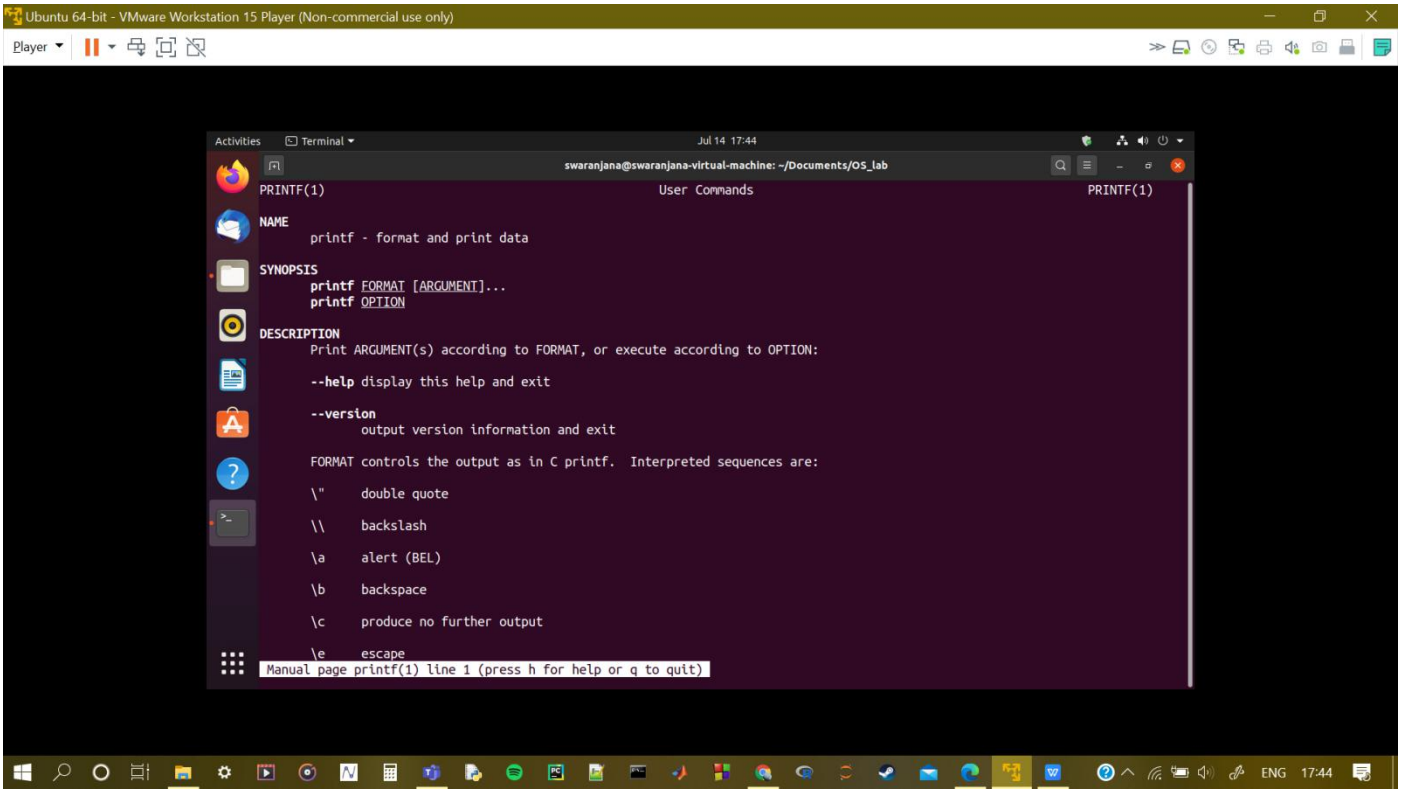
`$man scanf`

(displays the man page of scanf function in C)





1. Man page of rm command



2. Man page of printf command

9. ECHO COMMAND

DESCRIPTION:

This command is used to display the text.

SYNTAX:

$echo text/variable

EXAMPLE:

$echo hello

10. WC COMMAND

DESCRIPTION:

This command is used to display the number of lines, words and character.

SYNTAX:

wc – l filename

wc – w filename

wc – c filename

11. LIST COMMAND

DESCRIPTION:

This command is used to list the current file in the directory.

SYNTAX:

$ls filename

EXAMPLE:

$ls aaa

ATTRIBUTES:

$ls – l -> List of files in long format.

$ls – u -> Display the files in the order of last access time.

$ls – a -> It displays all the files and directories including hidden files.

$ls – p -> Put slash after the directories.

$ls – t -> Displays the file in order of modified time.

12. HEAD COMMAND

DESCRIPTION:

This command is used to display the specified line in the specified file.

SYNTAX:

$head ~count <filename>

13. TAIL COMMAND

DESCRIPTION:

This command is used to retrieve specified lines from given file in order we choose.

SYNTAX:

$tail [+/- count] <filename>

EXAMPLE:

$tail -3 book

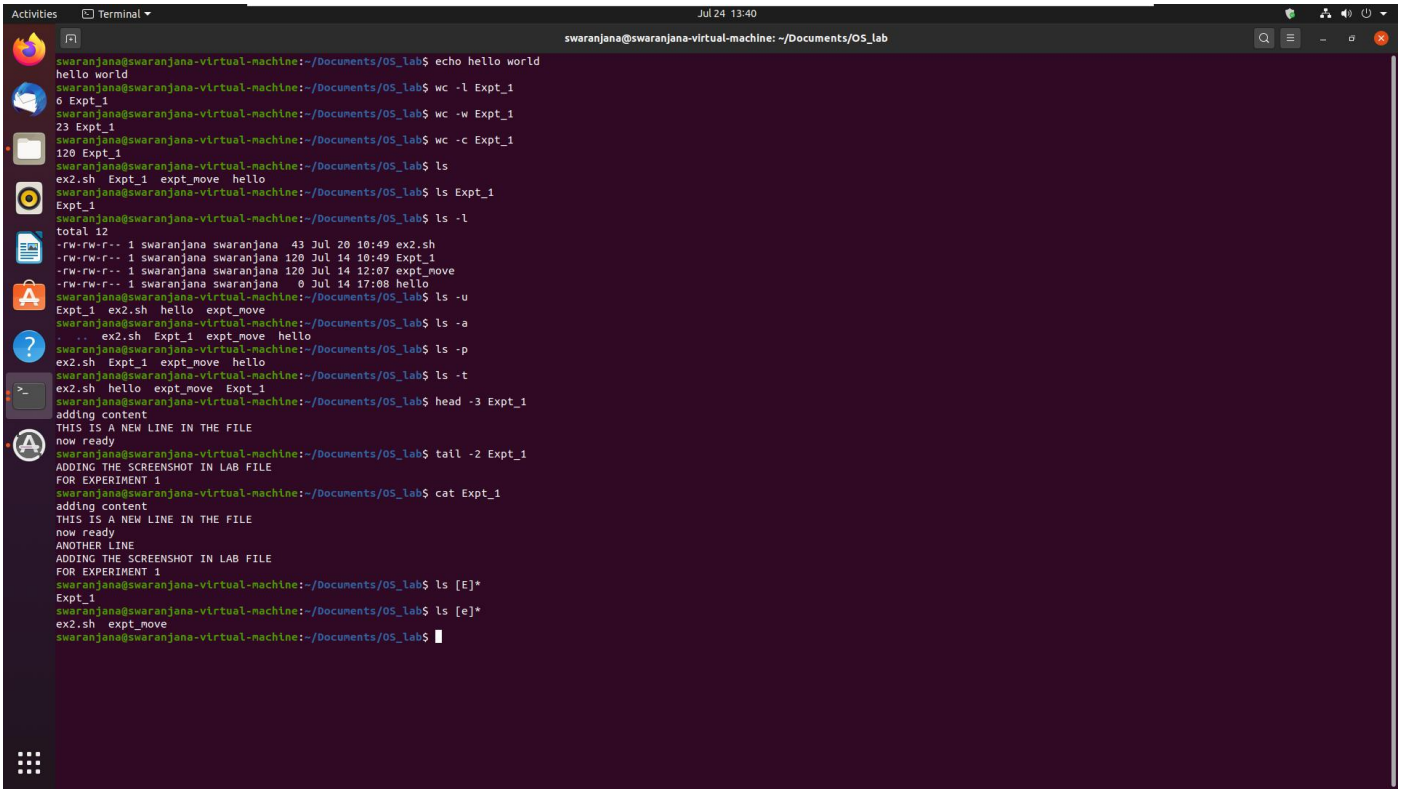
14. WILD CHAR COMMAND/PATTERN

DESCRIPTION:

It is used to display the file starting with specified character.

SYNTAX:

$ls [alphabet]*



WORKING WITH DIRECTORIES

1. CREATE A DIRECTORY

DESCRIPTION:

This command is used to create a new directory.

SYNTAX:

$mkdir directoryname

EXAMPLE:

$mkdir book

2. CHANGE COMMAND

DESCRIPTION:

This command is used to change the new directory.

SYNTAX:

$cd directoryname

EXAMPLE:

$cd flowers

3. REMOVE DIRECTORY COMMAND

DESCRIPTION:

This command is used to remove a directory.

SYNTAX:

$rm directoryname

EXAMPLE:

$rm fruits

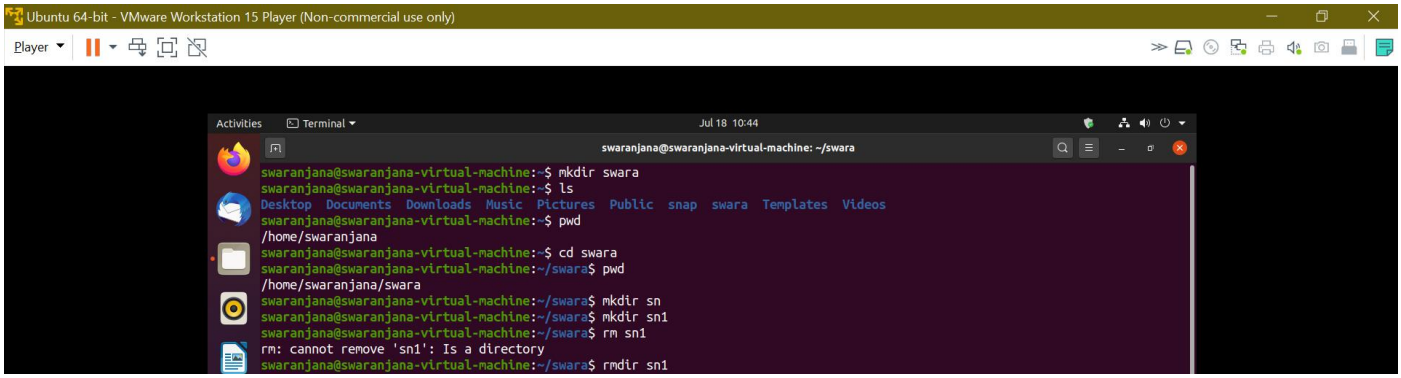
4. PATH COMMAND

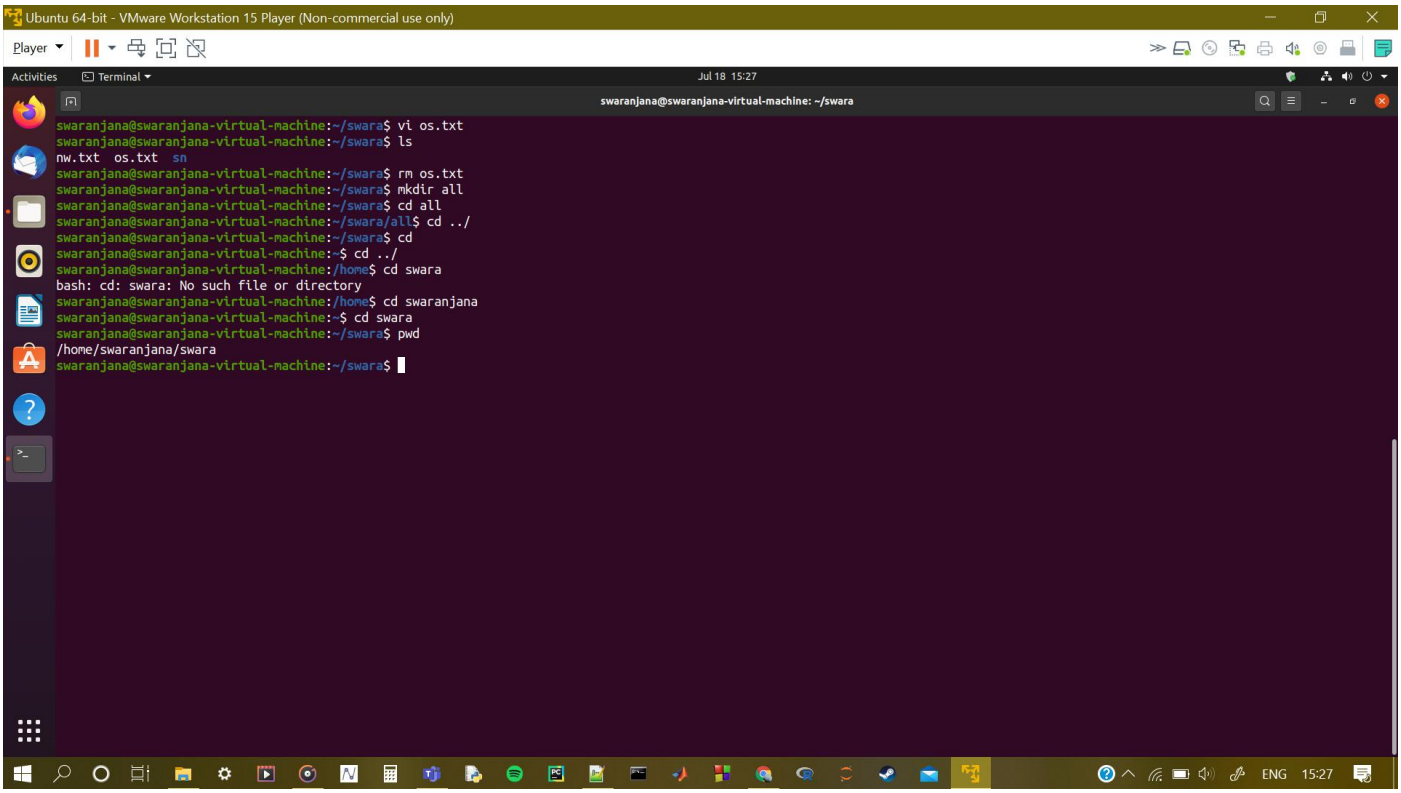
DESCRIPTION:

This command is used to display the path of the current file.

SYNTAX:

$pwd





5. GREP COMMAND

DESCRIPTION:

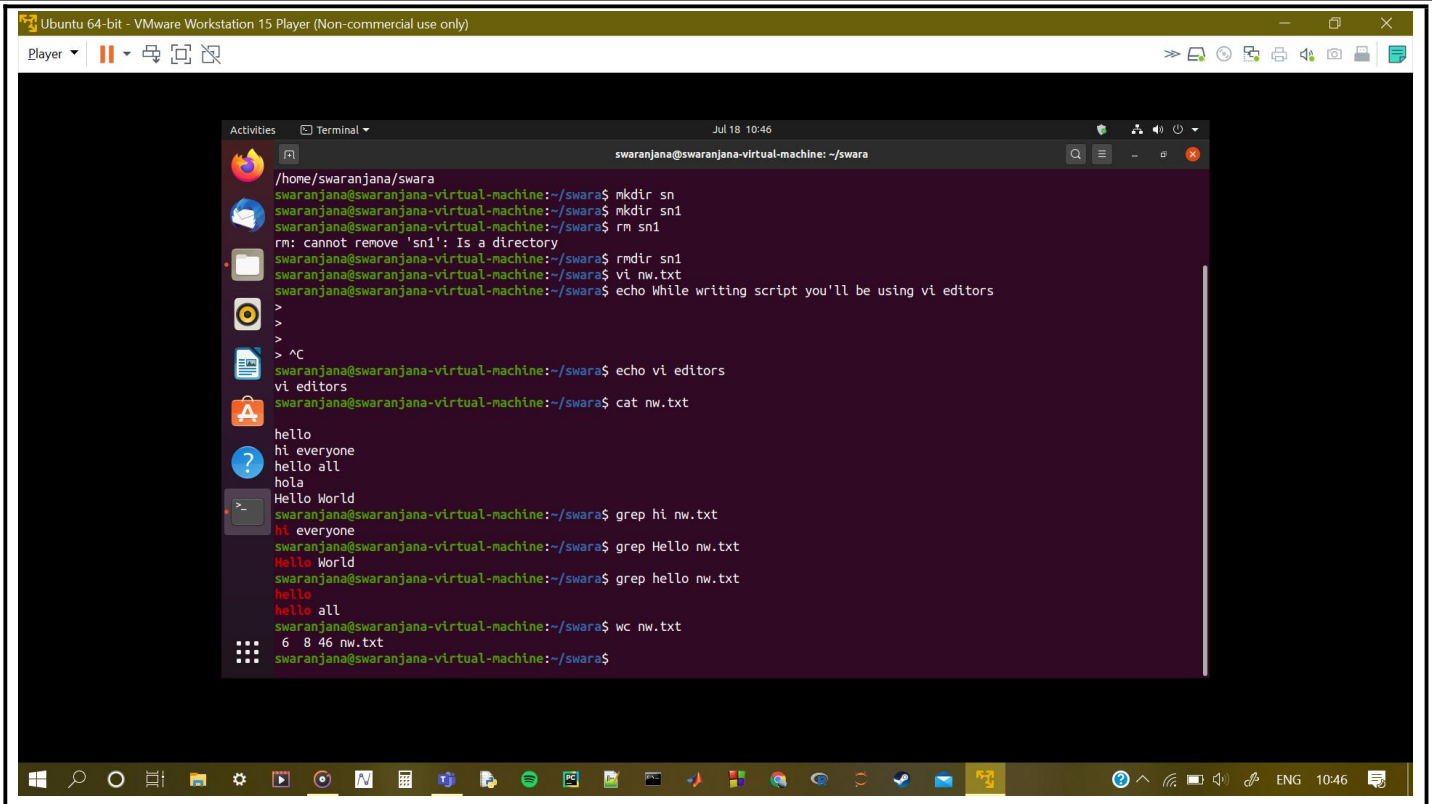
This command is used for certain kind of pattern matching.

SYNTAX:

$grep string filename

EXAMPLE:

$grep earth planet



6. VI COMMAND

DESCRIPTION:

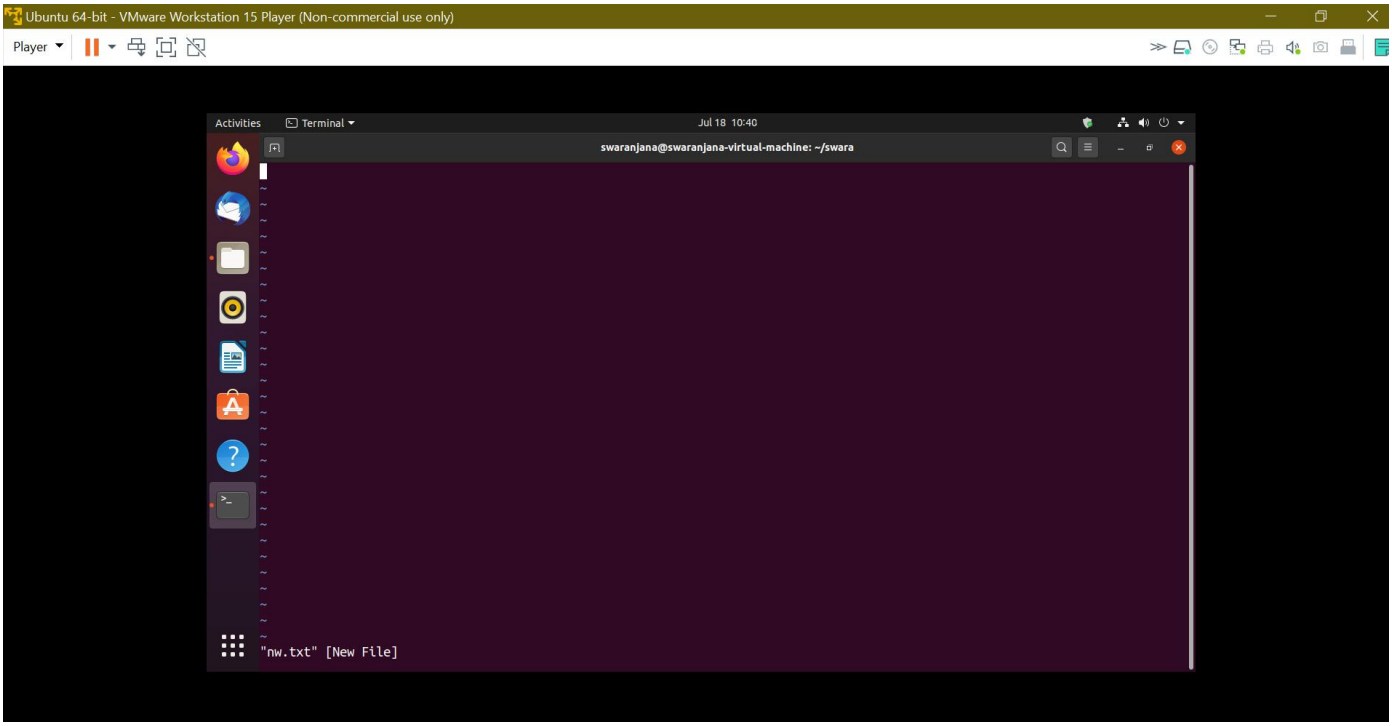
The **vi** command starts the visual mode of **ex**, the landmark editing program

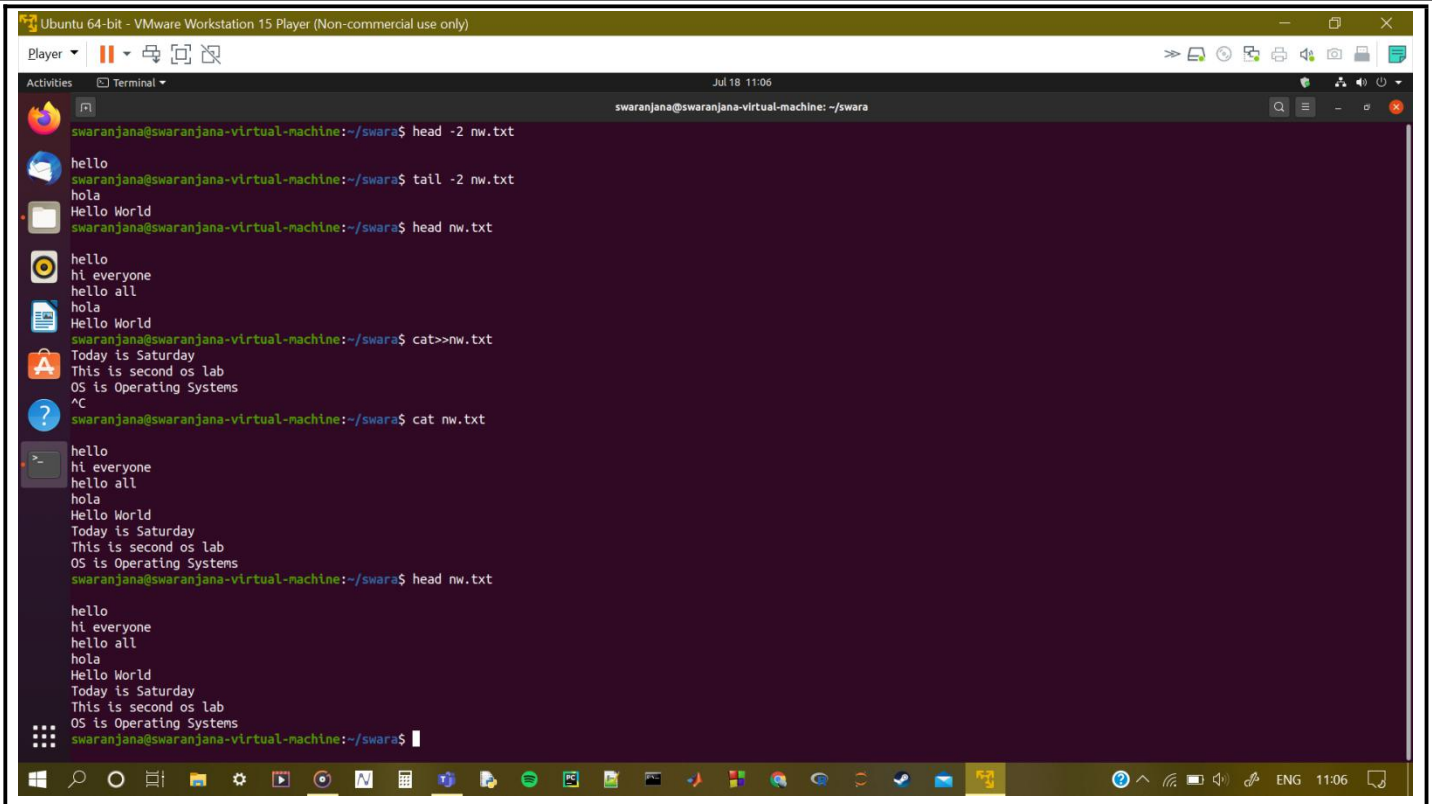
SYNTAX:

$ vi <filename>.<extension>

EXAMPLE:

$vi nw.txt





7. DATE COMMAND

DESCRIPTION

This command is used to display the date in month, year, day, hours, minutes and seconds.

SYNTAX

$date

8. WHO COMMAND

DESCRIPTION

This command is used to see who are the users connected to the server.

SYNTAX

$who

9. CALENDER COMMAND

DESCRIPTION

This command is used to display the calendar with month and year.

SYNTAX

$cal month year

EXAMPLE

$cal 01 2011

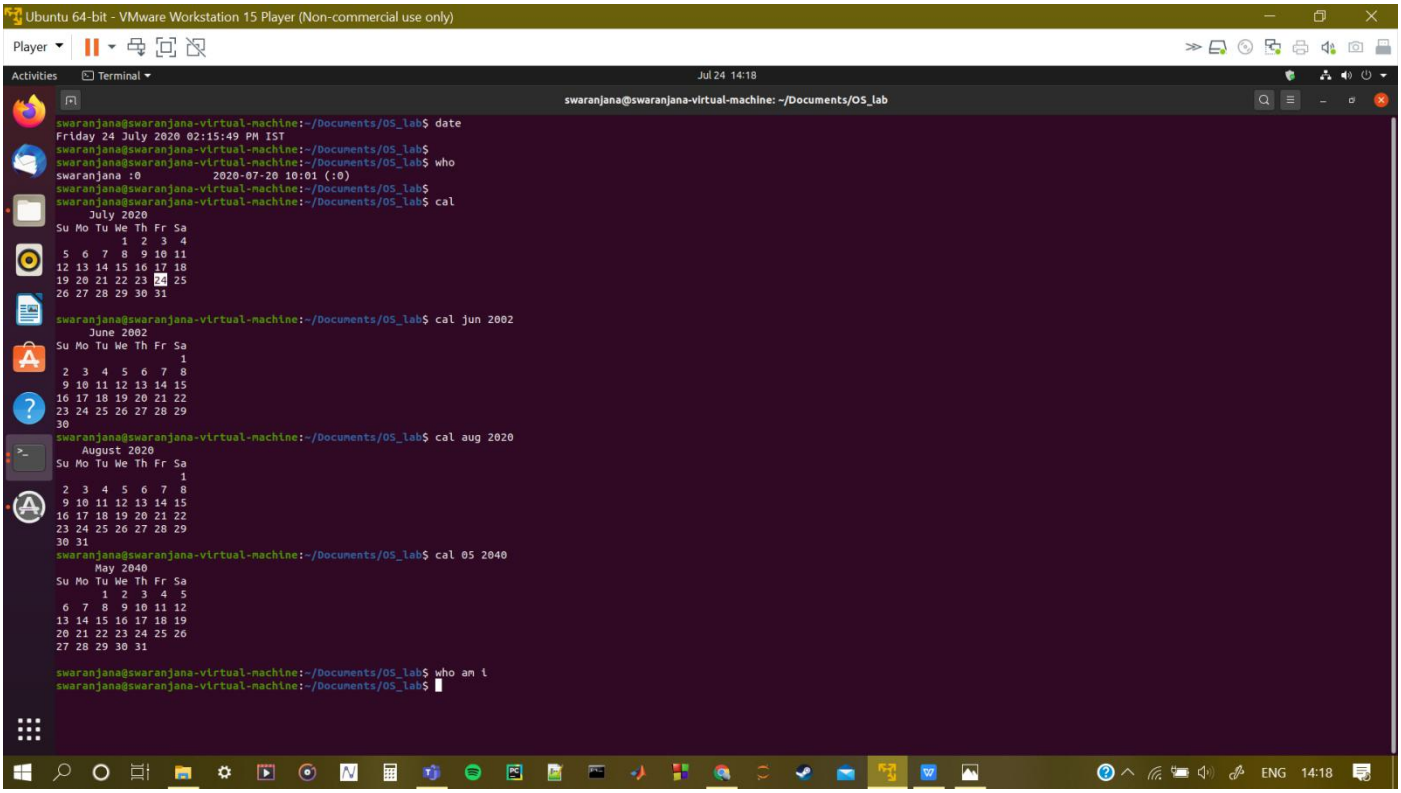
10. WHO AM I COMMAND

DESCRIPTION

This command is used to know the system number.

SYNTAX

$who am I



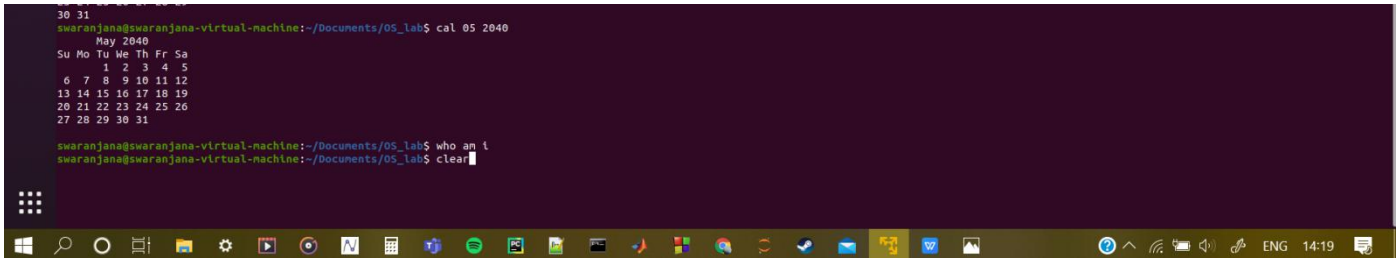
11. CLEAR COMMAND

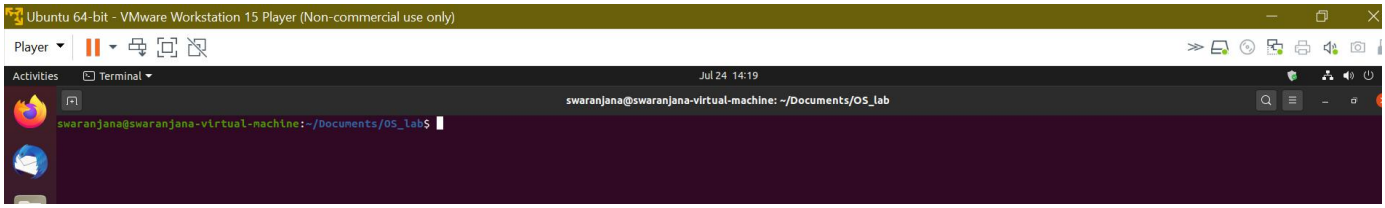
DESCRIPTION

This command is used to clear the screen.

SYNTAX

$clear





12. FILE COMMAND

DESCRIPTION

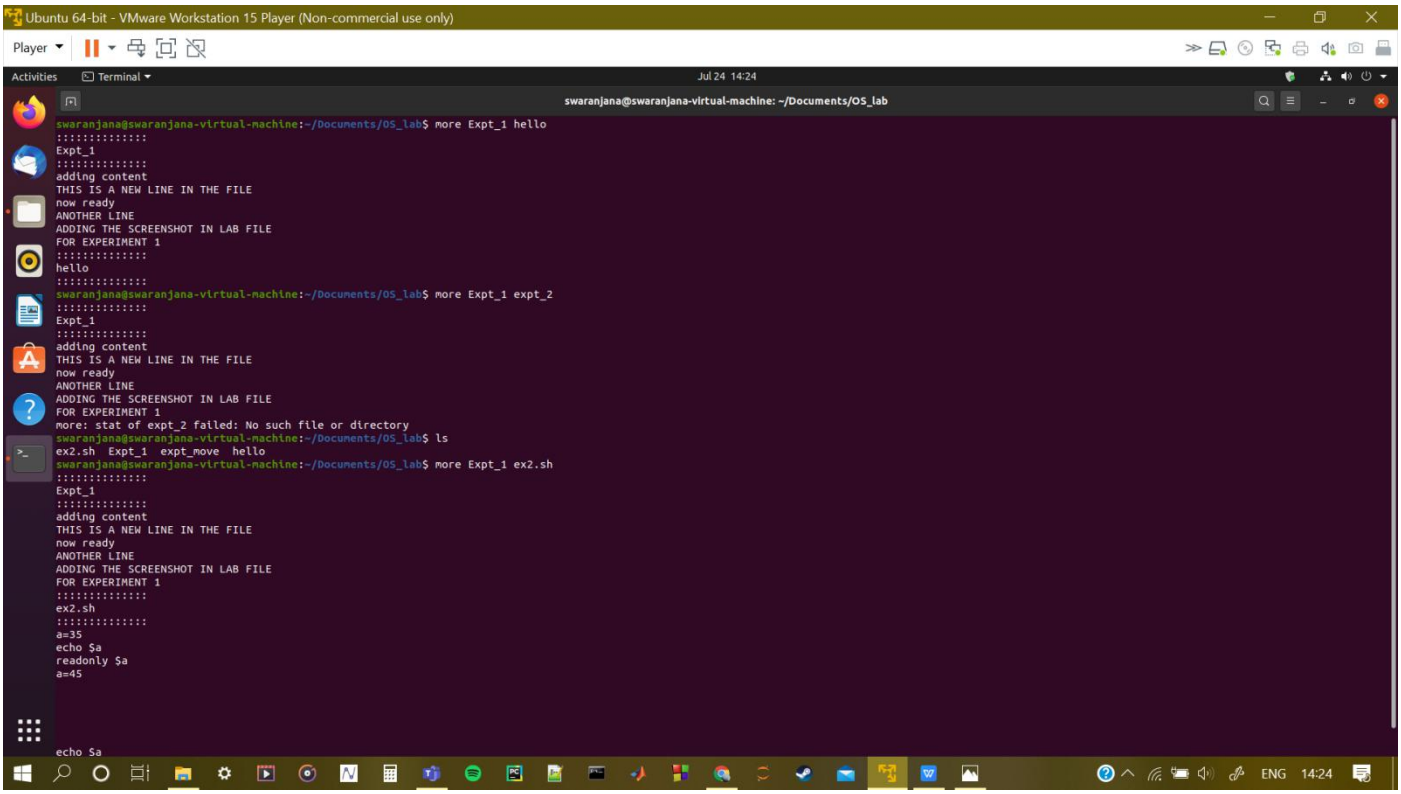
This command is used to display the contents of more than one file.

SYNTAX

$more <file1> <file2>

EXAMPLE

$more aaa bbb



13. SORT COMMAND

DESCRIPTION

This command is used to sort the contents of file in a predefined order.

SYNTAX

$sort filename

EXAMPLE

$sort name

ATTRIBUTES

-n -> Display the numerical value order.

-r -> Display the reverse order.

-m -> The case distinction is ignored.

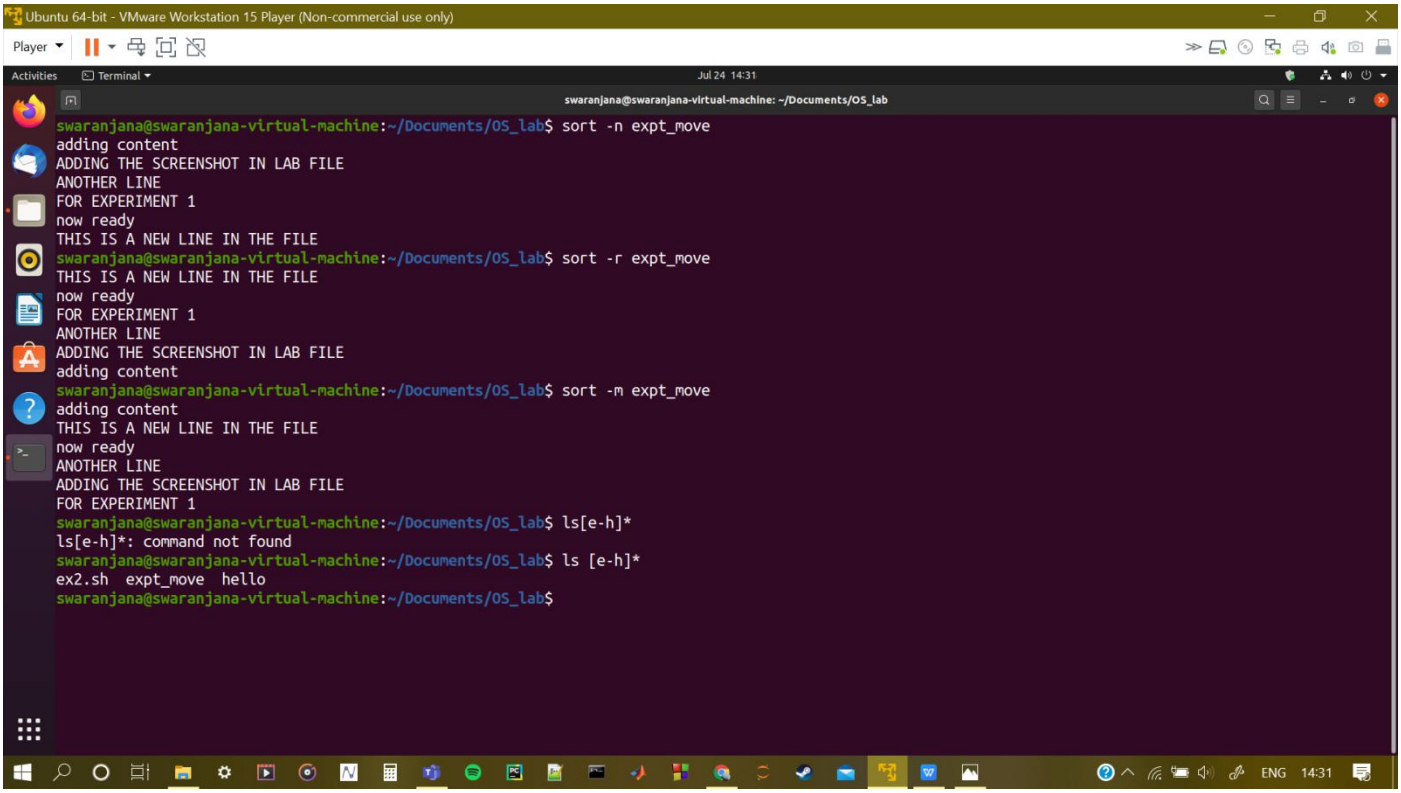
14. DISPLAY FILES IN A PATTERN

DESCRIPTION

It is used to display the file starting with a to m.

SYNTAX

$ls[a-m]*



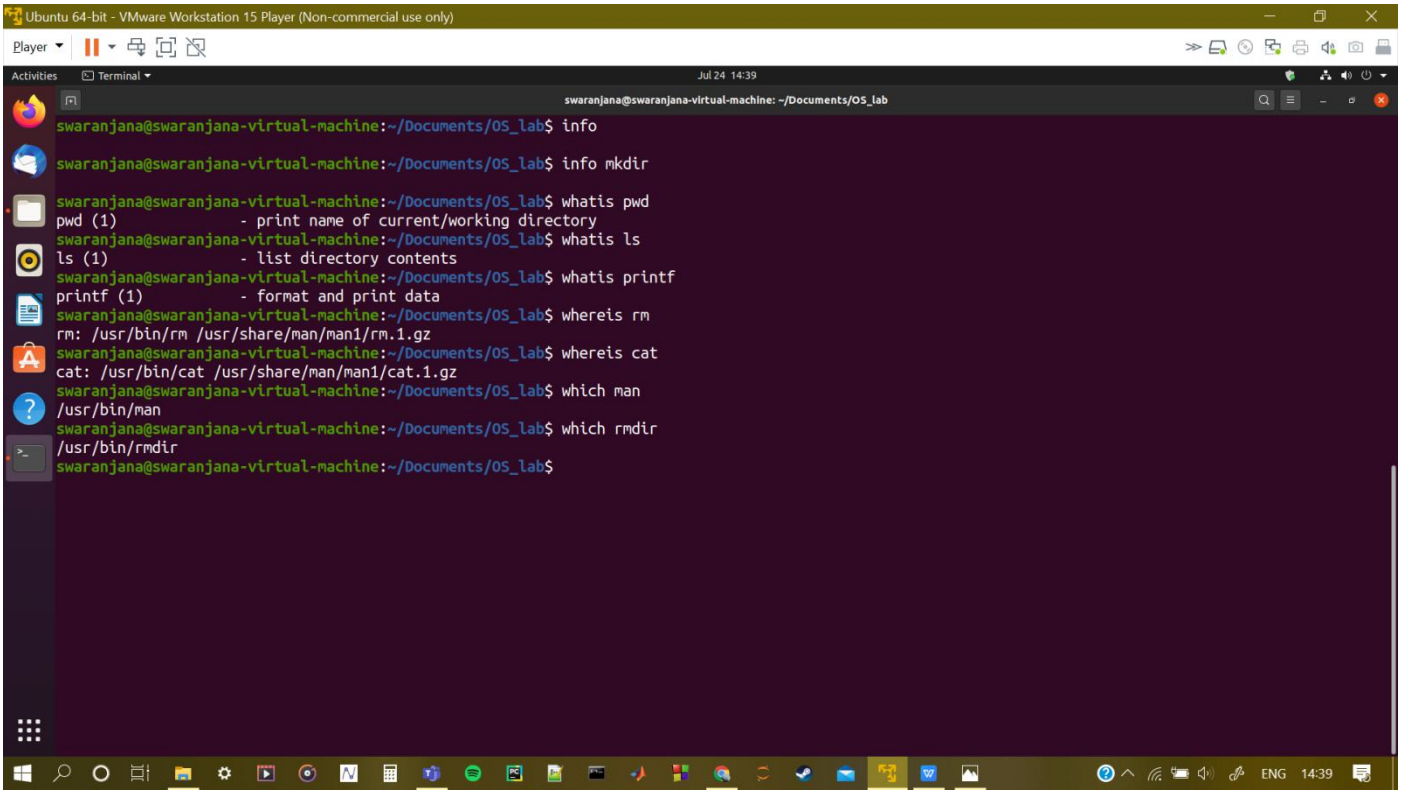
EXTRA COMMANDS

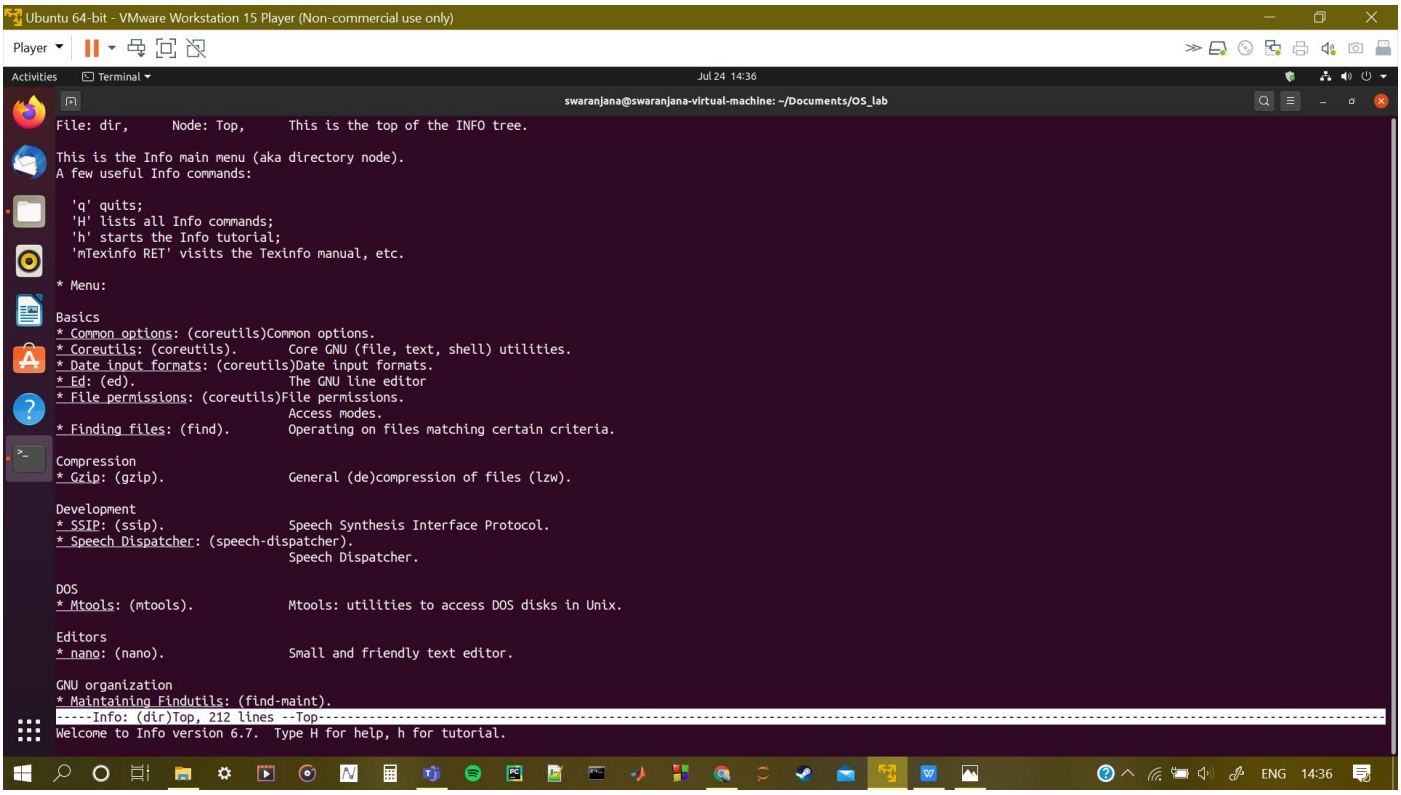
1. Command Name : **info**
   Description : read info documents
   Syntax : info (or) info commandName / functionName
   Example : info scanf

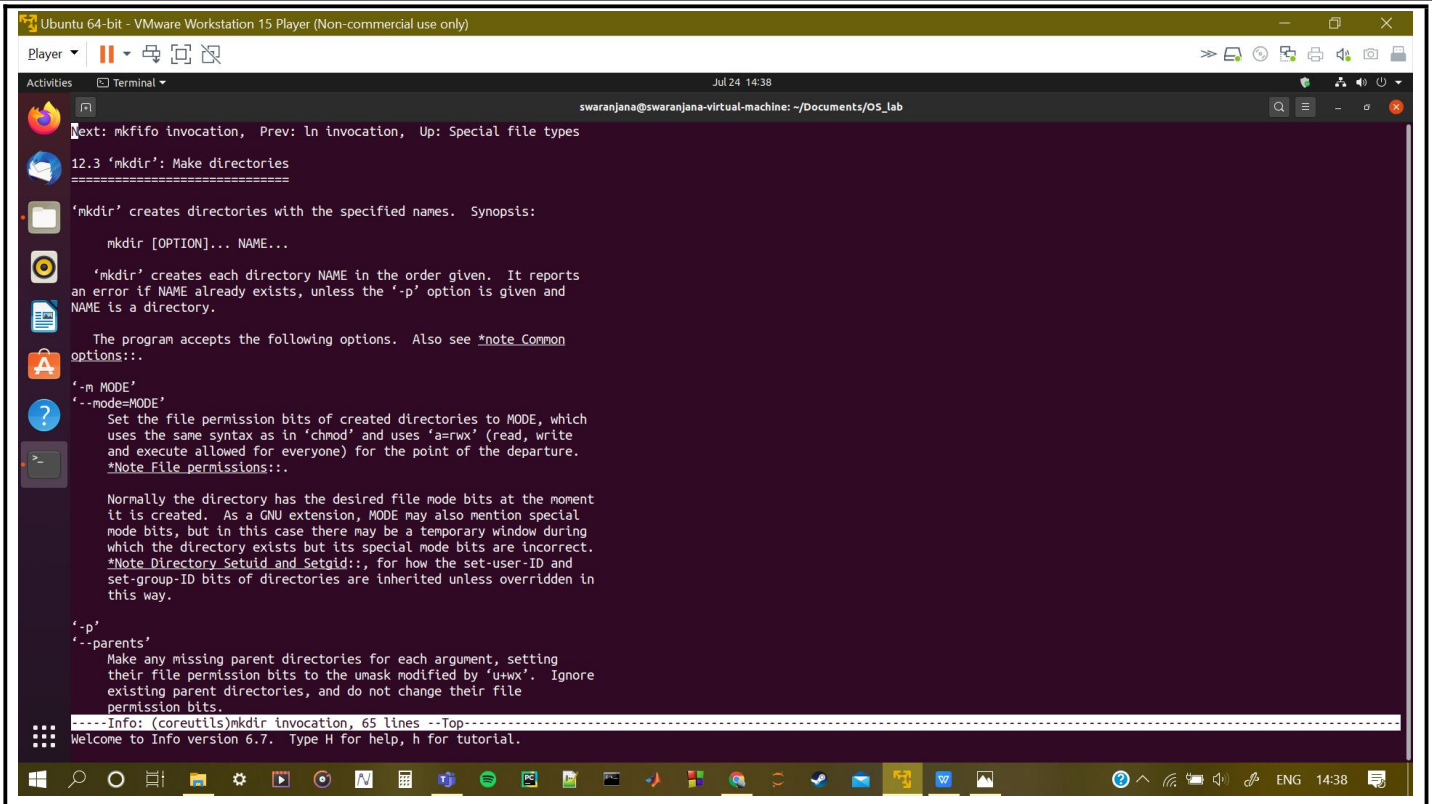
2. Command Name : **whatis**
   Description : search the whatis database for complete words
   Syntax : whatis commandName / functionName
   Example : whatis scanf
   (searches a set of database files containing short descriptions of system commands for keywords and displays the result on the standard output.)
   (scanf - input format conversion)

3. Command Name : **whereis**
   Description : locate the binary, source and manual page files for a command / function
   Syntax : whereis commandName
   Example : whereis fopen
   (fopen: /usr/share/man/man3/fopen.3.gz /usr/share/man/man3p/fopen.3p.gz)

4. Command Name : **which**
   Description : shows the full path of commands (OS commands)
   Syntax : which commandName1 [commandName2] ….
   Example : which man
   (/usr/bin/man)







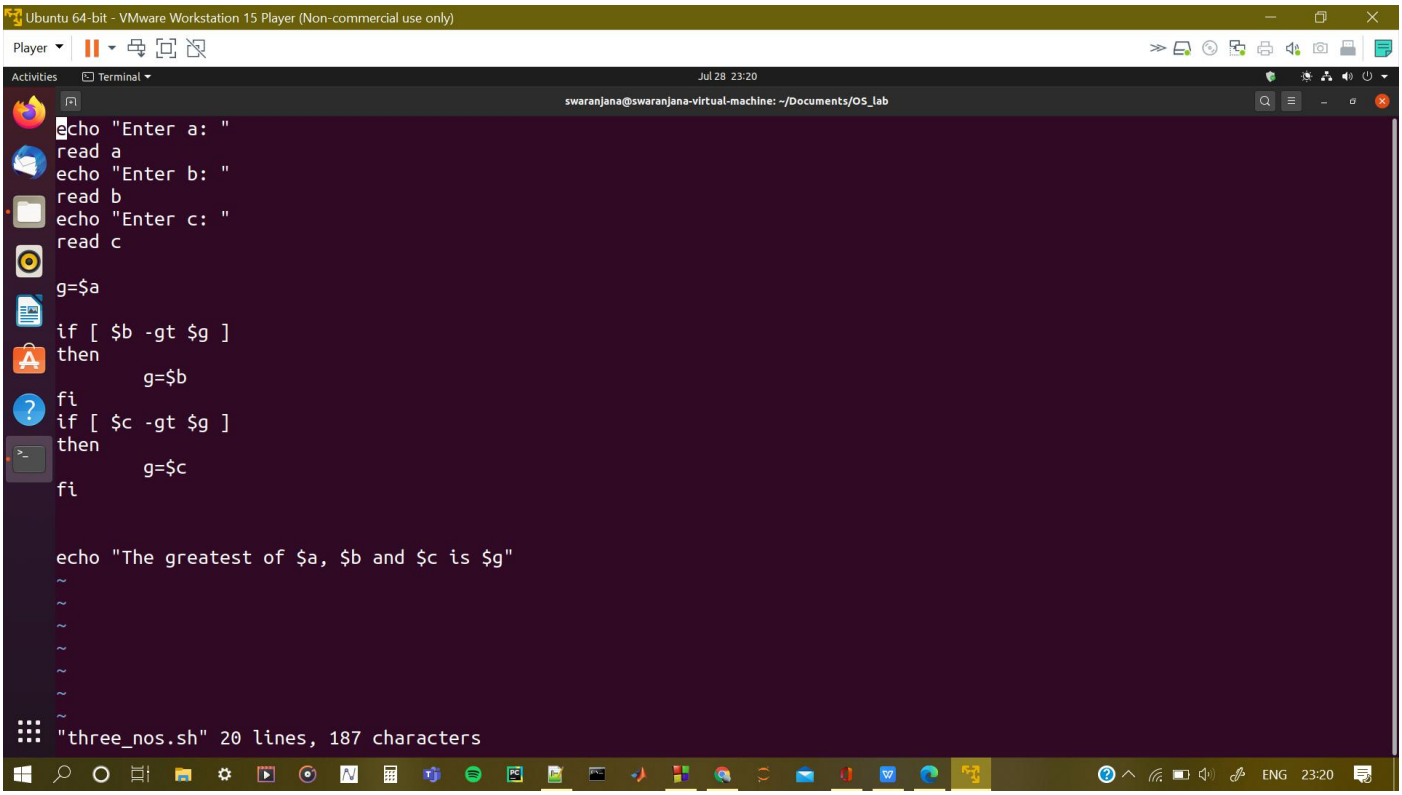
RESULT

Thus the basics of UNIX commands are studied and executed successfully.

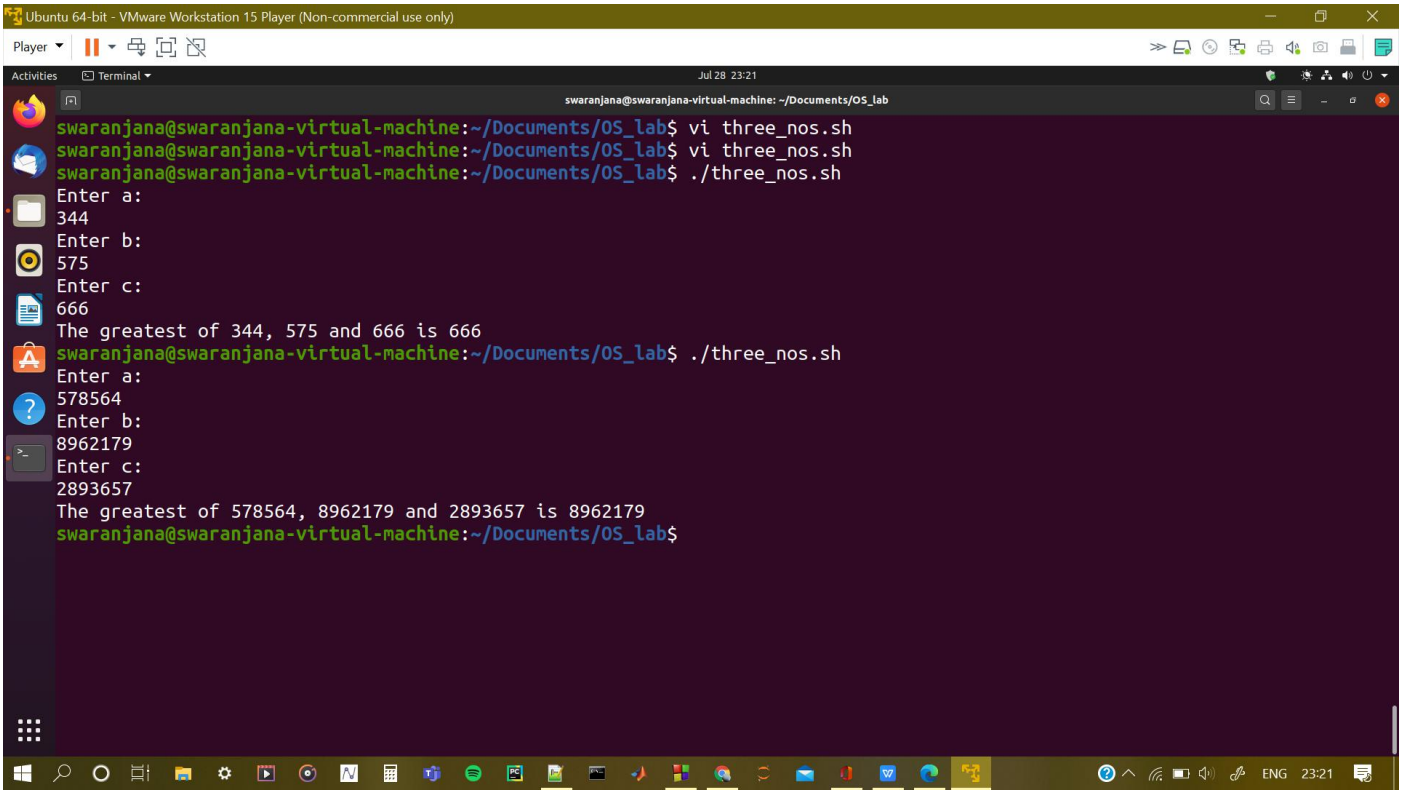
2A. Write a program to write the greatest of three numbers.

Code:

```
echo "Enter a: "
read a
echo "Enter b: "
read b
echo "Enter c: "
read c

g=$a

if [ $b -gt $g ]
then
    g=$b
fi
if [ $c -gt $g ]
then
    g=$c
fi


echo "The greatest of $a, $b and $c is $g"
```



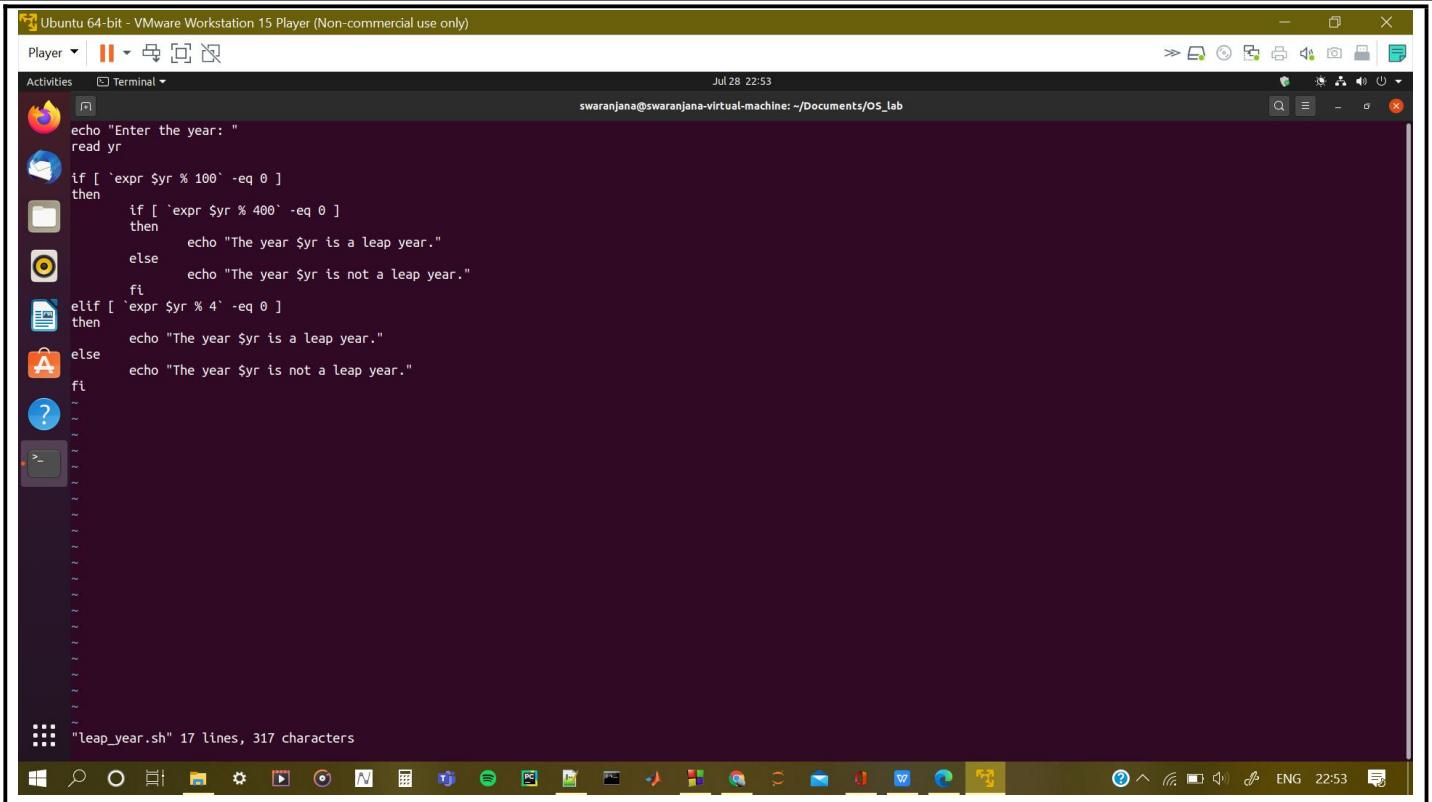
Output:



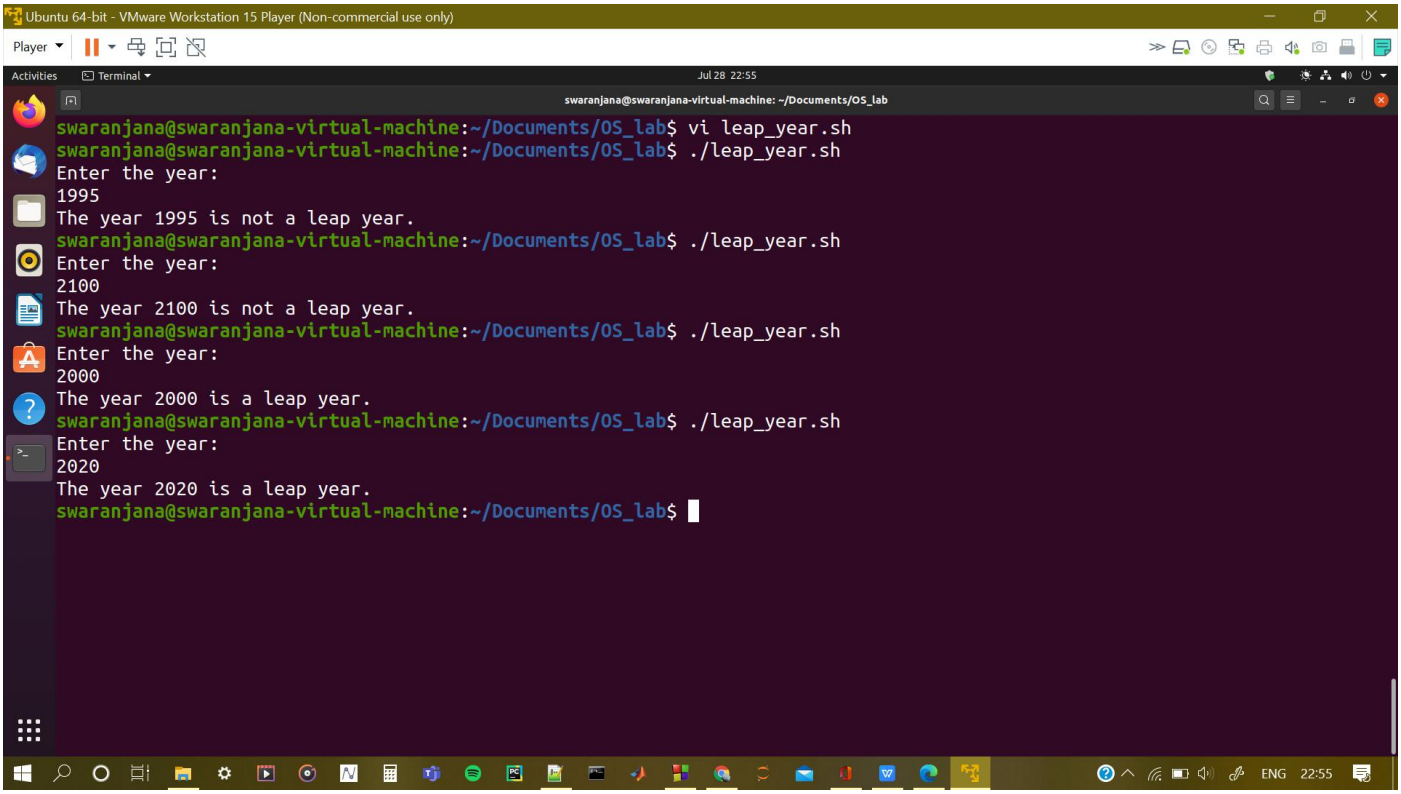
2B. Write a program to check whether a given year is a leap year or not.

Code:

```
echo "Enter the year: "
read yr


if [ `expr $yr % 100` -eq 0 ]
then
    if [ `expr $yr % 400` -eq 0 ]
    then
        echo "The year $yr is a leap year."
    else
        echo "The year $yr is not a leap year."
    fi
elif [ `expr $yr % 4` -eq 0 ]
then
    echo "The year $yr is a leap year."
else
    echo "The year $yr is not a leap year."
fi
```


Ubuntu 64-bit - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
Jul 28 22:53
swaranjana@swaranjana-virtual-machine: ~/Documents/OS_lab
echo "Enter the year: "
read yr
if [`expr $yr % 100` -eq 0]
then
if [`expr $yr % 400` -eq 0]
then
echo "The year $yr is a leap year."
else
echo "The year $yr is not a leap year."
fi
elif [`expr $yr % 4` -eq 0]
then
echo "The year $yr is a leap year."
else
echo "The year $yr is not a leap year."
fi
"leap_year.sh" 17 lines, 317 characters
ENG
22:53

Output:


Ubuntu 64-bit - VMware Workstation 15 Player (Non-commercial use only)
Player
Activities
Terminal
Jul 28 22:55
swaranjana@swaranjana-virtual-machine: ~/Documents/OS_lab
swaranjana@swaranjana-virtual-machine:~/Documents/OS_lab$ vi leap_year.sh
swaranjana@swaranjana-virtual-machine:~/Documents/OS_lab$ ./leap_year.sh
Enter the year:
1995
The year 1995 is not a leap year.
swaranjana@swaranjana-virtual-machine:~/Documents/OS_lab$ ./leap_year.sh
Enter the year:
2100
The year 2100 is not a leap year.
swaranjana@swaranjana-virtual-machine:~/Documents/OS_lab$ ./leap_year.sh
Enter the year:
2000
The year 2000 is a leap year.
swaranjana@swaranjana-virtual-machine:~/Documents/OS_lab$ ./leap_year.sh
Enter the year:
2020
The year 2020 is a leap year.
swaranjana@swaranjana-virtual-machine:~/Documents/OS_lab$
ENG
22:55