Assignment 2

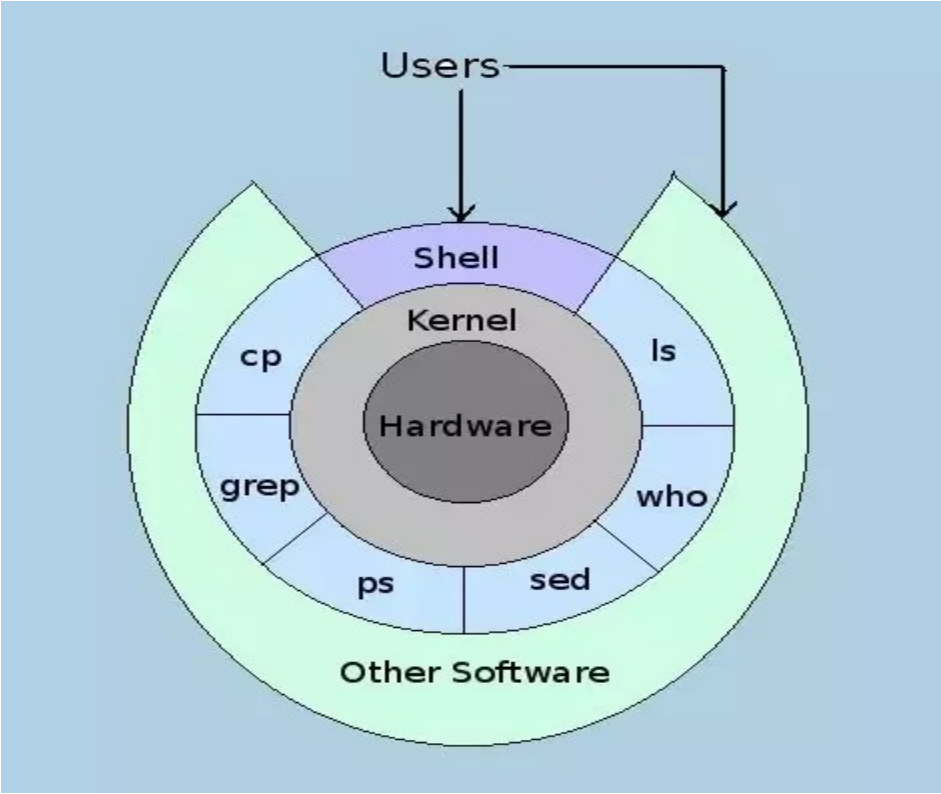
1) AIM: Creating Address Book using Shell Script

2) THEORY:

-) Shell

A Shell provides you with an interface to the Unix system. It gathers input from you and executes programs based on that input. When a program finishes executing, it displays that program's output.

Shell is an environment in which we can run our commands, programs, and shell scripts. There are different flavors of a shell, just as there are different flavors of operating systems. Each flavor of shell has its own set of recognized commands and functions.



Types of Shell:

- **The C Shell –** Bill Joy created it at the University of California at Berkeley. It incorporated features such as aliases and command history. It includes helpful programming features like built-in arithmetic and C-like expression syntax.
- **The Bourne Shell –**

It was written by Steve Bourne at AT&T Bell Labs. It is the original UNIX shell. It is faster and more preferred. It lacks features for interactive use like the ability to recall previous commands. It also lacks built-in arithmetic and logical expression handling. It is default shell for Solaris OS.

•The Korn Shell

It Was written by David Korn at AT&T Bell LabsIt is a superset of the Bourne shell.So it supports everything in the Bourne shell.It has interactive features. It includes features like built-in arithmetic and C-like arrays, functions, and string-manipulation facilities.It is faster than C shell. It is compatible with script written for C shell.

•GNU Bourne-Again Shell –

It is compatible to the Bourne shell. It includes features from Korn and Bourne shell.

For the GNU Bourne-Again shell the:

Shell Scripts

The basic concept of a shell script is a list of commands, which are listed in the order of execution. A good shell script will have comments, preceded by # sign, describing the steps.

There are conditional tests, such as value A is greater than value B, loops allowing us to go through massive amounts of data, files to read and store data, and variables to read and store data, and the script may include functions.

We are going to write many scripts in the next sections. It would be a simple text file in which we would put all our commands and several other required constructs that tell the shell environment what to do and when to do it.

Shell scripts and functions are both interpreted. This means they are not compiled.

Example Script

Assume we create a **test.sh** script. Note all the scripts would have the **.sh** extension. Before you add anything else to your script, you need to alert the system that a shell script is being started. This is done using the **shebang** construct. For example −

```
#!/bin/sh
```

This tells the system that the commands that follow are to be executed by the Bourne shell. *It's called a shebang because the # symbol is called a hash, and the ! symbol is called a bang*.

To create a script containing these commands, you put the shebang line first and then add the commands −

```
#!/bin/bash
pwd
ls
```

3)INPUT

1. start.sh

```
#!/bin/sh
HEIGHT=15
WIDTH=40
CHOICE_HEIGHT=6
BACKTITLE="Address Book"
TITLE="Address Book"
MENU="Select Options"
OPTIONS=(1 "Add"
        2 "List"
        3 "Find"
        4 "Delete"
        5 "Update"
        6 "Exit")
CHOICE=$(dialog --clear \ --backtitle "$BACKTITLE" \--title "$TITLE" \ --menu "$MENU" \
$HEIGHT $WIDTH $CHOICE_HEIGHT \ "${OPTIONS[@]}" \ 2>&1 >/dev/tty)
clear
BOOK="address-book.txt"
          echo "What operation do you want?"
          echo -e "add, list, find, del, update, exit: "
case $CHOICE in
       1)[ "$answer" = "add" ]
             ./add.sh
```

```
        ;;
    2) [ "$answer" = "list" ]
          ./list.sh
       ;;
    3)[ "$answer" = "find" ]
        ./find.sh
       ;;
    4)  "$answer" = "del" ]
          ./del.sh
          ;;
    5) [ "$answer" = "update" ]
          ./update.sh
          ;;
     6) [ "$answer" = "exit" ]
       ;;
esac
```

2.add.sh

```
#!/bin/sh
    # Name of address book
    BOOK="address-book.txt"

    # Ask the user for a name and assign to a variable
    echo -n "Enter Name of person: "
    read name

    # Ask the user for a phone number and assign to a variable
    echo -n "Enter Phone number: "
    read phone

    # Echo the answers and ask for confirmation
    echo "Do you want to enter values in address book ?"

    #  echo "Should I enter the values:"
    echo -e "$name ; $phone \n"
    echo -n "y/n: "
    read answer

    if [ "$answer" == "y" ]
    then
       # Write the values to the address book
       echo "$name ; $phone" >>$BOOK
       dialog --msgbox "Record Added Successfully...." 0 0
    clear
    else
       # Give the user a message
       echo "$name ; $phone NOT written to $BOOK"
```

```
    dialog --msgbox "Record Discarded...." 0 0
clear
fi
exit 0
```

3.find.sh

```
#!/bin/sh

    BOOK="address-book.txt"

    # Ask the user what to look for.
    echo -n "Enter person name to scerch :  "
    read find

    # Print the header before the answer
    echo "Name ; Phone number"
    grep -i $find $BOOK
```

4.list.sh

```
 #!/bin/sh

    BOOK="address-book.txt"
    # Display the format before the entries
    dialog --infobox "Wait for few seconds" 0 0
    sleep 2
    clear
    # Print the book with line numbers and paused with less
    nl --number-separator=":    " $BOOK | less
```

5.update.sh

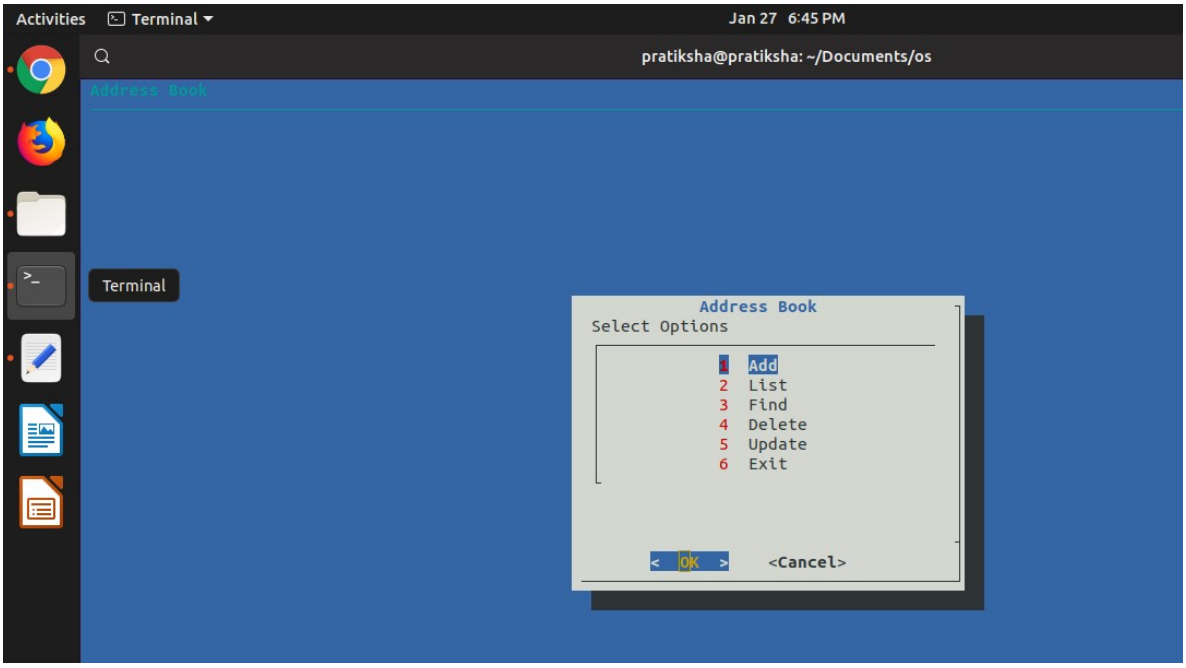
```
 #!/bin/sh

BOOK="address-book.txt"
    echo -n "Enter name of person to modify: "
    read find

    # Print the header before the answer
    if  grep -qF "$find" address-book.txt; then
       dialog --msgbox "Record Found" 0 0
       clear
    else
       dialog --msgbox "Record not found" 0 0
       clear
    fi

    echo "Name ; Phone number"
```

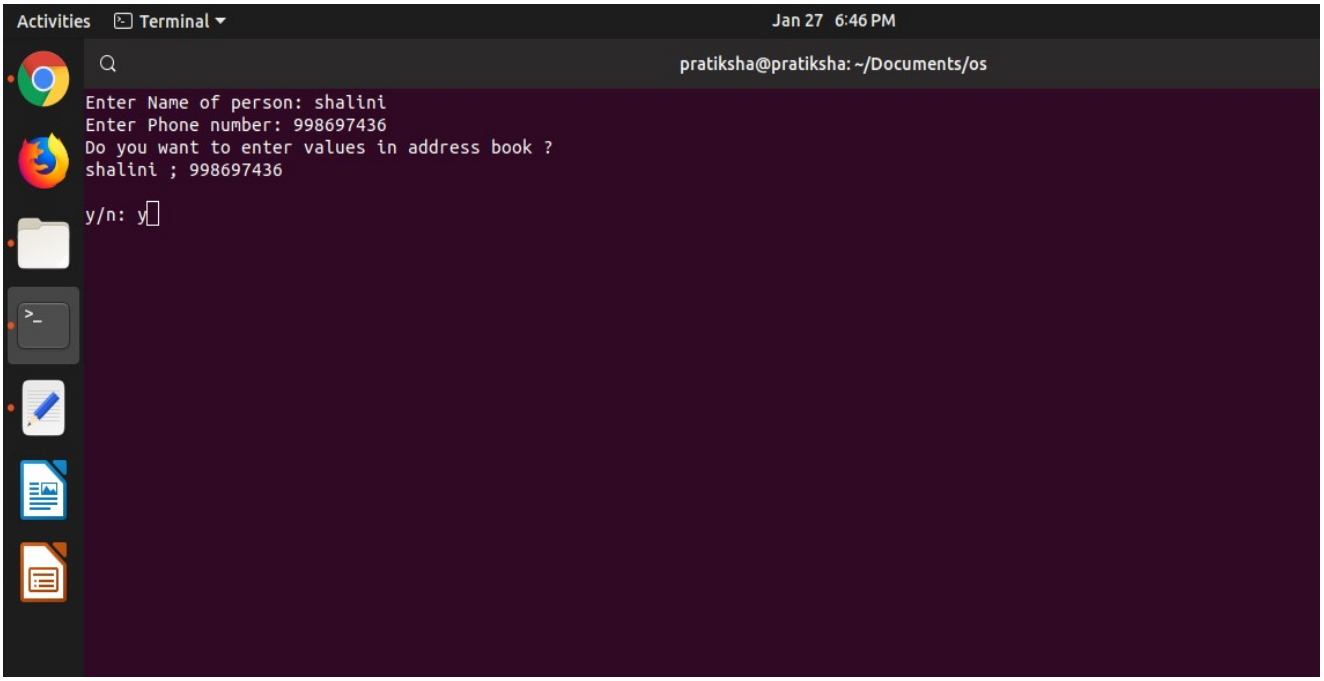
```
grep -i $find $BOOK

echo -n "Enter New name: "
read new_name

sed -i "s/$find/$new_name/g" address-book.txt
echo "Line Number   Name     Phone Number"
# Print the book with line numbers and paused with less
nl --number-separator="    " $BOOK
```

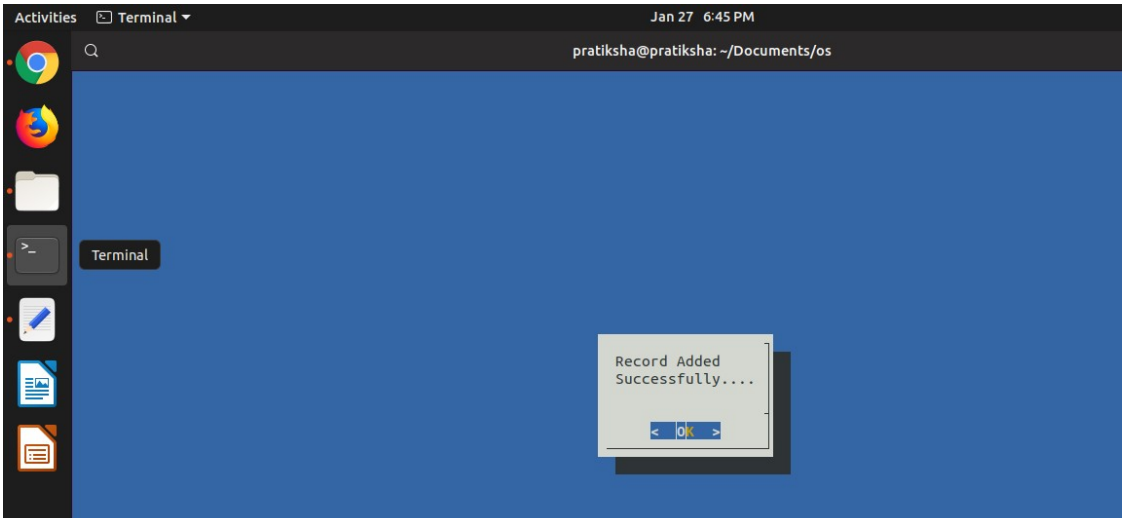
6.del.sh

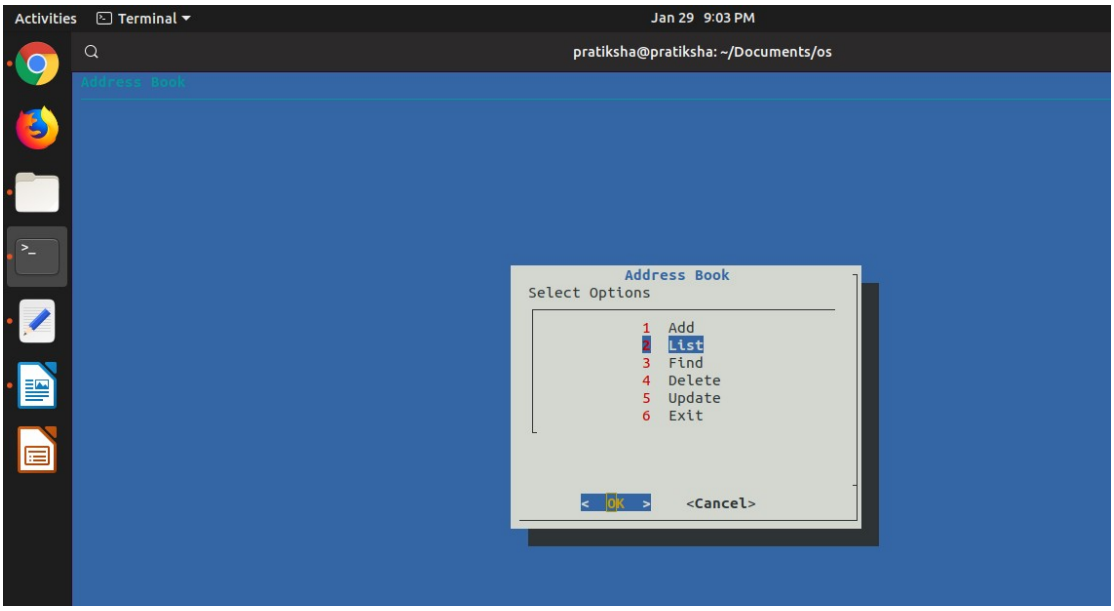
```
#!/bin/sh

BOOK="address-book.txt"
# Ask the user which line to delete
echo -n "Enter line number to be delete: "

read number
# Rename the file before deleting
mv $BOOK boo.txt

# Add line numbers and delete against that number
nl --number-separator=":" boo.txt | grep -v $number: | awk -F: '{print $2}' |  tee $BOOK

dialog --msgbox "Record deleted..." 0 0
clear
```

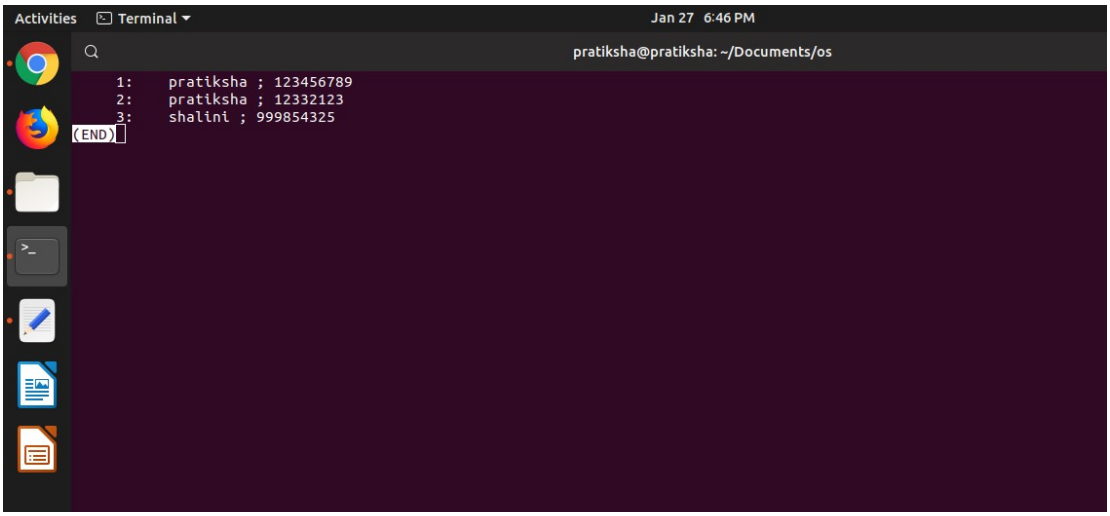
4)OUTPUT

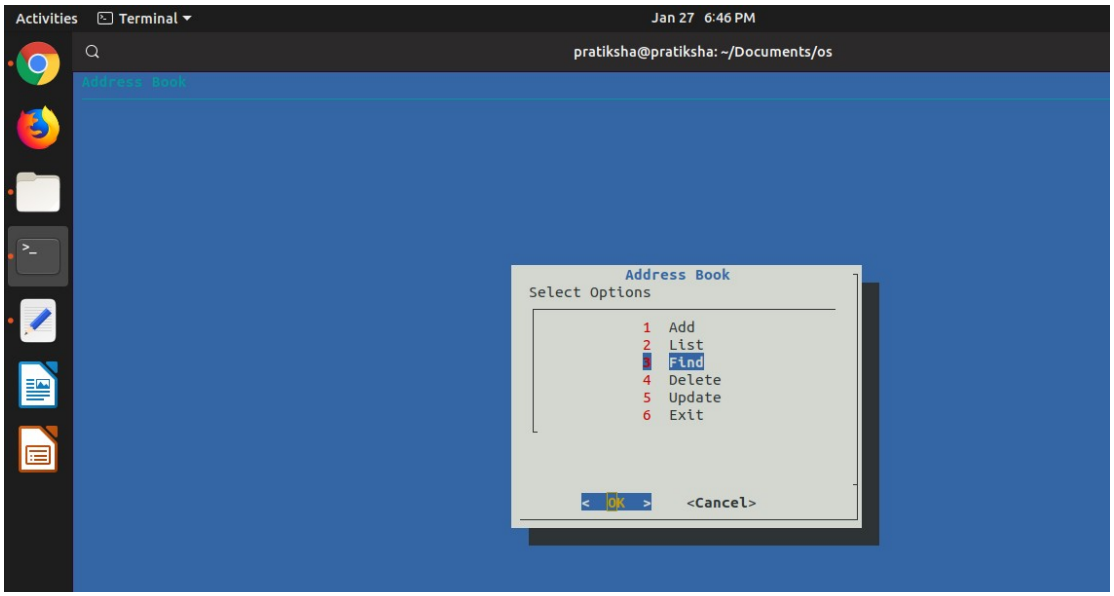

Activities
Terminal
Jan 27 6:45 PM
pratiksha@pratiksha: ~/Documents/os
Address Book
Terminal
Address Book
Select Options
1 Add
2 List
3 Find
4 Delete
5 Update
6 Exit
< OK >
<Cancel>

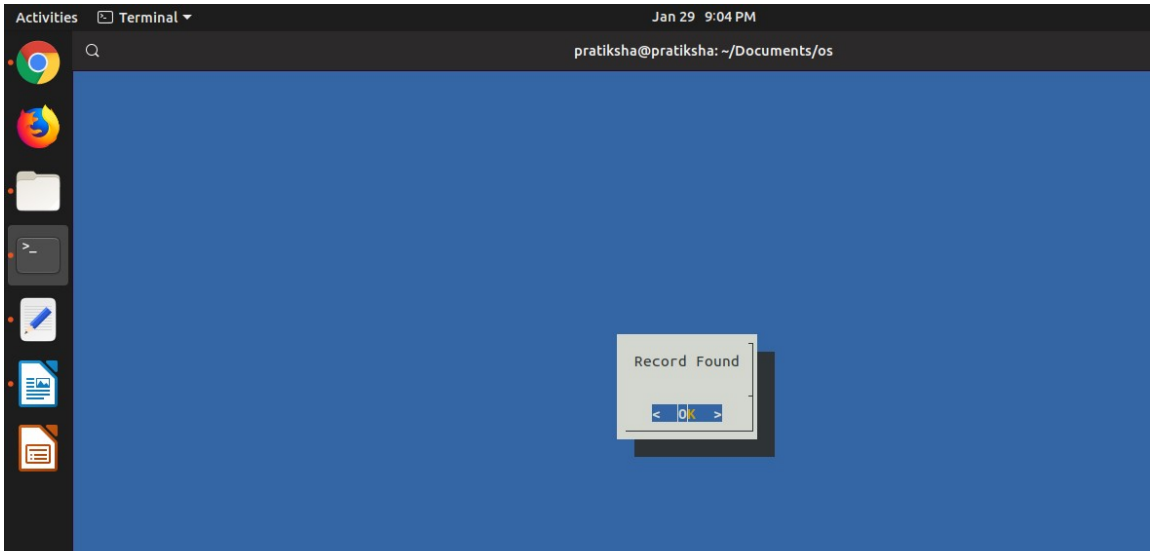

Activities
Terminal
Jan 27 6:46 PM
pratiksha@pratiksha: ~/Documents/os
Enter Name of person: shalini
Enter Phone number: 998697436
Do you want to enter values in address book ?
shalini ; 998697436
y/n: y

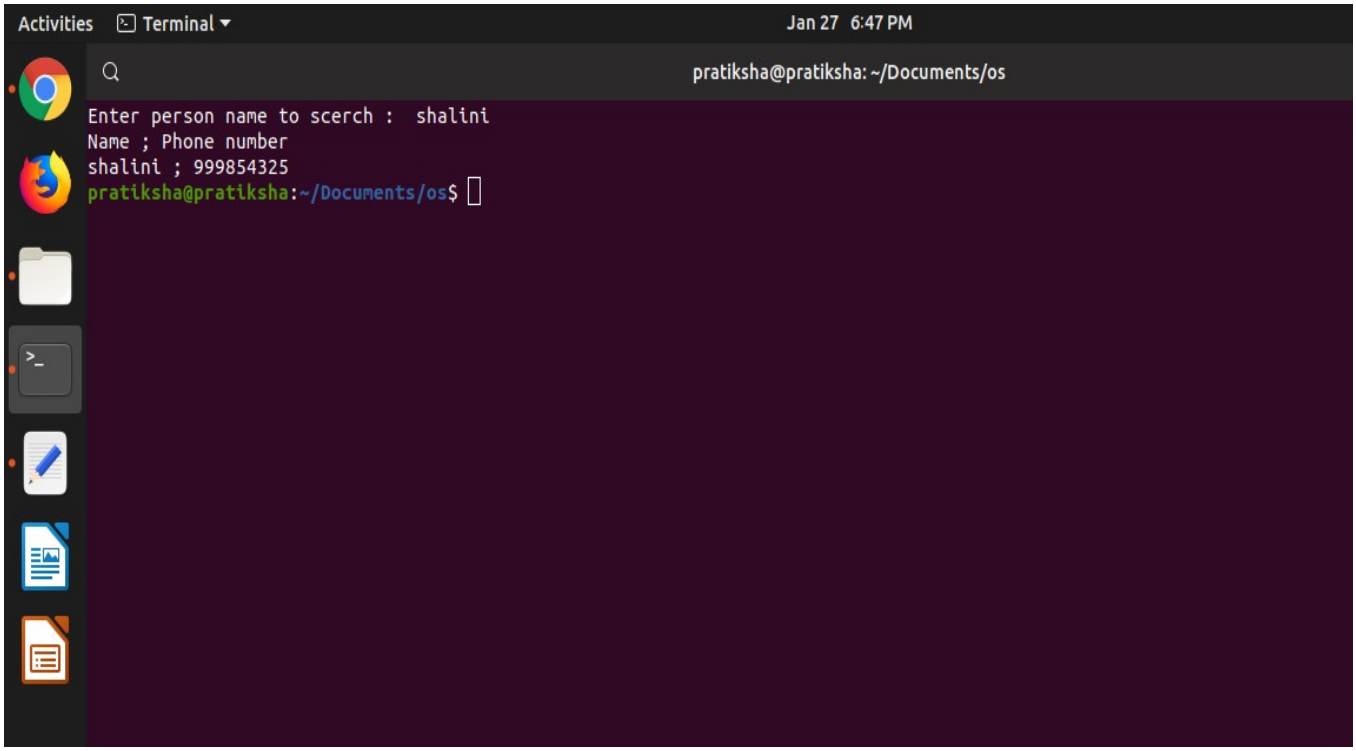

Activities
Terminal
Jan 27 6:45 PM
pratiksha@pratiksha: ~/Documents/os
Terminal
Record Added
Successfully....
< OK >

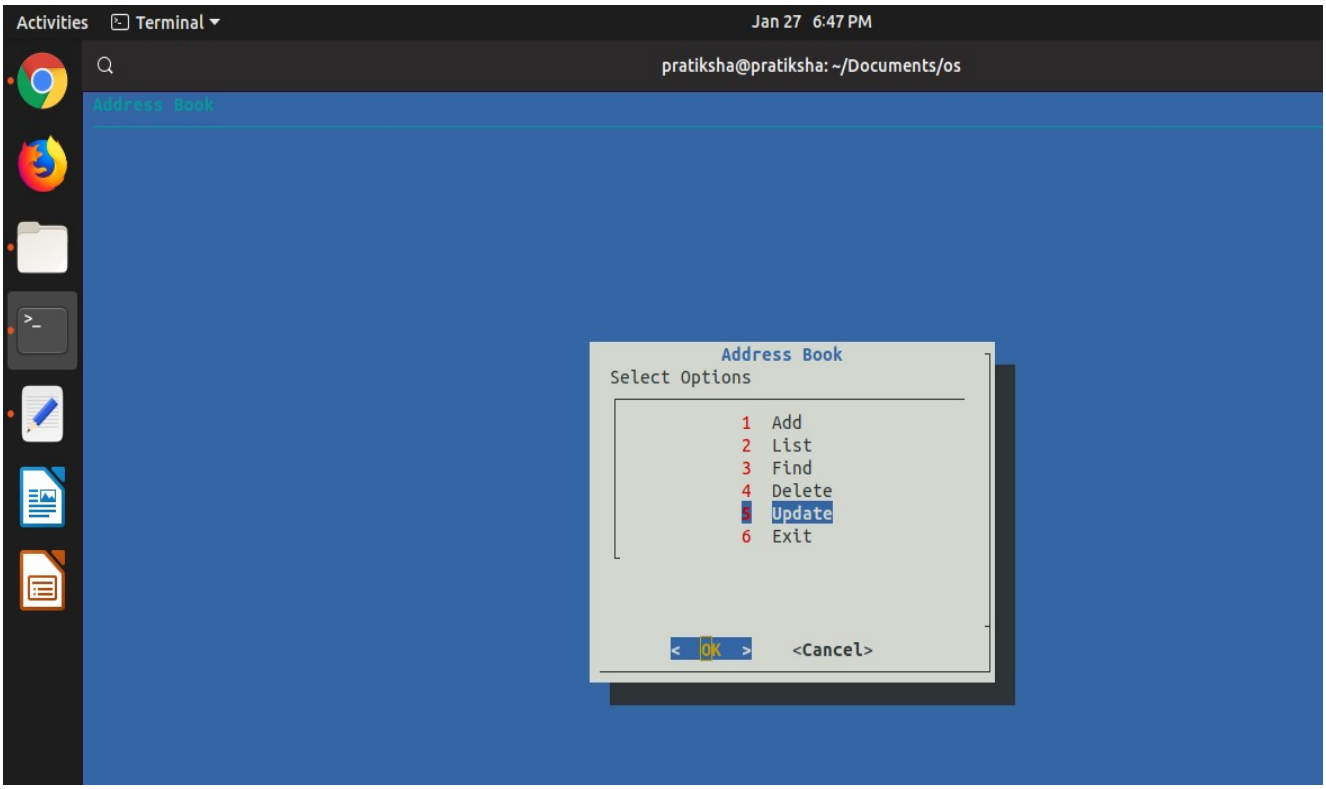

Activities
Terminal
Jan 29 9:03 PM
pratiksha@pratiksha: ~/Documents/os
Address Book
Address Book
Select Options
1 Add
2 List
3 Find
4 Delete
5 Update
6 Exit
< OK >
<Cancel>

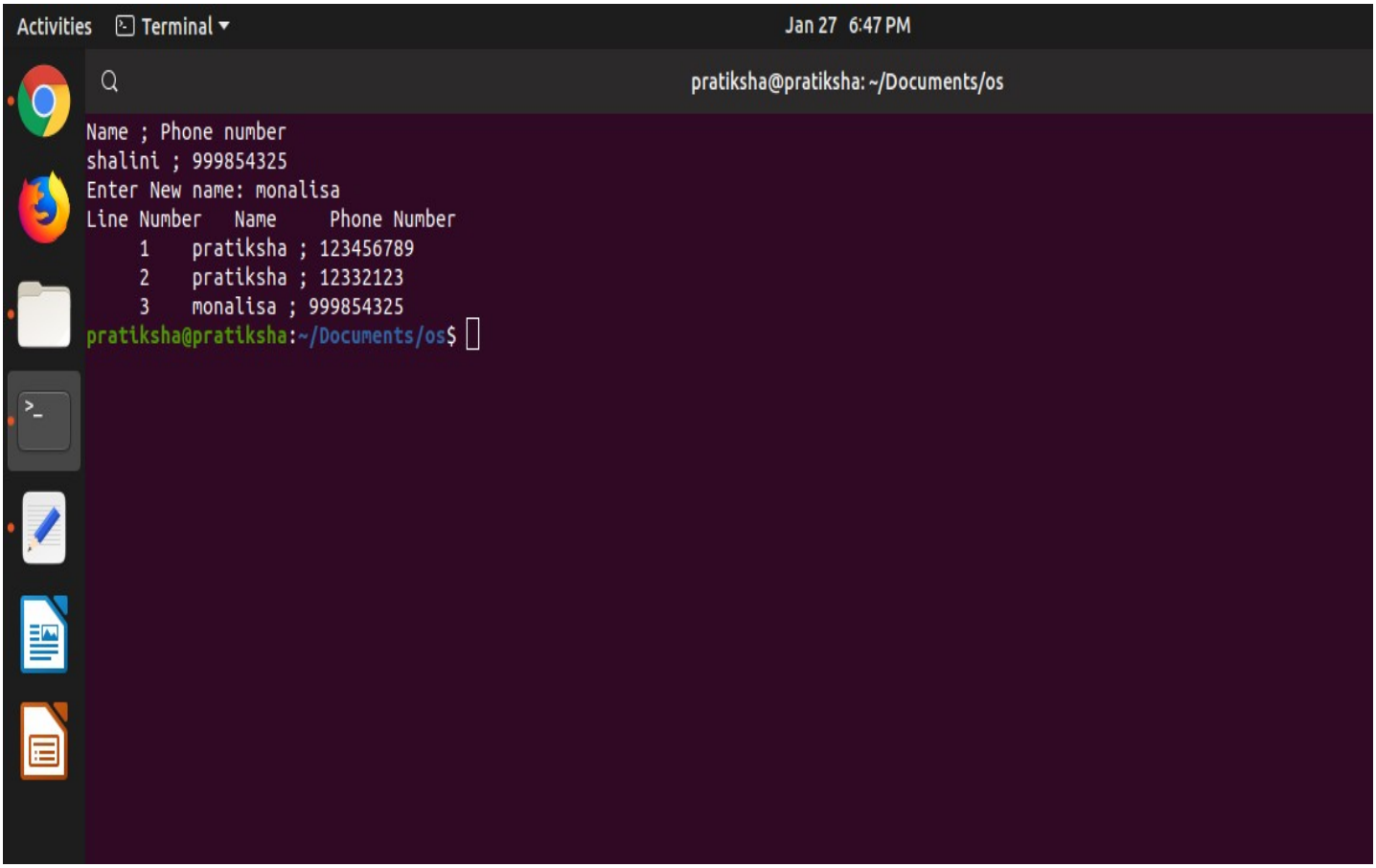

Activities
Terminal
Jan 27 6:46 PM
pratiksha@pratiksha: ~/Documents/os
1: pratiksha ; 123456789
2: pratiksha ; 12332123
3: shalini ; 999854325
(END)

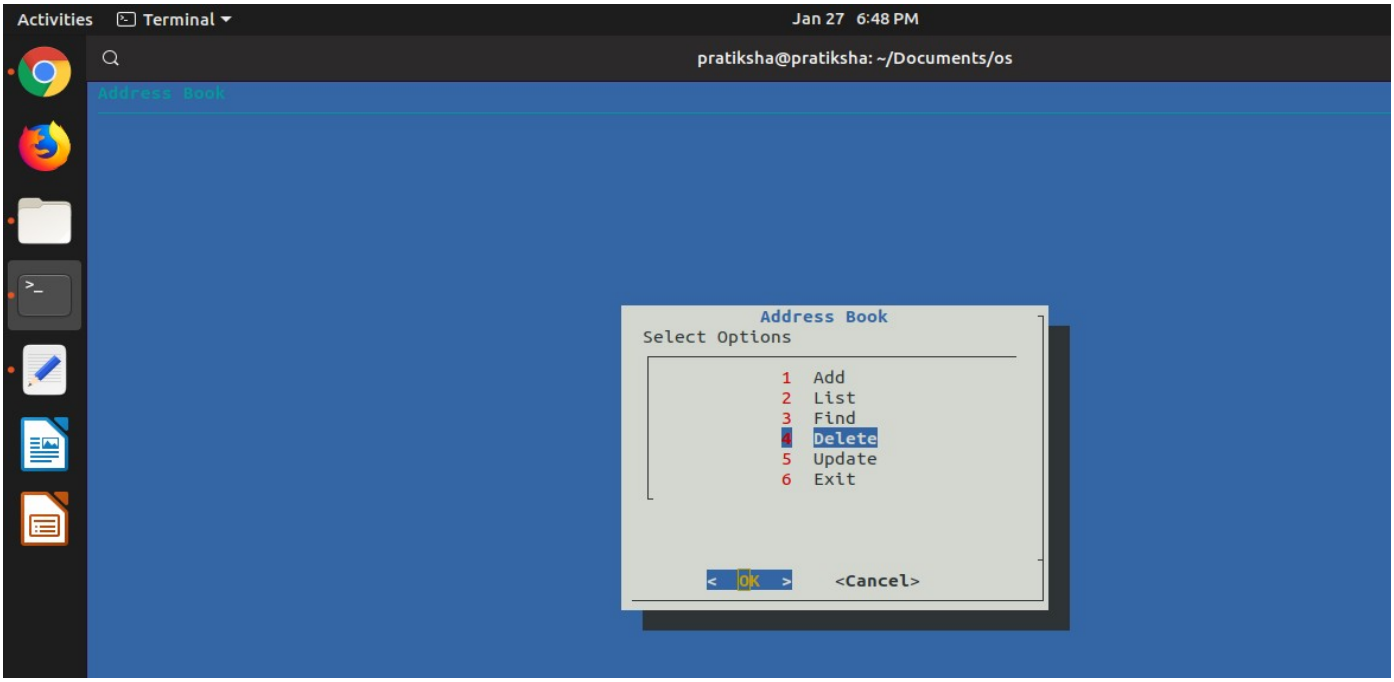

Activities
Terminal
Jan 27 6:46 PM
pratiksha@pratiksha: ~/Documents/os
Address Book
Address Book
Select Options
1 Add
2 List
3 Find
4 Delete
5 Update
6 Exit
< Ok >
<Cancel>

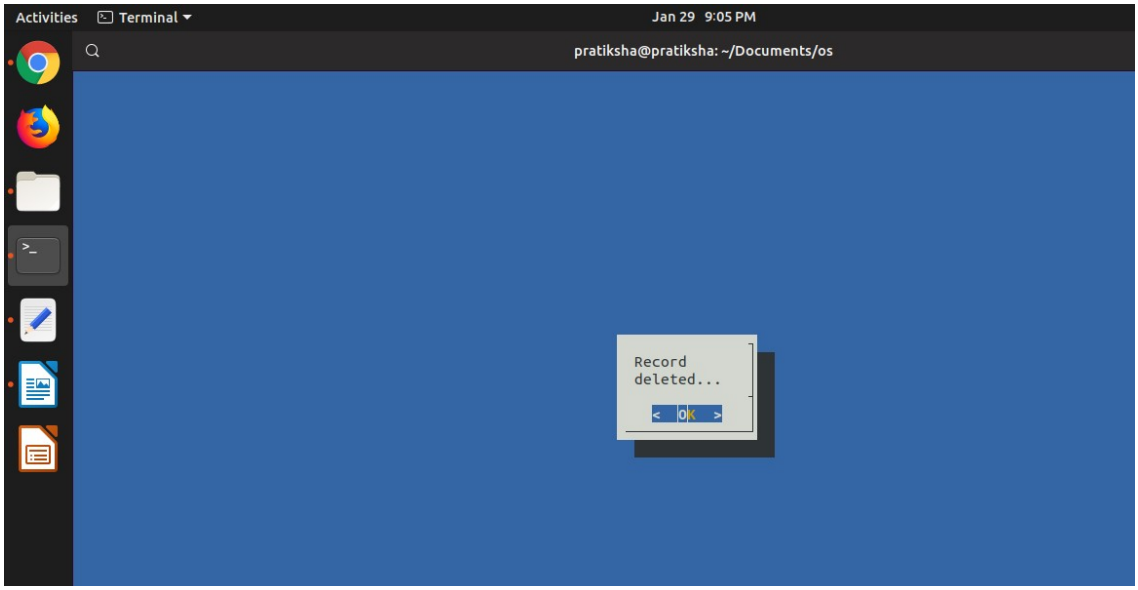

Activities
Terminal
Jan 29 9:04 PM
pratiksha@pratiksha: ~/Documents/os
Record Found
< OK >

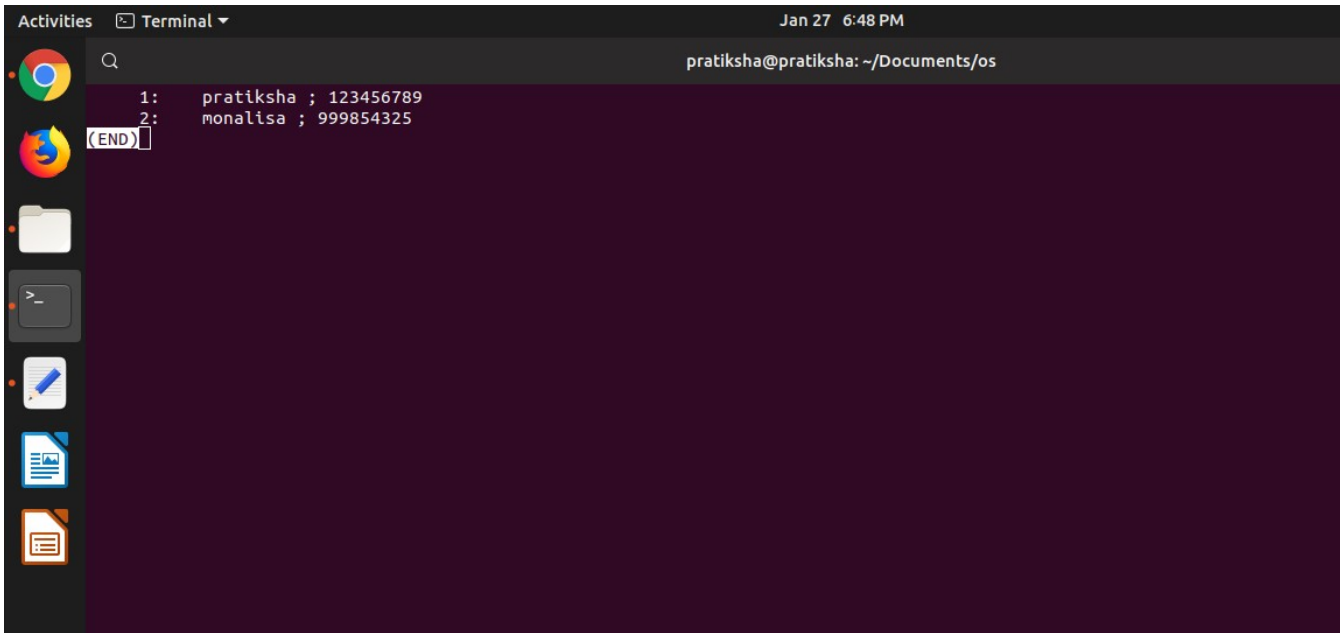

Activities
Terminal
Jan 27 6:47 PM
pratiksha@pratiksha: ~/Documents/os
Enter person name to scerch : shalini
Name ; Phone number
shalini ; 999854325
pratiksha@pratiksha:~/Documents/os$


Activities
Terminal
Jan 27 6:47 PM
pratiksha@pratiksha: ~/Documents/os
Address Book
Address Book
Select Options
1 Add
2 List
3 Find
4 Delete
5 Update
6 Exit
< OK >
<Cancel>


Activities
Terminal
Jan 27 6:47 PM
pratiksha@pratiksha: ~/Documents/os
Name ; Phone number
shalini ; 999854325
Enter New name: monalisa
Line Number Name Phone Number
1 pratiksha ; 123456789
2 pratiksha ; 12332123
3 monalisa ; 999854325
pratiksha@pratiksha:~/Documents/os$


Activities
Terminal
Jan 27 6:48 PM
pratiksha@pratiksha: ~/Documents/os
Address Book
Address Book
Select Options
1 Add
2 List
3 Find
4 Delete
5 Update
6 Exit
< OK >
<Cancel>


Activities
Terminal
Jan 29 9:05 PM
pratiksha@pratiksha: ~/Documents/os
Record
deleted...
< OK >


Activities
Terminal
Jan 27 6:48 PM
pratiksha@pratiksha: ~/Documents/os
1: pratiksha ; 123456789
2: monalisa ; 999854325
(END)