ANOM DEVGUN

GR:21810017, ROLL NO:13

COMP SY A1

Assignment 2: Address Book Using GUI and Shell Script.

AIM: Write a shell script program to create an address book

**What Is a Shell?**

Before jumping in and discussing how to program using a shell, let’s review the shell’s function and the

different shells available for Linux. A *shell* is a program that acts as the interface between you and the Linux

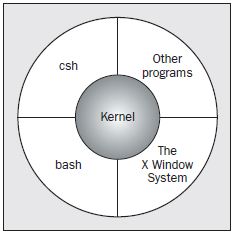
system, enabling you to enter commands for the operating system to execute. In that respect, it resembles the

Windows command prompt, but as mentioned earlier, Linux shells are much more powerful. For example,

input and output can be redirected using < and >, data piped between simultaneously executing programs

using |, and output from a subprocess grabbed by using $(...). On Linux it’s quite feasible to have multiple

shells installed, with different users able to pick the one they prefer.



Because Linux is so modular, you can slot in one of the many different shells in use, although most of

them are derived from the original Bourne shell. On Linux, the standard shell that is always installed as

/bin/sh is called *bash* (the GNU Bourne-Again SHell), from the GNU suite of tools. Because this is an

excellent shell that is always installed on Linux systems, is open source, and is portable to almost all

UNIX variants, bash is the shell we will be using. This chapter uses bash version 3 and mostly uses the

features common to all POSIX-compatible shells. We assume that the shell has been installed as /bin/sh

and that it is the default shell for your login. On most Linux distributions, the program /bin/sh, the

default shell, is actually a link to the program /bin/bash.

**WHAT Is a GUI?**

A **GUI** (graphical user interface) is a system of interactive visual components for computer [software](https://www.computerhope.com/jargon/s/software.htm). A GUI displays objects that convey information, and represent actions that can be taken by the user. The objects change color, size, or visibility when the user interacts with them.

GUI objects include [icons](https://www.computerhope.com/jargon/i/icon.htm), [cursors](https://www.computerhope.com/jargon/c/cursor.htm), and [buttons](https://www.computerhope.com/jargon/b/button.htm). These graphical elements are sometimes enhanced with sounds, or visual effects like [transparency](https://www.computerhope.com/jargon/t/transpar.htm) and [drop shadows](https://www.computerhope.com/jargon/d/dropshad.htm).

A GUI is considered to be more [user-friendly](https://www.computerhope.com/jargon/u/userfrie.htm) than a text-based [command-line interface](https://www.computerhope.com/jargon/c/commandi.htm), such as [MS-DOS](https://www.computerhope.com/jargon/m/msdos.htm), or the [shell](https://www.computerhope.com/jargon/s/shell.htm) of [Unix-like](https://www.computerhope.com/jargon/u/unix-like.htm) operating systems.

The GUI was first developed at [Xerox PARC](https://www.computerhope.com/jargon/x/xparc.htm) by [Alan Kay](https://www.computerhope.com/people/alan_kay.htm), [Douglas Engelbart](https://www.computerhope.com/people/douglas_engelbart.htm), and a group of other researchers in [1981](https://www.computerhope.com/history/1981.htm). Later, [Apple](https://www.computerhope.com/comp/apple.htm) introduced the [Lisa computer](https://www.computerhope.com/jargon/l/lisa-computer.htm) with a GUI on January 19, [1983](https://www.computerhope.com/history/1983.htm).

ADD: #!/bin/sh

zenity --entry --title="ADDITION" --text="Name:" >temp.txt; #enter data

name=$(cat temp.txt)

zenity --entry --title="ADDITION" --text="City:" >temp.txt;

city=$(cat temp.txt)

zenity --entry --title="ADDITION" --text="State" >temp.txt;

state=$(cat temp.txt)

echo " $name $city $state " >> address.txt #append data to eof address.txt

zenity --info --title="DONE" --text="DATA ADDED SUCCESSFULLY"

exit 0

DELETE:

#!/bin/sh

zenity --entry --title="Delete Record" --text="Enter Name of Record Holder:" >temp.txt; #input name

t=$(cat temp.txt) #read temp.txt

grep -vE $t address.txt > temp.txt #leaving aside search element copy everything else from address.txt to temp.txt

cp temp.txt address.txt #rename temp.txt to address.txt and override

zenity --info --title="Done!" --text="Deleted Successfully"

exit 0

FIND:

#!/bin/sh

zenity --entry --title=" SEARCH" --text="Enter Name Of Record Holder: " >temp.txt; #Input Search Data

fin=$(cat temp.txt)

grep "$fin" address.txt > temp.txt #Copy search element data to temp

f=$(cat temp.txt)

if [ $f = '']

then

zenity --info --title=" 404Lol " --text=" Record not Found. "

else

zenity --list --title=" FOUND " --column "Name" --column "City" --column " State " $f --height=300 --width=350 #Display Content of Temp

fi

exit 0

LIST:

#!/bin/sh

a=$(cat address.txt)

zenity --list --title="RECORD DISPLAY" --column "Name" --column "City" --column " State "$a --height=300 --width=350 #Display all content of address.txt

exit 0

