**ASSIGNMENT No. – 2**

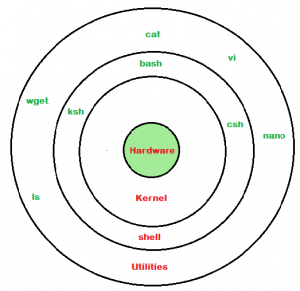
AIM - Creating Address Book using Shel Script .

OBJECTIVE – Write a program to implement an address book with the options given below .

1. Create address book
2. View address book
3. Insert Record
4. Delete a record
5. Modify record
6. Exit .

THEORY –

A shell is special user program which provide an interface to user to use operating system services. Shell accept human readable commands from user and convert them into something which kernel can understand. It is a command language interpreter that execute commands read from input devices such as keyboards or from files. The shell gets started when the user logs in or start the terminal.



Shell is broadly classified into two categories –

* Command Line Shell
* Graphical shell

**Command Line Shell**

Shell can be accessed by user using a command line interface. A special program called **Terminal** in linux/macOS or **Command Prompt** in Windows OS is provided to type in the human readable commands such as “cat”, “ls” etc. and then it is being execute. The result is then displayed on the terminal to the user.

**Graphical Shells**

Graphical shells provide means for manipulating programs based on graphical user interface (GUI), by allowing for operations such as opening, closing, moving and resizing windows, as well as switching focus between windows. Window OS or Ubuntu OS can be considered as good example which provide GUI to user for interacting with program. User do not need to type in command for every actions.A typical GUI in Ubuntu system –

GUI shell

There are several shells are available for Linux systems like –

* [BASH (Bourne Again SHell)](https://en.wikipedia.org/wiki/Bash_(Unix_shell)) – It is most widely used shell in Linux systems. It is used as default login shell in Linux systems and in macOS. It can also be installed on Windows OS.
* [CSH (C SHell)](https://en.wikipedia.org/wiki/C_shell) – The C shell’s syntax and usage are very similar to the C programming language.
* [KSH (Korn SHell)](https://en.wikipedia.org/wiki/Korn_shell) – The Korn Shell also was the base for the POSIX Shell standard specifications etc.

Each shell does the same job but understand different commands and provide different built in functions.

**Shell Scripting**

Usually shells are interactive that mean, they accept command as input from users and execute them. However some time we want to execute a bunch of commands routinely, so we have type in all commands each time in terminal.  
As shell can also take commands as input from file we can write these commands in a file and can execute them in shell to avoid this repetitive work. These files are called **Shell Scripts** or **Shell Programs**. Shell scripts are similar to the [**batch file**](https://en.wikipedia.org/wiki/Batch_file) in MS-DOS. Each shell script is saved with **.sh** file extension eg. **myscript.sh**

A shell script have syntax just like any other programming language. If you have any prior experience with any programming language like Python, C/C++ etc. it would be very easy to get started with it.  
A shell script comprises following elements –

* Shell Keywords – if, else, break etc.
* Shell commands – cd, ls, echo, pwd, touch etc.
* Functions
* Control flow – if..then..else, case and shell loops etc.

**Why do we need shell scripts**

There are many reasons to write shell scripts –

* To avoid repetitive work and automation
* System admins use shell scripting for routine backups
* System monitoring
* Adding new functionality to the shell etc.

**Advantages of shell scripts**

* The command and syntax are exactly the same as those directly entered in command line, so programmer do not need to switch to entirely different syntax
* Writing shell scripts are much quicker
* Quick start
* Interactive debugging etc.

**Disadvantages of shell scripts**

* Prone to costly errors, a single mistake can change the command which might be harmful
* Slow execution speed
* Design flaws within the language syntax or implementation
* Not well suited for large and complex task
* Provide minimal data structure unlike other scripting languages. etc

CODE –

echo "Menu"

echo "1. Create"

echo "2. Display"

echo "3. Insert"

echo "4. Delete"

echo "5. Search"

echo "6. Exit"

while true;

do

echo "ENTER CHOICE"

read choice

case $choice in

1)

echo "Enter number of records you want to add "

read n

for (( c=1 ; c<= $n ; c++ ))

do

echo "Enter your name"

read name

echo "Enter your city"

read city

echo "NAME :" $name "CITY :" $city>>"file.txt"

done

echo "ADDRESS BOOK CREATED SUCCESSFULLY" ;;

2) more "file.txt";;

3)echo "Enter your name"

read name

echo "Enter your city"

read city

echo "NAME :" $name "CITY :" $city>>"file.txt"

echo "DATA INSERTED SUCCESSFULLY";;

4)echo "Enter name to delete"

read name

grep -v $name "file.txt">"file2.txt"

mv "file2.txt" "file.txt"

echo "Data removed " ;;

5)

echo "Enter name to search"

read name

grep $name file.txt ;;

6)

echo "PRESS CTRL+C " ;;