

Assignment 1-B

Name - SOHAM PRASHANT SANT

Roll no - 32371

Title :- Shell Programming

AIM :

Write a program to implement an address book with options given below:

- a) Create address book.
- b) View address book.
- c) Insert a record.
- d) Delete a record.
- e) Modify a record.
- f) Exit.

OBJECTIVE:

This assignment helps the students understand the basic commands in Unix/Linux and how write the shell scripts.

Theory :

1. What is Shell Programming ?

A shell programming answer in brief is that it involves writing and executing scripts using a command-line interface (CLI) or shell. A shell is a program that provides a user interface for interacting with an operating system. Shell scripts are essentially a series of commands and instructions written in a scripting language that the shell interprets and executes.

Shell scripting is commonly used for automating tasks, managing system configurations, and running multiple commands sequentially or conditionally. Popular Unix-based shells include Bash, Zsh, and Csh, while Windows systems typically use PowerShell.

In summary, shell programming is about using a shell to write scripts that automate tasks and interact with the operating system through the command line.

2. Features of Shell Programming ?

- Command-line Interface (CLI): Shell programs provide a text-based interface where users can interact with the operating system by entering commands.
- Scripting Language: Shell scripts are written in a specific scripting language (e.g., Bash, Zsh) that the shell interprets and executes.
- Automation: Shell scripts are used for automating repetitive tasks, allowing users to save time and effort.

- **Command Execution:** The ability to execute system commands and utilities directly from the shell script.
- **Variables and Data Manipulation:** Shell scripts support variables to store and manipulate data, making it possible to work with user input and process data.
- **Flow Control:** Shell programming offers flow control mechanisms like conditionals (if-else) and loops (for, while) to control the execution of commands based on specific conditions.
- **File and Directory Operations:** Shell programming enables users to create, delete, move, and manipulate files and directories.
- **System Administration:** Shell programming is widely used in system administration tasks to manage and configure servers and systems.
- **Portability:** Shell scripts can run on different Unix-like systems, making them portable across various platforms.
- **Modularity:** Shell scripts can be divided into functions and modules, promoting code reusability and maintainability.

These features make shell programming a powerful tool for managing systems, automating tasks, and interacting with the operating system efficiently through the command-line interface.

3. Conditional looping constitute in shell brief description and syntax if-else,while,case-esac

Conditional looping in shell programming involves executing a block of code repeatedly based on certain conditions. There are three main constructs for conditional looping in shell scripts: if-else, while loop, and case-esac.

1. if-else:

The if-else statement allows you to execute different blocks of code based on whether a specific condition is true or false.

Syntax:

```
if [ condition ]; then
    # Code block executed if the condition is true
else
    # Code block executed if the condition is false
fi
```

Example:

```
#!/bin/bash
read -p "Enter a number: " num

if [ $num -gt 0 ]; then
    echo "The number is positive."
```

```
else
    echo "The number is non-positive."
fi
```

2. while loop:

The while loop executes a block of code repeatedly as long as a given condition is true.

Syntax:

```
while [ condition ]; do
    # Code block executed as long as the condition is true
done
```

Example:

```
#!/bin/bash
count=1

while [ $count -le 5 ]; do
    echo "Count: $count"
    count=$((count + 1))
done
```

3. case-esac:

The case-esac construct is used for multi-way branching based on the value of a variable.

Syntax:

```
case "$variable" in
    pattern1)
        # Code block executed if variable matches pattern1
        ;;
    pattern2)
        # Code block executed if variable matches pattern2
        ;;
    *)
        # Code block executed if variable doesn't match any previous patterns
        ;;
esac
```

Example:

```
#!/bin/bash
read -p "Enter a day (1-7): " day

case "$day" in
    1)
        echo "Sunday"
        ;;
```

```

2)
    echo "Monday"
    ;;
3)
    echo "Tuesday"
    ;;
4)
    echo "Wednesday"
    ;;
5)
    echo "Thursday"
    ;;
6)
    echo "Friday"
    ;;
7)
    echo "Saturday"
    ;;
*)
    echo "Invalid day entered."
    ;;
esac

```

These conditional looping constructs are essential for controlling the flow of execution in shell scripts based on specific conditions, making shell programming versatile and capable of handling various scenarios.

CODE :-

```

#!/bin/bash

# Create a directory named "DataBase" and move into it.
mkdir DataBase
cd DataBase

# Initialize the variable "ans" to 0.
ans=0

# Start an infinite loop to display the menu until the user chooses to exit (option 7).
while [ $ans -ne 7 ]; do
    # Display the menu options.
    echo -e "\n-----"
    echo "|          ***MENU***          |"
    echo "|-----|"
    echo "|1.Create Database           |"
    echo "|2.Display Database          |"
    echo "|3.Search Record             |"
    echo "|4.Insert Record             |"
    echo "|5.Delete Record             |"
    echo "|6.Modify Record             |"
    echo "|7.Exit                       |"
    echo "|-----|"
    echo -n "Enter your choice => "
    read ans

    # Use the "case" statement to execute the corresponding block of code based on the user's choice.
    case $ans in

```

```

1)
# Option 1: Create a new database.
echo -n "Name Your database => "
read name
touch "${db_name}.txt"
>"${db_name}.txt"

echo -n "Enter number of records => "
read total

i=0
uid=0

# Start a loop to input the records for the database.
while [ $i -lt $total ]; do
    echo -n "Enter name of student $(( i + 1 )) => "
    read name
    echo -n "Enter age of $name => "
    read age
    echo -n "Enter phone number of $name => "
    read phone
    echo -n "Enter pointer of $name => "
    read pointer
    uid=$(( $uid + 1 ))
    # Append the record to the database file.
    echo "${uid}|${name}|${age}|${phone}|${pointer}|" >> "${db_name}.txt"
    i=$(( $i + 1 ))
done

echo "Database Created"
;;

# Option 2: Display the contents of a database file.
2)
echo -n "Enter the name of the database to display => "
read db_name
if [ -f "${db_name}.txt" ]; then
    cat "${db_name}.txt"
else
    echo "Database not found!"
fi
;;

# Option 3: Search for a record in a database.
3)
echo -n "Enter the name of the database to search => "
read db_name
echo -n "Enter the name of the student to search => "
read search_name

if [ -f "${db_name}.txt" ]; then
    grep -i "${search_name}" "${db_name}.txt"
else
    echo "Database not found!"
fi
;;

# Option 4: Insert a new record into a database.
4)
echo -n "Enter the name of the database to insert record => "
read db_name

```

```

if [ -f "${db_name}.txt" ]; then
    echo -n "Enter name of the student => "
    read name
    echo -n "Enter age of $name => "
    read age
    echo -n "Enter phone number of $name => "
    read phone
    echo -n "Enter pointer of $name => "
    read pointer
    uid=$(( $uid + 1 ))
    # Append the new record to the database file.
    echo "${uid}|${name}|${age}|${phone}|${pointer}|" >> "${db_name}.txt"
    echo "Record Inserted"
else
    echo "Database not found!"
fi
;;

# Option 5: Delete a record from a database.
5)
    echo -n "Enter the name of the database to delete record => "
    read db_name

    if [ -f "${db_name}.txt" ]; then
        echo -n "Enter the name of the student to delete => "
        read delete_name
        # Use grep to filter out the record to be deleted and create a temporary file.
        grep -iv "${delete_name}" "${db_name}.txt" > "${db_name}_temp.txt"
        # Rename the temporary file as the new database file, effectively deleting the record.
        mv "${db_name}_temp.txt" "${db_name}.txt"
        echo "Record Deleted"
    else
        echo "Database not found!"
    fi
    ;;

# Option 6: Modify a record in a database.
6)
    echo -n "Enter the name of the database to modify record => "
    read db_name

    if [ -f "${db_name}.txt" ]; then
        echo -n "Enter the name of the student to modify => "
        read modify_name

        # Check if the record exists in the database.
        if grep -iq "${modify_name}" "${db_name}.txt"; then
            echo -n "Enter the attribute to modify (name, age, phone, pointer): "
            read attribute

            # Read the new value for the attribute.
            echo -n "Enter the new value for ${attribute}: "
            read new_value

            # Update the attribute of the record using sed.
            sed -i "s/\(^.*${modify_name}|.*\).*$/\1${new_value}/" "${db_name}.txt"
            echo "Record Modified"
        else
            echo "Record not found!"
        fi
    else
        echo "Database not found!"
    fi

```

```

        fi
        ;;

# Option 7: Exit the script.
7)
    echo "BYE !!"
    ;;

# For any invalid input, display an error message.
*)
    echo "Invalid choice. Please try again."
    ;;
esac
done

```

output :-

```

PS C:\Users\soham\OneDrive\Desktop> bash
soham@LAPTOP-VC5HPE1S:/mnt/c/Users/soham/OneDrive/Desktop$ ./text.sh

```

```

-----
|      ***MENU***      |
|-----|
|1.Create Database      |
|2.Display Database     |
|3.Search Record       |
|4.Insert Record       |
|5.Delete Record       |
|6.Modify Record       |
|7.Exit                |
|-----|
Enter your choice => 1
Name Your database => N10
Enter number of records => 5
Enter name of student 1 => Soham Sant
Enter age of Soham Sant => 19
Enter phone number of Soham Sant => 897645231
Enter pointer of Soham Sant => 8.5
Enter name of student 2 => Aditya Ghadge
Enter age of Aditya Ghadge => 19
Enter phone number of Aditya Ghadge => 876541237
Enter pointer of Aditya Ghadge => 8.9
Enter name of student 3 => Nandini Nikam
Enter age of Nandini Nikam => 19
Enter phone number of Nandini Nikam => 754896312
Enter pointer of Nandini Nikam => 8.2
Enter name of student 4 => Mega Nadar
Enter age of Mega Nadar => 21
Enter phone number of Mega Nadar =>
Enter pointer of Mega Nadar => 8.5
Enter name of student 5 => Isha Wagh
Enter age of Isha Wagh => 20
Enter phone number of Isha Wagh => 7863242465
Enter pointer of Isha Wagh => 8.6
Database Created

-----
|      ***MENU***      |
|-----|

```

```

|1.Create Database      |
|2.Display Database    |
|3.Search Record       |
|4.Insert Record       |
|5.Delete Record       |
|6.Modify Record       |
|7.Exit                |
-----
Enter your choice => 2
Enter the name of the database to display => N10
1|Soham Sant|19|897645231|8.5|
2|Aditya Ghadge|19|876541237|8.9|
3|Nandini Nikam|19|754896312|8.2|
4|Mega Nadar|21||8.5|
5|Isha Wagh|20|7863242465|8.6|

-----
|      ***MENU***      |
-----
|1.Create Database      |
|2.Display Database    |
|3.Search Record       |
|4.Insert Record       |
|5.Delete Record       |
|6.Modify Record       |
|7.Exit                |
-----
Enter your choice => 6
Enter the name of the database to modify record => N10
Enter the name of the student to modify => Mega Nadar
Enter the attribute to modify (name, age, phone, pointer): phone
Enter the new value for phone: 78642513
Record Modified

-----
|      ***MENU***      |
-----
|1.Create Database      |
|2.Display Database    |
|3.Search Record       |
|4.Insert Record       |
|5.Delete Record       |
|6.Modify Record       |
|7.Exit                |
-----
Enter your choice => 3
Enter the name of the database to search => N10
Enter the name of the student to search => soham
1|Soham Sant|19|897645231|8.5|

-----
|      ***MENU***      |
-----
|1.Create Database      |
|2.Display Database    |
|3.Search Record       |
|4.Insert Record       |
|5.Delete Record       |
|6.Modify Record       |
|7.Exit                |
-----
Enter your choice => 4

```



```

Enter the name of the database to insert record => N10
Enter name of the student => Kunal Wagh
Enter age of Kunal Wagh => 21
Enter phone number of Kunal Wagh => 865476312
Enter pointer of Kunal Wagh => 9.5
Record Inserted

```

```

-----
|      ***MENU***      |
|-----|
|1.Create Database    |
|2.Display Database   |
|3.Search Record     |
|4.Insert Record      |
|5.Delete Record      |
|6.Modify Record      |
|7.Exit              |
|-----|

```

```

Enter your choice => 5
Enter the name of the database to delete record => N10
Enter the name of the student to delete => soham
Record Deleted

```

```

-----
|      ***MENU***      |
|-----|
|1.Create Database    |
|2.Display Database   |
|3.Search Record     |
|4.Insert Record      |
|5.Delete Record      |
|6.Modify Record      |
|7.Exit              |
|-----|

```

```

Enter your choice => 2
Enter the name of the database to display => N10
2|Aditya Ghadge|19|876541237|8.9|
3|Nandini Nikam|19|754896312|8.2|
4|Mega Nadar|21||8.5|78642513|
5|Isha Wagh|20|7863242465|8.6|
6|Kunal Wagh|21|865476312|9.5|

```

```

-----
|      ***MENU***      |
|-----|
|1.Create Database    |
|2.Display Database   |
|3.Search Record     |
|4.Insert Record      |
|5.Delete Record      |
|6.Modify Record      |
|7.Exit              |
|-----|

```

```

Enter your choice => 7
BYE !!

```

Conclusion :-

we learned the fundamentals of shell scripting, including reading user input, using conditional statements, and employing file handling techniques.