Experiment 4: Shell Programming

Aim

The aim of this experiment is to learn the basics of shell programming in Linux by writing simple shell scripts. The purpose is to understand how automation of tasks can be achieved using shell commands inside scripts.

Requirements

1. Linux operating system (Ubuntu, Fedora, etc.)
2. Access to a terminal
3. Text editor (gedit, nano, or vi) to write shell scripts
4. Knowledge of basic Linux commands
5. Shell (commonly bash) installed

Procedure

1. Open a terminal in Linux.
2. Create a new shell script file, e.g., nano script.sh.
3. Write a simple shell program inside it, for example:

```
#!/bin/bash

echo "Hello, this is my first shell script"
```

4. Save the file and exit.
5. Change file permission to make it executable:

```
chmod +x script.sh
```

6. Run the script using:

```
./script.sh
```

7. Write and run other shell programs such as:

Program to display current date and time (date).

Program to show present working directory (pwd).

Program to print all files in a directory (ls).

Program using loop (for, while).

Program using conditional statements (if-else).

LAB TASKS

TASK i – Hello script

COMMAND

```
#!/bin/bash
echo "Hello, World!"
```

TASK ii. **Personalized Greeting**

COMMAND

```
#!/bin/bash
echo "Enter your name: "
read name    # 'read' takes user input
echo "Hello, $name! Welcome to Shell Scripting."
```

TASK iii. Arithmetic Operations

```
#!/bin/bash
echo "Enter first number: "
read num1
echo "Enter second number: "
read num2

echo "Addition: $((num1 + num2))"
echo "Subtraction: $((num1 - num2))"
echo "Multiplication: $((num1 * num2))"
echo "Division: $((num1 / num2))"
```

- $(()) is **arithmetic expansion** in bash.

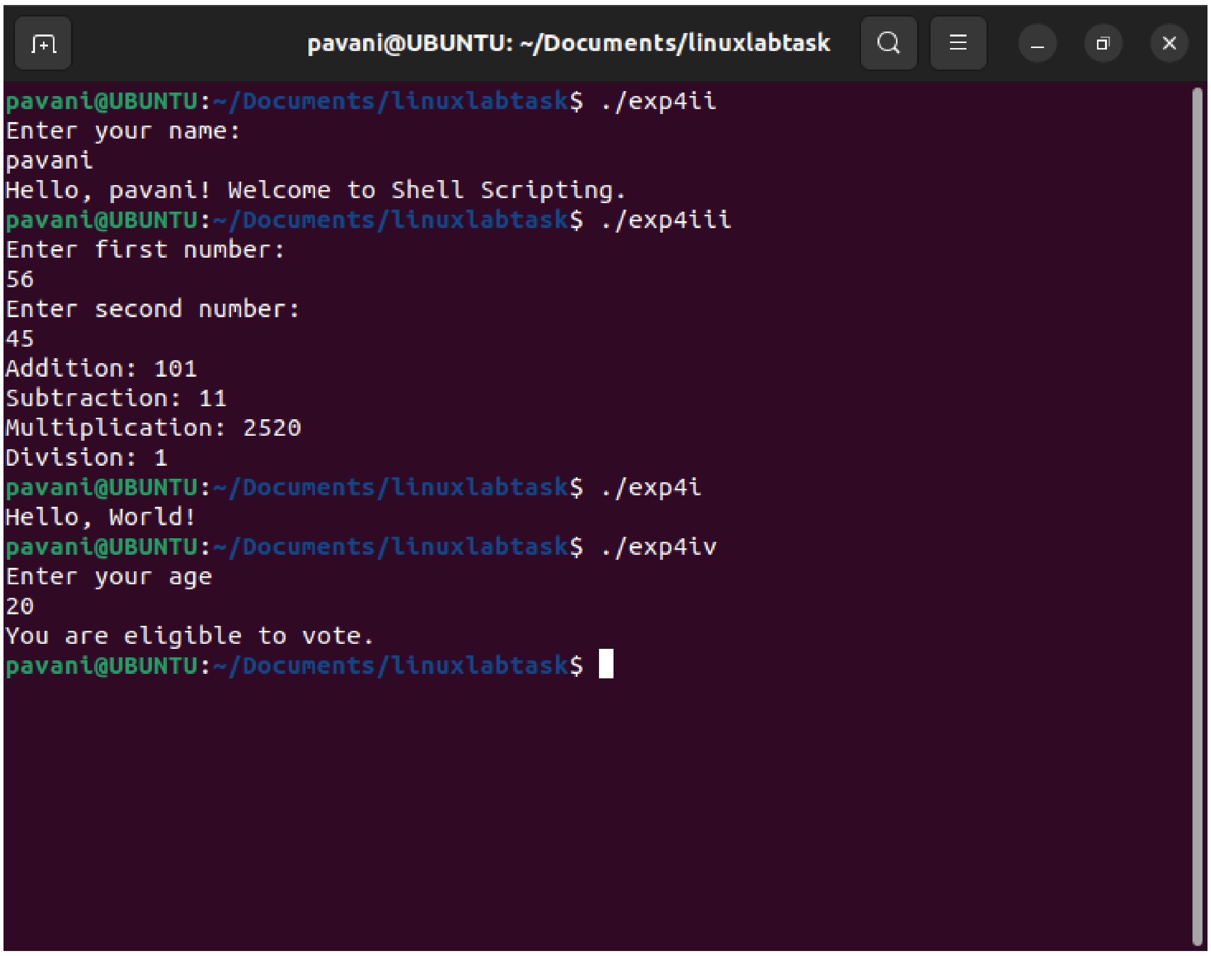
TASK iv. Voting Eligibility

```
#!/bin/bash
echo "Enter your age: "
read age

if [ $age -ge 18 ]
then
```

```
    echo "You are eligible to vote."
else
    echo "Sorry, you are not eligible to vote."
```

OUTPUTS OF TASK i,ii,iii,iv



```
pavani@UBUNTU:~/Documents/linuxlabtask$ ./exp4ii
Enter your name:
pavani
Hello, pavani! Welcome to Shell Scripting.
pavani@UBUNTU:~/Documents/linuxlabtask$ ./exp4iii
Enter first number:
56
Enter second number:
45
Addition: 101
Subtraction: 11
Multiplication: 2520
Division: 1
pavani@UBUNTU:~/Documents/linuxlabtask$ ./exp4i
Hello, World!
pavani@UBUNTU:~/Documents/linuxlabtask$ ./exp4iv
Enter your age
20
You are eligible to vote.
pavani@UBUNTU:~/Documents/linuxlabtask$
```

Observation

Shell scripts execute commands step by step automatically.

#!/bin/bash defines the interpreter for the script.

Permissions must be set using chmod +x before execution.

Scripts can use variables, loops, and conditions like in programming languages.

Conclusion

From this experiment, we conclude that shell programming is a powerful way to automate tasks in Linux. By writing shell scripts, repetitive tasks can be executed with a single command. Shell scripts also support variables, loops, and conditions, which make them useful for both system administration and programming tasks.