

Name: Nivedita Londhe

PRN: 22420003

Batch: EN-4

Practical No. 3

Title: Shell Scripting

Part A: Shell scripting of Arithmetic Operations

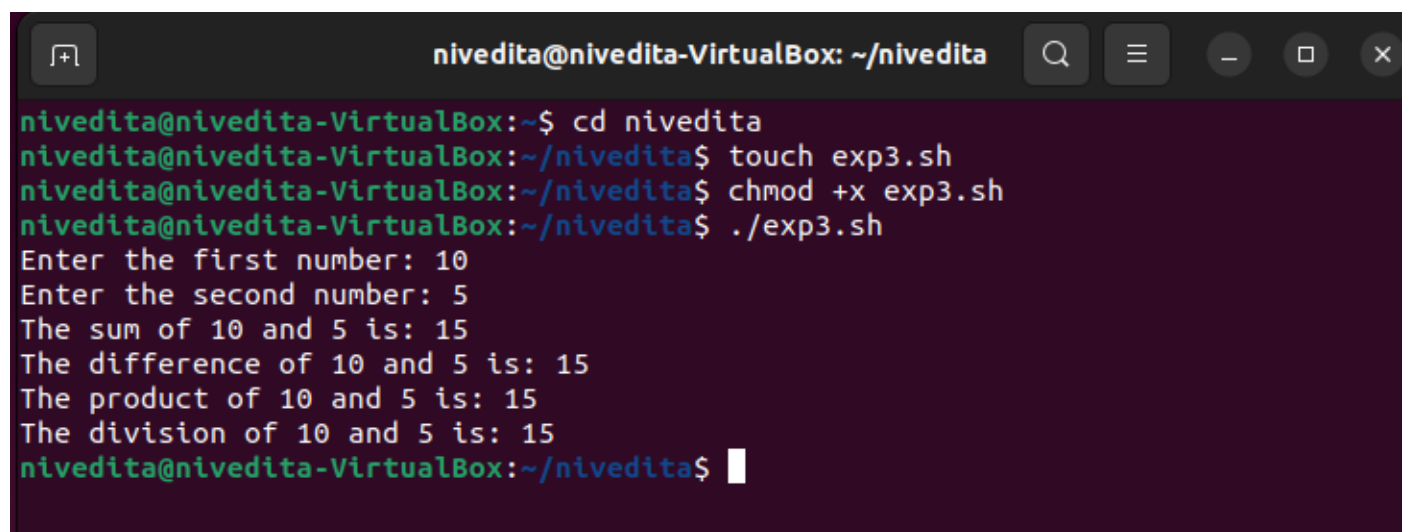
Commands:

- **touch** : This command is used to create an exp3.sh empty file.
- **chmod +x exp3.sh** : Make the script executable.
- **./exp3.sh** : Run the script by typing.

Code:

```
#!/bin/bash
read -p "Enter the first number: " num1
read -p "Enter the second number: " num2
sum=$((num1 + num2))
sub=$((num1 - num2))
mul=$((num1 * num2))
div=$((num1 / num2))
echo "The sum of $num1 and $num2 is: $sum"
echo "The difference of $num1 and $num2 is: $sum"
echo "The product of $num1 and $num2 is: $sum"
echo "The division of $num1 and $num2 is: $sum"
```

Result:



The screenshot shows a terminal window titled "nivedita@nivedita-VirtualBox: ~/nivedita". The user has navigated to the directory ~/nivedita and executed the following commands: `cd nivedita`, `touch exp3.sh`, `chmod +x exp3.sh`, and `./exp3.sh`. The script prompts for two numbers, 10 and 5, and then displays the results of arithmetic operations: sum (15), difference (5), product (50), and division (2). The terminal output is as follows:

```
nivedita@nivedita-VirtualBox:~$ cd nivedita
nivedita@nivedita-VirtualBox:~/nivedita$ touch exp3.sh
nivedita@nivedita-VirtualBox:~/nivedita$ chmod +x exp3.sh
nivedita@nivedita-VirtualBox:~/nivedita$ ./exp3.sh
Enter the first number: 10
Enter the second number: 5
The sum of 10 and 5 is: 15
The difference of 10 and 5 is: 5
The product of 10 and 5 is: 50
The division of 10 and 5 is: 2
nivedita@nivedita-VirtualBox:~/nivedita$
```

Part B: Shell scripting of Conditional statements

Commands:

- **`touch`** : This command is used to create an empty file.
- **`chmod +x your_script.sh`** : Make the script executable.
- **`./your_script.sh`** : Run the script by typing.

Code:

```
#!/bin/bash
read -p "Enter a number: " num
if (($num > 0)); then
echo "$num is positive."
elif (($num < 0)); then
echo "$num is negative."
else
echo "$num is zero."
fi
```

Result:



```
nivedita@nivedita-VirtualBox: ~/nivedita
nivedita@nivedita-VirtualBox:~/nivedita$ touch ifElse.sh
nivedita@nivedita-VirtualBox:~/nivedita$ chmod +x ifElse.sh
nivedita@nivedita-VirtualBox:~/nivedita$ ./ifElse.sh
Enter a number: 4
4 is positive.
nivedita@nivedita-VirtualBox:~/nivedita$ ./ifElse.sh
Enter a number: -10
-10 is negative.
nivedita@nivedita-VirtualBox:~/nivedita$ ./ifElse.sh
Enter a number: 0
0 is zero.
nivedita@nivedita-VirtualBox:~/nivedita$
```

Part C: Shell scripting using loops and switch case

Loops:

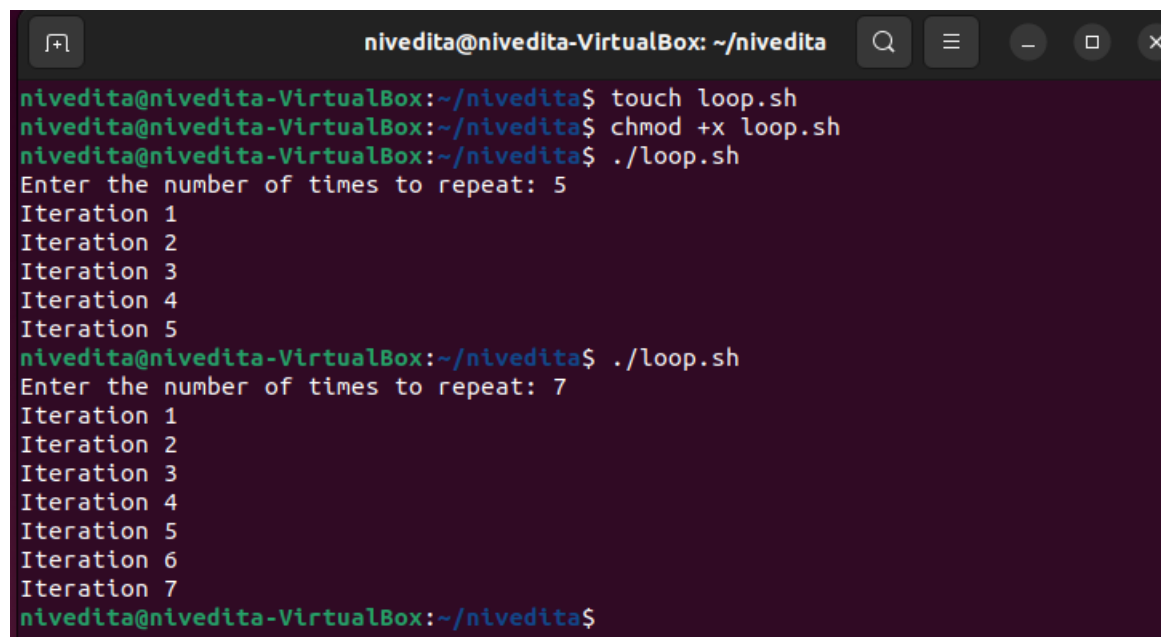
Commands:

- **touch** : This command is used to create an empty file.
- **chmod +x your_script.sh** : Make the script executable.
- **./your_script.sh** : Run the script by typing.
- **ls** : Lists the files and directories in the current directory.

Code:

```
#!/bin/bash
read -p "Enter the number of times to repeat: " num_repeats
for (( i=1; i<=num_repeats; i++ )); do
    echo "Iteration $i"
done
```

Result:



```
nivedita@nivedita-VirtualBox: ~/nivedita
nivedita@nivedita-VirtualBox:~/nivedita$ touch loop.sh
nivedita@nivedita-VirtualBox:~/nivedita$ chmod +x loop.sh
nivedita@nivedita-VirtualBox:~/nivedita$ ./loop.sh
Enter the number of times to repeat: 5
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
nivedita@nivedita-VirtualBox:~/nivedita$ ./loop.sh
Enter the number of times to repeat: 7
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Iteration 7
nivedita@nivedita-VirtualBox:~/nivedita$
```

Switch Case:**Code:**

```
#!/bin/bash
read -p "Enter a choice (1 for addition, 2 for subtraction): " choice
case $choice in
  1)
    read -p "Enter the first number: " num1
    read -p "Enter the second number: " num2
    result=$((num1 + num2))
    echo "The sum is: $result"
    ;;
  2)
    read -p "Enter the first number: " num1
    read -p "Enter the second number: " num2
    result=$((num1 - num2))
    echo "The difference is: $result"
    ;;
  *)
    echo "Invalid choice. Please enter 1 or 2."
    ;;
esac
```

Result:


```
nivedita@nivedita-VirtualBox: ~/nivedita
nivedita@nivedita-VirtualBox:~/nivedita$ touch switchCase.sh
nivedita@nivedita-VirtualBox:~/nivedita$ chmod +x switchCase.sh
nivedita@nivedita-VirtualBox:~/nivedita$ ./switchCase.sh
Enter a choice (1 for addition, 2 for subtraction): 1
Enter the first number: 5
Enter the second number: 2
The sum is: 7
nivedita@nivedita-VirtualBox:~/nivedita$ ./switchCase.sh
Enter a choice (1 for addition, 2 for subtraction): 2
Enter the first number: 3
Enter the second number: 5
The difference is: -2
nivedita@nivedita-VirtualBox:~/nivedita$ ls
exp3.sh  ifElse.sh  switchCase.sh
nivedita@nivedita-VirtualBox:~/nivedita$
```

Conclusion:

We created a shell script program that used different programming elements like conditional statements, arithmetic operations, loops, and switch cases. This helped us understand the importance of shell scripts in the Linux environment.

Questions

1. Write significance of Shell script in Linux

Shell scripts are essential in Linux for several reasons:

- **Automation:** Shell scripts automate repetitive tasks, allowing users to execute multiple commands in sequence without manual intervention. This reduces human error and saves time.
- **Customization:** Users can tailor their Linux environment by creating scripts that configure their system or software exactly as needed.
- **Batch Processing:** Shell scripts can process large volumes of data or files in bulk, making them ideal for administrative tasks, such as backups, system monitoring, or log management.
- **Task Scheduling:** Scripts can be scheduled to run at specific times using tools like cron, making them ideal for regular maintenance tasks.
- **Efficiency:** Shell scripts are lightweight and can be executed faster than other types of scripts or compiled programs, especially for small tasks.
- **Portability:** Since shell scripts are text files, they can be easily shared and executed on any Linux system with the appropriate shell interpreter.

2. Write types of shells

There are several types of shells available in Linux, each with its unique features:

- **Bourne Shell (sh):**
 - The original Unix shell, created by Stephen Bourne.
 - Known for its simplicity and speed.
 - It's the default shell on many Unix-like systems.
- **Bourne Again Shell (bash):**
 - An enhanced version of the Bourne Shell.
 - Offers features like command-line editing, command history, and improved scripting capabilities.
 - It's the default shell in most Linux distributions.

➤ **Korn Shell (ksh):**

- Developed by David Korn, it combines features of the Bourne Shell with those of the C Shell (csh).
- Known for its scripting efficiency and advanced features like associative arrays and built-in floating-point arithmetic.

➤ **C Shell (csh):**

- Developed by Bill Joy, it has a syntax similar to the C programming language.

➤ **Z Shell (zsh):**

- Combines features from bash, ksh, and tcsh.
- Known for its powerful scripting capabilities, advanced auto- completion, and customization options.
- Popular among power users due to its flexibility and feature- rich environment.

