

PRACTICAL NO: 2

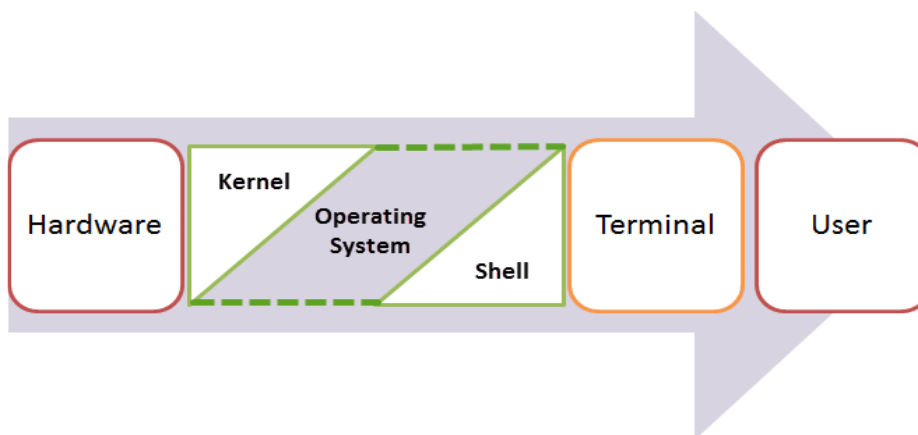
2.1 Study the basics of shell programming.

Aim: Study the basics of shell programming.

What is a Shell?

An Operating is made of many components, but its two prime components are

- Kernel
- Shell



A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.

A shell in a Linux operating system takes input from you in the form of commands, processes it, and then gives an output. It is the interface through which a user works on the programs, commands, and scripts. A shell is accessed by a terminal which runs it.

When you run the terminal, the Shell issues **a command prompt (usually \$)**, where you can type your input, which is then executed when you hit the Enter key. The output or the result is thereafter displayed on the terminal.

The Shell wraps around the delicate interior of an Operating system protecting it from accidental damage. Hence the name **Shell**

Types of Shell

There are two main shells in Linux:

1. Bourne Shell: The prompt for this shell is \$ and its derivatives are listed below:

- POSIX shell also is known as sh
- Korn Shell also known as sh
- **Bourne Again Shell** also known as bash (most popular)

2. The C shell: The prompt for this shell is %, and its subcategories are:

- C shell also is known as csh
- Tops C shell also is known as tcsh

What is Shell Scripting?

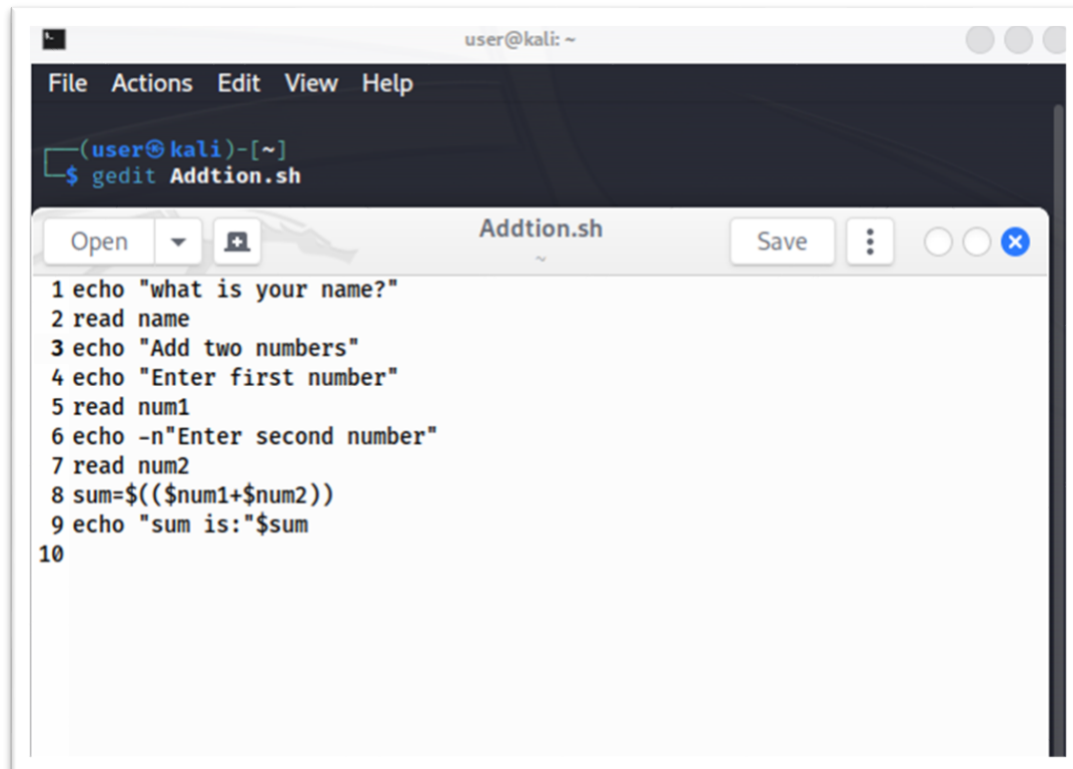
Shell scripting is writing a series of command for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script, which can be stored and executed anytime. This reduces the effort required by the end user.

Let us understand the steps in creating a Shell Script :-

- 1. Create a file using** TERMINAL with the **gedit** command
(**gedit File_name.sh**)
- 2. Name** script file with **File_name .sh**
- 3. Write** some code.
- 4. Save** the script file
- 5. For executing** the script type **bash File_name.sh**

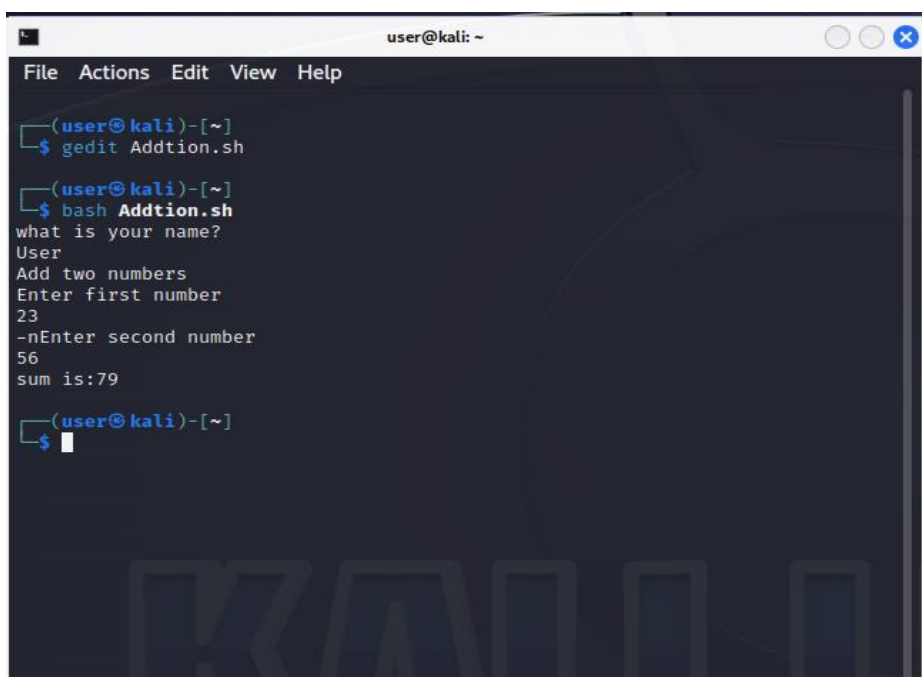
2.2 Adding of Two Numbers

Step1 create and save the shell script with gedit command



```
user@kali: ~  
File Actions Edit View Help  
(user@kali)-[~]  
$ gedit Addtion.sh  
Addtion.sh  
1 echo "what is your name?"  
2 read name  
3 echo "Add two numbers"  
4 echo "Enter first number"  
5 read num1  
6 echo -n"Enter second number"  
7 read num2  
8 sum=$((num1+num2))  
9 echo "sum is:$sum"  
10
```

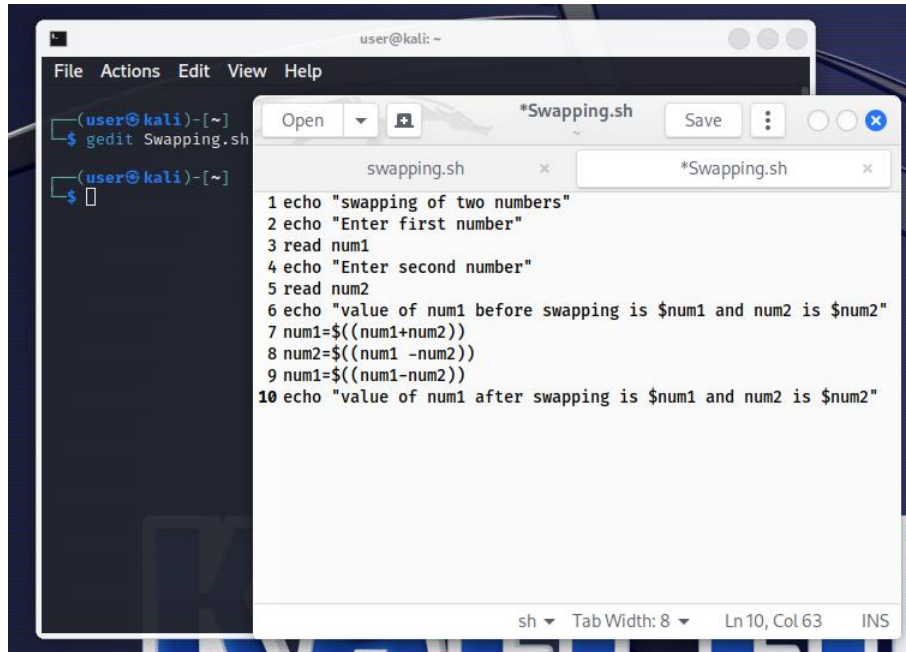
Step 2: Run saved shell script with bash command



```
user@kali: ~  
File Actions Edit View Help  
(user@kali)-[~]  
$ gedit Addtion.sh  
(user@kali)-[~]  
$ bash Addtion.sh  
what is your name?  
User  
Add two numbers  
Enter first number  
23  
-nEnter second number  
56  
sum is:79  
(user@kali)-[~]  
$
```

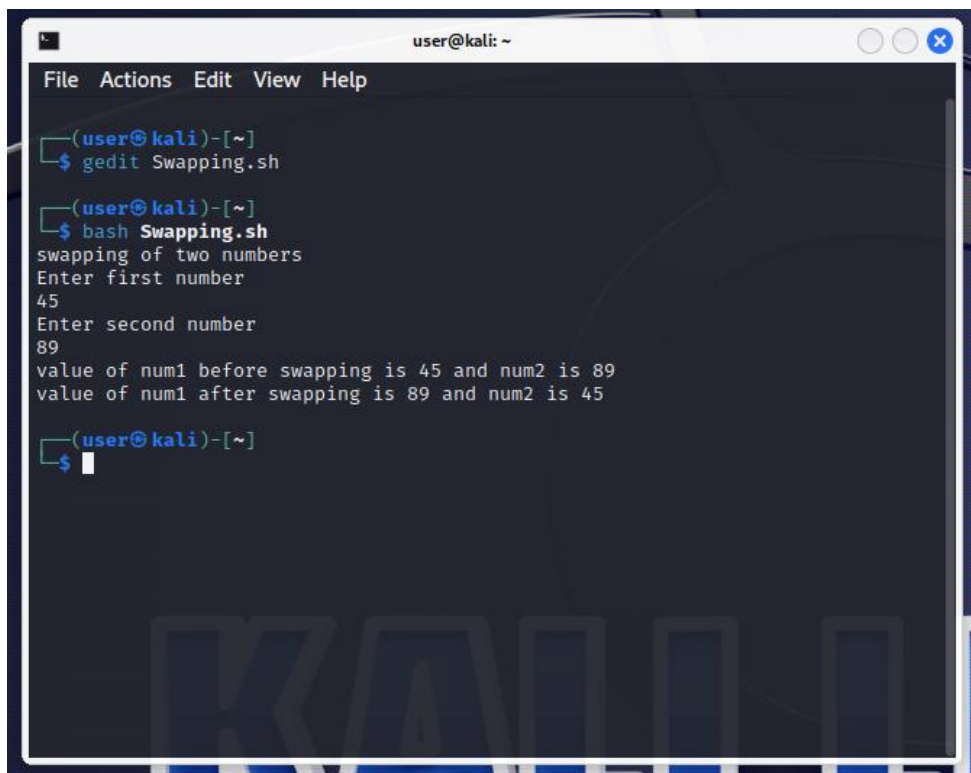
2.3 swap Two Variables without Using Third Variable

Step 1: create and save the shell script with gedit command



```
user@kali: ~  
File Actions Edit View Help  
$ gedit Swapping.sh  
$  
swapping.sh  
1 echo "swapping of two numbers"  
2 echo "Enter first number"  
3 read num1  
4 echo "Enter second number"  
5 read num2  
6 echo "value of num1 before swapping is $num1 and num2 is $num2"  
7 num1=$((num1+num2))  
8 num2=$((num1 - num2))  
9 num1=$((num1-num2))  
10 echo "value of num1 after swapping is $num1 and num2 is $num2"
```

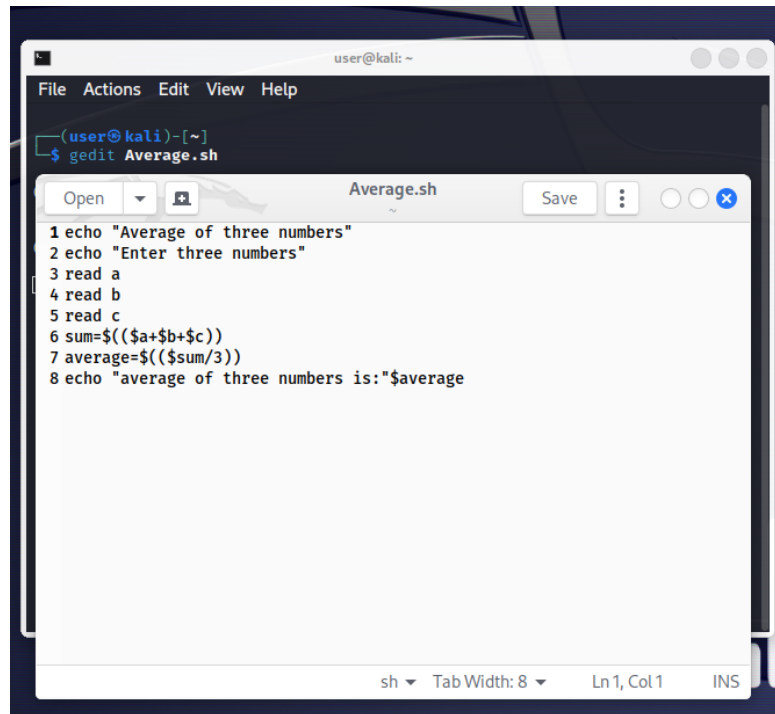
Step 2: Run the shell script with bash command



```
user@kali: ~  
File Actions Edit View Help  
$ gedit Swapping.sh  
$ bash Swapping.sh  
swapping of two numbers  
Enter first number  
45  
Enter second number  
89  
value of num1 before swapping is 45 and num2 is 89  
value of num1 after swapping is 89 and num2 is 45  
$
```

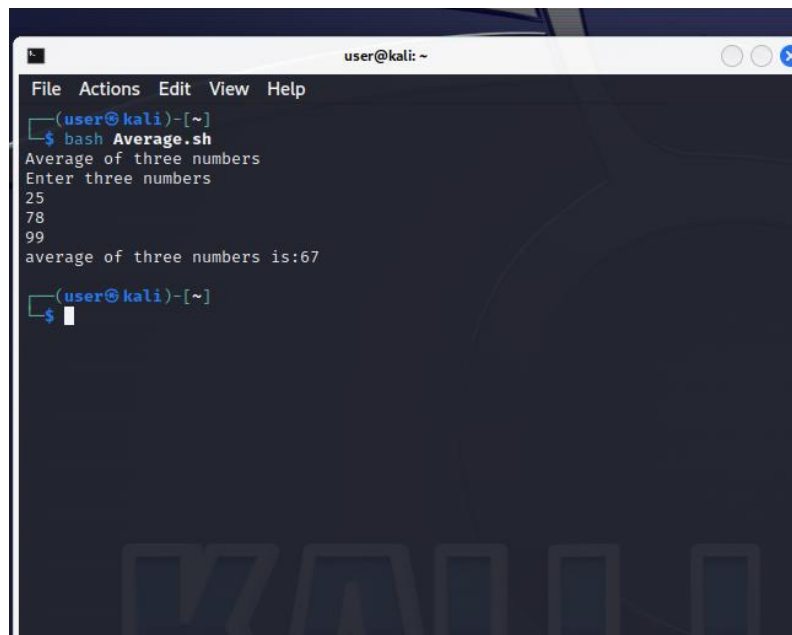
2.4 Average of 3 Numbers

Step 1: create and save the shell script with gedit command



```
user@kali: ~  
File Actions Edit View Help  
(user@kali)-[~]  
$ gedit Average.sh  
Average.sh  
1 echo "Average of three numbers"  
2 echo "Enter three numbers"  
3 read a  
4 read b  
5 read c  
6 sum=$((a+b+c))  
7 average=$((sum/3))  
8 echo "average of three numbers is:$average"  
sh Tab Width: 8 Ln 1, Col 1 INS
```

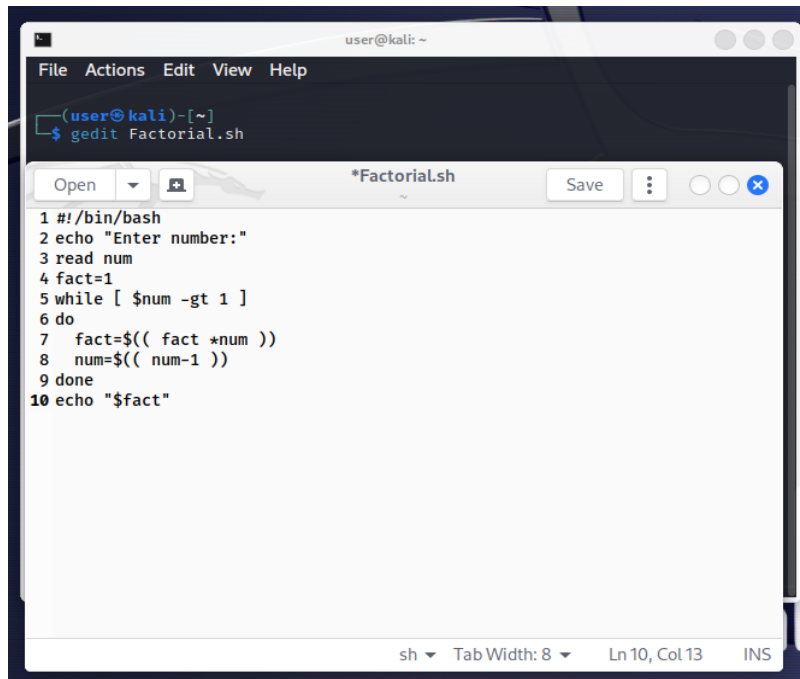
Step 2: Run the shell script with bash command



```
user@kali: ~  
File Actions Edit View Help  
(user@kali)-[~]  
$ bash Average.sh  
Average of three numbers  
Enter three numbers  
25  
78  
99  
average of three numbers is:67  
(user@kali)-[~]  
$
```

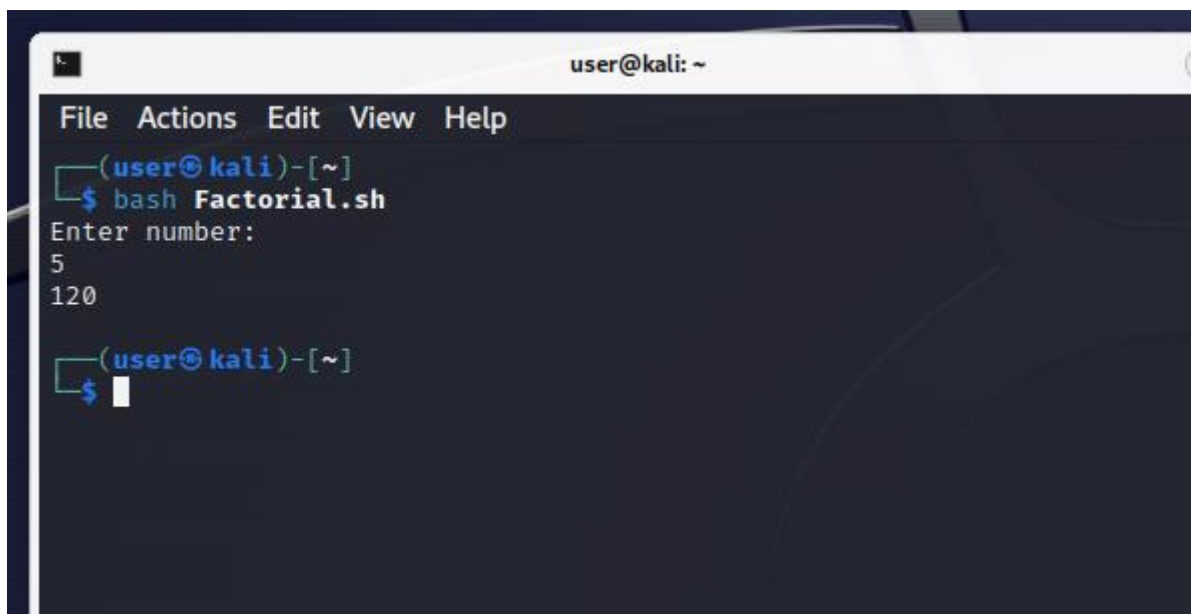
2.5 Calculate Factorial of Given Number

Step 1: create and save the shell script with gedit command



```
user@kali: ~  
File Actions Edit View Help  
(user@kali)-[~]  
$ gedit Factorial.sh  
Open *Factorial.sh Save  
1 #!/bin/bash  
2 echo "Enter number:"  
3 read num  
4 fact=1  
5 while [ $num -gt 1 ]  
6 do  
7 fact=$(( fact *num ))  
8 num=$(( num-1 ))  
9 done  
10 echo "$fact"  
sh Tab Width: 8 Ln 10, Col 13 INS
```

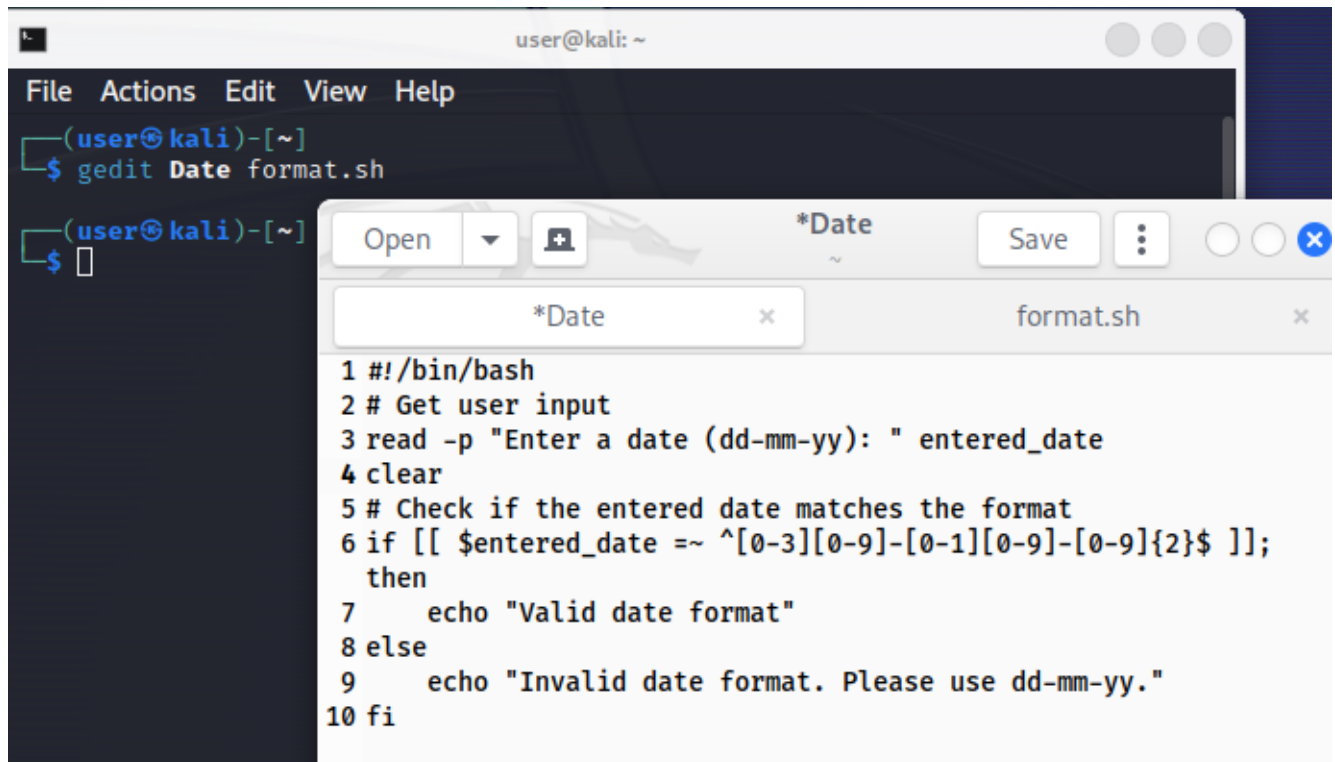
Step 2: Run the shell script with bash command



```
user@kali: ~  
File Actions Edit View Help  
(user@kali)-[~]  
$ bash Factorial.sh  
Enter number:  
5  
120  
(user@kali)-[~]  
$
```

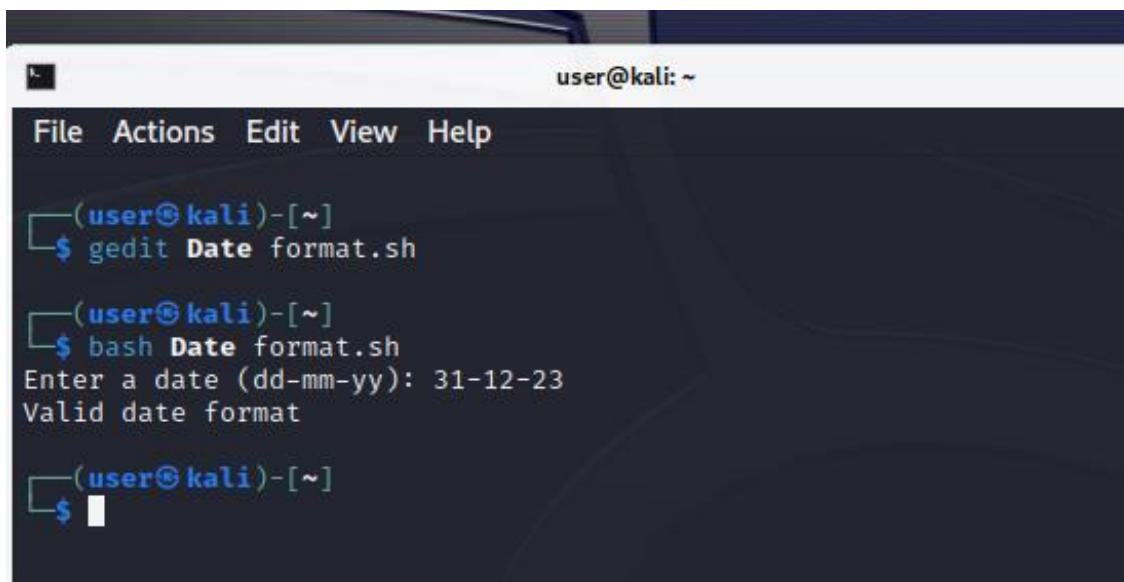
2.6 To validate the entered date. (eg. Date format is: dd-mm-yy)

Step 1: create and save the shell script with gedit command



```
user@kali: ~  
File Actions Edit View Help  
(user@kali)-[~]  
$ gedit Date format.sh  
(user@kali)-[~]  
$  
1 #!/bin/bash  
2 # Get user input  
3 read -p "Enter a date (dd-mm-yy): " entered_date  
4 clear  
5 # Check if the entered date matches the format  
6 if [[ $entered_date =~ ^[0-3][0-9]-[0-1][0-9]-[0-9]{2}$ ]];  
7     then  
8         echo "Valid date format"  
9     else  
10        echo "Invalid date format. Please use dd-mm-yy."  
11    fi
```

Step 2: Run the shell script with bash command



```
user@kali: ~  
File Actions Edit View Help  
(user@kali)-[~]  
$ gedit Date format.sh  
(user@kali)-[~]  
$ bash Date format.sh  
Enter a date (dd-mm-yy): 31-12-23  
Valid date format  
(user@kali)-[~]  
$
```