Shell Scripting in Linux

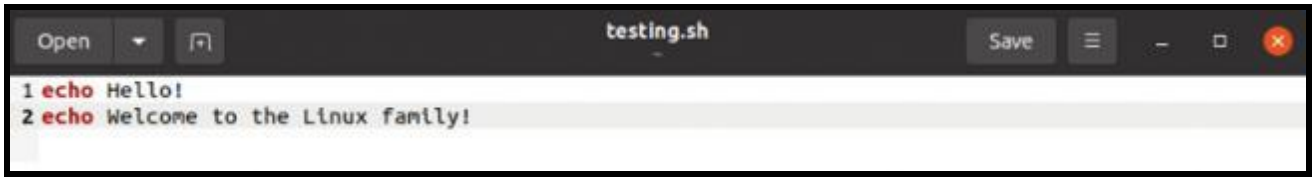
Shell Scripting is defined as an open-source program that's run by Linux or Unix shell. The user can write and store series of commands for the shell script to be executed by the Linux shell at any time or whenever it is needed .

Shell scripting makes programming in Linux easy. There are many methos to write shell scripts in Linux explained in the following:

Method 1: Using the Default Text Editor

To create a shell script using the default text editor, follow the steps given below:

***Step** 1:* Open the text editor in Linux, then create a text file having a ".sh" extension. Then type a simple script .



***Step** 2:* Open the terminal (don't change the current directory). Then write the command below, to give executable access to the file created.

```
chmod +x testing.sh
```

***Step** 3:* Execute the below given command in the terminal:

```
./testing.sh
```

This is a simple technique to create a shell script in Linux using the default **editor**.

Method 2: Using the Vim Text Editor Tool

Using Vim text editor tool, which is a tool used to write scripts in a Linux system, use the following command to install Vim, then open the vim tool

```
sudo apt install vim
```

***Step** 1:* For opening the editor, type vi or vim :

```
vim
```

***Step** 2:* open the terminal. Then create a bash file using **vi** command:

```
vi testing.sh
```

step 3: Press **i** from the keyboard and get into the Insert mode. then, type the following commands in it.

```
#! /bin/bash

echo “Welcome to the Linux family.”
```

***Step** 4:* Press **Esc** to exit this mode. Then type **:w** to **save** the script or use

:wq command that will also **save a file and exit** the text editor, Once saved, the shell script will appear as below.

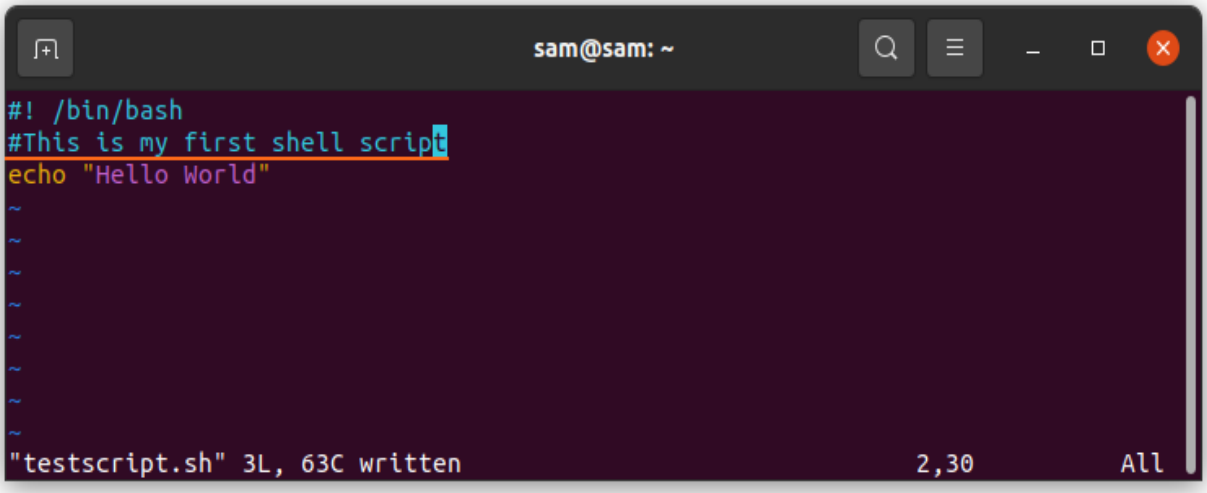
To **exit** without saving the changes type **q!** after the colon “ **:** “ and hit **Enter** .

To **rename** an existing file use the command **:w [newfilename]** that is used by adding the new name after the commands.

***Step** 5:* Go back to the console, and write:

```
bash testing.sh
```

This command will execute the shell file and will display the output in the terminal. Another example:



Adding Comments in Shell

Just like in many other programming languages, we use # (the hashtag symbol) to add comments in the shell script.

In the Vim editor, open the text file. And write a comment as below.

```
#! /bin/bash
# This is the testing script!
echo “Welcome to the Linux family.”
```

The above example illustrates using # symbol to add comments in a shell script.

Using Shell Variables

Variables are a value that can store information in the form of numbers and characters. Shell variables can store data as well. For example,

```
var ="Hello"
echo $var
```

The above lines will create a shell variable “**var**” and then will print it. Example:

```
#! /bin/bash
```

```
variable=“This is the testing shell script.”
echo $variable
```

To get the value of the variable, execute the command below.

```
bash script_example.sh
```

Example to write a script using a variable.

```
#!/bin/sh
echo "what is your name?"
read name
echo "How do you do, $name?"
read remark
echo "I am $remark too!"
```

Run the script file and enter the “name” as “Robin”.

```
bash scriptsample.sh
what is your name?
Robin
```

As you enter the input, the script reads the name and replies:

```
How do you do, Robin?
```

Then enter the “remark” as “good”. And you’ll notice that the script repeats the remark.

```
good
I am good, too!
```

The second line of the above set of codes shows the response from the script.