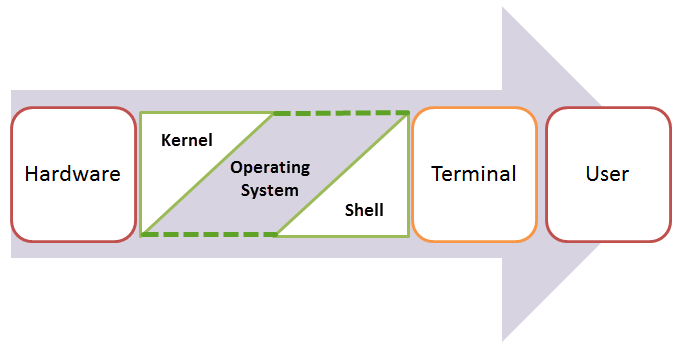
# **Introduction to Shell Scripting**

## **What is a Shell?**

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

* Kernel
* Shell



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 program's, 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**. The **Bourne Shell**: The prompt for this shell is $ and its derivatives are listed below:

* POSIX shell also known as sh
* Korn Shell also known as sh
* **B**ourne **A**gain **SH**ell also known as bash (most popular)

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

* C shell also known as csh
* Tops C shell also known as tcsh

We will discuss bash shell based shell scripting in this tutorial.

## **What is Shell scripting and why do I need it?**

**Writing a series of command** for the shell to execute **is called shell scripting**.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** a **vi** editor(or any other editor). Name script file with **extension .sh**
2. **Start** the script with **#! /bin/sh**
3. Write some code.
4. Save the script file as filename.sh
5. For **executing** the script type **bash filename.sh**

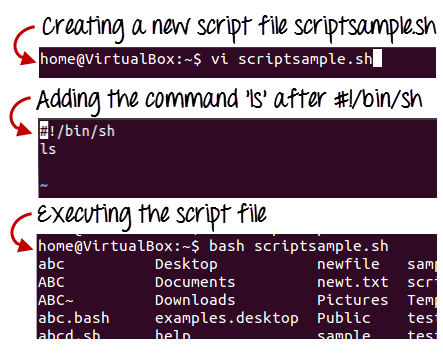
"#!" is an operator called shebang which directs the script to the interpreter location. So, if we use"#! /bin/sh" the script gets directed to the bourne-shell.

Let create a small script -

***#!/bin/sh***

***ls***

Let's see the steps to create it -



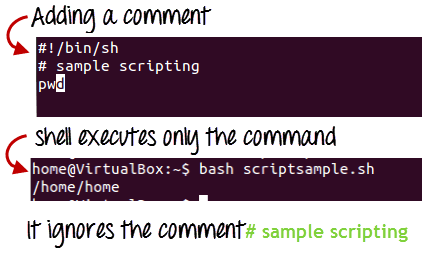
Command 'ls' is executed when we execute the scripsample.sh file.

## **Adding shell comments**

Commenting is important in any program. In Shell, the syntax to add a comment is

**#comment**

Let understand this with an example



## **What are Shell Variables?**

As discussed earlier, Variables store data in the form of characters and numbers. Similarly, Shell variables are used to store information and they can by the shell only.

For example, the following creates a shell variable and then prints it:

**variable ="Hello"**

**echo $variable**

Below is a small script which will use a variable.

**#!/bin/sh**

**echo "what is your name?"**

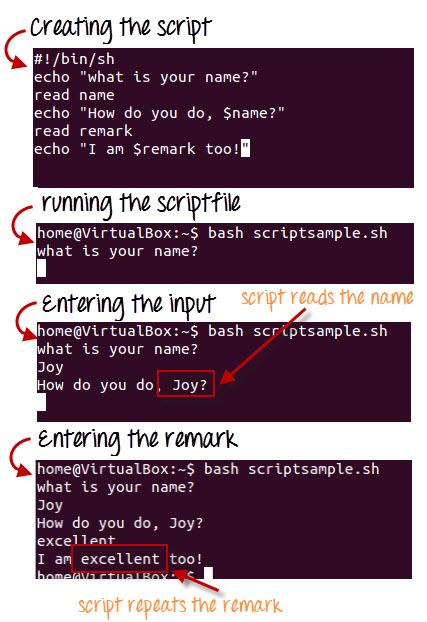
**read name**

**echo "How do you do, $name?"**

**read remark**

**echo "I am $remark too!"**

Let's understand, the steps to create and execute the script



As you see, the program picked the value of the variable 'name' as Joy and 'remark' as excellent.

This is a simple script. You can develop advanced scripts which contain conditional statements, loops and functions. Shell scripting will make your life easy and Linux administration a breeze.



### **Summary:**

* Kernel is the nucleus of the operating systems and it communicates between hardware and software
* Shell is a program which interprets user commands through CLI like Terminal
* The Bourne shell and the C shell are the most used shells in Linux
* Shell scripting is writing a series of command for the shell to execute
* Shell variables store the value of a string or a number for the shell to read
* Shell scripting can help you create complex programs containing conditional statements, loops and functions