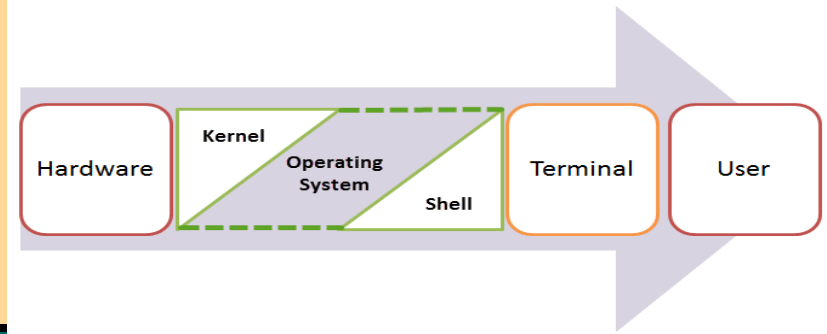


Systems Programming

Shell Scripting

What is a Shell?



- An Operating system is made of many components but its two prime components are -
 - Kernel
 - 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.
 - Shell
 - 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

- **1. The Bourne Shell:**

- The prompt for this shell is \$ and its derivatives are listed below:
 - POSIX shell (Portable Operating System Interface) also known as psh
 - Korn Shell also known as ksh
 - **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 known as csh
 - Tops C shell also known as tcsh

Shell scripting

- 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..
- steps in creating a Shell Script
 - Create a file using a vi editor(or any other editor). Name script file with extension .sh
 - Start the script with `#!/bin/sh`
 - Write some code.
 - Save the script file as filename.sh
 - 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`

Creating a new script file `scriptsample.sh`

```
home@VirtualBox:~$ vi scriptsample.sh
```

Adding the command 'ls' after `#!/bin/sh`

```
#!/bin/sh  
ls  
  
~
```

Executing the script file

```
home@VirtualBox:~$ bash scriptsample.sh  
abc      Desktop      newfile      samp  
ABC      Documents    newt.txt     scri  
ABC~     Downloads    Pictures     Temp  
abc.bash  examples.desktop  Public      test  
abcd.sh  help         sample      test
```

Adding shell comments

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

Adding a comment

```
#!/bin/sh  
# sample scripting  
pwd
```

shell executes only the command

```
home@VirtualBox:~$ bash scriptsample.sh  
/home/home
```

It ignores the comment # sample scripting

Shell Variables

- Shell variables are used to store information and they can be used 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!"**

Shell Scripts

- 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

Creating the script

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

running the scriptfile

```
home@VirtualBox:~$ bash scriptsample.sh
what is your name?
```

Entering the input

script reads the name

```
home@VirtualBox:~$ bash scriptsample.sh
what is your name?
Joy
How do you do, Joy?
```

Entering the remark

```
home@VirtualBox:~$ bash scriptsample.sh
what is your name?
Joy
How do you do, Joy?
excellent
I am excellent too!
```

script repeats the remark

Virtual Terminal

- **Virtual Terminals are similar to Terminal** that you have been using so far. They are **used for executing commands and offering input**. The only difference is that you **cannot use the mouse with the Virtual Terminals**. Therefore, you need to know the keyboard shortcuts.
- Virtual Terminals enable a **number of users to work on different programs at same time on the same computer**. This is the reason they are one of the most **distinguished feature of Linux**

Starting a Virtual Terminal

Starting a Virtual terminal

1) Press Ctrl+Alt+F1

```
Ubuntu 11.10 VirtualBox tty1
VirtualBox login: _
```

2) Enter User ID and Password

```
Ubuntu 11.10 VirtualBox tty1
VirtualBox login: home
Password: _
```

3) Now the Virtual Terminal is ready to work on

```
Ubuntu 11.10 VirtualBox tty1
VirtualBox login: home
Password:
Last login: Mon Sep 17 23:22:09 IST 2012 on tty1
Welcome to Ubuntu 11.10 (GNU/Linux 3.0.0-12-generic i686)

 * Documentation:  https://help.ubuntu.com/

178 packages can be updated.
178 updates are security updates.

home@VirtualBox:~$
```

Navigating through virtual terminal

- ***Ctrl + Alt + F (1 to 6) key***
- F1 being the first while F6 being the last virtual terminal.
- You can work on all of at the same time.
- In order to know which virtual terminal you are working on, note **tty** given at the top.
- tty is the teletype number which you can also know by typing the command "tty".

The seventh terminal

The seventh terminal is the one which we have been using so far in Linux tutorials. It can be accessed by pressing the below given key combination.

Ctrl + Alt + F7

Virtual Terminal shortcuts

Shortcut	Function
Home or Ctrl + a	Move the cursor to the start of the current line
End or Ctrl + e	Move the cursor to the end of the current line
Tab	Autocomplete commands
Ctrl + u	Erase the current line
Ctrl + w	Delete the word before the cursor
Ctrl + k	Delete the line from the cursor position to the end
reset	Reset the terminal
history	List of commands executed by the user
Arrow up	Scroll up in history and enter to execute
Arrow down	Scroll down in history and enter to execute
Ctrl + d	Logout from the terminal
Ctrl + Alt + Del	Reboot the system