**Introduction to Linux/Unix/FreeBSD/Mac OSX: - Shell**

A Unix/Linux Shell is a command line interpreter which interacts with kernel on the user’s behalf. User can directly enter command on the shell or as sequence of commands in a file (typically called the shell script) to interact with the system and hardware.  
Shell also provides wildcard interpretation, piping of commands, command substitution and control structures for conditional operation.

There are many different types of shell available for Linux and Unix but bash is most predominantly used amongst all of them for Linux. In Unix, bash is also used now a day, but in older days Unix used to have default shell as sh (Bourne Shell) which is a predecessor to Bash Shell (Bourne again Shell)

Other types of shell which one can encounter is ash, tcsh, csh(C- Shell), ksh(Korn Shell) etc.  
  
All the shells have different set of commands which can be run, and different ways of handling conditional operator, however the shells are mainly based on two broad categories  
  
1) c-Shell categories -> tcsh, csh  
2) Bourne shell -> sh, bash, ksh, pdsh  
  
In this course we will learn and practice based on bash shell as its most widely used and most popular at the moment. If you have some other shell based on bourne shell categories, but different than bash shell then most of the concept and command would still apply except for few.

Note: csh or c-Shell is completely different and it is not in scope of this course.

The Unix/Linux/Mac OS X shell is the main front end using which a user can interact with the operating system (There are GUI programs to interact as well like X11, but it’s not in this course scope.)

Interaction with the shell help in interpretation of the User command, expands user’s argument and present in a format which the kernel can understand. Shell  
uses system call as an interface to interact with the kernel.  
Shell hides the nitty gritty of the operating system and manages the details under hood while providing a clean and simple interface to the user to interact.  
  
On some host like Mac OSX or Xwindow based Linux, users may have never interacted with the Shell directly, but the shell are always built in these operating system and a Shell gets active as soon as the user logs in to the system.  
  
In Linux/Unix/Mac OS X the default shell a user logs in is present in the user profile

First Unix Shell (sh) was written by Ken Thompson at Bell Labs from year 1971 to 1975.

The Bourne Shell was rewritten by Stephen Bourne and was released in the year 1979 along with Unix release.

C-shell (csh) was written by Bill Joy from university of California, Berkeley as a graduate student in late 1970’s

**Configuration Files:-**

When a user logs into the system a shell is invoked. The type of shell depends on user’s preference in /etc/passwd which is being set by system administrator during the time of user account(login) creation.

A typical entry in /etc/passwd for the user root is given below: -

*root:x:0:0:root:/root:****/bin/bash***

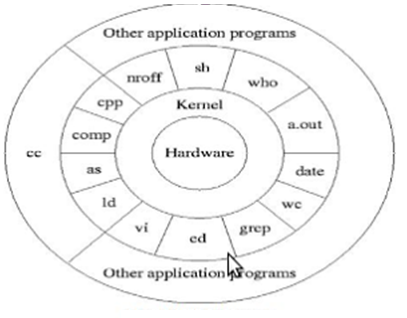
The explanation of the above line with each field separated by “:” is as follows.

Username : Password required : user ID: group ID: Description of User: Home directory: **Shell to be used**

|  |  |
| --- | --- |
| **User** | |
| **Shell** | **Applications** |
| **System** **Call** | |
| **Kernel** | |
| **Hardware** | |

Shell layer with respect to Linux/Unix/FreeBsd/Mac OSX Operating system

Kernel is the heart of any operating system. In case of OS likeUbuntu/Fedora/CentOS the kernel is Linux. The kernel manages various hardware resources in the system like CPU, memory, disk, display, serial port, Network, Different type of I/O’s and almost everything which is physically present**.** Without the kernel an OS cannot be envisioned.   
  
In contrast, the Shell interacts with kernel on user’s behalf and massaged and formats the output in the way Kernel understands.

A typical layout of the Shell, kernel and different applications in the system would look like below.