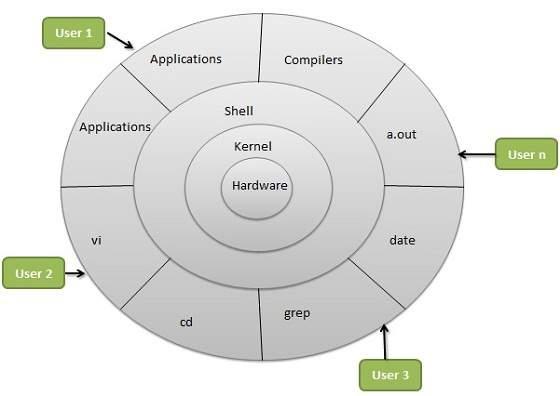
**Lab 3- LINUX ARCHITECTURE**

**Linux Architecture**

The architecture of a Linux System consists of the following layers −

* **Hardware layer** − Hardware consists of all peripheral devices (RAM/ HDD/ CPU etc).
* **Kernel** − It is the core component of Operating System, interacts directly with hardware, provides low level services to upper layer components.
* **Shell** − An interface to kernel, hiding complexity of kernel's functions from users. The shell takes commands from the user and executes kernel's functions.
* **Utilities** − Utility programs that provide the user most of the functionalities of an operating systems



Linux Architecture

**Kernel**

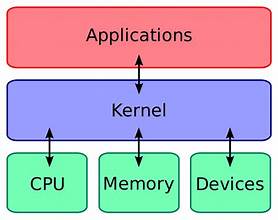
Thekernelis the core component of an operating system. This provides a platform for programs and various services to run on top of it. The Linux kernel is modifiable according to the user’s needs. Overall, the Linux Operating System and Linux kernel together provide a strong and user-friendly platform.

In a General-Purpose Computer running many processes simultaneously, we need a middle layer to manage the distribution of the computer’s hardware resources efficiently and fairly among all the various processes running on the computer. This middle layer is referred to as the **kernel**. The kernel virtualizes the computer’s common hardware resources to provide each process with its own virtual resources. This makes the process seem as if it is the sole process running on the machine. The kernel is also responsible for preventing and mitigating conflicts between different processes.

**Types of Kernel**

There are some of the important kernel types which are mentioned below:

* Monolithic Kernel
* Micro kernels:
* Exo kernels
* Hybrid kernels



Understanding the Linux Kernel

**Monolithic Kernel**

Monolithic kernels are larger than microkernels because they house both kernel and user services in the same address space. Monolithic kernels use a faster system call communication protocol than microkernels to execute processes between the hardware and software. They're less flexible than microkernels and require more work; admins must reconstruct the entire kernel to support a new service.

Linux Kernel’s uses monolithic kernel architecture

**Micro Kernel**

Microkernels have all their services in separate address spaces from the kernel. Microkernels use message passing for their communication protocol, which sends data packets, signals and functions to the correct processes. Microkernels also provide greater flexibility than monolithic kernels; to add a new service, admins modify the user address space for a microkernel.

A kernel where the kernel and user services are kept in separate address areas.MINIX 3 is an example of a microkernel.

**Exo Kernel**

Exokernels are unique in that they expose hardware resources directly to applications. Instead of abstracting hardware functionality like other kernel types, exokernels enable applications to implement their own abstractions and management policies. This means that application developers can make the most efficient use of resources for each program. Exokernels also come with library OSes, which can export different APIs.

Exokernels provide applications with better control and flexibility, which can add to performance gains.

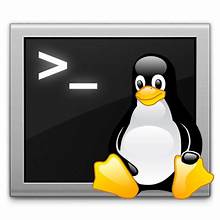
**Hybrid Kernel**

Apple developed the XNU OS kernel in 1996 as a hybrid of the Mach and Berkeley Software Distribution (BSD) kernels and paired it with an Objective-C application programming interface ([API](https://www.techtarget.com/searchapparchitecture/definition/application-program-interface-API)). Because it's a combination of the monolithic kernel and microkernel, it has increased modularity, and parts of the OS gain memory protection.A mix of monolithic and microkernel designs that combines the performance of a monolithic kernel with the modularity of a microkernel. Windows NT and macOS (since version X) are examples of hybrid kernels.

**Shell**

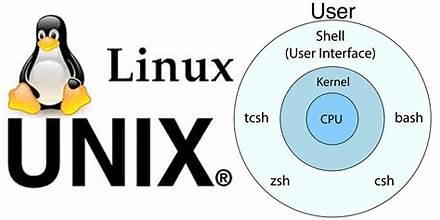
A shell is a special user program that provides an interface for the user to use operating system services. A shell is a program through which users can interact with the Operating System.

Shell accepts human-readable commands from users and converts them into something which the kernel can understand. It is a command language interpreter that executes commands read from input devices such as keyboards or from files. The shell gets started when the user logs in or starts the terminal.



**Types of Shell**

* The C Shell
* The Bourne Shell
* The Korn Shell
* GNU Bourne-Again Shell (BASH)



Shell in Linux

**The C Shell**

Bill Joy created it at the University of California at Berkeley. It incorporated features such as aliases and command history. It includes helpful programming features like built-in arithmetic and C-like expression syntax. In C shell:

Command full-path name is /bin/csh,  
 Non-root user default prompt is hostname %,   
 Root user default prompt is hostname #.

**The Bourne Shell**

It was written by Steve Bourne at AT&T Bell Labs. It is the original UNIX shell. It is faster and more preferred. It lacks features for interactive use like the ability to recall previous commands. It also lacks built-in arithmetic and logical expression handling. It is default shell for Solaris OS. For the Bourne shell the:

Command full-path name is /bin/sh and /sbin/sh,  
 Non-root user default prompt is $,  
 Root user default prompt is #.

**The Korn Shell**

It was written by David Korn at AT&T Bell Labs. It is a superset of the Bourne shell. So it supports everything in the Bourne shell.It has interactive features. It includes features like built-in arithmetic and C-like arrays, functions, and string-manipulation facilities. It is faster than C shell. It is compatible with script written for C shell. For the Korn shell the:

Command full-path name is /bin/ksh,  
 Non-root user default prompt is $,  
 Root user default prompt is #.

**GNU Bourne-Again Shell**

It is compatible to the Bourne shell. It includes features from Korn and Bourne shell.

Command full-path name is /bin/bash,  
Default prompt for a non-root user is bash-g.gg$   
(g.ggindicates the shell version number like bash-3.50$),  
Root user default prompt is bash-g.gg#.