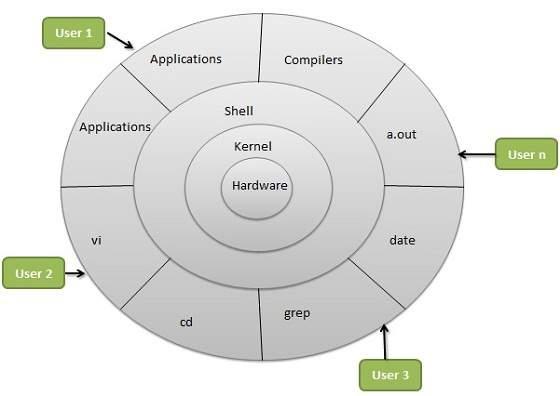
**Linux Architecture**



Linux architecture consists of several layers:

1. **Kernel**: The core component that manages hardware resources and system processes.
2. **System Libraries**: Standard functions that applications use to communicate with the kernel.
3. **System Utilities**: Tools and programs for performing specific tasks (e.g., file management, networking).
4. **User Interface**: The interface users interact with, which can be command-line (CLI) or graphical (GUI).
5. **Applications**: Software programs that run on Linux, utilizing libraries and utilities.

**Kernel**

The Linux kernel is the central component of the Linux operating system. It acts as a bridge between hardware and software, managing system resources and enabling communication between applications and the hardware.

A microkernel delegates user processes and kernel services in different address spaces.

A monolithic kernel implements services in the same address space.

A hybrid kernel, such as the Microsoft Windows NT and Apple XNU kernels, attempts to combine the behaviors and benefits of microkernel and monolithic kernel architectures.

A nanokernel focuses on providing minimal services limited to low-level hardware management, delegating most other services to higher-level modules.

An exokernel exposes hardware resources directly to applications, giving them more control over hardware.

A multi-kernel uses multiple kernels to manage different hardware resources, commonly used in distributed environments.

**Types of Kernel**

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

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

**Here are some important types of kernels:**

**1. Monolithic Kernel: All core services run in a single address space, allowing efficient communication. This design offers high performance but may be less stable, as a failure can crash the entire system.**

**2. Microkernel: Only essential functions are kept in the kernel, with most services running in user space. This improves stability and security but may introduce performance overhead due to more context switching.**

**3. Exokernel: Exposes hardware resources directly to applications, giving them control over resource management. This minimalistic approach can enhance performance but requires developers to manage hardware interactions effectively.**

**4. Hybrid Kernel: Combines elements of monolithic and microkernel architectures. Essential services are retained in the kernel for performance, while other services may run in user space, balancing efficiency and modularity.**

**SHELL**

**A shell is a program through which users can interact with the Operating System.**

**Linux provides commonly 4 types of Shells**

**The C Shell**

**The Bourne Shell**

**The Korn Shell**

**GNU Bourne-Again Shell (BASH)**

**The 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(sh)**

**Command full-path name is /bin/sh and /sbin/sh,**

**Non-root user default prompt is $,**

**Root user default prompt is #.**

**The Korn Shell**

**Command full-path name is /bin/ksh, non-root user default prompt is $,**

**Root user default prompt is #.**

**GNU Bourne-Again Shell**

**Command full-path name is /bin/bash,**

**Default prompt for a non-root user is bash-g.gg$**

**(g.gg indicates the shell version number like bash-3.50$),**

**Root user default prompt is bash-g.gg#.**