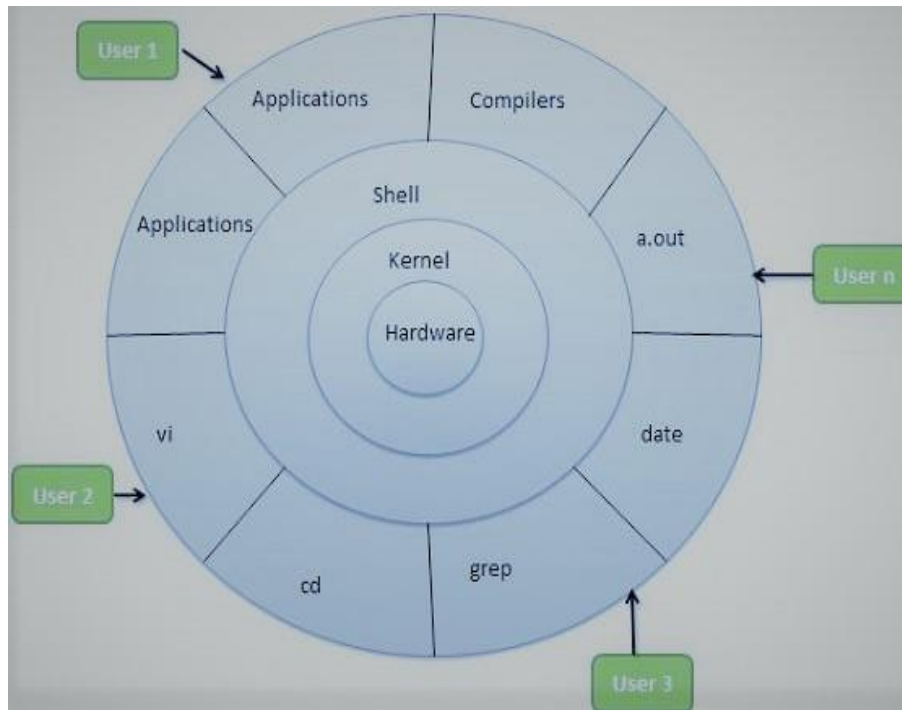# GETTING STARTED

WITH

LINUX ARCHITECTURE

➢ LINUX ARCHITECTURE:

THE SHELL:

- Shell is like a Container.
- It's an Interface between users and Kernel.
- In Linux, we usually mean a command line shell.
- The default shell in Linux **BASH**, the **GNU Bourne Again Shell**.
- The shell is most commonly used to run commands.

TYPES OF LINUX SHELLS:

- GNOME (GNU Network Object Model Environment)
- KDE (K Desktop Environment)
- Sh (Bourne Shell)
- Bash (Bourne Again Shell)
- ksh (Korn Shell)
- scheme shell
- csh and tcsh (tenex-c-shell)
- zsh (z shell)
    - **Find your shell:**
        $echo $0

LINUX KERNEL:

- Kernel is a heart of Linux OS.
- The Linux kernel is the main component of a Linux operating system.
- It is the core interface between a computer's hardware and its processes.
- What the kernel does:
  - Memory management
  - Process Management
  - Device drivers
  - System calls and security

WHERE THE KERNEL FITS WITHIN THE OS:

- To put the kernel in context, you can think of a Linux machine as having 3 layers:
  - The hardware
  - The Linux kernel
  - User processes

KERNEL SPACE:

- The kernel provides abstraction for security, hardware, and internal data structures.

USER SPACE & PROCESSES:

USER SPACE:

- User Space consists of all processes running outside the Kernel.
- Most Unix-like operating systems (including Linux) come pre-packaged with all kinds of utilities, programming languages, and graphical tools - these are user space applications.

PROCESSES:

- A process is an instance of a computer program that is currently being executed.
- Each Process has its own a private virtual memory space, a security context, a state, a process id number.

TYPES OF PROCESS:

USER PROCESSES:

- Most processes in the system are user processes.
- A user process is one that is initiated by a regular user account and runs in user space.

DAEMON PROCESSES:

- A daemon process is an application that is designed to run in the background, typically managing some kind of ongoing service.
- A daemon process might listen for an incoming request for access to a service.
- For example, the httpd daemon listens for requests to view web pages.
- daemon processes are typically managed as services by the root user.

KERNEL PROCESSES:

- Kernel processes execute only in kernel space. They are similar to daemon processes.
- The primary difference is that kernel processes have full access to kernel data structures, which makes them more powerful than daemon processes that run-in user space.
- Kernel processes also are not as flexible as daemon processes. You can change the behaviour of a daemon process by changing configuration files and reloading the service.
- Changing kernel processes, however, may require recompiling the kernel.