Understanding Linux Privilege Levels:

Linux, like many modern operating System, employs a privilege separation machanism to enhance security and stability.

Kernel Mode:

Kernal Mode, also known as supervisor mode or privileged mode, is the higest privilege level in the Linux operating system.

In this mode, the executing code has unrestricted access to all system resources, including hardware and memory.

Direct Hardware Access: The kernal can directly with hardware devices, such as disk drives, network interfaces and graphics cards.

Memory Management: The Kernal has ability to manage memory allocation and deallocation.

Process Scheduling: The Kernal is responsible for scheduling processes and managing CPU time.

System call: The Kernal provides a set of system calls that user applications calls that user applications ~~call~~ can invoke to request services.

Inter-Process Communication (IPC): The Kernal facilitates communication between processes through mechanisms like pipes, message queues and shared memory.

User Mode:

It is the lower privilege level in the Linux operating system. Applications and processes run in this mode, which

restricts their access to system resource and hardware.

Application Execution: User mode allows applications to execute codes, but they cannot directly access hardware or critical system resources.

Limited Memory Access: Each user mode process operates within its Virtual memory space.

System call Invocation: User mode applications can request services from the kernel through system calls.

File Operations: User mode applications can perform file operations.

User Interface Interaction: Applications running in user mode can interact with the graphical user interface or command line, allowing user to inputput data and receive input.

Example of operations permitted at each level.

→ Imagine a text editor application that a user is running. The applications operates in user mode and allows the user to create and edit text files. When the user saves a file, application call the 'write()' system call to save the data. This call transitions the execution to Kernal mode, where the Kernel checks permissions, writes the data to the disk, and then returns control to the text editor in user mode.