

CPU Modes

In an operating system, instructions executed by the CPU can be classified into privileged and non-privileged instructions. These classifications help the operating system ensure security, stability, and efficient resource management.

Privileged Mode (Kernel Mode)

- Privileged instructions can only be executed by the OS kernel or a privileged process, such as a device driver
- These instructions typically perform operations that require direct access to hardware or other privileged resources, such as setting up memory mappings or accessing I/O devices
- Characteristics
 - If any attempt is made to execute a Privileged Instruction in User Mode, it will not be executed and will be treated as an illegal instruction => The Hardware traps it in OS
 - Before transferring the control to any User Program, it is the responsibility of the Operating System to ensure that the Timer is set to interrupt. Thus, if the timer interrupts then the Operating System regains control
 - Privileged Instructions are used by the Operating System to achieve correct operation
- Examples
 - I/O instructions and Halt instructions
 - Turn off all Interrupts
 - Set the Timer
 - Context Switching
 - Clear the Memory or Remove a process from the Memory
 - Modify entries in the Device-status table
- Role of OS
 - Controlling access to privileged instructions
 - Memory protection
 - Interrupt handling: the execution of privileged instructions or exceptions is handled by OS through interrupt handling
 - Virtualization: allows OS to create a simulated environment where processes can execute privileged instructions without having direct access to the underlying hardware. Thus, it creates a more secure and isolated execution environment for privileged instructions by limiting process access to authorized hardware resources only.

Unprivileged Mode (User Mode)

- Non-privileged instructions can be executed by any process, including user-level processes
- used for performing computations, accessing user-level resources such as files and memory, and managing process control
- are executed in user mode, which provides limited access to system resources and ensures that processes cannot interfere with one another
- Examples
 - Reading the status of Processor

- Reading the System Time
 - Generate any Trap Instruction
 - Sending the final printout of Printer
- In order to change the mode from Privileged to Non-Privileged, we require a Non-privileged Instruction that does not generate any interrupt

References

- [Privileged and Non-Privileged Instructions in Operating System - GeeksforGeeks](#)