

Operating System Operations -

Operating systems are interrupt driven.

Events are always signaled by occurrence of an interrupt or a trap.

A trap (or Exception) is a software generated interrupt caused either by an error or specific request from a user program, that an OS service is performed.

For each interrupt, separate segment of code in the operating system determine what action should be taken.

Interrupt Service routine is provided to deal with the interrupt.

→ If a process stuck in infinite loop, this could prevent the correct operation of many other process.

A OS must ensure that an incorrect / malicious program cannot cause other programs to execute incorrectly.

Dual Mode & Multimode Operation -

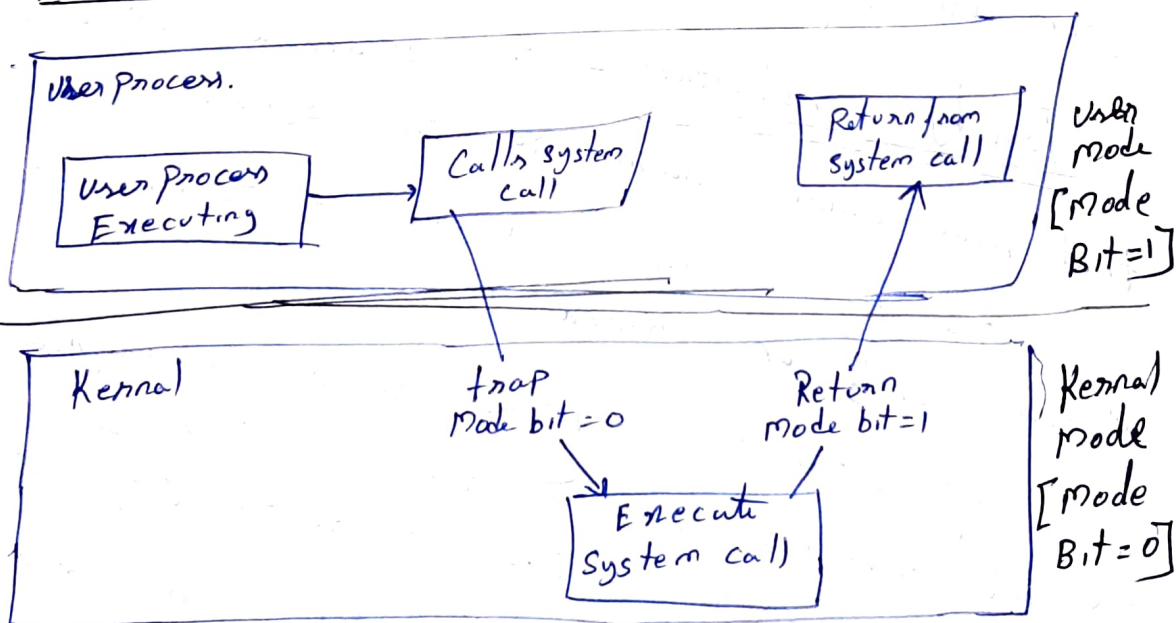


Fig 1-10 Transition from user to kernel mode.

To ensure proper execution of OS, we should distinguish between execution of OS code & user defined code.

2 Mode of Operations

↳ User Mode

↳ Kernel Mode / Supervisor Mode / System Mode / Privileged Mode.

A Bit called Mode Bit is added to the hardware of the computer to indicate current mode.

For Kernel, Mode Bit = 0

For User, Mode Bit = 1.

When a user application request a service from OS via sys call, the sys transition from user to kernel mode to fulfill the Request.

Privileged Instructions - The machine instructions that may cause harm to the system. The H/w allows privileged instructions to execute only on kernel mode.

If attempt is made to run in user mode, the h/w does not execute the instruction & treats it as illegal & traps it to the OS.

example of Priv. Inst. →

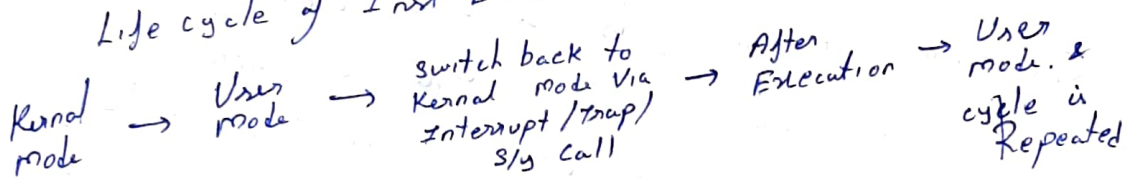
- * Instruction switch to kernel mode.
- * I/O control.
- * Timers Management & Interrupt Management.

Virtual Machine Manager (VMM)

The CPU that support virtualization frequently, have separate mode to indicate the VMM & the virtualization mgmt s/w is in control of the s/w.

VMM has high privilege than user mode but fewer than kernel.

Life cycle of Inst Execution



Timer - To ensure OS maintain control over the CPU.

We cant allow a user prog to get stuck in infinite loop or to fail to call s/w services & never return ctrl to the OS. To accomplish this goal, we can use a timer.

A timer can be set to interrupt the computer after a specified period. Timer may be fixed or variable.

(eg) For 10 bit counter with 1ms clock allows interval from 1ms to 1024 ms.

Before turning over control to user, the OS ensures that the timer is set to interrupt.

Ex If a counter is initialized to 420. For every second, the timer interrupt & counter is decremented by 1. As long as counter is '+ve', control is returned to user program. When counter is '-ve', OS terminate the program for exceeding the assigned time limit.

Process Management -

Process Example

- ↳ Program in Execution
- ↳ A time shared User Program - Compiler.
- ↳ User Executing program in PC.
- ↳ S/y Task like sending o/p to printer.

A process needs certain resources, including CPU time, memory files & I/O devices to accomplish its task.

Process must be sequential. The execution of line by line instruction execution until it completes the process. The CPU executes specify next instruction to execute.

The OS is responsible for the following activities under process management.

- * Scheduling processes & threads on the CPUs.
 - * Creating & deleting both user & s/y processes.
 - * Suspending & resuming processes.
 - * Providing mechanism for process synchronization.
 - * Providing mechanism for process communication.
-