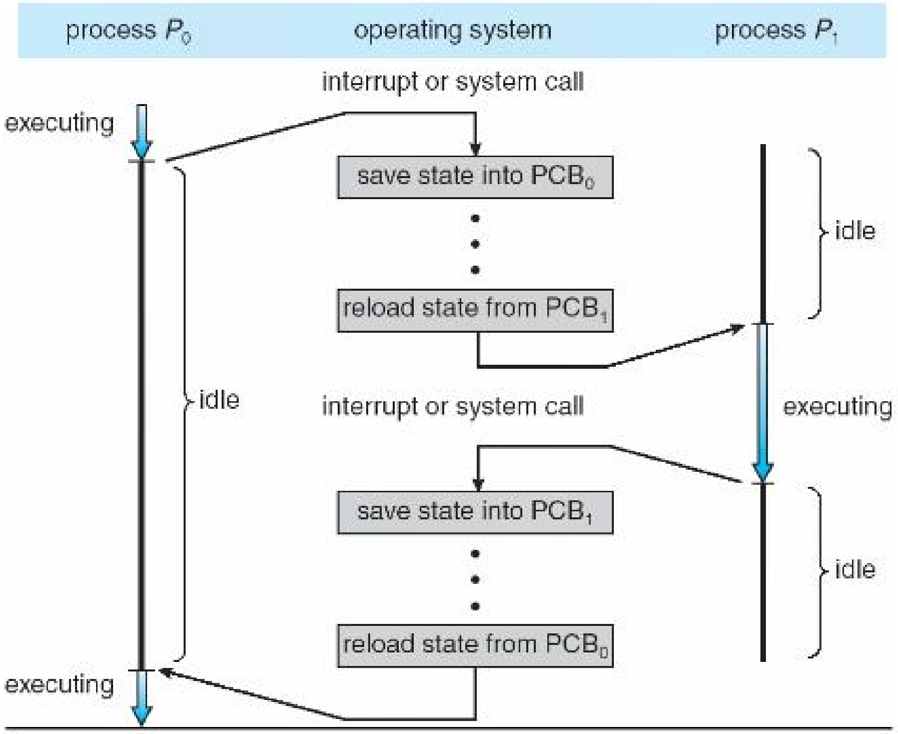
What is Context Switching?

- Involves switching of the CPU from one process or task to another.
- The execution of the process that is present in the **running state is suspended** by the kernel and another process that is present in **the ready state is executed** by the CPU.
- It is one of the essential features of the multitasking operating system.
- The processes are switched so fast – gives an illusion to the user that all the processes are being executed at the same time.
- Context switching can happen due to the following reasons:
    - a process of high priority comes in the ready state
    - an interruption occurs–process in the running state should be stopped
    - transition between the user mode and kernel mode is required



- **What is Critical Section Problem?**

- A critical section is a segment of code which can be accessed by a **signal** process at a specific point of time.
- The section consists of shared data resources that is required to be accessed by other processes.
- The **entry** to the critical section is handled by the **wait()** function, and it is represented as P().
- The **exit** from a critical section is controlled by the **signal()** function, represented as V().

- In the critical section, only a single process can be executed. Other processes, waiting to execute their critical section, need to wait until the current process completes its execution.

Process Synchronization

- **Rules for Critical Section**

- **Mutual Exclusion:**
  - It implies that only one process can be inside the critical section at any time.
  - If any other processes require the critical section, they must wait until it is free.
- **Progress:**
  - If a process is not using the critical section, then it should not stop any other process from accessing it.
  - In other words, any process can enter a critical section if it is free.
- **Bounded Waiting:**
  - Bounded waiting means that each process must have a limited waiting time.
  - It should not wait endlessly to access the critical section.

Process Synchronization

- **Race Condition**

- Race condition is a situation where-
  - The final output produced depends on the execution order of instructions of different processes.

  - Several processes compete with each other.

OS Services/Functions

Briefly, the OS typically provides services in the following areas:

- **Program development:** The OS provides a variety of facilities and services, such as editors and debuggers, to assist the programmer in creating programs. and are referred to as application program development tools.
- **Program execution:** A number of steps need to be performed to execute a program. Instructions and data must be loaded into main memory, I/O devices and files must be initialized, and other resources must be prepared. The OS handles these scheduling duties for the user.
- **Access to I/O devices**: Each I/O device requires its own peculiar set of instructions or control signals for operation. The OS provides a uniform interface that hides these details so that programmers can access such devices using simple reads and writes.
- **Controlled access to files:** For file access, the OS must reflect a detailed understanding of not only the nature of the I/O device (disk drive, tape drive) but also the structure of the data contained in the files on the storage medium.
- **System access:** For shared or public systems, the OS controls access to the system as a whole and to specific system resources. The access function must provide protection of resources and data from unauthorized users and must resolve conflicts for resource contention.
- **Error detection and response:** A variety of errors can occur while a computer system is running. These include internal and external hardware errors, such as a memory error, or a device failure or malfunction; and various software errors, such as division by zero, attempt to access forbidden memory location, and inability of the OS to grant the request of an application.
- **Accounting: A good OS will collect usage statistics for various resources and** monitor performance parameters such as response time.