

CPU Scheduling:

CPU scheduling is the operating system process of determining which ready task gets to use the processor, aiming to maximize CPU utilization, throughput, and fairness while minimizing waiting and turnaround times. It enables multitasking by switching the CPU among processes, using algorithms like FCFS, SJF, and Round Robin.

Key Concepts and Algorithms

- **Pre-emptive Scheduling:** The OS can interrupt a running process to allocate the CPU to a higher-priority task.
- **Non Pre-emptive Scheduling :** A running process keeps the CPU until it terminates or voluntarily waits.

Common Algorithms:

- **First-Come, First-Served (FCFS):** Processes are executed in the order they arrive; simple but can lead to long wait times (convoy effect).
- **Shortest Job First (SJF/SJN):** Selects the waiting process with the smallest burst time, minimizing average waiting time.
- **Round Robin:** Each process gets a small, fixed unit of time (time quantum) in a cyclic manner; good for time-sharing systems.
- **Priority Scheduling:** Allocates the CPU to the process with the highest priority.
- **Shortest Remaining Time First (SRTF):** A preemptive version of SJF.
- **Multilevel Queue/Feedback Queue:** Uses multiple queues with different priorities and allows processes to move between them.

Scheduling Criteria (Goals) When Scheduling Decisions Occur

- **Maximize CPU Utilization:** Keep the CPU as busy as possible.
- **Maximize Throughput:** Complete the maximum number of processes per time unit.
- **Minimize Turnaround Time:** Reduce the total time from submission to completion.
- **Minimize Waiting Time:** Reduce the time a process spends waiting in the ready queue.
- **Minimize Response Time:** Reduce the time from submission to the first response.

When Scheduling Decisions Occur

1. A process switches from running to waiting (e.g., I/O request).
2. A process switches from running to ready (e.g., interrupt).

3. A process switches from waiting to ready (e.g., I/O completion).
4. A process terminates.