CPU Scheduling

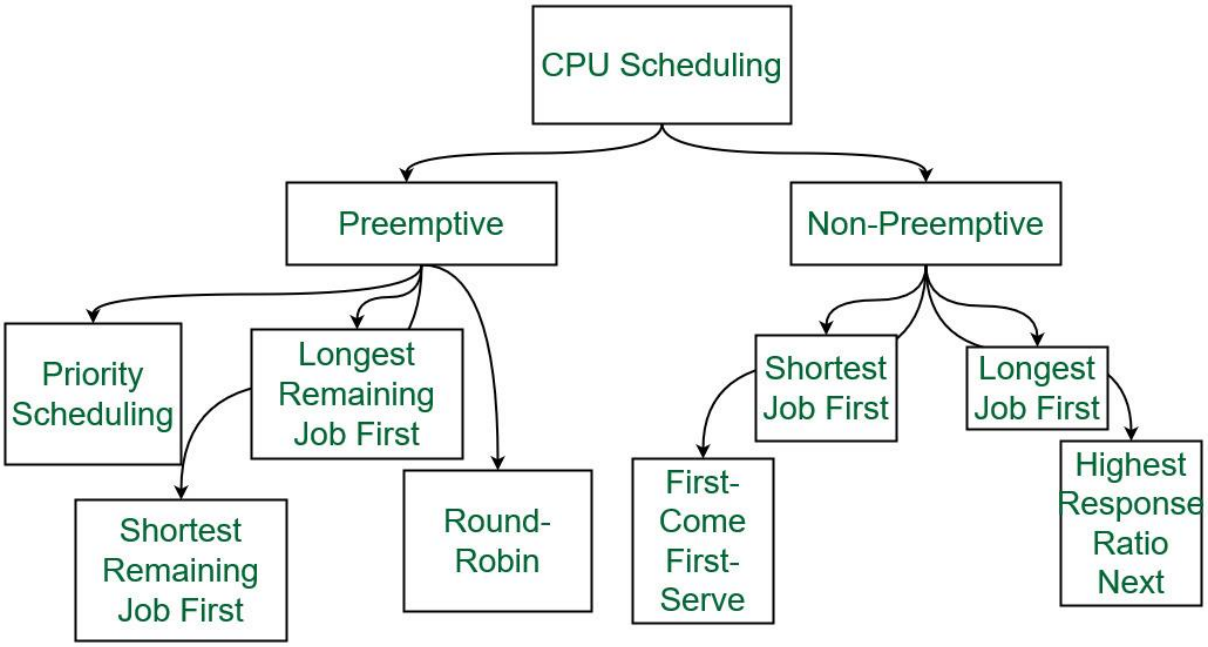
Overview

CPU scheduling is the task performed by the CPU that decides the way and order in which processes should be executed. There are two types of CPU scheduling - Preemptive, and non-preemptive. The criteria the CPU takes into consideration while "scheduling" these processes are - CPU utilization, throughput, turnaround time, waiting time, and response time.

There are mainly two types of scheduling methods:

Preemptive Scheduling: Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to the ready state.

Non-Preemptive Scheduling: Non-Preemptive scheduling is used when a process terminates , or when a process switches from running state to waiting state.



Types of CPU scheduling Algorithms:

There are several types of CPU scheduling algorithms, each with its own strengths and weaknesses. Some of the most common types of CPU scheduling algorithms include:

- **First-Come, First-Served (FCFS):** The simplest CPU scheduling algorithm, in which processes are executed in the order in which they arrive.

- **Shortest Job First (SJF):** This algorithm schedules processes based on their expected CPU burst time. The process with the shortest expected burst time is executed first.

- **Priority Scheduling:** This algorithm assigns a priority value to each process, and processes with higher priority values are executed first.

- **Round Robin (RR):** This algorithm allocates a fixed time slice (quantum) to each process in a cyclic order. If a process has not completed its execution within its allotted time slice, it is preempted, and the next process in the queue is executed.

- **Multilevel Queue Scheduling:** This algorithm divides the processes into separate queues based on their characteristics. Each queue has its own scheduling algorithm, and processes are moved between queues based on their priority or other criteria.

- **Multilevel Feedback Queue Scheduling:** This algorithm is an extension of multilevel queue scheduling that allows processes to move between queues based on their behavior. For example, a process that uses a lot of CPU time may be moved to a lower-priority queue to make room for other processes.

Comparison between Various CPU Scheduling Algorithms:

١. **First-Come, First-Served (FCFS):** Simple and easy to implement. This may result in long waiting times for processes with longer CPU burst times, causing poor response times.

٢. **Shortest Job First (SJF):** Results in shorter waiting times for processes with shorter CPU burst times. Difficult to predict CPU burst times accurately, which can lead to starvation for longer processes.

٣. **Priority Scheduling:** Can ensure that high-priority processes are executed first. This may result in lower-priority processes being starved of CPU time.

٤. **Round Robin (RR):** Ensures that all processes get an equal share of CPU time. This may result in high overhead due to frequent context switching.

٥. **Multilevel Queue Scheduling:** Allows for the separation of processes based on their characteristics, improving overall system performance. Requires more complex implementation and management.

٦. **Multilevel Feedback Queue Scheduling:** Allows for processes to be moved between queues based on their behavior, preventing starvation and ensuring that processes are executed fairly. More complex and difficult to implement compared to other scheduling algorithms.

Overall, there is no one-size-fits-all solution when it comes to CPU scheduling algorithms. The choice of algorithm depends on the specific needs of the system being used, such as the nature of the processes, the desired response times, and the overall system performance requirements.

Producer: Elham Jafari

Computer Engineering