

Scheduling algorithms are used by operating systems to manage the allocation of resources to processes. The primary objective of scheduling algorithms is to maximize the utilization of the CPU and minimize the waiting time of processes in the ready queue <sup>1</sup>. There are several scheduling algorithms that can be used to schedule jobs, each with its own advantages and disadvantages. Some of the most common algorithms include:

**First Come First Serve (FCFS):** This is the simplest algorithm to implement. It schedules processes in the order in which they arrive in the ready queue. The first process to arrive is the first to be executed <sup>123</sup>.

**Shortest Job First (SJF):** This algorithm schedules processes based on their burst time. The process with the shortest burst time is executed first <sup>123</sup>.

**Round Robin (RR):** This algorithm schedules processes in a circular queue. Each process is given a fixed time slice, called a quantum, to execute. If the process completes its execution within the quantum, it is removed from the queue. Otherwise, it is moved to the end of the queue <sup>123</sup>.

**Priority Scheduling:** This algorithm schedules processes based on their priority. The process with the highest priority is executed first. Processes with the same priority are scheduled in a round-robin fashion <sup>123</sup>.

**Multilevel Queue Scheduling:** This algorithm divides the ready queue into several separate queues, each with its own scheduling algorithm. Processes are assigned to a queue based on their characteristics, such as priority or memory requirements 2.

**Multilevel Feedback Queue Scheduling:** This algorithm is similar to multilevel queue scheduling, but it 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 2.

These algorithms are designed to balance the trade-off between maximizing CPU utilization and minimizing the waiting time of processes in the ready queue. Each algorithm has its own strengths and weaknesses, and the choice of algorithm depends on the specific requirements of the system