A scheduling algorithm is used to estimate the CPU time required to allocate to the processes and threads. The prime goal of any CPU scheduling algorithm is to keep the CPU as busy as possible for improving CPU utilization.

Scheduling Algorithms

1. First Come First Serve(FCFS): As the name implies that the jobs are executed on a first come first serve basis. It is a simple algorithm based on FIFO that's first in first out. The process that comes first in the ready queue can access the CPU first. If the arrival time is less, then the process will get the CPU soon. It can suffer from the convoy effect if the burst time of the first process is the longest among all the jobs.

2. Shortest Job First (SJF): Also known as the shortest job first or shortest job next is a non-preemptive type algorithm that is easy to implement in batch systems and is best in minimizing the waiting time. It follows the strategies where the process that is having the lowest execution time is chosen for the next execution.

3. Longest Job First Scheduling(LJF): The longest job first (LJF) is the non-preemptive version. This algorithm is also based upon the burst time of the processes. The processes are put into the ready queue according to their burst times and the process with the largest burst time is processed first.

4. Longest Remaining Time First Scheduling (LRTF): The preemptive version of LJF is LRTF. Here the process with the maximum remaining CPU time will be considered first and then processed. And after some time interval, it will check if another process having more Burst Time arrived up to that time or not. If any other process has more remaining burst time, so the running process will get pre-empted by that process.

5. Shortest Remaining Time First(SRTF): This algorithm is based on SJF and this is the preemptive version of SJF. In this scheduling algorithm, the process with the smallest remaining burst time is executed first and it may be preempted with a new job that has arrived with a shorter execution time.

6. Round Robin(RR): It is a preemptive scheduling algorithm in which each process is given a fixed time called quantum to execute. At this time one process is allowed to execute for a quantum and then preempts and then another process is executed. In this way, there is context switching between processes to save states of these preempted processes.

7. Priority Scheduling: It is a non-preemptive algorithm that works in batch systems and, each process is assigned with a priority and the process with the highest priority is executed first. This can lead to starvation for other processes.
8. Multiple Levels Queues Scheduling: In this scheduling, multiple queues have their scheduling Algorithms and are maintained with the processes that possess the same characteristics. For this, priorities are assigned to each queue for the jobs to be executed.
9. Multilevel-Feedback-Queue Scheduler: It defines several queues and the scheduling algorithms for each queue. This algorithm is used to determine when to upgrade a process when to demote a process, and also determine the queue in which a process will enter and when that process needs service.

Comparative Analysis of Different CPU Scheduling Algorithms

Here is a brief comparison between different CPU scheduling algorithms:

Algorithm	Allocation is	Complexity	Average waiting time (AWT)	Preemption	Starvation	Performance
FCFS	According to the arrival time of the processes, the CPU is allocated.	Not complex	Large.	No	No	Slow performance
SJF	Based on the lowest CPU burst time (BT).	More complex than FCFS	Smaller than FCFS	No	Yes	Minimum Average Waiting Time

Algorithm	Allocation is	Complexity	Average waiting time (AWT)	Preemption	Starvation	Performance
LJFS	Based on the highest CPU burst time (BT)	More complex than FCFS	Depending on some measures e.g., arrival time, process size, etc.	No	Yes	Big turn-around time
LRTF	Same as LJFS the allocation of the CPU is based on the highest CPU burst time (BT). But it is preemptive	More complex than FCFS	Depending on some measures e.g., arrival time, process size, etc.	Yes	Yes	The preference is given to the longer jobs
SRTF	Same as SJF the allocation of the CPU is based on the lowest CPU burst time (BT). But it is preemptive.	More complex than FCFS	Depending on some measures e.g., arrival time, process size, etc	Yes	Yes	The preference is given to the short jobs

Algorithm	Allocation is	Complexity	Average waiting time (AWT)	Preemption	Starvation	Performance
RR	According to the order of the process arrives with fixed time quantum (TQ)	The complexity depends on TQ size	Large as compared to SJF and Priority scheduling.	No	No	Each process has given a fairly fixed time
Priority Pre-emptive	According to the priority. The bigger priority task executes first	This type is less complex	Smaller than FCFS	Yes	Yes	Well performance but contain a starvation problem
Priority non-preemptive	According to the priority. with monitoring the new incoming higher priority jobs	This type is less complex than Priority preemptive	preemptive Smaller than FCFS	No	Yes	Most beneficial with batch systems

Algorithm	Allocation is	Complexity	Average waiting time (AWT)	Preemption	Starvation	Performance
MLQ	According to the process that resides in the bigger queue priority	More complex than the priority scheduling algorithms	Smaller than FCFS	No	Yes	Good performance but contain a starvation problem
MFLQ	According to the process of a bigger priority queue.	It is the most Complex but its complexity rate depends on the TQ size	Smaller than all scheduling types in many cases	No	No	Good performan