Unit-II
Lecture: 9
(Scheduling)

CPU scheduling is the basis of multiprogrammed operating systems.
The sole purpose of implementing process scheduling for a CPU is to utilize the processor and other resources to the fullest at any given time.

CPU-I/O Burst Cycle:

The success of CPU scheduling depends on an observed property of processes:

- process execution consists of a cycle of CPU execution and I/O wait.
- Processes alternate between these two states.

Process execution begins with a CPU burst. That is followed by an I/O burst, which is followed by another CPU burst, then another I/O burst and so on. Eventually, the final CPU burst ends with a system request to terminate execution.

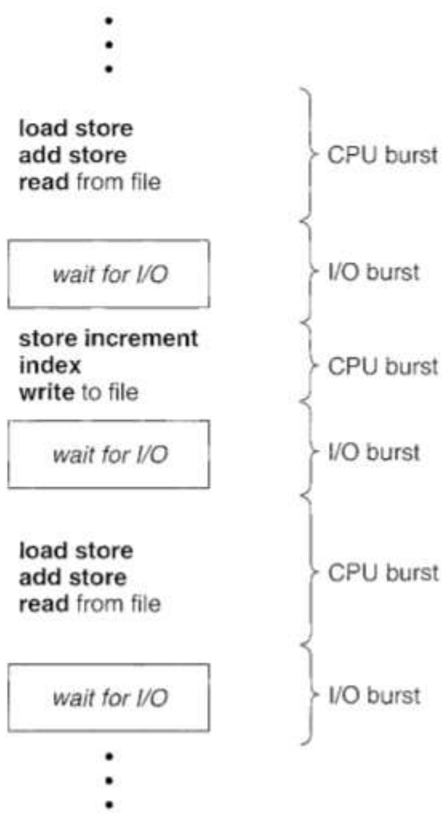


Fig: Alternating Sequence of CPU and I/O bursts

Thus, any process would typically require both CPU time and I/O time. So, while it is using the I/O resources, CPU remains idle and vice versa. To utilize the resources efficiently we can schedule other processes to utilize out idle resources.

What are Scheduling Queues?

- All processes, upon entering into the system, are stored in the Job Queue.
- Processes in the Ready state are placed in the Ready Queue.
- Processes waiting for a device to become available are placed in the Device Queues. There are unique device queues available for each I/O device.

A new process is initially put in the Ready Queue. It waits in the ready queue until it is selected for execution (or dispatched). Once the process is assigned to the CPU and is executing, one of the following several events can occur:

- The process could issue an I/O request and then be placed in the I/O queue.
- The process could create a new sub-process and wait for its termination.
- The process could be removed forcibly from the CPU, as a result of an interrupt and be put back in the ready queue.

Types of Schedulers:

1. **Long Term or Job Scheduler:** Long term scheduler runs less frequently. It brings the new process to the "Ready State".
    - It controls Degree of Multiprogramming i.e. number of process present in the ready state at any point of time.
    - Long term scheduler make a careful selection of both IO and CPU bound process.
        - IO bound tasks are which use much of their time in input and output operations while CPU bound processes are which spend their time on CPU.
    - The job scheduler increases efficiency by maintaining the balance between the two.

2. **Short Term or CPU scheduler:** It is responsible for selecting one process from ready state for scheduling it on the running state. It runs very frequently. The primary aim of this scheduler is to enhance the CPU performance and increase process execution rate.
    - Short term scheduler only selects the process to schedule it does not load the process on running. Here is when all the scheduling algorithms are used.
    - The CPU scheduler is responsible for ensuring there is no starvation owing to high burst time processes.
        - Dispatcher is responsible for loading the process selected by Short term scheduler on the CPU (Ready to running state).

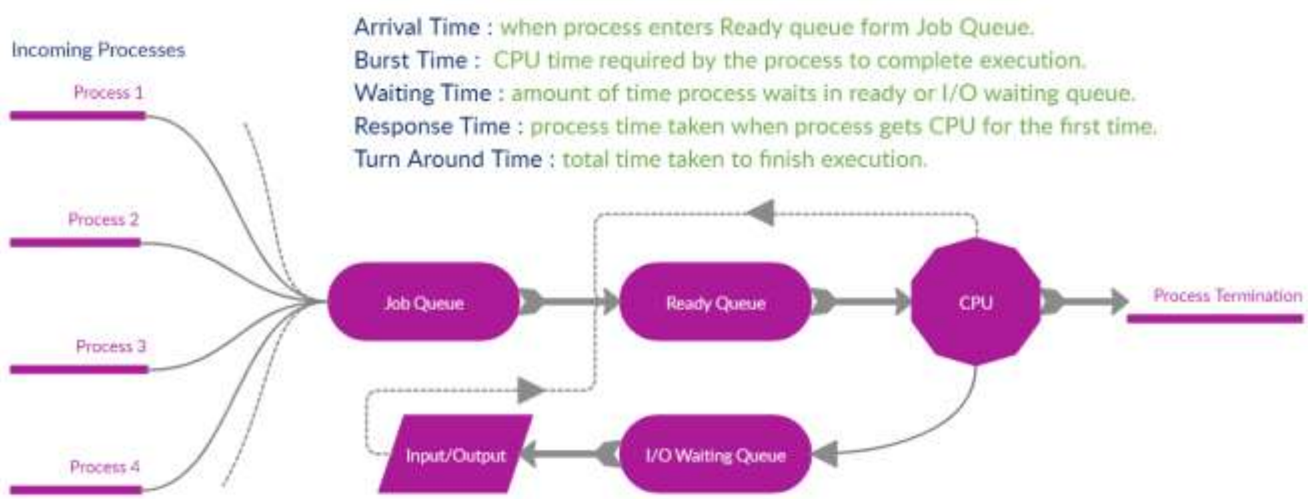
3. **Medium-term scheduler:** It is responsible for suspending and resuming the processes. It mainly does swapping (moving processes from main memory to disk and vice versa). It is helpful in maintaining a perfect balance between the I/O bound and the CPU bound. It reduces the degree of multiprogramming.

Objectives of a Process Scheduling Algorithms:

- Keeping CPU as busy as possible i.e. maximum CPU utilization.
- Make CPU allocation aboveboard.
- Minimizing the Turnaround time, waiting time and response time.

Turnaround time is the total time taken by the process to complete execution; it includes burst time (service time) and waiting time.

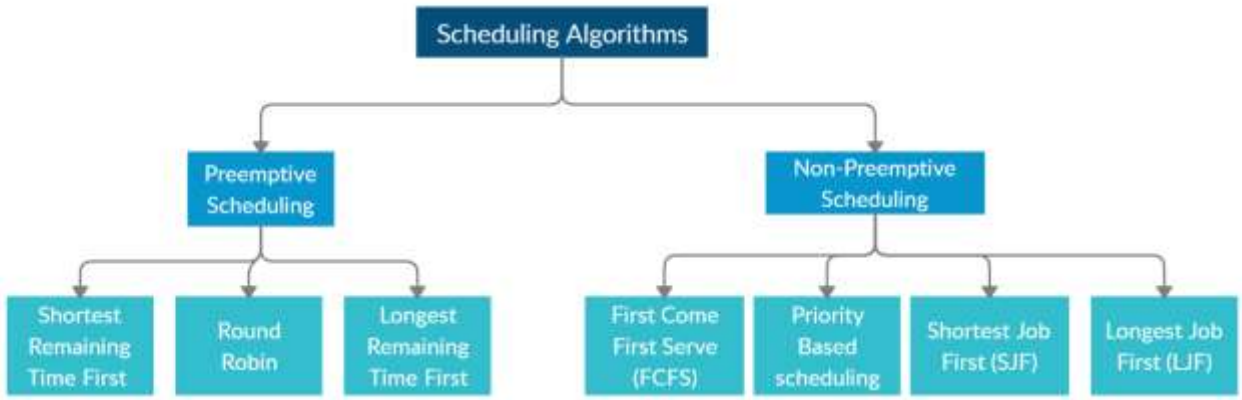
Turnaround time = Service time + waiting time



Preemptive Scheduling and Non-Preemptive Scheduling:

These are the two techniques to schedule the incoming processes:

- **Non Pre-empting Scheduling:** when the currently executing process gives up the CPU voluntarily.
- **Pre-emptive Scheduling**: When the operating system decides to favour another process, pre-empting the currently executing process.



Non- Preemptive Scheduling Algorithms —

1. **First Come First Serve(FCFS)**: Schedule the processes on the basis of their arrival time.

2. **Priority Based Scheduling:** Each process has a priority assigned to it, processes with higher priority gets the processor first. If processes have different Arrival Time in Priority Based Scheduling then we have to implement it using Preemptive Scheduling technique.

3. **Shortest Job First (SJF):** Processes with shortest Burst Time gets the CPU first. If we have different Arrival Time we will have to use Preemptive Scheduling technique.

4. **Longest Job First:** Processes with longest Burst Time gets the CPU first. This is similar to shortest job first scheduling algorithm.

Preemptive Scheduling Algorithms —

1. **Shortest Remaining Time First (SRTF):** Allot the processor to the process that has shortest remaining time first. Remaining time calculated as:

   Remaining Time = Total Burst time -CPU time already utilized by the process

2. **Round Robin:** Allot the processor to each of the process for a fixed time and then *"Context Switch"* to next processes in queue in a cyclic manner.

3. **Longest Remaining Time First:** Allot the processor to the process that has longest remaining time.