Chapter 3 CPU Scheduling algorithms

CPU Scheduling is a process of determining which process will own CPU for execution while another process is on hold. The main task of CPU scheduling is to make sure that whenever the CPU remains idle, the OS at least select one of the processes available in the ready queue for execution. The selection process will be carried out by the CPU scheduler. It selects one of the processes in memory that are ready for execution.

Scheduling of processes/work is done to finish the work on time. **CPU Scheduling** is a process that allows one process to use the CPU while another process is delayed (in standby) due to unavailability of any resources such as I / O etc, thus making full use of the CPU. The purpose of CPU Scheduling is to make the system more efficient, faster, and fairer.

Whenever the CPU becomes idle, the operating system must select one of the processes in the **ready queue** to be executed. The selection process is carried out by the short-term scheduler (or CPU scheduler). The scheduler selects from among the processes in memory that are ready to execute and allocates the CPU to one of them.

Types of CPU Scheduling

1. Long term scheduler

Long term scheduler is also known as job scheduler. It chooses the processes from secondary memory and puts them in the ready queue in main memory.

2. Short term scheduler

Short term scheduler is also known as CPU scheduler. OS use short term scheduler to selects one of the process from the ready queue and dispatch it to the CPU for the execution.

Short term scheduler use scheduling algorithm to select a process from ready queue.

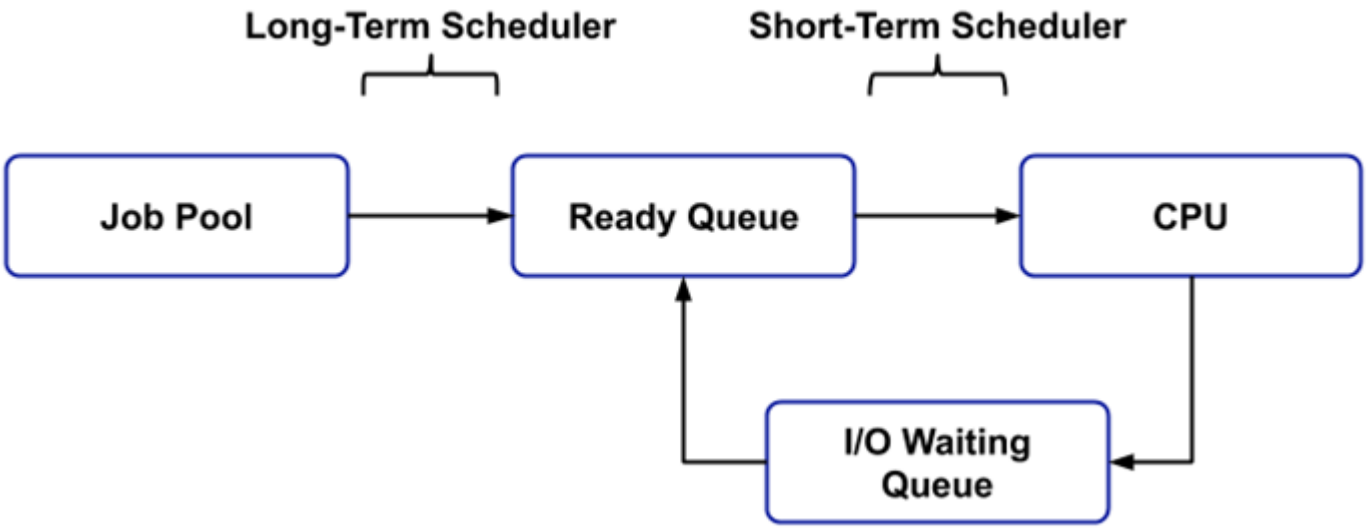
Starvation problem may occur if the short term scheduler makes some mistakes while choosing the process. As, selecting of that process which having very high burst time.

3. Medium term scheduler

It involves in swapping-out and swapping-in in-between the main memory and secondary memory.

It suspended the process and transfers it toward secondary memory from main memory and resumes the suspended process from secondary memory to primary memory.

So, in short medium term scheduler is responsible for suspending the process from main memory to secondary memory and resuming the processes from secondary to main memory.

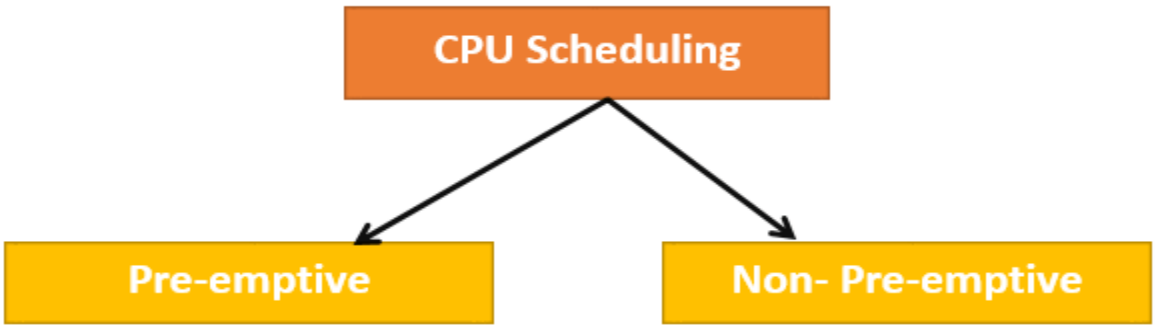


CPU scheduling decisions may take place under the following four circumstances:

1. When a process switches from the **running** state to the **waiting** state(for I/O request or invocation of wait for the termination of one of the child processes).
2. When a process switches from the **running** state to the **ready** state (for example, when an interrupt occurs).
3. When a process switches from the **waiting** state to the **ready** state(for example, completion of I/O).
4. When a process **terminates**.

What are the different types of CPU Scheduling Algorithms?

There are mainly two types of scheduling methods:



1. **Preemptive Scheduling:**

Preemptive scheduling is used when a process switches from running state to ready state or from the waiting state to ready state. **That process stays in the ready queue till it gets its next chance to execute.** Algorithms based on preemptive scheduling are: Round Robin (RR),Shortest Remaining Time First (SRTF), etc.

2. **Non-Preemptive Scheduling:**

Under non-preemptive scheduling, once the CPU has been allocated to a process, the process keeps the CPU until it releases the **CPU either by terminating or by switching to the waiting state.** Non-preemptive Scheduling is used when a process terminates, or a process switches from running to the waiting state. In this scheduling, once the resources (CPU cycles) are allocated to a process, the process holds the CPU till it gets terminated or reaches a waiting state. **In the case of non-preemptive scheduling does not interrupt a process running CPU in the middle of the execution. Instead, it waits till the process completes its CPU burst time, and then it can allocate the CPU to another process.**

Algorithms based on non-preemptive scheduling are: Shortest Job First (SJF basically non preemptive), etc.

- Considering that there may be hundreds of programs that need to work, the OS must launch the program, stop it, switch to another program, etc. The way the OS configures the system to run another in the CPU is called "context switching". If the OS keeps context-switching programs in and out of the provided CPUs, it can give the user a tricky idea that he or she can run any programs he or she wants to run, all at once.
- So now that we know we can run 1 program at a given CPU, and we know we can change the operating system and remove another one using the context switch, how do we choose which programs we need. run, and with what program? That's where **scheduling** comes in!

Objectives of Process Scheduling Algorithm:

- Utilization of CPU at maximum level. **Keep CPU as busy as possible**.
- **Allocation of CPU should be fair**.
- **Throughput should be Maximum**. i.e. Number of processes that complete their execution per time unit should be maximized.
- **Minimum turnaround time**, i.e. time taken by a process to finish execution should be the least.
- There should be a **minimum waiting time** and the process should not starve in the ready queue.
- **Minimum response time.** It means that the time when a process produces the first response should be as less as possible.

What are the different terminologies to take care of in any CPU Scheduling algorithm?

- Arrival Time: Time at which the process arrives in the ready queue.
- Completion Time = Time at which Process terminate (see at gantt chart).
- Burst Time**:** Time required by a process for CPU execution.
- Turn Around Time: Time Difference between completion time and arrival time.
- Turnaround Time = Completion time- Arrival time OR waiting time+ Burst Time
- Waiting Time = Turnaround time – Burst Time
- Response Time = In Non-Preemptive Response Time= Waiting time
- Avg. Turnaround Time = Sum of turnaround time of all processes/ total processes

Turn Around Time = Completion Time – Arrival Time

- **Waiting Time(W.T):** Time Difference between turn around time and burst time.

Waiting Time = Turn Around Time – Burst Time

CPU scheduling Algorithms

1- The shortest job first (SJF) Algorithm.
2- Shortest Remaining Time First (SRTF).

1-The shortest job first (SJF) or shortest job next, is a scheduling policy that selects the waiting process with the smallest execution time to execute next. SJN, also known as Shortest Job Next (SJN), can be preemptive or non-preemptive.

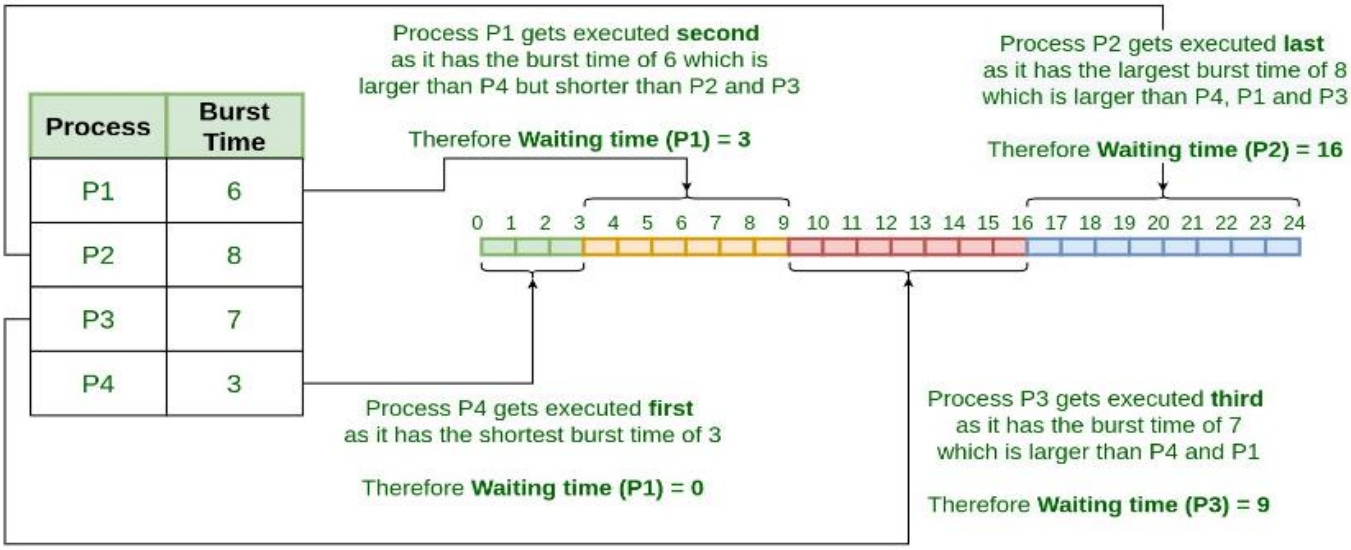
Characteristics of SJF Scheduling:

- Shortest Job first has the advantage of having a minimum average waiting time among all scheduling algorithms.
- It is a Greedy Algorithm.
- It may cause starvation if shorter processes keep coming. This problem can be solved using the concept of ageing.
- It is practically infeasible as Operating System may not know burst times and therefore may not sort them. While it is not possible to predict execution time, several methods can be used to estimate the execution time for a job, such as a weighted average of previous execution times.
- SJF can be used in specialized environments where accurate estimates of running time are available.

Algorithm:

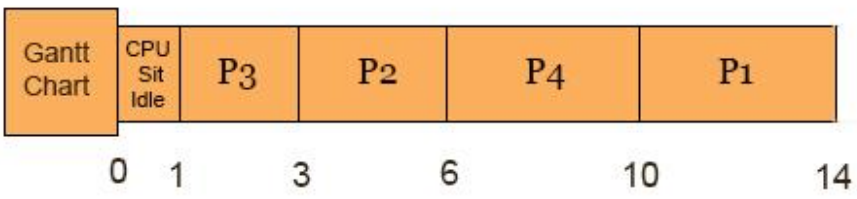
- Sort all the processes according to the arrival time.
- Then select that process that has minimum arrival time and minimum Burst time.
- After completion of the process make a pool of processes that arrives afterward till the completion of the previous process and select that process among the pool which is having minimum Burst time.

Shortest Job First (SJF) Scheduling Algorithm



Process	Burst Time
P1	6
P2	8
P3	7
P4	3

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time	Response Time	Avg. Time Calculations
P1	1	3	6	5	2	2	Average Waiting Time = (2+4+0+6)/4 = 3
P2	2	4	10	8	4	4	Average Turnaround Time = (5+8+2+10)/4 = 6.25
P3	1	2	2	2	0	0	
P4	4	4	14	10	6	6	



Homework:

Non Preemptive Shortest Job First Scheduling:

In every example Given Fields are

- Process No.
- Arrival Time.
- Burst Time.

Find the following fields:

- Completion Time
- Turnaround Time
- Waiting Time
- Response Time
- Average Turnaround Time

2- Shortest Remaining Time First

- is the preemptive version of Shortest Job Next (SJN) algorithm, where the processor is allocated to the job closest to completion.
- This algorithm requires advanced concept and knowledge of CPU time required to process the job in an interactive system, and hence can't be implemented there. But, in a batch system where it is desirable to give preference to short jobs, SRT algorithm is used.

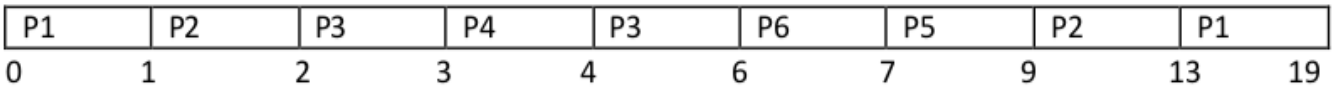
Process No.	Arrival Time (AT)	Burst Time (BT)
P1	0	8
P2	1	4
P3	2	9
P4	3	5

Gantt Chart

Result

P_No.	AT	BT	CT	TAT	WT	First Response (FR)	RT
P1	0	8	17	(17 – 0) = 17	(17 – 8) = 9	0	(0 – 0) = 0
P2	1	4	5	(5 – 1) = 4	(4 – 4) = 0	1	(1 – 1) = 0
P3	2	9	26	(26 – 2) = 24	(24 – 9) = 15	17	(17 – 2) = 15
P4	3	5	10	(10 – 3) = 7	(7 – 5) = 2	5	(5 – 3) = 2

PID	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	6	9	9	3
P2	0	8	24	24	16
P3	0	7	16	16	9
P4	0	3	3	3	0



At AT = 0, P1 Process is picked from the ready queue and executed for 1 unit of time. In SRTF a process should not be executed till completion. It is better to execute process per unit time and meanwhile check for any other processes is in queue.

At AT = 1, the processes P1 and P2 are in the queue. We need to check which process is taking less time. Here, P2 is taking less time. So, P2 will execute for 1 unit of time.

At AT = 2, the processes P1, P2 and P3 are in the queue. Since process P3 is taking lesser time compared to P1 and P2, process P3 will execute for 1unit time.

At AT = 3, P4 has 1 unit burst time. So, P4 will execute and finish the process execution.

At AT = 4, the processes P1, P2, P3 and P5 are in the queue. Here, the processes P3 and P5 have the same burst time. In this case, the process which is scheduled first will execute first. So, P3 will execute first and finish the process execution.

At AT = 5, all the processes are in queue except P4 and P3 as they have completed their execution. So now

the process that has the least burst time will execute first. P6 will execute first after that P5, then P2 and at last P1 will execute.

Note: It is not going to be Convoy effect. Starvation doesn't happen here because the shortest process will execute first

PID	Arrival Time	Burst Time	Completion time	Turn Around time (CT-AT)	Waiting Time (TAT-BT)
sP1	0	7	19	19	12
P2	1	5	13	12	7
P3	2	3	6	4	1
P4	3	1	4	1	0
P5	4	2	9	5	3
P6	5	1	7	2	1