Shortest Job First (SJF) Scheduling

Till now, we were scheduling the processes according to their arrival time (in FCFS scheduling). However, SJF scheduling algorithm, schedules the processes according to their burst time.

In SJF scheduling, the process with the lowest burst time, among the list of available processes in the ready queue, is going to be scheduled next.

However, it is very difficult to predict the burst time needed for a process hence this algorithm is very difficult to implement in the system.

Advantages of SJF

1. Maximum throughput
2. Minimum average waiting and turnaround time

Disadvantages of SJF

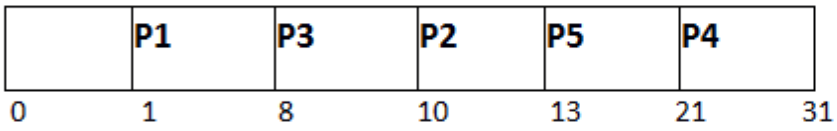
1. May suffer with the problem of starvation
2. It is not implementable because the exact Burst time for a process can't be known in advance.

There are different techniques available by which, the CPU burst time of the process can be determined.

Example

In the following example, there are five jobs named as P1, P2, P3, P4 and P5. Their arrival time and burst time are given in the table below.

PID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
1	1	7	8	7	0
2	3	3	13	10	7
3	6	2	10	4	2
4	7	10	31	24	14
5	9	8	21	12	4



Since, No Process arrives at time 0 hence; there will be an empty slot in the **Gantt chart** from time 0 to 1 (the time at which the first process arrives).

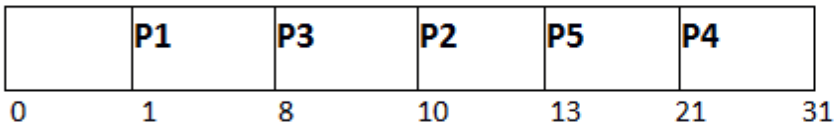
According to the algorithm, the OS schedules the process which is having the lowest burst time among the available processes in the ready queue.

Till now, we have only one process in the ready queue hence the scheduler will schedule this to the processor no matter what is its burst time.

This will be executed till 8 units of time. Till then we have three more processes arrived in the ready queue hence the scheduler will choose the process with the lowest burst time.

Among the processes given in the table, P3 will be executed next since it is having the lowest burst time among all the available processes.

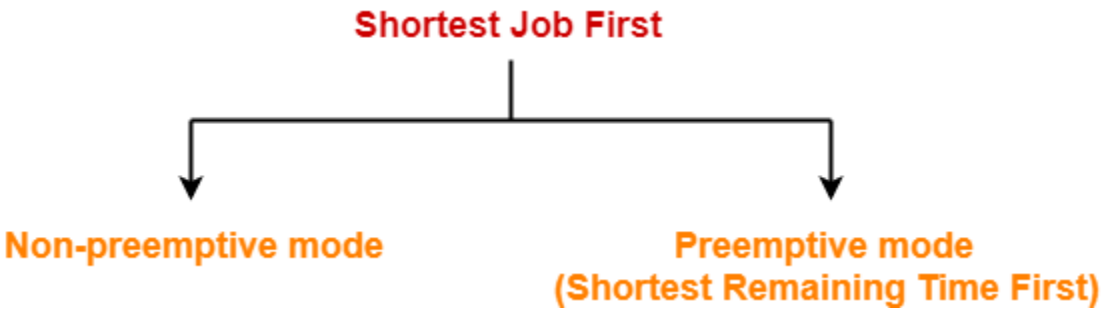
So that's how the procedure will go on in **shortest job first (SJF)** scheduling algorithm.



Avg Waiting Time = 27/5

In SJF Scheduling,

- **Out of all the available processes, CPU is assigned to the process having smallest burst time.**
- **In case of a tie, it is broken by FCFS Scheduling.**



- SJF Scheduling can be used in both preemptive and non-preemptive mode.
- Preemptive mode of Shortest Job First is called as **Shortest Remaining Time First (SRTF)**

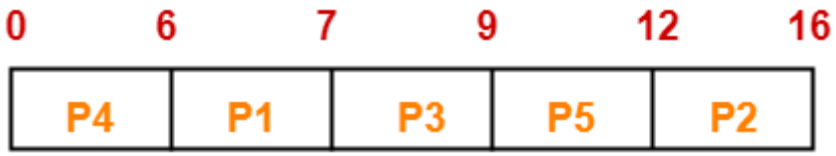
Advantages-

- SRTF is optimal and guarantees the minimum average waiting time.
- It provides a standard for other algorithms since no other algorithm performs better than it.

Disadvantages-

- It cannot be implemented practically since burst time of the processes cannot be known in advance.
- It leads to starvation for processes with larger burst time.
- Priorities cannot be set for the processes.
- Processes with larger burst time have poor response time.

PRACTICE PROBLEMS BASED ON SJF SCHEDULING-



Gantt Chart

Problem-01:

Consider the set of 5 processes whose arrival time and burst time are given below-

Process Id	Arrival time	Burst time
P1	3	1
P2	1	4
P3	4	2
P4	0	6
P5	2	3

If the CPU scheduling policy is SJF non-preemptive, calculate the average waiting time and average turn around time.

Solution-

Gantt Chart-

Now, we know-

- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

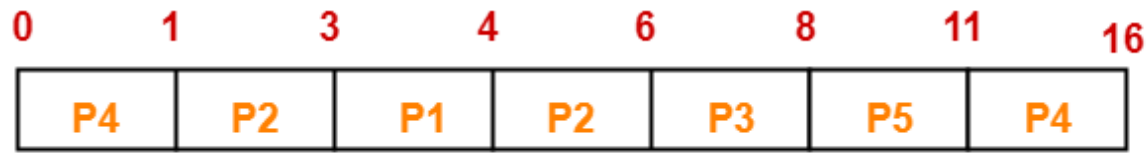
Process Id	Exit time	Turn Around time	Waiting time
P1	7	7	4
P2	16	16	15
P3	9	9	5
P4	6	6	6
P5	12	12	10

Now,

- Average Turn Around time = (4 + 15 + 5 + 6 + 10) / 5 = 40 / 5 = 8 unit
- Average waiting time = (3 + 11 + 3 + 0 + 7) / 5 = 24 / 5 = 4.8 unit

Problem-02:

Consider the set of 5 processes whose arrival time and burst time are given below-



Gantt Chart

Process Id	Arrival time	Burst time
P1	3	1
P2	1	4
P3	4	2
P4	0	6
P5	2	3

If the CPU scheduling policy is SJF preemptive, calculate the average waiting time and average turn around time.

Solution-

Gantt Chart-

Now, we know-

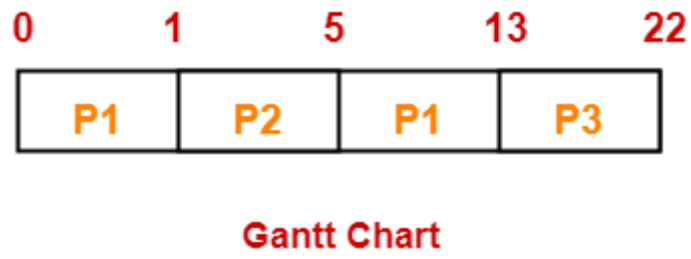
- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

Process Id	Exit time	Turn Around time	Waiting time
P1	4	4	1
P2	6	6	5
P3	8	8	4
P4	16	16	16
P5	11	11	9

Now,

- Average Turn Around time = (1 + 5 + 4 + 16 + 9) / 5 = 35 / 5 = 7 unit
- Average waiting time = (0 + 1 + 2 + 10 + 6) / 5 = 19 / 5 = 3.8 unit

Problem-03:



Gantt Chart

Consider the set of 3 processes whose arrival time and burst time are given below-

Process Id	Arrival time	Burst time
P1	0	9
P2	1	4
P3	2	9

If the CPU scheduling policy is SRTF, calculate the average waiting time and average turn around time.

Solution-

Gantt Chart-

Now, we know-

- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

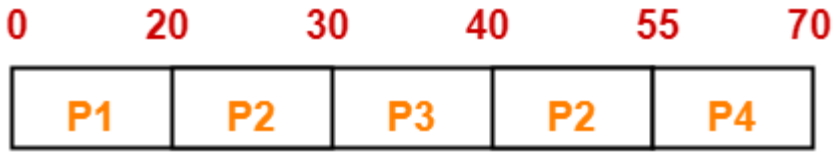
Process Id	Exit time	Turn Around time	Waiting time
P1	13	13 – 0 = 13	13 – 9 = 4
P2	5	5 – 1 = 4	4 – 4 = 0
P3	22	22- 2 = 20	20 – 9 = 11

Now,

- Average Turn Around time = (13 + 4 + 20) / 3 = 37 / 3 = 12.33 unit
- Average waiting time = (4 + 0 + 11) / 3 = 15 / 3 = 5 unit

Problem-04:

Consider the set of 4 processes whose arrival time and burst time are given below-



Gantt Chart

Process Id	Arrival time	Burst time
P1	0	20
P2	15	25
P3	30	10
P4	45	15

If the CPU scheduling policy is SRTF, calculate the waiting time of process P2.

Solution-

Gantt Chart-

Now, we know-

- Turn Around time = Exit time – Arrival time
- Waiting time = Turn Around time – Burst time

Thus,

- Turn Around Time of process P2 = 55 – 15 = 40 unit
- Waiting time of process P2 = 40 – 25 = 15 unit

Round Robin Scheduling Algorithm

we are going to learn about the most efficient CPU Process Scheduling Algorithm named Round Robin CPU Process Scheduling. This algorithm is very special because it is going to remove all the Flaws which we have detected in the previous CPU Process Scheduling Algorithms.

There is a lot of popularity for this Round Robin CPU Scheduling is because Round Robin works only in Pre Emptive state. This makes it very reliable.

Round Robin CPU Scheduling

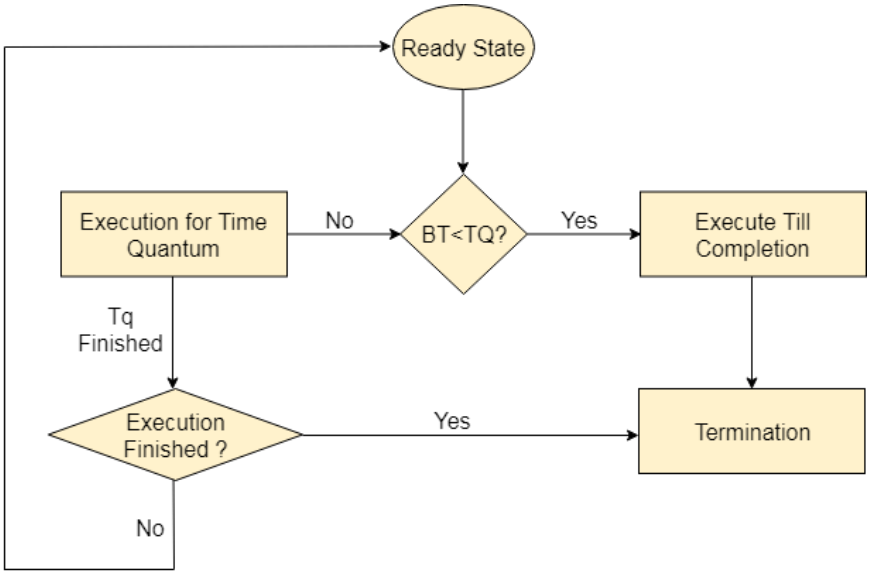
Round Robin CPU Scheduling is the most important CPU Scheduling Algorithm which is ever used in the history of CPU Scheduling Algorithms. Round Robin CPU Scheduling uses Time Quantum (TQ). The Time Quantum is something which is removed from the Burst Time and lets the chunk of process to be completed.

Time Sharing is the main emphasis of the algorithm. Each step of this algorithm is carried out cyclically. The system defines a specific time slice, known as a time quantum.

First, the processes which are eligible to enter the ready queue enter the ready queue. After entering the first process in Ready Queue is executed for a Time Quantum chunk of time. After execution is complete, the process is removed from the ready queue. Even now the process requires some time to complete its execution, then the process is added to Ready Queue.

The Ready Queue does not hold processes which already present in the Ready Queue. The Ready Queue is designed in such a manner that it does not hold non unique processes. By holding same processes Redundancy of the processes increases.

After, the process execution is complete, the Ready Queue does not take the completed process for holding.



Advantages

The Advantages of Round Robin CPU Scheduling are:

1. A fair amount of CPU is allocated to each job.
2. Because it doesn't depend on the burst time, it can truly be implemented in the system.
3. It is not affected by the convoy effect or the starvation problem as occurred in First Come First Serve CPU Scheduling Algorithm.

Disadvantages

The Disadvantages of Round Robin CPU Scheduling are:

1. Low Operating System slicing times will result in decreased CPU output.
2. Round Robin CPU Scheduling approach takes longer to swap contexts.
3. Time quantum has a significant impact on its performance.
4. The procedures cannot have priorities established.

Examples:

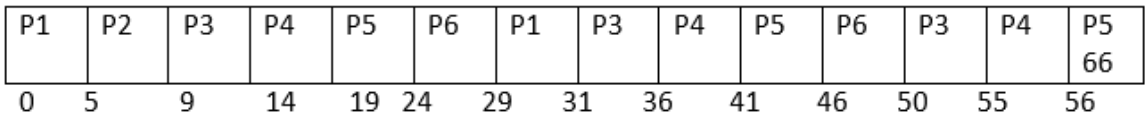
1. S. No	Process ID	Arrival Time	Burst Time
2. 1	P 1	0	7
3. 2	P 2	1	4
4. 3	P 3	2	15
5. 4	P 4	3	11
6. 5	P 5	4	20
7. 6	P 6	4	9

Assume Time Quantum TQ = 5

Ready Queue:

1. P1, P2, P3, P4, P5, P6, P1, P3, P4, P5, P6, P3, P4, P5

Gantt chart:



Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	7	31	31	24
P2	1	4	9	8	4
P3	2	15	55	53	38
P4	3	11	56	53	42
P5	4	20	66	62	42
P6	4	9	50	46	37

Average Completion Time

1. Average Completion Time = (31 +9 + 55 +56 +66 + 50) / 6
2. Average Completion Time = 267 / 6
3. Average Completion Time = 44.5

Average Waiting Time

1. Average Waiting Time = (5 + 26 + 5 + 42 + 42 + 37) / 6
2. Average Waiting Time = 157 / 6

3. Average Waiting Time = 26.16667

Average Turn Around Time

1. Average Turn Around Time = (31 + 8 + 53 + 53 + 62 + 46) / 6
2. Average Turn Around Time = 253 / 6
3. Average Turn Around Time = 42.16667