NAME - PAYAL SUTHAR

Roll NO. - 26

SEM. — MCA 1st SEM

Assignment - 2

(CPU Scheduling)

## CPU SCHEDULING :-

CPU Scheduling is a process that allows one process to use the CPU while another process is delayed due to unavailability of any resources such as I/O etc, thus making full use of the CPU.

The purpose of CPU scheduling is make the system more efficient, faster, and fairer.

⇒ CPU Scheduling is the task performed by the CPU that decides the way and order in which processes should be executed.

There are two types of CPU Scheduling:-

(1) Pre-emptive:-

⇒ CPU resources are allocated to a process for only a limited period of time and then those resources are taken back. A running process could be interrupted to execute a higher priority process.

⇒ Preemptive scheduling is used when a process switches from the running state to the ready state or from the waiting state to the ready state.

Algorithms based on preemptive scheduling are:-

1) Round Robin (RR)
2) Shortest Remaining Time First (SRTF)
3) Priority (Preemptive version)

* Advantage :-

1) Each occurrence prevents the completion of ongoing tasks.
2) The average response time is improved.

* Disadvantage :-

1) Limited computational resources must be used.
2) The low priority process would have to wait if multiple high-priority processes arrived at the same time.

(1) Round Robin :- 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.

Example :-

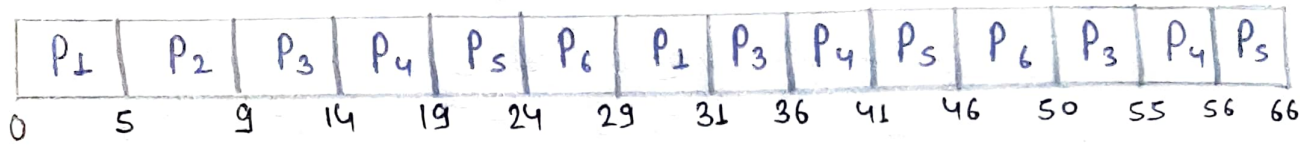
TQ - 5

| Process ID | Arrival Time | Burst Time |
|---|---|---|
| $P_1$ | 0 | 7 |
| $P_2$ | 1 | 4 |
| $P_3$ | 2 | 15 |
| $P_4$ | 3 | 11 |
| $P_5$ | 4 | 20 |
| $P_6$ | 4 | 9 |

Ready Queue :-

| $P_1$ | $P_2$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_1$ | $P_3$ | $P_4$ | $P_5$ | $P_6$ | $P_3$ | $P_4$ | $P_5$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Gantt chart :-

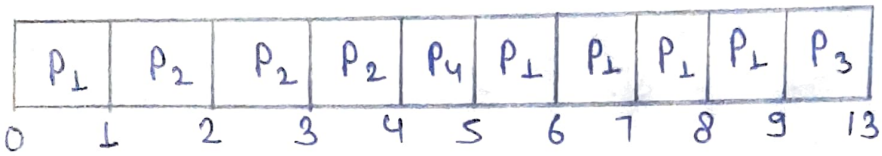



| Process ID | AT | BT | CT | TAT | WT |
|---|---|---|---|---|---|
| $P_1$ | 0 | 7 | 31 | 31 | 24 |
| $P_2$ | 1 | 4 | 9 | 8 | 4 |
| $P_3$ | 2 | 15 | 55 | 53 | 38 |
| $P_4$ | 3 | 11 | 56 | 53 | 42 |
| $P_5$ | 4 | 20 | 66 | 62 | 42 |
| $P_6$ | 4 | 9 | 50 | 46 | 37 |

(2) SRTF :- This algo. is the preemptive version of SJF scheduling. In SRTF, the execution of the process can be stopped after certain amount of time. At the arrival of every process, the short term scheduler, Schedules the process with the least remaining burst time among the list of available processes and the running process.

| Process ID | Arrival Time | Burst Time |
|---|---|---|
| $P_1$ | 0 | 5 4 3 2 1 0 |
| $P_2$ | 1 | 3 2 1 0 |
| $P_3$ | 2 | 4 0 |
| $P_4$ | 4 | 1 0 |

Gantt chart -




| P. ID | AT | BT | CT | TAT | WT |
|---|---|---|---|---|---|
| $P_1$ | 0 | 5 | 9 | 9 | 4 |
| $P_2$ | 1 | 3 | 4 | 4 | 0 |
| $P_3$ | 2 | 4 | 13 | 13 | 7 |
| $P_4$ | 4 | 1 | 5 | 5 | 0 |

Turn around time :- Completion time - Arrival time

waiting time :- TAT - BT

Response time :- {CPU first time - AT}

Avg. TAT :- $24/4 = 6$

Avg. WT :- $11/4 = 2.75$

Avg. RT :- $7/4 = 1.75$

(2) NON-preemptive :-

=> 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. It waits till the process completes its CPU burst time, and then it can allocate the CPU to another process.

* Advantage :-

1) It has a minimal scheduling burden.
2) It is a very easy procedure.
3) Less computational resources are used.
4) It has a high throughput rate.

* Disadvantage :-

1) Its response time to the process is super.
2) Bugs can cause a computer to freeze up.

Algorithms based on non preemptive scheduling are :-

1) FCFS (First Come first serve)
2) SJF (Shortest job first)

1) FCFS :-

(First Come first Serve)-

The FCFS algorithm assigns the CPU to the process that arrives first in the ready Queue. This means that the process that requests the CPU for its execution first will get the CPU allocated first. This is managed through the FIFO Queue.

The lesser the arrival time of processes in the ready Queue. the sooner the process gets the CPU.

* Advantage :-

1) It is the simplest form of a scheduling algorithm.
2) Its implementation is easy and it is first-come, first serve.

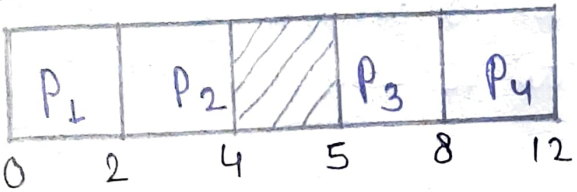
* Disadvantage :-

1) Due to the non-preemptive nature of the algorithm, short processes at the end of the Queue have to wait for long processes that are present at the front of the queue to finish.

2) There is a high average waiting time that causes a starvation problem.

| Process ID | Arrival Time | Burst Time |
|---|---|---|
| $P_1$ | 0 | 2 |
| $P_2$ | 1 | 2 |
| $P_3$ | 5 | 3 |
| $P_4$ | 6 | 4 |

Grantt chart




| P. ID | AT | BT | CT | TAT | WT |
|---|---|---|---|---|---|
| $P_1$ | 0 | 2 | 2 | 2 | 0 |
| $P_2$ | 1 | 2 | 4 | 3 | 1 |
| $P_3$ | 5 | 3 | 8 | 3 | 0 |
| $P_4$ | 6 | 4 | 12 | 6 | 2 |

Avg. TAT :- $14/4 = 3.5$

Avg. WT :- $3/4 = 0.75$

2) SJF :-

(Shortest Job first) -

SJF is an algorithm in which the process having the smallest execution time is chosen for the next execution.

It significantly reduces the average waiting time for other processes awaiting execution.

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

* Advantage of SJF :-

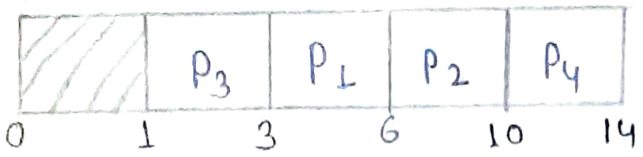
1) Maximum throughput
2) Minimum average waiting and turnaround time.

* Disadvantage 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.

| Process ID | Arrival time | Burst time |
|---|---|---|
| $P_1$ | 1 | 3 |
| $P_2$ | 2 | 4 |
| $P_3$ | 1 | 2 |
| $P_4$ | 4 | 4 |

Gantt chart:-




| Process ID | AT | BT | CT | TAT | WT |
|---|---|---|---|---|---|
| $P_1$ | 1 | 3 | 6 | 5 | 2 |
| $P_2$ | 2 | 4 | 10 | 8 | 4 |
| $P_3$ | 1 | 2 | 3 | 2 | 0 |
| $P_4$ | 4 | 4 | 14 | 10 | 6 |

* Avg Turn around time :-

$$\Rightarrow \frac{(5+8+2+10)}{4}$$

$$\Rightarrow \frac{25}{4} \Rightarrow 6.25$$

* Avg Completion time :-

$$\Rightarrow \frac{(6+10+3+14)}{4}$$

$$\Rightarrow \frac{33}{4} \Rightarrow 8.25$$

* Avg. waiting time :-

$$\Rightarrow \frac{(2+4+0+6)}{4} \Rightarrow \frac{12}{4} \Rightarrow 3$$