CPU Scheduling in Operating Systems

- Difficulty Level : Easy
- Last Updated : 28 Jun, 2021

Scheduling of processes/work is done to finish the work on time.

Below are different time with respect to a process.

Arrival Time: *Time at which the process arrives in the ready queue.*
Completion Time: *Time at which process completes its execution.*
Burst Time: *Time required by a process for CPU execution.*
Turn Around Time: *Time Difference between completion time and arrival time.*
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

Why do we need scheduling?
A typical process involves both I/O time and CPU time. In a uni programming system like MS-DOS, time spent waiting for I/O is wasted and CPU is free during this time. In multi programming systems, one process can use CPU while another is waiting for I/O. This is possible only with process scheduling.

Objectives of Process Scheduling Algorithm
Max CPU utilization [Keep CPU as busy as possible]
Fair allocation of CPU.
Max throughput [Number of processes that complete their execution per time unit]
Min turnaround time [Time taken by a process to finish execution]
Min waiting time [Time a process waits in ready queue]
Min response time [Time when a process produces first response]

Different Scheduling Algorithms
*First Come First Serve (FCFS)**:*** Simplest scheduling algorithm that schedules according to arrival times of processes. First come first serve scheduling algorithm states that the process that requests the CPU first is allocated the CPU first. It is implemented by using the FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue. FCFS is a non-preemptive scheduling algorithm.
Note: First come first serve suffers from convoy effect.
*Shortest Job First (SJF)**:*** Process which have the shortest burst time are scheduled first.If two processes have the same bust time then FCFS is used to break the tie. It is a non-preemptive scheduling algorithm.

Longest Job First (LJF): It is similar to SJF scheduling algorithm. But, in this scheduling algorithm, we give priority to the process having the longest burst time. This is non-preemptive in nature i.e., when any process starts executing, can't be interrupted before complete execution.
Shortest Remaining Time First (SRTF): It is preemptive mode of SJF algorithm in which jobs are schedule according to shortest remaining time.
Longest Remaining Time First (LRTF): It is preemptive mode of LJF algorithm in which we give priority to the process having largest burst time remaining.
Round Robin Scheduling: Each process is assigned a fixed time(Time Quantum/Time Slice) in cyclic way.It is designed especially for the time-sharing system. The ready queue is treated as a circular queue. The CPU scheduler goes around the ready queue, allocating the CPU to each process for a time interval of up to 1-time quantum. To implement Round Robin scheduling, we keep the ready queue as a FIFO queue of processes. New processes are added to the tail of the ready queue. The CPU scheduler picks the first process from the ready queue, sets a timer to interrupt after 1-time quantum, and dispatches the process. One of two things will then happen. The process may have a CPU burst of less than 1-time quantum. In this case, the process itself will release the CPU voluntarily. The scheduler will then proceed to the next process in the ready queue. Otherwise, if the CPU burst of the currently running process is longer than 1-time quantum, the timer will go off and will cause an interrupt to the operating system. A context switch will be executed, and the process will be put at the tail of the ready queue. The CPU scheduler will then select the next process in the ready queue.
Priority Based scheduling (Non-Preemptive): In this scheduling, processes are scheduled according to their priorities, i.e., highest priority process is scheduled first. If priorities of two processes match, then schedule according to arrival time. Here starvation of process is possible.
Highest Response Ratio Next (HRRN): In this scheduling, processes with highest response ratio is scheduled. This algorithm avoids starvation.

```
Response Ratio = (Waiting Time + Burst time) / Burst time
```

Multilevel Queue Scheduling: According to the priority of process, processes are placed in the different queues. Generally high priority process are placed in the top level queue. Only after completion of processes from top level queue, lower level queued processes are scheduled. It can suffer from starvation.
Multi level Feedback Queue Scheduling: It allows the process to move in between queues. The idea is to separate processes according to the characteristics of their CPU bursts. If a process uses too much CPU time, it is moved to a lower-priority queue.

Some useful facts about Scheduling Algorithms:

1. FCFS can cause long waiting times, especially when the first job takes too much CPU time.
2. Both SJF and Shortest Remaining time first algorithms may cause starvation. Consider a situation when the long process is there in the ready queue and shorter processes keep coming.
3. If time quantum for Round Robin scheduling is very large, then it behaves same as FCFS scheduling.
4. SJF is optimal in terms of average waiting time for a given set of processes,i.e., average waiting time is minimum with this scheduling, but problems are, how to know/predict the time of next job.

Exercise:

1. Consider a system which requires 40-time units of burst time. The Multilevel feedback queue scheduling is used and time quantum is 2 unit for the top queue and is incremented by 5 unit at each level, then in what queue the process will terminate the execution?

2. Which of the following is false about SJF?
   S1: It causes minimum average waiting time

S2: It can cause starvation
(A) Only S1
(B) Only S2
(C) Both S1 and S2
(D) Neither S1 nor S2
Answer (D)
S1 is true SJF will always give minimum average waiting time.
S2 is true SJF can cause starvation.

3. Consider the following table of arrival time and burst time for three processes P0, P1 and P2. (GATE-CS-2011)

Process	Arrival time	Burst Time
P0	0 ms	9 ms
P1	1 ms	4 ms
P2	2 ms	9 ms

The pre-emptive shortest job first scheduling algorithm is used. Scheduling is carried out only at arrival or completion of processes. What is the average waiting time for the three processes?
(A) 5.0 ms
(B) 4.33 ms
(C) 6.33
(D) 7.33
Solution :
Answer: – (A)
Process P0 is allocated processor at 0 ms as there is no other process in the ready queue. P0 is preempted after 1 ms as P1 arrives at 1 ms and burst time for P1 is less than remaining time of P0. P1 runs for 4ms. P2 arrived at 2 ms but P1 continued as burst time of P2 is longer than P1. After P1 completes, P0 is scheduled again as the remaining time for P0 is less than the burst time of P2.
P0 waits for 4 ms, P1 waits for 0 ms and P2 waits for 11 ms. So average waiting time is (0+4+11)/3 = 5.

7. Consider the following set of processes, with the arrival times and the CPU-burst times given in milliseconds (GATE-CS-2004)

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	3
P4	4	1

What is the average turnaround time for these processes with the preemptive shortest remaining processing time first (SRPT) algorithm ?
(A) 5.50
(B) 5.75
(C) 6.00
(D) 6.25
Answer (A)
Solution:
The following is Gantt Chart of execution

P1	P2	P4	P3	P1
1	4	5	8	12

Turn Around Time = Completion Time – Arrival Time
Avg Turn Around Time = (12 + 3 + 6+ 1)/4 = 5.50

12. An operating system uses the Shortest Remaining Time first (SRTF) process scheduling algorithm. Consider the arrival times and execution times for the following processes:

13. Process	Execution time	Arrival time
14. P1	20	0
15. P2	25	15
16. P3	10	30
P4	15	45

What is the total waiting time for process P2?
(A) 5
(B) 15
(C) 40
(D) 55
Answer (B)
At time 0, P1 is the only process, P1 runs for 15 time units.
At time 15, P2 arrives, but P1 has the shortest remaining time. So P1 continues for 5 more time units.
At time 20, P2 is the only process. So it runs for 10 time units
At time 30, P3 is the shortest remaining time process. So it runs for 10 time units
At time 40, P2 runs as it is the only process. P2 runs for 5 time units.
At time 45, P3 arrives, but P2 has the shortest remaining time. So P2 continues for 10 more time units.
P2 completes its execution at time 55

```
Total waiting time for P2 = Completion time - (Arrival time + Execution time)
                          = 55 - (15 + 25)
```

= 15