Name:- Kumar Deshmukh
Roll:- 33143

Assignment 3

Aim:- Implement a C program for CPU Scheduling Algorithms:-
(i) SJF (Pre-emptive)
(ii) Round Robin with different Arrival Times

Theory

What is CPU Scheduling?

CPU scheduling is a process of determining which process will own the CPU for execution while another process is on hold. The main task of CPU scheduling is to ensure that whenever the CPU remains idle, the OS at least selects one of the available process in the Ready Queue for execution. The selection process is carried out by the CPU scheduler using some scheduling algorithm.

The scheduling algorithms may be of 2 types -
Preemptive & Non-Premptive

Premptive Scheduling

In premptive scheduling, the CPU scheduler can stop the currently running process into waiting state & replace it with another process. This is usually done to make the higher priority process execute first. Also, there are time slots given.

Non-Premptive Scheduling

In this type of scheduling, the scheduler cannot replace the currently running process mid-way. The selected sheduled process executes fully then another process gets allocated to the CPU.

Scheduling Algorithms

The sequence of steps followed by the Scheduler to schedule processes to the CPU are called Scheduling Algorithms.

Some examples of Scheduling Algorithms are:-

SJF, FCFS, SRTF, Round Robin, Priority Scheduling, HRRN

(Shortest Job First) (First-Come First-Serve) (Shortest Remaining Time First) (Highest Response Ratio Next)

Also, there are multi-queue scheduling algorithms that may use a combination of above algorithms.

The real-world Operating Systems use more complex scheduling algorithms

---

Shortest Job First (SJF)

- In SJF scheduling algorithm, at any time, the process in the Ready Queue with the shortest burst time is scheduled first and then other processes in increasing order of burst times
- It significantly reduces the average waiting time for other processes awaiting execution
- However, it causes starvation of processes present in Ready Queue with higher burst times
- Also, the burst time can't be precisely known at all times for all processes, which makes implementing SJF impractical
- SJF may be pre-emptive or non-premptive

  Pre-emptive SJF behaves just like SRTF (Shortest Remaining Time First)

Round Robin (RR)

- Round Robin Scheduling algorithm is a Pre-emptive algorithm that is said to be one of the fairest algorithms in terms of starvation
- The main factor of Round Robin is the Time quantum (Q). It is the amount of time slot given to each process
- The algorithm cycles through the processes available in Ready Queue allowing each to execute for at max. the time quantum (Q) & then it gets preempted by the next available scheduled process. This cycling (round travelling) reduces startvation
- The time quantum should be selected carefully as it impacts the whole performance of Round Robin. It should be small but not too small (or else a lot of time gets spent in context switching)

Example

Q = 2s

Pid	AT	BT	CT	TAT	WT	RT
1	0	4	8	8	4	0
2	1	5	20	19	14	1
3	2	6	24	22	16	2
4	3	3	19	16	13	5
5	4	2	12	8	6	6
6	5	4	22	17	13	9

Context switches :- 12

Avg. WT = 11 s

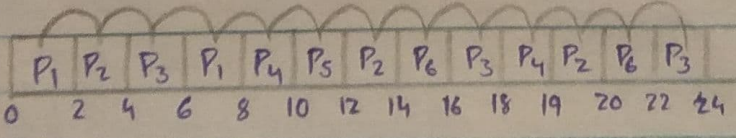
Avg. TAT = 15 s

Scheduling length = 24 s

Throughput = 6/24 = 0.25s

Cycling Queue: P_1 P_2 P_3 P_1 P_4 P_5 P_2 P_6 P_3 P_4 P_2 P_6 P_3

Gantt Chart



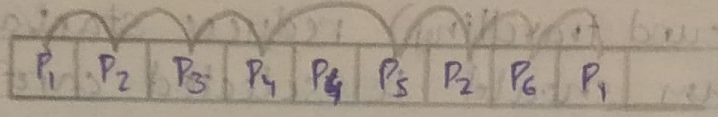
(Preemptive)

SJF example

Pid	AT	BT
1	0	6
2	1	4
3	2	1
4	3	2
5	4	1
6	5	3

CT	TAT	WT	RT
17	17	11	0
9	8	4	0
3	1	0	0
5	2	0	0
6	2	1	1
12	7	4	4

Gantt chart :-



Context switches = 7

Avg. TAT = 6.1667 s Avg WT = 3.33 s

Scheduling length = 17s

Throughput = 6/17 = 0.3529 s

Conclusion

We implemented SJF (Premptive) & Round Robin Scheduling algorithms