Mitali Shrivastav
Div: 03
RNo: 375

Ass No 3 13

Problem Statement: Write a CPP program to implement following scheduling algorithms: FCFS, SJF, priority & Round Robin.

Theory:

CPU scheduling:
Scheduling of processes/work is done to finish the work on time.
Below are different time with respect to a process:

1) Arrival Time: Time at which process completes in execution.
2) Burst Time: Time required by a process for CPU execution.
3) Completion Time: Time at which process completes in execution.
4) Turn Around Time: Difference bet" completion & arrival time.
5) Waiting time: Time difference bet" turn around time and burst time.

Why do we need scheduling:
A typical process involves both I/O time & CPU time. In unit programming system like MS-DOS, time spent waiting for I/O is wasted & CPU is free during this time. This is only possible with process scheduling.

Different CPU Scheduling Algorithms:

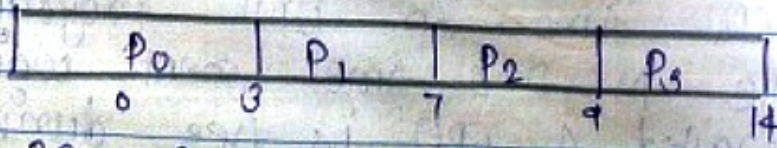
1) First Come First Serve:

Simplest scheduling algorithm that schedule according to arrival time of processes. First serve scheduling algorithm states that the process that request the CPU first is allocated to the CPU first. It is implemented using FIFO queue. When a process enter a ready queue its PCB is linked into tail of queue. When CPU is free it is allocated to the process at head of queue. The running process is then removed from queue. It is non-premptive.

Example:

Process	Arrival time	Completion time
P_0	0	3
P_1	1	4
P_2	2	2
P_3	3	5

Gantt chart:



Waiting time:- $P_0 = 0$, $P_1 = 2$, $P_2 = 5$, $P_3 = 6$.

Avg. waiting time = $(0+2+5+6)/4 = 8.25$ s.

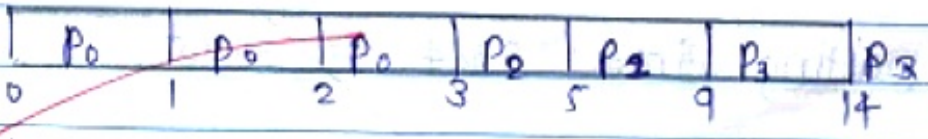
2) Shortest Job First [preemptive]:

The process with smallest amount of time remaining until completion is selected to execute since currently executing process is the one with the shortest amount of time remaining by definition, and since that time should only reduce as execution progresses, processes will always run until complete or new process is added that requires a smaller amount of time.

example:

Process	Arrival time	Burst time
P_0	0	8
P_1	1	4
P_2	2	2
P_3	3	1

Gantt Chart



Waiting time: $P_0 = 0$, $P_1 = 4$, $P_2 = 1$, $P_3 = 6$

Avg. waiting time: $(0+1+4+6)/9 = 2.75$s.

3) Priority Scheduling [Premptive]:

Priority scheduling is a me of scheduling process that is based on priority. In this algorithm, the schedular selects the tasks to work as per the priori The processes with higher priority should be carried out first whereas job with equal priorities are carried out on a round-robin. Priority depends upon memory requirements.

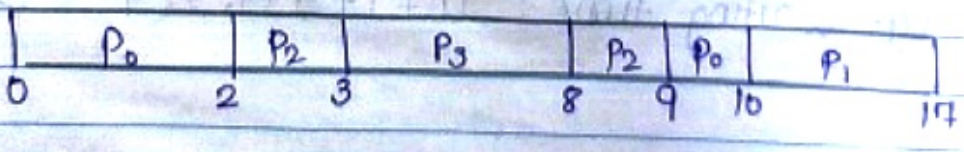
example:

Process	Arrival time	Priority	Burst time
P_0	0	3	3
P_1	1	1	4
P_2	2	2	2
P_3	3	1	3

Waiting time: $P_0 = 7$, $P_1 = 9$, $P_2 = 7$, $P_3 = 0$

Avg. waiting time: 23/4 = 5.75 s

Gantt chart:



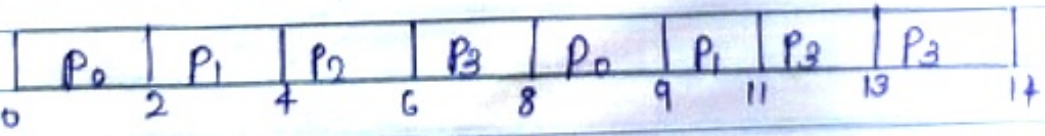
4) Round-robbin:

Round robbin is one of the algorithms employed by process and network schedulers in computing. As the term is generally used time slices are assigned to each process in equal portions and in circular order, handling all processes without priority. Round-robbin scheduling is simple, easy to implement, and starvation-free. Round-robbin scheduling can be applied to other scheduling problems. Such as data packets scheduling in computer networks. It is an operating system concept.

example:

Process	Arrival time	Burst time
P_0	0	3
P_1	1	4
P_2	2	2
P_3	3	5

Gantt Chart:



Waiting time : $P_0 = 6$, $P_1 = 6$, $P_2 = 2$, $P_3 = 6$

Avg. waiting time: $20/4 = 5$ s.

Conclusion: We studied & implemented scheduling