Date
Page

Os Term Paper

Toppic- Round Robin scheduling

Name - Swati Kumari
Roll. No. - 33200118004
Sem - 5th Dep - CSE

☆ Round Robin scheduling :-

- This type of scheduling is used for multiple operating system or we can say time sharing operating system.

Ex:- In multiple operating System, so many process are running parallel, but in actually every process has fixed time that is called time Quantum.

- Time Quantum - The operating System allocate some unit of time for each and every process. that time is called time Quantum.

All process are switching, and process of switching is too fast. So It seems like parallel.

- This scheduling is almost similar to FCFS (first come first search) but with some time quantum.
- It is always Preemptive scheduling Algorithm.
- Round Robin scheduling is simple, easy to implement and starvation - free.
- Basic fundamental of this scheduling is, each process takes an equal share of time during running or execution.
- In this scheduling Algorithm scheduler forces the process out of the CPU once the time quotum expires.

Ex:- If time slot is 100 ms, and process 1 takes a total time of 200 ms to complete, the round-robin scheduler will suspend the job after 100 ms and give other (process) job 2 their time on the CPU.

Once the other process have had their equal share (100 ms) process 1 will get another allocation of CPU time and the cycle will repeat until the job finished and doesn't need more time on the CPU.

Numerical Example of Round Robin scheduling

T.Q = 3 ms.

Process	AT	BT
P_1	0	8
P_2	5	2
P_3	1	7
P_4	6	3
P_5	8	5

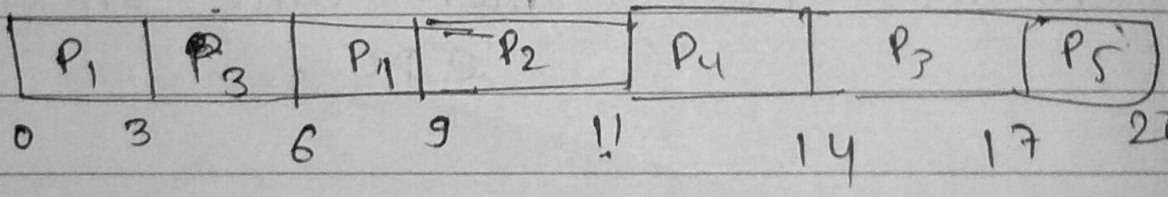
A.T = Arrival time

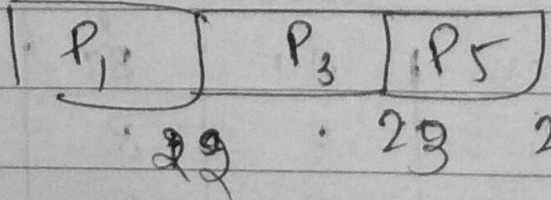
B.T = Burst time

Solve:-

	AT	B.T	Remaining B.T.	
P_1	0	8	5 -3	2
P_2	5	2		
P_3	1	7	4 -3	1
P_4	6	3		
P_5	8	5-3	2	

Ready Queue P_1 P_3 P_1 P_2 P_4 P_3 P_5 P_1 P_3





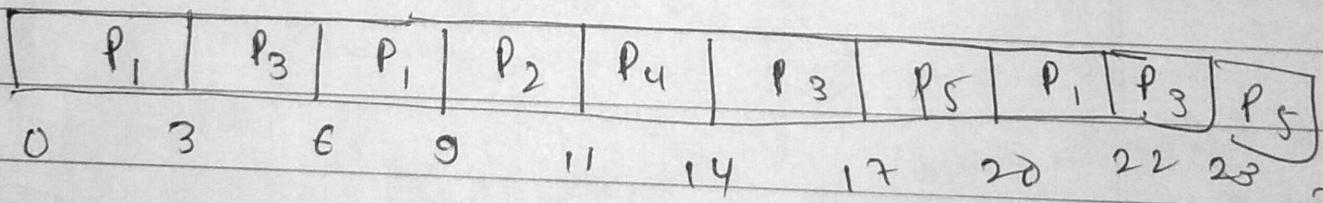
At Beginning Allocate time is 3ms and Process (P_1) has 0 Arrival time

So, P_1 goes to ready Queue; but P_1 has 8 ms B.T, ~~after 3 ms, P_1 go~~

after P_1 execute 3 ms, Another process come in ready Queue which A.T is less than or equal to 3 ms, means P_3 come, P_3 execute 3 - 6 ms, (at that time P_1 is Waiting in Ready Queue,

The process are running Circular way until All process are completely executed.

So, execution series are :-



$P_1 \rightarrow P_3 \rightarrow P_1 \rightarrow P_2 \rightarrow P_4 \rightarrow P_3 \rightarrow P_5 \rightarrow P_1 \rightarrow P_3 \rightarrow P_5$

Process	AT	BT	CT	TT	WT	RT
P_1	0	8	22	22	14	0
P_2	5	2	11	6	4	4
P_3	1	7	23	22	15	2
P_4	6	3	14	8	5	5
P_5	8	5	25	17	2	9

- If TQ is very large, context switching is low. It takes less time.

2