19UCS130

Assignment No. 03

Qu.1 Differentiate between preemptive and non-preemptive scheduling.

Preemtive	Non-Preemptive
1) Process can be interrupted in between.	1) Interrupt until it teminate itself or its time is up.
2) It has overhead of scheduling the process.	2) It does not have overhead.
3) It is flexible.	3) It is rigid.
4) CPU utilization is more efficient compared to Non-Preemptive scheduling.	4) CPU utilization is less efficient compared to preemptive scheduling.
5) Examples:- Shortest Remaining Time First, Round Robin, etc.	5) Example:- First come First serve, Shortest Job first, Priority scheduling, etc.

19UCS130

Qu.2 which of the following scheduling algorithms could result in starvation?

a) First-come, first-served
b) Shortest job first
c) Round robin
d) Priority.

Ans:- b) Shortest Job First
d) Priority.

a) In First-come, fist-served a process with very large burst time come before other process, the other process will have to wait long time but it is clear than other process will have to wait long time but it is clear than other process will definitely get their chance to execute, so it will not suffer from starvation.

b) In Shortest Job First, if process with shortest process time keep priority process suffer from starvation.

c) In Round-robin there is a fixed time quantum and every process will get their chance to execute, so no starvation.

d) In priority scheduling, if higher priority process will keep on coming then low priority process will suffer from starvation.

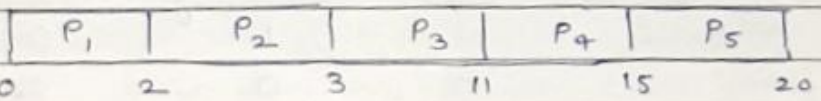
Que.3 Consider the following set of processes, with the length of the CPU burst given in milliseconds..

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

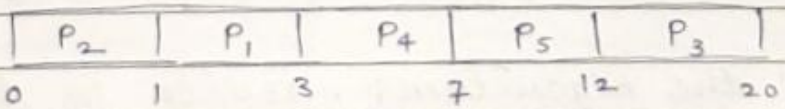
The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

a) Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms:

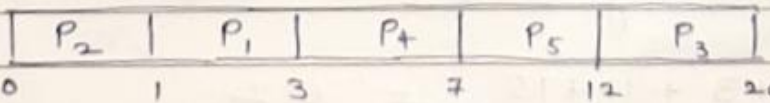
i) FCFS :-



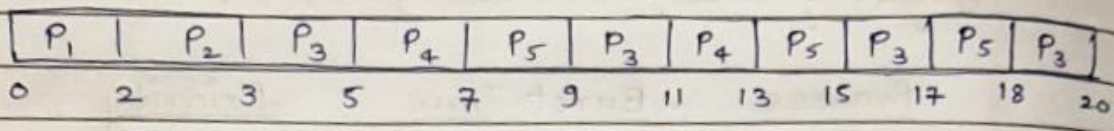
ii) SJF :-



iii) Nonpreemptive priority :-



iv) RR (quantum = 2)



b) what is the turnaround time of each process for each of the scheduling algorithms in part a?

→

Process	FCFS	SJF	Priority	RR
P_1	2	3	3	2
P_2	3	1	1	3
P_3	11	20	20	20
P_4	15	7	7	13
P_5	20	12	12	18

c) what is the waiting time of each process for each of these scheduling algorithms?

→

Process	FCFS	SJF	Priority	RR
P_1	0	1	1	0
P_2	2	0	0	2
P_3	3	12	12	12
P_4	11	3	3	9
P_5	15	7	7	13

d) which of the algorithms results in the minimum average waiting time (over all processes)?

i) Average waiting time for FCFS

$$= \frac{0+2+3+11+15}{5} = 6.2$$

iii) Average waiting time for SJF

$$= \frac{1+0+12+3+7}{5} = 4.6$$

iii) Average waiting time for Priority

$$= \frac{1+0+12+3+7}{5} = 4.6$$

iv) Average waiting time for RR,

$$= \frac{0+2+12+9+13}{5} = 7.2$$

→ As, SJF and priority scheduling algorithm have same average waiting time. Therefore, SJF and priority has minimum average waiting time over all processes.

Qu. 4 Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use nonpreemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

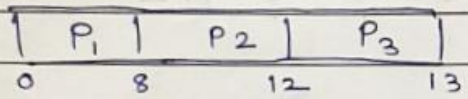
Process	Arrival time	Burst time
P_1	0.0	8
P_2	0.4	4
P_3	1.0	1

a) what is the average turnaround time for these processes with the FCFS scheduling algorithm?

→

Turnaround Time = completion Time − Arrival Time | Waiting time = turnaround time − Burst time

FCFS Gantt chart :-



Process	completion time	Turnaround Time	waiting time
P_1	8	8 − 0 = 8	8 − 8 = 0
P_2	12	12 − 0.4 = 11.6	11.6 − 4 = 7.6
P_3	13	13 − 1.0 = 12	12 − 1 = 11

Average turnaround time

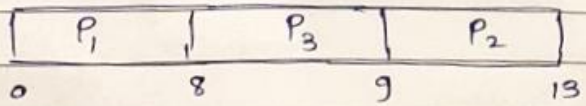
$$= \frac{8 + 11.6 + 12}{3}$$

$$= \frac{31.6}{3}$$

$$= 10.53$$

b) what is the average turnaround time for these processes with the SJF scheduling algorithm?

→ Gantt chart



Process	Completion time	Turnaround time	Waiting time
P_1	8	8 - 0 = 8	0
P_2	13	13 - 0.4 = 12.6	8.6
P_3	9	9 - 1.0 = 8	7

Average turnaround time

$$= \frac{8 + 12.6 + 8}{3}$$

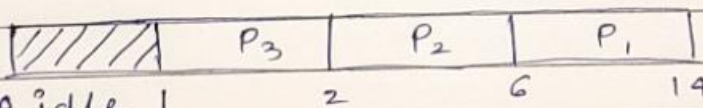
$$= \frac{28.6}{3}$$

$$= 9.53$$

c) The SJF algorithm is suppose to improve performance, but notice that we chose to run process P_1 at time 0 because we did not know that two shorter processes would arrive soon. Compute what the average turnaround time will be if the CPU is left idle for the first 1 unit and the SJF scheduling is used. Remember that processes P_1 and P_2 are waiting during this idle time, so their waiting time may increase. This algorithm could be called future-knowledge scheduling.

→

Gantt chart :-



Process	Arrival time	Burst time	CT	Turnaround Time	Waiting time
P_1	0.0	8	14	14	6
P_2	0.4	4	6	5.6	1.6
P_3	1.0	1	2	1	0

Average turnaround time

$$= \frac{14 + 5.6 + 1}{3}$$

$$= \frac{20.6}{3}$$

$$= 6.81$$