

## Instructions

Q1-Explain the difference between preemptive and nonpreemptive scheduling.

Preemptive scheduling is when a process can be interrupted and moved back to the ready queue if a higher priority process arrives.

Nonpreemptive Scheduling: once a process starts executing, it runs until it either completes or yields voluntarily.

Q2-Consider the following set of processes, with the length of the CPU burst time given in milliseconds:

Process	Burst Time	Priority
P1	2	2
P2	1	1
P3	8	4
P4	4	2
P5	5	3

The processes are assumed to have arrived in the order P1 , P2 , P3 , P4 , P5 , all at time 0.

a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, non-preemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2).

Chart : **FCFS**

p1   p2   p3   p4   p5
0    2    3    11    15    20

Chart: **SJF**

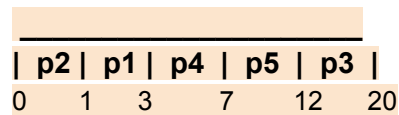
p2   p1   p4   p5   p3
0    1    3    7    12    20

Chart: **Priority Scheduling (nonpreemptive, "larger priority number implies a higher priority" )**

p3   p5   p1   p4   p2
0    8    13    15    19    20

\*Just incase i miss read the instructions, making one with the shorter priority

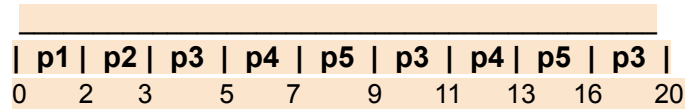
Chart: **Priority Scheduling (nonpreemptive, "shorter priority number implies a higher priority" )**



Avg waiting time:  $(0 + 1 + 3 + 7 + 12)/5 = 4.6$  ms

Avg TAT:  $(1 + 3 + 7 + 12 + 20)/5 = 8.6$  ms

Chart: **Round Robin (Quantum = 2)**



b. What is the turnaround time of each process for each of the scheduling algorithms in part a?

Turnaround Time = Completion Time - Arrival Time

Since all processes arrive at time 0, Turnaround Time = Completion Time

Chart : **FCFS**

Process	Completion Time	Turnaround Time (TAT)
p1	2	2
p2	3	3
p3	11	11
p4	15	15
p5	20	20

Chart: **SJF**

Process	Completion Time	Turnaround Time (TAT)
p2	1	1
p1	3	3
p4	7	7
p5	12	12
p3	20	20

Chart: **Priority Scheduling (nonpreemptive, higher number = higher priority)**

Process	Completion Time	Turnaround Time (TAT)
p2	8	8
p1	13	13
p4	15	15
p5	19	19
p3	20	20

Chart: **Priority Scheduling (nonpreemptive, lower number = higher priority)**

Process	Completion Time	Turnaround Time (TAT)
p2	1	1
p1	3	3
p4	7	7
p5	12	12
p3	20	20

Chart: **Round Robin (Quantum = 2)**

Process	Completion Time	Turnaround Time (TAT)
p1	2	2
p2	3	3
p3	20	20
p4	13	13
p5	16	16

c. What is the waiting time of each process for each of these scheduling algorithms?

Waiting Time=Turnaround Time-Burst Time

FCFS

Process	Burst Time	Completion	Turnaround Time	Waiting Time
P1	2	2	2	0
P2	1	3	3	2
P3	8	11	11	3
P4	4	15	15	11
P5	5	20	20	15

SJF

Process	Burst Time	Completion	Turnaround Time	Waiting Time
P2	1	1	1	0
P1	2	3	3	1
P4	4	7	7	3
P5	5	12	12	7
P3	8	20	20	12

\*Lower # priority is the same as the SJF, I will continue with the Highest # priority

Priority (Higher Number = Higher Priority)

Process	Brust Time	Completion	Turnaround Time	Waiting Time
P3	8	8	8	0
P5	5	13	13	8
P1	2	15	15	13
P4	4	19	19	15
P2	1	20	20	19

Round Robin (q = 2)

Process	Brust Time	Completion	Turnaround Time	Waiting Time
P1	2	2	2	0
P2	1	3	3	2
P3	8	20	20	12
P4	4	13	13	9
P5	5	16	16	11

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

Average Waiting time						
Algorithm	P1	P2	P3	P4	P5	Avg WT
FCFS	0	2	3	7	10	4.4
SJF	1	0	12	3	7	4.6
Priority	0	8	13	17	19	11
RR(q=2)	0	2	12	9	11	6.8

The average with the minimum average waiting time is **FCFS** with a waiting time of **4.4ms**.

Q3-Consider two processes, P1 and P2 , where  $p1 = 50$ ,  $t1 = 25$ ,  $p2 = 75$ , and  $t2 = 30$ .

a. Can these two processes be scheduled using rate-monotonic scheduling? Illustrate your answer using a Gantt chart such as the ones in Figure 5.21–Figure 5.24.

**\* I can't find the images mentioned in the questions. I will continue with the charts I've been doing from q # 1.**

**Gantt chart: RMS**

**RMS is a fixed-priority scheduling algorithm where processes with shorter periods get higher priority.**

```
| P1 | P1 | P1 | P1 | P1 | P2 | P2 | P2 | P2 | P2 | P2 | P1 | P1 | P1 | P1 | P1 | P2 | P2 | P2 | P2 |
0  5  10  15  20   25  30  35  40  45  50  55  60  65  70  75  80  85  90  95  100
```

**No, these two processes can't be scheduled using rate-monotonic scheduling.**

**P1 misses its deadline at  $t = 50$  ms, so RMS fails**

b. Illustrate the scheduling of these two processes using earliest-deadline-first (EDF) scheduling.

**Gantt chart: EDF**

```
| P1 | P1 | P1 | P1 | P1 | P2 | P2 | P2 | P2 | P2 | P2 | P1 | P1 | P1 | P1 | P1 | P2 | P2 | P2 | P2 |
0  5  10  15  20   25  30  35  40  45  50  55  60  65  70  75  80  85  90  95  100
```

**The charts look similar but this process is more dynamic and does meet the deadlines.**

Q4- Write a program (c++ or java) that computes turnaround time and average wait time for the processes listed in Q2 for FCFS and SJF scheduling . Upload your solution to GitHub and submit the link to your program.

Example Output:

----- FCFS -----

Process ID	Waiting Time	Turnaround Time
1	0	8

2		0		5
3		5		8
4		7		13

```
C:\Users\richa\.jdk\openjdk-23.0.1\bin\java.exe ...
```

```
----- FCFS -----
```

```
Process ID | Waiting Time | Turnaround Time
```

1		0		2
2		2		3
3		3		11
4		11		15
5		15		20

```
Average Waiting Time: 6.2
```

```
Average Turnaround Time: 9.8
```

```
----- SJF -----
```

```
Process ID | Waiting Time | Turnaround Time
```

2		0		1
1		1		3
4		3		7
5		7		12
3		12		20

```
Average Waiting Time: 4.6
```

```
Average Turnaround Time: 8.4
```