COMP 3411

Assignment 4

Martin Atanacio - T00684924

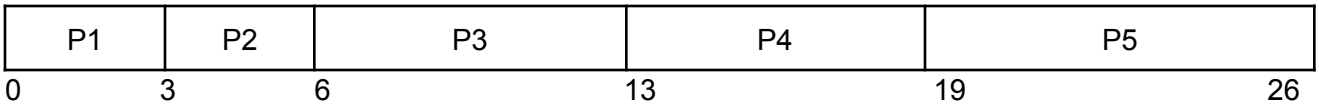
Turnaround time per process = completion time – arrival time

Wait time per process = turnaround time – burst time

Process	Arrival Time	Burst Time	Priority
P1	0	3	3
P2	2	3	3
P3	4	7	5
P4	5	6	8
P5	7	7	5

Question 1 - FCFS

Gantt Chart

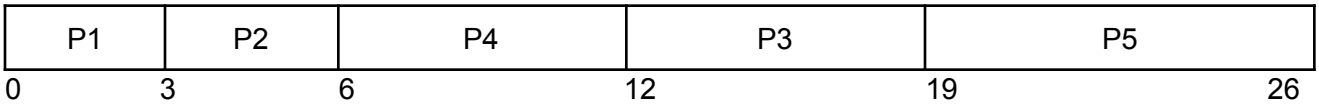


Solution

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	3	3	3-0 = 3	3-3 = 0
P2	2	3	6	6-2 = 4	4-3 = 1
P3	4	7	13	13-4 = 9	9-7 = 2
P4	5	6	19	19-4 = 14	14-6 = 8
P5	7	7	26	26-7 = 19	19-7 = 12
Averages				49/5 = 9.8	23/5 = 4.6

Question 2 - SJF

Gantt Chart



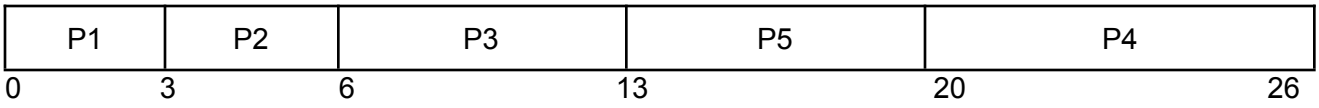
Solution

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	3	3	3-0 = 3	3-3 = 0
P2	2	3	6	6-2 = 4	4-3 = 1
P3	4	7	19	19-4 = 15	15-7 = 8
P4	5	6	12	12-5 = 7	7-6 = 1
P5	7	7	26	26-7 = 19	19-7 = 12
Averages				48/5 = 9.6	22/5 = 4.4

Question 3 - Priority Preemptive

(Low No. = Higher Priority)

Gantt Chart

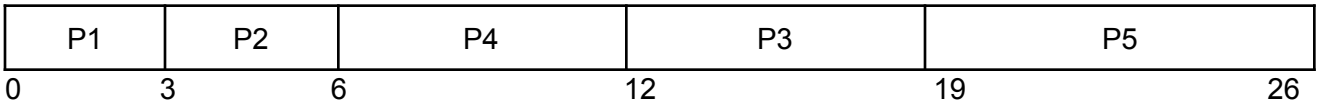


Solution

Process	Arrival Time	Burst Time	Priority	Completion Time	Turnaround Time	Waiting Time
P1	0	3	3	3	3-0 = 3	3-3 = 0
P2	2	3	3	6	6-2 = 4	4-3 = 1
P3	4	7	5	13	13-4 = 9	9-7 = 2
P4	5	6	8	26	26-5 = 21	21-6 = 15
P5	7	7	5	20	20-7 = 13	13-7 = 6
Averages					50/5 = 10	24/5 = 4.8

Question 4 - SRTF

Gantt Chart

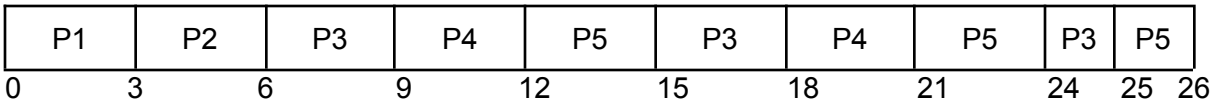


Solution

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	3	3	3-0 = 3	3-3 = 0
P2	2	3	6	6-2 = 4	4-3 = 1
P3	4	7	19	19-4 = 15	15-7 = 8
P4	5	6	12	12-5 = 7	7-6 = 1
P5	7	7	26	26-7 = 19	19-7 = 12
Averages				48/5 = 9.6	22/5 = 4.4

Question 5 - RR
(Quantum = 3)

Gantt Chart



Solution

Process	Arrival Time	Burst Time	Completion Time	Turnaround Time	Waiting Time
P1	0	3	3	3-0 = 3	3-3 = 0
P2	2	3	6	6-2 = 4	4-3 = 1
P3	4	7	25	25-4 = 21	21-7 = 14
P4	5	6	21	21-5 = 16	16-6 = 10
P5	7	7	26	26-7 = 19	19-7 = 12
Averages				63/5 = 12.6	37/5 = 7.4

Question 6

Starvation happens when a process is unable to obtain the required resources from the CPU for an extended period of time, while other processes keep getting scheduled ahead of it. Thus, making a process delayed or stuck indefinitely.

1. **FCFS** can cause starvation because it schedules processes in the order that they arrive. If a process with a high burst time arrives first and other short processes arrive afterward, the long process will use the CPU for an extended period of time, forcing other processes to wait and possibly leading to starvation.

2. **SJF** can also lead to starvation for longer processes if shorter processes keep arriving. The long processes might continuously be delayed by the arrival of shorter ones, causing them to starve.

3. **SRTF** is a preemptive version of SJF where the process that is currently executing can be preempted if a new process with a shorter remaining time arrives. Similar to SJF, as shorter processes arrive, longer processes are continuously being preempted and can lead to starvation.

4. **RR** is preemptive and it works by providing each process with a given a fixed time slice to execute. Every process gets a fair chance to execute, however, processes with varying execution times might not get enough CPU time if the time slice is too small. However, compared to other algorithms, RR is less prone to starvation.

5. **Priority Scheduling** assigns priorities to processes and executes higher-priority processes before lower-priority ones. This can lead to starvation if a process with a low priority keeps getting preempted by higher-priority processes, thus, reducing the chances of the low-priority process executing and causing it to starve.