Assignment 2 (MTL458)

Authors: Hanish Goyal [2020MT10805] — Harshvardhan Patel [2020MT10808]

1 Introduction

This report consists of the results, of the experiments we conducted on the various scheduling strategies, namely, First Come First Serve (FCFS), Shortest Job First (SJF), Shortest Job Remaining First (SJRF), Round Robin (RR) and Multi-Level Feedback Queue (MLFQ).
Note: The experiments were conducted within a Linux-based environment; therefore, it is imperative to execute them exclusively on Linux-based systems for verification purposes.

Description of the Input: The input for the program is produced by the program itself. The input consists of two sets of Workloads. Each workload is a set of 10 Jobs. These Jobs are produced so that the waiting time between two jobs follows *Exponential Distribution*. The main difference between the two sets of workloads, $W1$ and $W2$, is that the parameter λ of exponential distribution is small for $W1$ and large for $W2$. The parameter for $W1$ and $W2$ be λ_1 and λ_2 respectively. Since, $\lambda_1 = 0.1$ and $\lambda_2 = 1$ in our experiments, the expected waiting time in $W1$ is 10 times greater than in $W2$. So, the Jobs in $W2$ are very densely packed, and the Jobs in $W1$ are sparsely packed. For both these kinds of workload we will see which of the following methods perform the best.

Process ID 1	Arrival Time	Burst Length
P1	0.9263	20.00
P2	4.6338	21.00
P3	18.6485	21.00
P4	24.5991	7.00
P5	42.4917	24.00

Table 1: Workload 1 ($W1$)

Process ID 1	Arrival Time	Burst Length
P1	0.4044	20.00
P2	0.7192	11.00
P3	1.9913	23.00
P4	2.1990	8.00
P5	3.0527	12.00

Table 2: Workload 2 ($W2$)

2 First Come First Serve

The following **Gantt-Chart** shows how the 5 Jobs in $W1$ are scheduled.

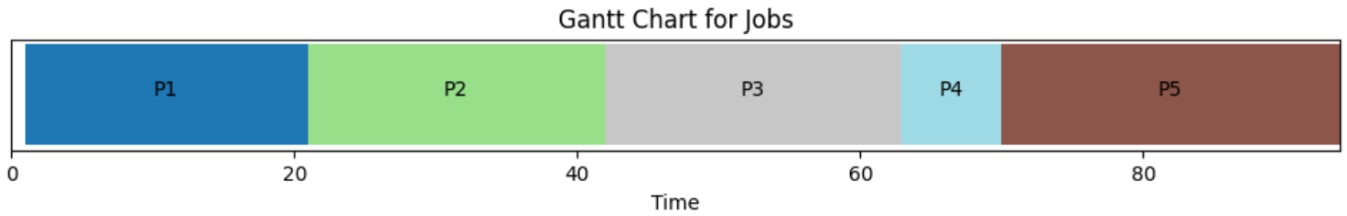

Figure 1: Your Image Caption

Average Turnaround Time : 39.666
Average Response Time : 21.066

The following **Gantt-Chart** shows how the 5 Jobs in $W2$ are scheduled.

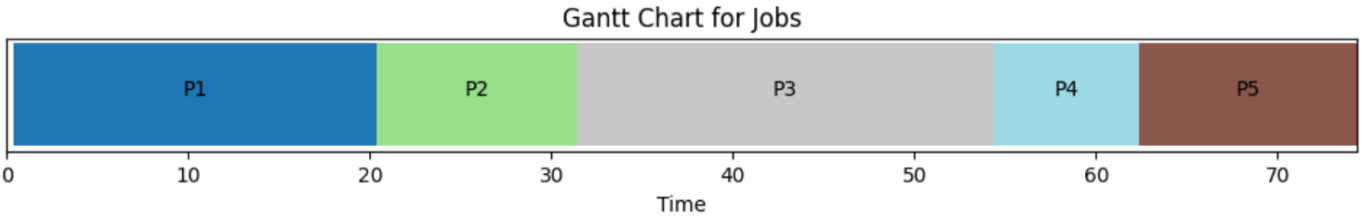

Figure 2: Your Image Caption

Average Turnaround Time : 46.931
Average Response Time : 32.131

The above observations show us that the FCFS strategy works better when the processes are well-separated. As a result, both the factors, Turnaround Time and Response Time, are less for $W1$. However, since the strategy is too trivial, it can be seen that both values are greater than other strategies.

3 Shortest Job First

The following **Gantt-Chart** shows how the 5 Jobs in $W1$ are scheduled.

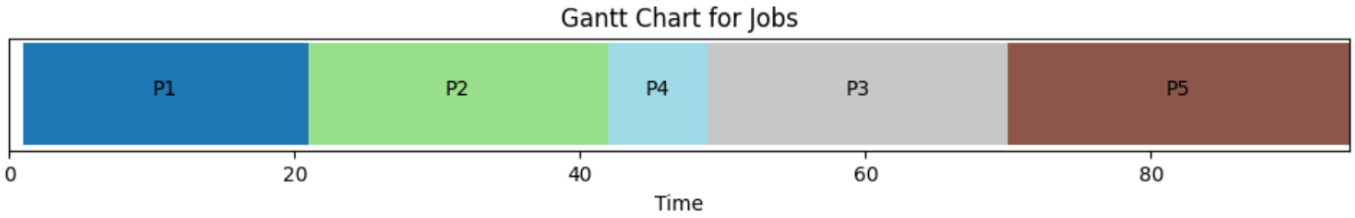

Figure 3: Your Image Caption

Average Turnaround Time : 36.866
Average Response Time : 18.266

The following **Gantt-Chart** shows how the 5 Jobs in $W2$ are scheduled.

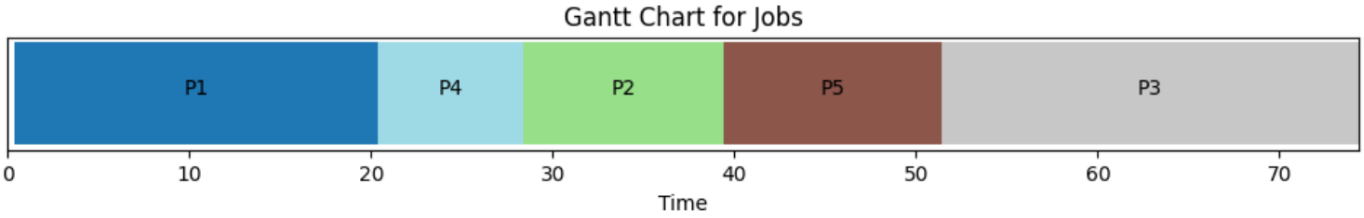

Figure 4: Your Image Caption

Average Turnaround Time : 41.131
Average Response Time : 26.331

Again, the above observations show that the SJF strategy works better for the case when the Jobs are well-separated. Also, one can see that in the case of overlapping Jobs, there is a sharp-decrease in the Turnaround Time without compromising the Response Time. Hence, this strategy is better than the FCFS strategy if you want all the Jobs to be completed as early as possible.

4 Shortest Remaining Time First

The following **Gantt-Chart** shows how the 5 Jobs in $W1$ are scheduled.

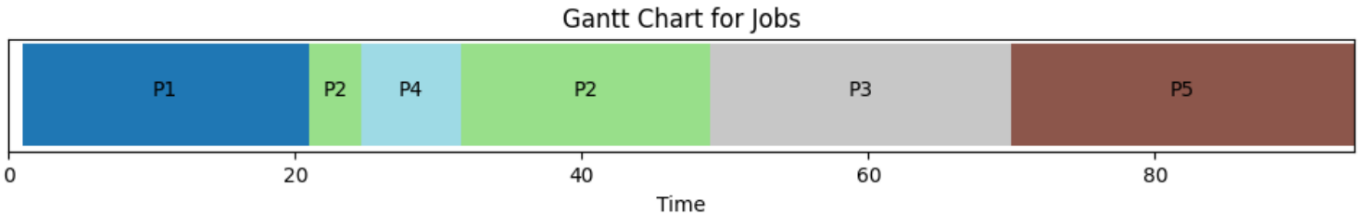

Figure 5: Your Image Caption

Average Turnaround Time : 34.801
Average Response Time : 14.801

The following **Gantt-Chart** shows how the 5 Jobs in $W2$ are scheduled.

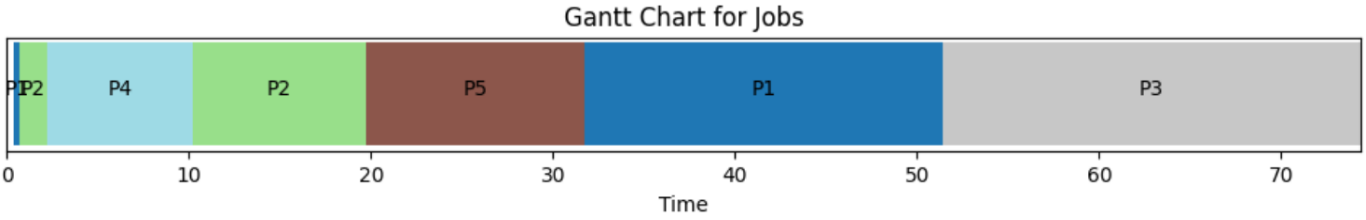

Figure 6: Your Image Caption

Average Turnaround Time : 35.816
Average Response Time : 13.216

SRJF is a greedy scheduling algorithm whose only objective is to minimise the Turnaround Time for all the Jobs. This can be verified easily by observing that the Turnaround Time in this case is the least among all the algorithms. Regarding fairness, i.e., Response Time, we can note that SRJF does not perform as well as RR or MLFQ since there is a trade-off between these two evaluation parameters. Also, the value of the evaluation parameters in the two cases doesn't differ much. Hence, SRJF performs equally for $W1$ and $W2$ type Workloads.

5 Round Robin

The following **Gantt-Chart** shows how the 5 Jobs in $W1$ are scheduled when the TS of RR is 6.

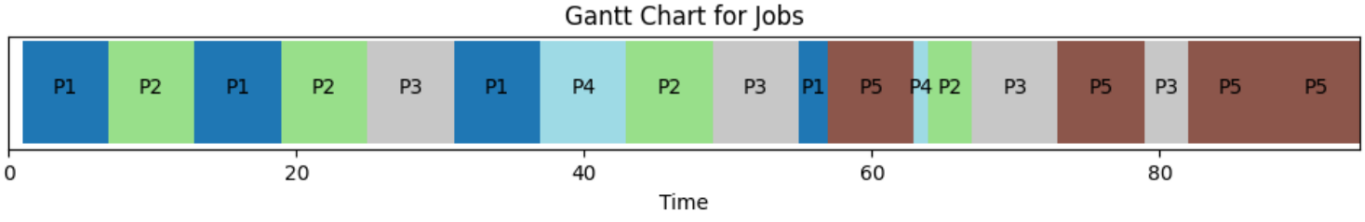

Figure 7: Your Image Caption

Average Turnaround Time : 54.466
Average Response Time : 7.066

The following **Gantt-Chart** shows how the 5 Jobs in $W2$ are scheduled when the TS of RR is 6.

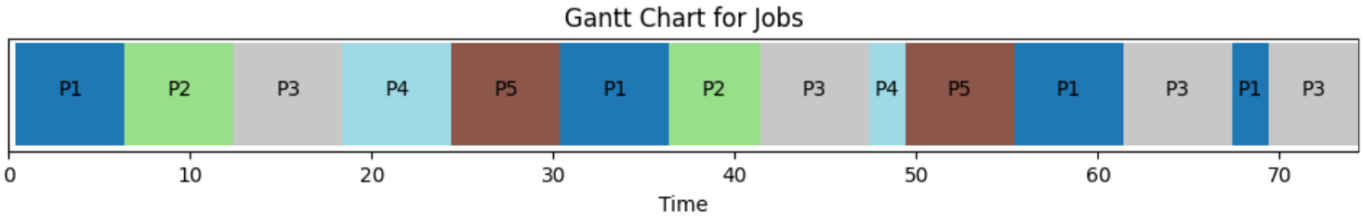

Figure 8: Your Image Caption

Average Turnaround Time : 56.331
Average Response Time : 10.731

We can easily observe that the Response Times for both types of Workloads are far less than those in any other strategies till now. Also, there is a little spike in the Turnaround Time. This is due to the main idea behind RR, i.e. to ensure Fairness over greedily completing the jobs.
Here, we experimented with different values of the Time Slice parameter. Considering $W1$ as the Workload, the following plots show the variation of Turnaround Time and Response Time while we vary the value of the Time Slice from 2 to 6.

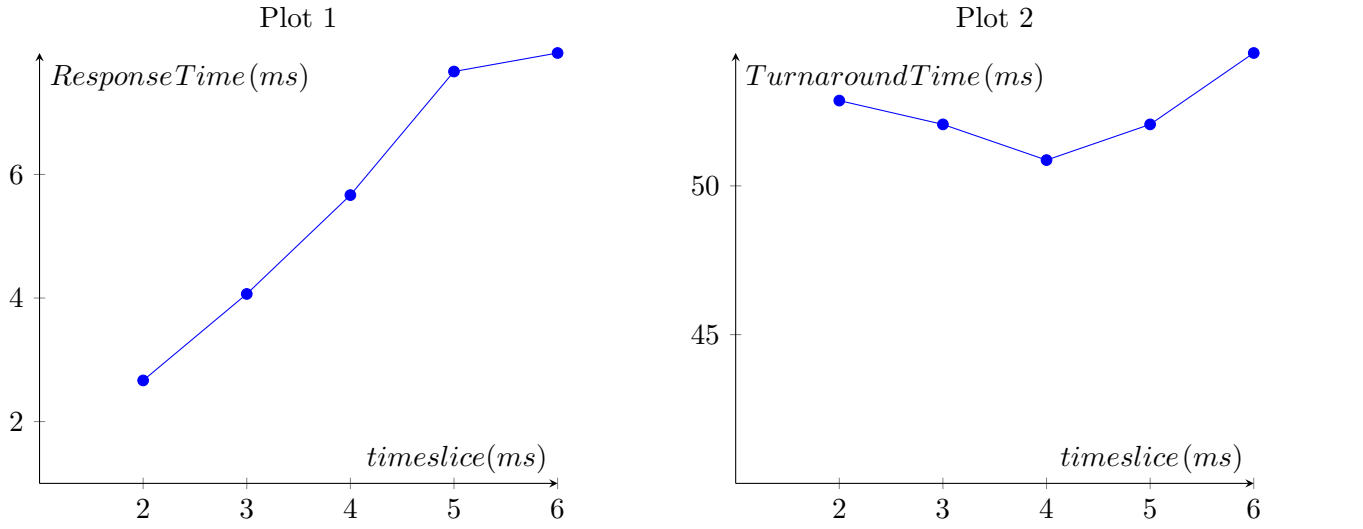

Figure 9: Plot 1 and Plot 2 show the variation of Response Time and Turnaround Time with TS respectively

Here, the Turnaround Time does not vary as much with the small changes in Time Slice. But the Response Time shows a sharp-decrease with decreasing TS. Hence, we can conclude that smaller TS results in better Response Times. But, one must always keep in mind that decreasing the value of TS indefinitely won't be a good thing for the processor. This is the case since we are not considering the resources we spent on a Context-Switch. Context-Switch is not easy to perform, and a very small TS will make things worse for the processor as Context-Switch will become more frequent.

6 Multi Level Feedback Queue

The following **Gantt-Chart** shows how the 5 Jobs in $W1$ are scheduled, when the $TS1 = 3$, $TS2 = 4$, $TS3 = 6$ and $BP = 11$.

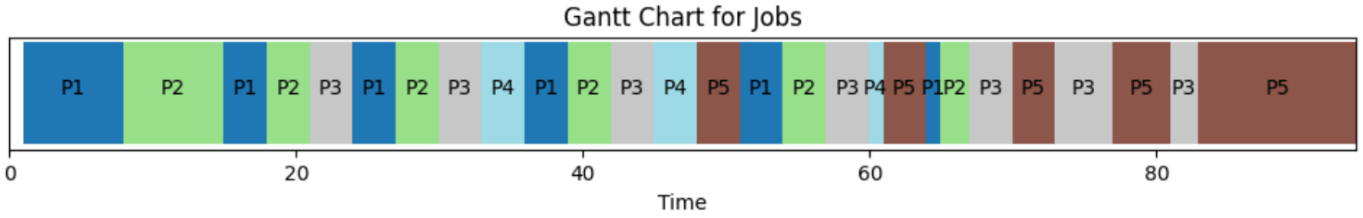

Figure 10: Your Image Caption

Average Turnaround Time : 55.666
Average Response Time : 3.866

The following **Gantt-Chart** shows how the 5 Jobs in $W2$ are scheduled, when the $TS1 = 3$, $TS2 = 4$, $TS3 = 6$ and $BP = 11$.

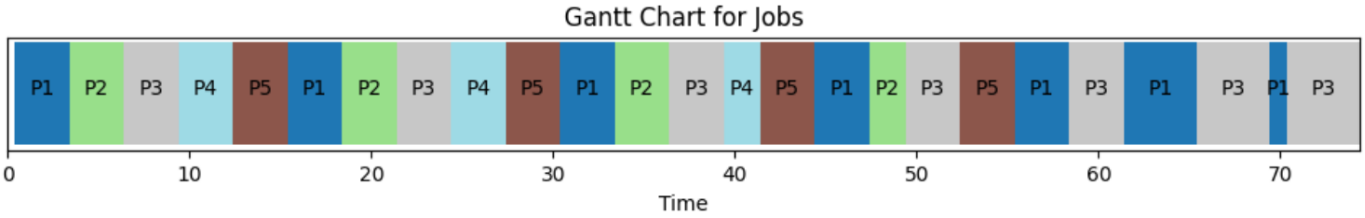

Figure 11: Your Image Caption

Average Turnaround Time : 56.531
Average Response Time : 4.731

We can see that the MLFQ strategy, the lowest Response Time is achieved. Almost similar Response Times and Turnaround Times have been achieved for both types of Workloads. Along with the sharp decrease in the Response Times, one can also note the sharp increase in Turnaround Times. But still, the Turnaround Times achieved in this case is better than the RR case. Hence, one can conclude that to get better responses from processes without taking too long to finish them, MLFQ is the best strategy available.
We further went on to experiment on different cases of parameters in MLFQ. The following table summarises the result.

Time Slice 1	Time Slice 2	Time Slice 3	Boost Parameter	Turnaround Time	Response Time
1	2	4	7	53.066	1.266
2	3	5	9	53.666	1.666
3	4	6	11	55.666	3.866

Table 3: Turnaround Time and Response Time Variation for $W1$ Workload.

Time Slice 1	Time Slice 2	Time Slice 3	Boost Parameter	Turnaround Time	Response Time
1	2	4	7	55.731	0.731
2	3	5	9	56.331	2.731
3	4	6	11	56.531	4.731

Table 4: Turnaround Time and Response Time Variation for $W2$ Workload.

Just as with the case of RR, here, too, the Turnaround Time doesn't vary much by changing the values of the parameters. Even the Response Time doesn't improve much in the case of $W1$. But for the $W2$ case, we can see that reducing the parameters can improve the response time. Hence, we can conclude that reducing the above parameter values can improve the Response Time for the Workload when the Jobs are densely packed. However, as previously mentioned, it's crucial to remember that managing Context-Switches is complex. Excessive Context Switching can lead to suboptimal scheduler design.