Ali Emir Güzey - Section 3 - 22003186
Mustafa Kaan Koç - Section 3 - 21903358
28.04.2023

Project 2 Report

In this project, we implemented a CPU-scheduling simulation using the `pthread` library in the C programming language. We also utilized synchronization tools from the same library, such as mutex locks and condition variables. This report consists of the results acquired from the experiments made on the implemented CPU-scheduling simulation named `mps.c`.

Experiments are done in 3 different setups. First, FCFS, SJF, and RR (q=20) algorithms were run with a single ready queue and step-by-step increasing processor count ( ). Second, these three algorithms were run with a ready queue for each processor and step-by-step increasing processor count, where the Round Robin method was used to select which queue to put the newcomer burst ( ). The third setup was the same as the second setup, except the Load Balance method was used to pick the queue for the next burst instead of the Round Robin method ( ). The same input file was used in all experiments (see App.).

In FCFS (First Come, First Served) algorithm's single queue method (S), as the number of processors (N) increased, the average turnaround time gradually decreased down to a certain level (~88 ms). When N ≥ 4, the average turnaround time converged to a specific value (~88 ms). In the multi-queue Round Robin Method (RM) approach, the average turnaround time behaved as in the single-queue method. In the multi-queue Load Balance Method (LM), the average turnaround time of the processes maintains a roughly flat line with minor fluctuations as N increases. For all N values, the single-queue approach performed better than both of the multi-queue approaches, except N=1, where multi-queue is not applicable. Between the multi-queue approaches, RM's performance increased as the number of processors increased, while LM provided a consistent performance regardless of the processor count. However, for increasing values of N, RM was a better approach than LM.

In SJF (Shortest Job First) algorithm's single queue method (S), as the number of processors (N) rose, the average turnaround time fell until it converged to a specific value. In the multi-queue Round Robin Method (RM), the behavior of the average turnaround time of the processes was similar to the values in the single-queue method. In the Load Balance Method (LM), the average turnaround time went up and down as N increased. The average turnaround time in (S) was lower than in (RM) until N = 6 except for N=1 (Notice that N ≠ 1 in the multi-queue methods). The average turnaround times in (S) and (RM) resembled each other after N ≥ 6. However, the value of the average turnaround time in (S) was always smaller than the values in (LM) except for N=1. The average turnaround time in (RM) and (LM) methods were similar when N = 2, but values in (LM) were greater than the values in (RM) when n is greater than 2.

In the Round Robin (RR) algorithm's single queue method (S), as the number of processors (N) increased, the average turnaround time of the processes went down until it started converging to a particular value. In both the multi-queue Load Balance Method (LM) and multi-queue Round Robin Method (RM), the average turnaround time acted the same as in the single-queue method. The average turnaround time in S and RM were quite similar to each other when 4 < N except N=1 (there is no value when N=1 in a multi-queue method), but values in S were smaller than values in RM until N reached 5. The values of the average turnaround time in S were lower than the values in LM for each N value except N=1 because N cannot be equal to 1 in a multi-queue approach. The average turnaround times in RM were always smaller than the average turnaround time in LM.

In Round Robin Method's single queue, as the time slice (Q) increased, the average turnaround time of the processes decreased when N=1. However, the values for other N values did not differ by much. In the Round Robin Method (RM), as Q values rose, the average turnaround time fell when N=1, but the average turnaround time did not react the same for other N values; they were similar to each other. In the Load Balance Method (LM), when Q = 10, the average turnaround time converged to ~128 ms. When Q = 20, the average turnaround time converged to ~127 ms. When Q = 40, the average turnaround time converged to ~ 118 ms.

In the single queue approach, the speed of the algorithms in terms of lower turnaround times was as SJF > FCFS > RR when N=1. As the N value increased, the speed of the algorithms got closer to each other at ~87 ms.

In the Round Robin multi-queue approach, the RR algorithm was slower than both SJF and FCFS algorithms in terms of the average turnaround time. SJF and FCFS were almost identical in terms of average turnaround times.

In the Load Balance Method multi-queue approach, the RR algorithm was the slowest one among other algorithms. SJF and FCFS algorithms fluctuated and changed places for different N values.

The exact results from the experiments can be found in the following tables and plots.

Table 1: Turnaround times for FCFS algorithm

ALGORITHM	N	SAP	QS	TURNAROUND(MS)
FCFS	1	S	NA	591.62
FCFS	2	S	NA	101.965
FCFS	3	S	NA	89
FCFS	4	S	NA	87.86
FCFS	5	S	NA	87.793
FCFS	6	S	NA	87.724
FCFS	7	S	NA	87.379
FCFS	8	S	NA	87.448
FCFS	9	S	NA	87.586
FCFS	10	S	NA	87.578
FCFS	2	M	RM	119.517
FCFS	2	M	LM	117.31
FCFS	3	M	RM	95.551
FCFS	3	M	LM	113.689
FCFS	4	M	RM	90.068
FCFS	4	M	LM	115.068
FCFS	5	M	RM	87.689
FCFS	5	M	LM	115
FCFS	6	M	RM	88.034
FCFS	6	M	LM	113.206
FCFS	7	M	RM	88.034
FCFS	7	M	LM	119.137
FCFS	8	M	RM	87.965
FCFS	8	M	LM	113.172
FCFS	9	M	RM	87.931
FCFS	9	M	LM	119.482
FCFS	10	M	RM	87.655
FCFS	10	M	LM	114.655

Table 2: Turnaround times for SJF algorithm

ALGORITHM	N	SAP	QS	TURNAROUND(MS)
SJF	1	S	NA	339.448
SJF	2	S	NA	98.586
SJF	3	S	NA	89.068
SJF	4	S	NA	88.241
SJF	5	S	NA	87.655
SJF	6	S	NA	87.517
SJF	7	S	NA	87.344
SJF	8	S	NA	87.655
SJF	9	S	NA	87.724
SJF	10	S	NA	87.517
SJF	2	M	RM	119.517
SJF	2	M	LM	121.517
SJF	3	M	RM	96.137
SJF	3	M	LM	120.448
SJF	4	M	RM	90.379
SJF	4	M	LM	114.793
SJF	5	M	RM	88.034
SJF	5	M	LM	112.896
SJF	6	M	RM	87.931
SJF	6	M	LM	119.379
SJF	7	M	RM	87.896
SJF	7	M	LM	119.379
SJF	8	M	RM	87.862
SJF	8	M	LM	113.137
SJF	9	M	RM	87.862
SJF	9	M	LM	113.344
SJF	10	M	RM	87.862
SJF	10	M	LM	113.448

Table 3: Turnaround times for RR algorithm when q=10

ALGORITHM	N	SAP	QS	Q	TURNAROUND(MS)
RR	1	S	NA	10	728.586
RR	2	S	NA	10	113.379
RR	3	S	NA	10	93.275
RR	4	S	NA	10	90.896
RR	5	S	NA	10	90
RR	6	S	NA	10	89.482
RR	7	S	NA	10	89.448
RR	8	S	NA	10	90
RR	9	S	NA	10	89.551
RR	10	S	NA	10	89.103
RR	2	M	RM	10	140.724
RR	2	M	LM	10	147.068
RR	3	M	RM	10	104.137
RR	3	M	LM	10	131.034
RR	4	M	RM	10	95.62
RR	4	M	LM	10	128.896
RR	5	M	RM	10	89.896
RR	5	M	LM	10	128.275
RR	6	M	RM	10	89.931
RR	6	M	LM	10	128.413
RR	7	M	RM	10	89.586
RR	7	M	LM	10	128.482
RR	8	M	RM	10	89.655
RR	8	M	LM	10	128.827
RR	9	M	RM	10	89.551
RR	9	M	LM	10	127.758
RR	10	M	RM	10	89.896
RR	10	M	LM	10	128.413

Table 4: Turnaround times for RR algorithm when q=20

ALGORITHM	N	SAP	QS	Q	TURNAROUND(MS)
RR	1	S	NA	20	696.724
RR	2	S	NA	20	109.827
RR	3	S	NA	20	92.586
RR	4	S	NA	20	89.551
RR	5	S	NA	20	89.241
RR	6	S	NA	20	89.689
RR	7	S	NA	20	89.137
RR	8	S	NA	20	88.517
RR	9	S	NA	20	89.344
RR	10	S	NA	20	88.551
RR	2	M	RM	20	134.586
RR	2	M	LM	20	135.896
RR	3	M	RM	20	104.103
RR	3	M	LM	20	130.034
RR	4	M	RM	20	92.793
RR	4	M	LM	20	127.206
RR	5	M	RM	20	90.034
RR	5	M	LM	20	126.448
RR	6	M	RM	20	89.862
RR	6	M	LM	20	127.241
RR	7	M	RM	20	89.551
RR	7	M	LM	20	127.793
RR	8	M	RM	20	88.896
RR	8	M	LM	20	127.034
RR	9	M	RM	20	88.827
RR	9	M	LM	20	126.172
RR	10	M	RM	20	88.482
RR	10	M	LM	20	126.827

Table 5: Turnaround times for RR algorithm when q=40

ALGORITHM	N	SAP	QS	Q	TURNAROUND(MS)
RR	1	S	NA	40	602.62
RR	2	S	NA	40	104.448
RR	3	S	NA	40	90.206
RR	4	S	NA	40	88.517
RR	5	S	NA	40	88.034
RR	6	S	NA	40	88.068
RR	7	S	NA	40	87.931
RR	8	S	NA	40	87.862
RR	9	S	NA	40	88.206
RR	10	S	NA	40	88.068
RR	2	M	RM	40	128.413
RR	2	M	LM	40	127.965
RR	3	M	RM	40	100.31
RR	3	M	LM	40	125.206
RR	4	M	RM	40	92.275
RR	4	M	LM	40	118.62
RR	5	M	RM	40	88.206
RR	5	M	LM	40	117.827
RR	6	M	RM	40	88.551
RR	6	M	LM	40	118
RR	7	M	RM	40	88.655
RR	7	M	LM	40	118.586
RR	8	M	RM	40	88.241
RR	8	M	LM	40	120.551
RR	9	M	RM	40	88.31
RR	9	M	LM	40	118.206
RR	10	M	RM	40	88.241
RR	10	M	LM	40	117.689

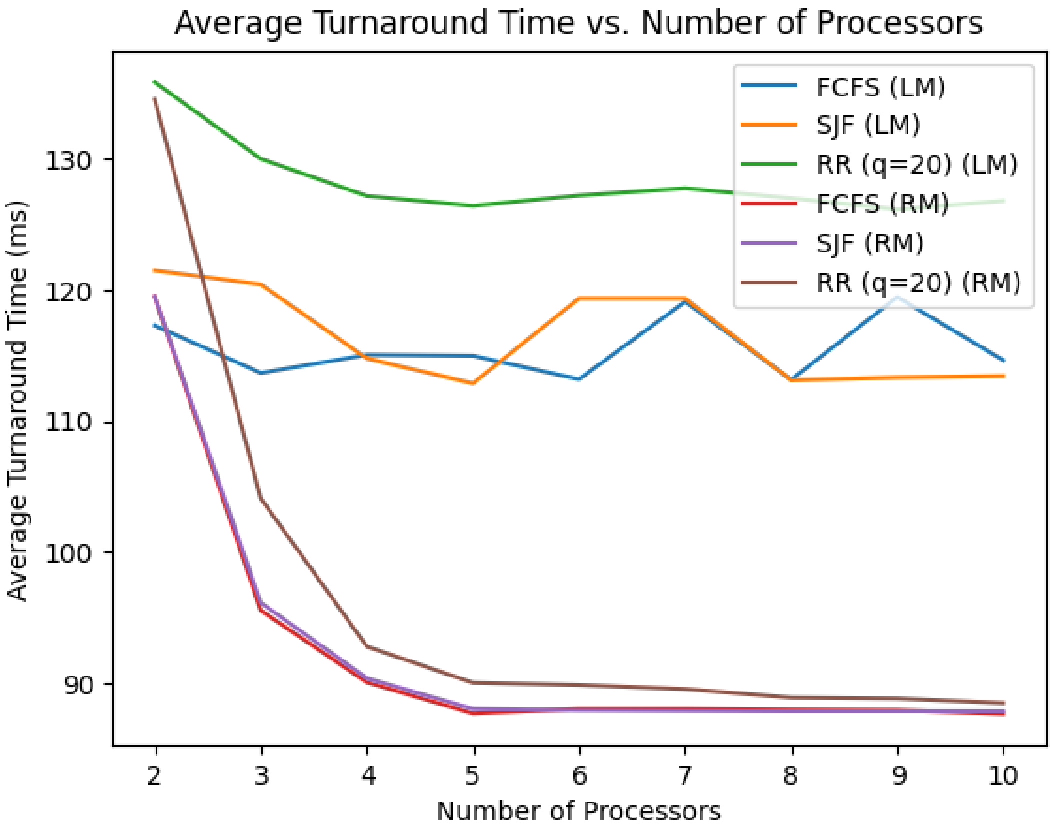

Figure 1: Average turnaround time vs. number of processes for multi-queue

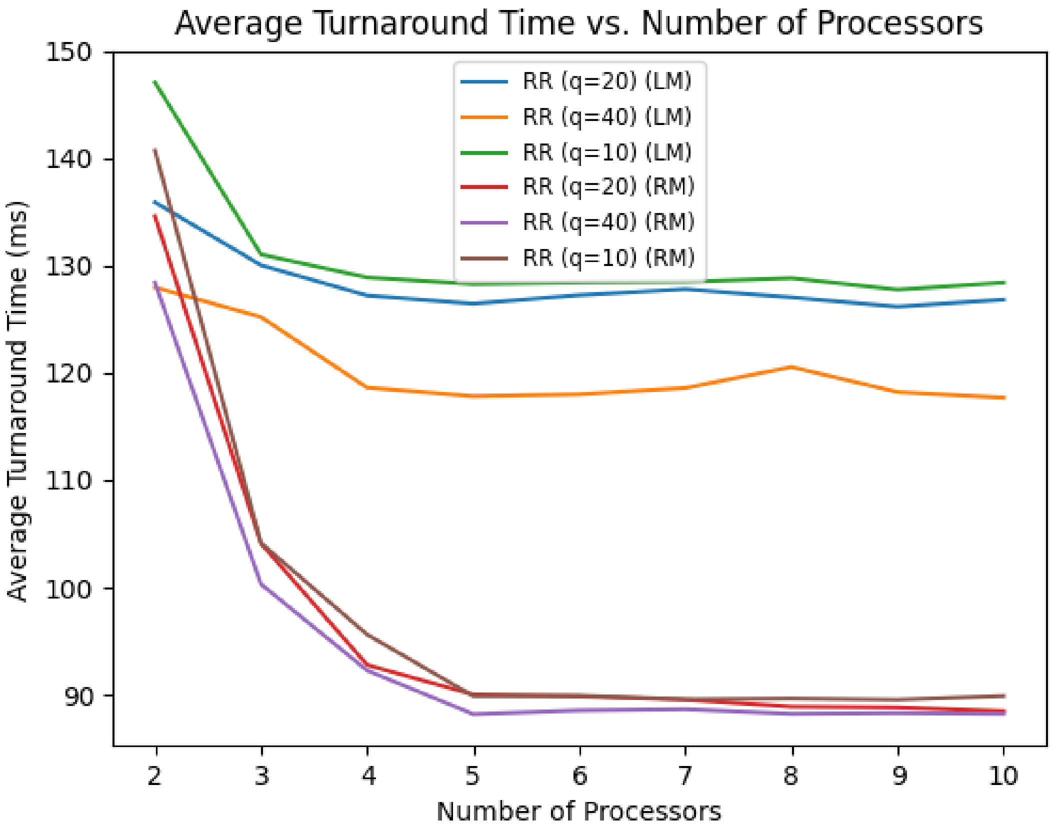

Figure 2: Average turnaround time vs. number of processes for multi-queue of RR

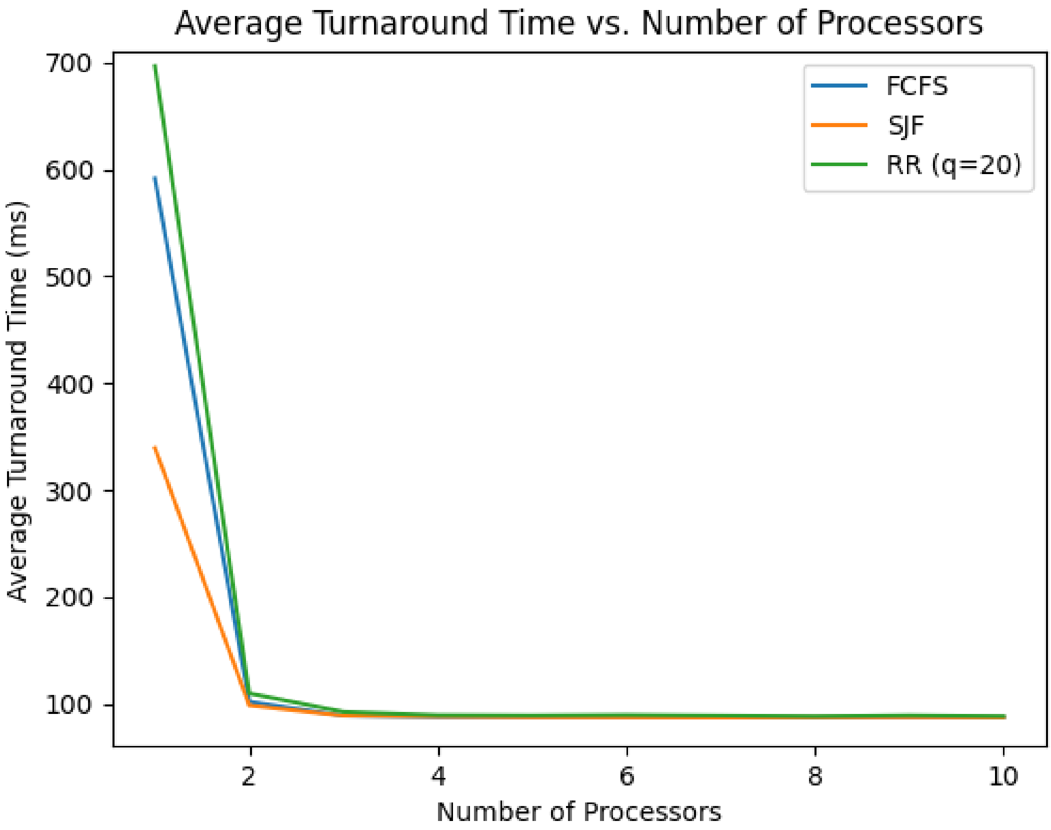

Figure 3: Average turnaround time vs. number of processes for single-queue

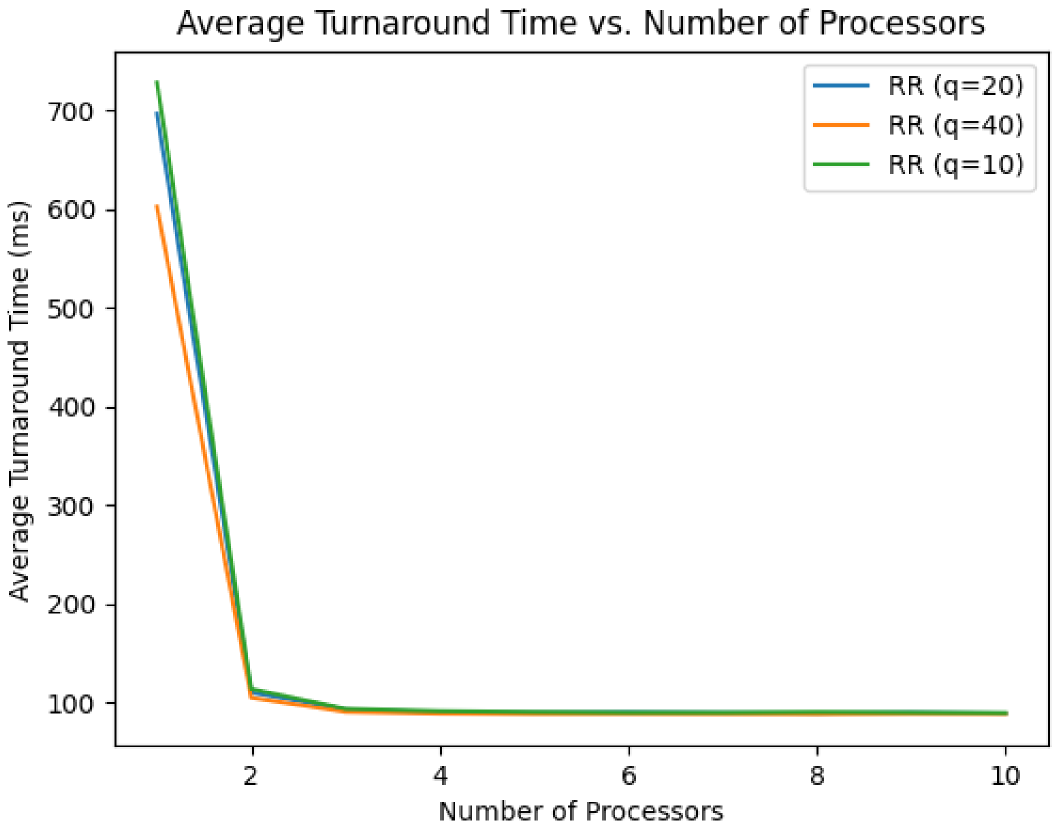

Figure 4: Average turnaround time vs. number of processes for single-queue of RR

APPENDIX (Input File)

```
PL 20
IAT 10
PL 80
IAT 50
PL 40
IAT 10
PL 120
IAT 20
PL 60
IAT 100
PL 80
IAT 30
PL 50
IAT 10
PL 80
IAT 20
PL 200
IAT 10
PL 40
IAT 100
PL 60
IAT 40
PL 30
IAT 80
PL 70
IAT 50
PL 30
IAT 50
PL 120
IAT 30
PL 40
IAT 20
PL 80
IAT 20
PL 160
IAT 100
PL 300
IAT 130
PL 20
IAT 50
PL 70
IAT 50
PL 180
IAT 20
PL 70
IAT 150
PL 130
```

```
IAT 120
PL 40
IAT 120
PL 60
IAT 10
PL 50
IAT 50
PL 20
IAT 80
PL 40
IAT 70
PL 200
```