CS342 Operating Systems

Fall 2024

PROJECT #2

MULTI-PROCESSOR SCHEDULING SIMULATION

Nov 13, 2024

Elif Sorguç - 22003782

Habibe Yılmaz - 22002766

EXPERIMENT RESULTS

These experiments are done by inputfile.txt that is given as an example in the project pdf. There are 7 processes.

inputfile.txt:

```
PL 80
IAT 100
PL 200
IAT 10
PL 50
IAT 70
PL 120
IAT 250
PL 100
IAT 20
PL 40
IAT 30
PL 80
```

num_processors	multi_queue	NA/RM/LM	scheduling_algo	avg_turnaround (ms)
1	-	NA	FCFS	184.40
2	-	NA	FCFS	103.00
4	-	NA	FCFS	100.50
1	-	NA	SJF	185.60
2	-	NA	SJF	102.67
4	-	NA	SJF	101.00
1	+	NA	FCFS	193.20
2	+	NA	FCFS	121.50
4	+	NA	FCFS	102.33
1	+	NA	SJF	192.40
2	+	NA	SJF	120.83
4	+	NA	SJF	100.83

num_processors	multi_queue	NA/RM/LM	scheduling_algo	avg_turnaround (ms)
2	+	RM	FCFS	119.83
2	+	RM	SJF	120.17
4	+	RM	FCFS	101.17
4	+	RM	SJF	100.33
2	+	LM	FCFS	120.67
2	+	LM	SJF	120.50
4	+	LM	FCFS	101.50
4	+	LM	SJF	101.33

For SJF: 6 processors and its division in multi queue with LM

pid	cpu	burst	arv	finish	waiting	turnaround
0	0	80	1	81	0	80
1	1	200	102	307	5	205
2	2	50	114	168	4	54
3	3	120	188	313	5	125
4	4	100	443	547	4	104
5	5	40	468	510	2	42
6	0	80	499	584	3	85

avg_turnaround (ms): 99.29

For SJF: 6 processors and its division in multi queue with RM

pid	cpu	burst	arv	finish	waiting	turnaround
0	0	80	1	83	2	82
1	1	200	101	305	4	204
2	2	50	111	161	0	50
3	3	120	185	305	0	120
4	4	100	438	543	5	105
5	5	40	463	504	1	41
6	0	80	495	577	2	82

avg_turnaround (ms): 97.71

For FCFS: 6 processors and its division in multi queue with LM

pid	cpu	burst	arv	finish	waiting	turnaround
0	0	80	1	82	1	81
1	1	200	103	308	5	205
2	2	50	115	170	5	55
3	3	120	189	313	4	124
4	4	100	443	546	3	103
5	5	40	468	508	0	40
6	0	80	500	581	1	81

avg_turnaround (ms): 98.43

For FCFS: 6 processors and its division in multi queue with RM

pid	cpu	burst	arv	finish	waiting	turnaround
0	0	80	1	84	3	83
1	1	200	107	311	4	204
2	2	50	119	174	5	55
3	3	120	192	313	1	121
4	4	100	444	545	1	101
5	5	40	468	508	0	40
6	0	80	503	588	5	85

avg_turnaround (ms): 98.43

EXPERIMENT INTERPRETATIONS

1. Single vs. Multi-Queue Approach

When comparing single and multi-queue approaches, it appears that the multi-queue configuration generally gives lower turnaround times than the single-queue setup, especially as the number of processors increases. This is due to each processor having its own queue, which reduces load for a single shared queue and allows for better parallel processing.

2. Effect of Processor Count on Turnaround Time

For both FCFS and SJF scheduling algorithms, the **average turnaround time tends to decrease** as the number of **processors increases**. This is expected as having more processors allows more processes to be executed simultaneously, and that is reducing overall waiting time in the queue.

For instance, under the single-queue FCFS configuration, increasing processors from 1 to 4 reduced the turnaround time significantly (from 184.4 ms to 100.5 ms). The same trend is observed for the SJF algorithm.

Similarly, in the multi-queue FCFS setup, the average turnaround time improves from 193.2 ms (1 processor) to 102.33 ms (4 processors).

3. Comparison of FCFS vs. SJF Scheduling Algorithms

SJF generally **results in lower turnaround times** compared to **FCFS**, as seen across various configurations. This is consistent with the theory since SJF prioritizes shorter processes, leading to lower average waiting and turnaround times.

For example, with 2 processors in a single-queue setup, SJF produced a turnaround time of 102.67 ms, while FCFS resulted in 103 ms. While the difference is small here, it’s consistent that SJF usually performs better or similarly to FCFS.

Under multi-queue settings with RR queue allocation, SJF provided slightly better turnaround times than FCFS.

4. Effect of Queue Selection Methods (RM vs. LM) in Multi-Queue Approach

When using the Round-Robin (RM) and Load-Balancing (LM) methods for queue selection in the multi-queue approach, there doesn’t appear to be a huge difference in turnaround times for both FCFS and SJF. However, LM shows a slight advantage in certain cases.

For instance, with 4 processors, Load-Balancing FCFS resulted in an average turnaround time of 101.5 ms compared to 101.17 ms for Round-Robin FCFS.

The Load-Balancing method is intended to distribute workloads more evenly across processors, which should theoretically give better results when workloads vary significantly. However, in this experiment, the impact is minimal, which shows that the workloads were fairly balanced or that the RM method also distributed jobs effectively in this case.

CONCLUSION

In summary, SJF consistently outperforms FCFS in terms of reducing turnaround time across various configurations. Multi-queue setups tend to perform better than single-queue setups, especially as the number of processors increases, due to reduced contention. The difference between RM and LM queue allocation methods is minor in these experiments, though LM might show benefits with more uneven workloads.