2022-2023 SPRING

CS 342 - OPERATING SYSTEMS PROJECT 02

Multiprocessor Scheduling, Threads, Synchronisation

28.04.2023

Group Members:

Emre Karataş - 22001641

İpek Öztaş - 22003250

A) PERFORMANCE OF SINGLE QUEUE APPROACH VS MULTI QUEUE APPROACH(*):

***Note that every integer value represented as time in this report is in ms.**

In this part of the report, multiprocessor scheduling by using single queue approach vs multi queue approach will be evaluated based on the FCFS algorithm with processor number (N) = 2.

Here are the results obtained from the Linux command line by running mps.c on a multi queue approach.

Single Queue Approach:

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	2	80	0	80	0	80
2	1	200	100	301	1	201
3	2	50	116	166	0	50
4	1	120	190	311	1	121
Average turnaround time: 99 ms						

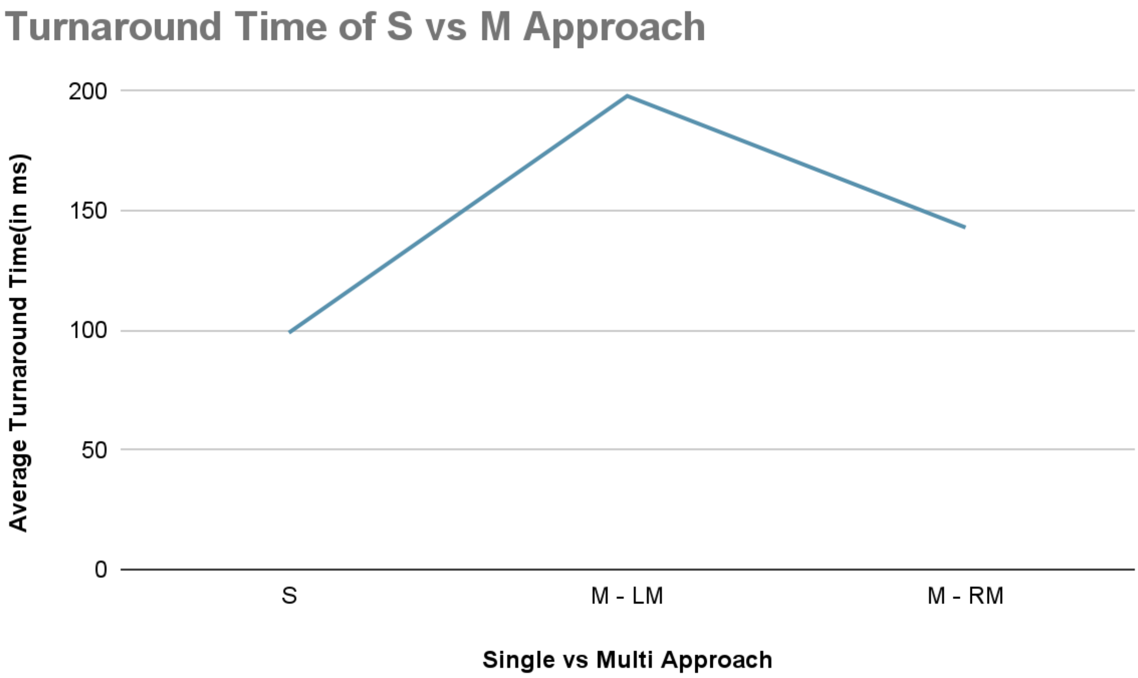
Multi-queue Approach with LM Method:

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	2	80	0	83	3	83
2	1	200	100	301	1	201
3	2	50	111	173	12	62
4	1	120	195	421	106	226
Average turnaround time: 143 ms						

Multi-queue Approach with RM Method:

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	2	80	0	83	0	80
2	2	200	100	300	0	200
3	2	50	123	351	178	228
4	2	120	193	477	164	284
Average turnaround time: 198 ms						

Here are the visual representations (plot) of the above result tables:



Since there are only two threads running concurrently and processing single and multi queues, for a small set of bursts it is hard to differentiate average turnaround times. Processes generally have better performance in the multi queue approach compared to single queue approach. In the sense of multi queue approach, LM method has also better performance since it always chooses the burst which has the least load. The RM method works as an RR algorithm and it may lead to unwanted results in some scenarios, which are explained in the coming section of this report.

Performance of the S vs M approach may also depend on the characteristics of the bursts set processed. Some of the cases IAT are small and most of the bursts in the process may have similar burst length, which may give chance to single approach in terms of better processing performance.

B) PERFORMANCE OF SCHEDULING ALGORITHMS (FCFS, SJF, RR) IN TERMS OF AVERAGE TURNAROUND TIME(*):

***Note that every integer value represented as time in this report is in ms.**

Performance of scheduling algorithms may depend on N (processor number) nd/or R(quantum number value, if the algorithm is RR) for all of the three algorithms. For now on, assuming all algorithms run on a fixed N value.Performance on different N values and/or R(quantum number value, if the algorithm is RR) will be checked in the following sections of this report.

Performance of these scheduling algorithms may also depend on the identifications of the bursts(ie. threads). That is, it may depend on the order of the processes taking account of their burst values (PL) and Interarrival times (IAT).

To mention that, for every algorithm:

First-Come-First-Served (FCFS): The average turnaround time for FCFS can be relatively high compared to other scheduling algorithms, especially when long processes arrive before shorter ones. In such cases, the short processes have to wait for the long process to finish, leading to a high average turnaround time (also known as the convoy effect).

Shortest Job First (SJF): SJF generally leads to lower turnaround times since every time it selects the job which has the shortest burst time.However, SJF is not always practical,as it requires knowledge of the burst times of all processes. Even though the values of PL and IAT are known in our scenario (by using sample in.txt file provided by the project instructions), this may not be available in every real-world scenario.

Round Robin (RR): The average turnaround time for RR depends on the chosen time quantum. If the time quantum is too large, RR may perform similarly to FCFS, leading to a high average turnaround time. If the time quantum is too small, the system may spend excessive time on context switching, reducing overall efficiency. When the time quantum is well-tuned, RR can provide a balance between fairness and efficiency.

Noting that 10<= Q <= 100 for the scenario given in the project instructions, higher quantum number (Q) values (ex. +70) leads to higher Turnaround time whereas lower Q values (ex. 20) will lead to even higher turnaround times due to overloading operations such as switching.

For plotting and making comparisons, N =4 and Q =20 for the RR algorithm) Q = 0 (For FCFS and SJF algorithms)

First-Come-First-Served (FCFS):

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	1	80	0	80	0	80
2	4	50	110	162	2	52
3	1	200	100	302	2	202
4	4	120	181	302	1	121
5	2	40	457	499	2	42
6	4	100	431	532	1	101
7	3	80	496	577	1	81
Average turnaround time: 97 ms						

Shortest Job First (SJF):

It’s worth to imply that the SJF algorithm evaluated in this report is non-preemptive.In a non-preemptive scheduling algorithm, once a process starts executing, it continues to run until it is completed. It cannot be interrupted or preempted by another process with a shorter burst time.In the case of non-preemptive SJF, the scheduler selects the process with the shortest burst time from the ready queue and allows it to run to completion. The scheduler makes the decision of which process to run only when the current process is finished or when a new process arrives, and the CPU is idle.

Here are the results for N = 4 and RR = 0 (Since it is NOT a RR algorithm.)

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	4	80	0	80	0	80
2	1	200	100	301	1	201
3	3	50	115	166	1	51
4	4	120	185	306	1	121
5	3	100	436	539	3	103
6	1	40	457	498	1	41
7	4	80	488	570	2	82

Average turnaround time: 100 ms

Since SJF works in a non-preemptive manner and the inter-arrival times between bursts are relatively long, in this scenario; SJF works similar to FCFS. If it is not the case, it would be obvious to see fluctuations between SJF and FCFS, and one would expect turnaround times in SJF would be lower compared to FCFS.

Round Robin (RR):

For N = 4 and Q = 20:

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	2	80	0	83	3	83
2	3	200	100	301	11	211
3	4	50	115	166	1	51
4	4	120	185	312	7	127
5	3	100	436	545	9	109
6	1	40	457	498	2	42
7	2	80	487	580	13	93
Average turnaround time: 102 ms						

Since N = 4, the differentiations between RR and SJF & FCFS are actually not clearly visible. For the explanations that CS 342 students used in class assumes that cpu number (N) is 1. Whereas in this situation N=4, the algorithm efficiency is not clearly seen.

To see differentiations in a clear way, take same algorithms and Q values with N =1.

First-Come-First-Served (FCFS):

pid	cpu	burstlen	arv	finish	waiting time	turnaround

1	1	80	0	97	0	17
2	1	200	100	311	11	211
3	1	50	111	362	201	251
4	1	120	213	514	181	301
5	1	100	463	615	52	152
6	1	40	514	655	101	141
7	1	80	544	739	115	195
Average turnaround time: 192 ms						

Shortest Job First (SJF):

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	1	80	0	85	5	85
2	1	200	100	302	2	202
3	1	50	113	360	197	247
4	1	120	186	484	178	298
5	1	100	451	707	156	256
6	1	40	477	525	8	48
7	1	80	508	606	18	98
Average turnaround time: 176 ms						

Round Robin (RR):

For N = 1 and Q = 20:

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	1	80	0	80	0	80
2	1	200	100	310	10	210
3	1	50	170	370	200	200
4	1	120	420	670	250	250

5	1	100	440	730	290	290
6	1	40	470	790	320	320
7	1	80	500	850	350	350
Average turnaround time: 243 ms						

Overall results for N = 1 suggest that SJF works better than FCFS since in this scenario just one cpu (thread) is working. Also note that average turnaround times for every three algorithms is higher in the case N = 1 compared to case N = 4.

The reason behind that is the threads may work concurrently, which leads to better performance in terms of turnaround time.

For the RR algorithm, R = 20 leads to higher turnaround times compared to FCFS and SJF because as mentioned above,this is because of time loss due to overloading and switching problems.

It’s worth checking the RR algorithm when Q = 80.

Round Robin (RR):

N =1 and Q = 80

pid	cpu	burstlen	arv	finish	waiting time	turnaround
1	1	80	0	80	0	80
2	1	200	100	280	0	180
3	1	50	170	330	60	160
4	1	120	420	450	0	30
5	1	100	440	570	10	130
6	1	40	470	610	0	140
7	1	80	500	690	0	190
Average turnaround time: 130 ms						

As obvious, when Q is higher, one could reach better turnaround times since as mentioned above, risk of overloading is reduced.

Also note that Q = 80 is not too much for this particular burst set. It is well tuned since the maximum length of this burst set is 200ms. If Q is close to 200, it would act as an FCFS algorithm.

Here are the visual representations (plot) of the above result tables: