CS 342 Operating Systems

Spring 2023

Project 2: Multiprocessor Scheduling, Threads, Synchronization

Group Members

Esad İsmail Tök – 21801679

Elifnur Alsaç - 21601098

Introduction:

- In this repot we examine the relations among queue approaches and scheduling algorithms.
- The queue approaches that are under the investigation are: Single Queue Approach, Multi Queue Approach with Raund Robin Method, and Multi Queue Approach with Load Balancing Method.
- The scheduling algorithms that are under investigation are: First Come First Served (FCFS), Shortest Job First (SJF) without preemption, and Round Robin. In order to investigate the effect of the length of the time quantum in the Round Robin, we will take 10ms, 20ms, and 40ms as time quantum and investigate separately for Round Robin algorithm.
- The same randomly distributed burst lengths and interrarival times are used for the each comparison and there are 7 burst that are scheduled in each test.
- The comparisons will be according to the average turnaround times as stated in the project manual.

- Three cases are examined in this report:
  - Comparisons among 5 different scheduling algorithms namely; FCFS, SJF, RR (Q=10), RR (Q=20), RR (Q=40) according to their average turnaround time in a Single Queue Approach with 1, 2, 4, and 6 as the CPU counts.
  - Comparisons among 5 different scheduling algorithms namely; FCFS, SJF, RR (Q=10), RR (Q=20), RR (Q=40) according to their average turnaround time in a Multi Queue Approach (RM) with 1, 2, 4, and 6 as the CPU counts.
  - Comparisons among 5 different scheduling algorithms namely; FCFS, SJF, RR (Q=10), RR (Q=20), RR (Q=40) according to their average turnaround time in a Multi Queue Approach (LM) with 1, 2, 4, and 6 as the CPU counts.

- The resulting investigations of the above cases are plotted into three separate graphs and evaluated in the relevant sections throughout the report.

Case 1: Comparisons among 5 Different Scheduling Algorithms in a Single Queue Approach with Different CPU Counts

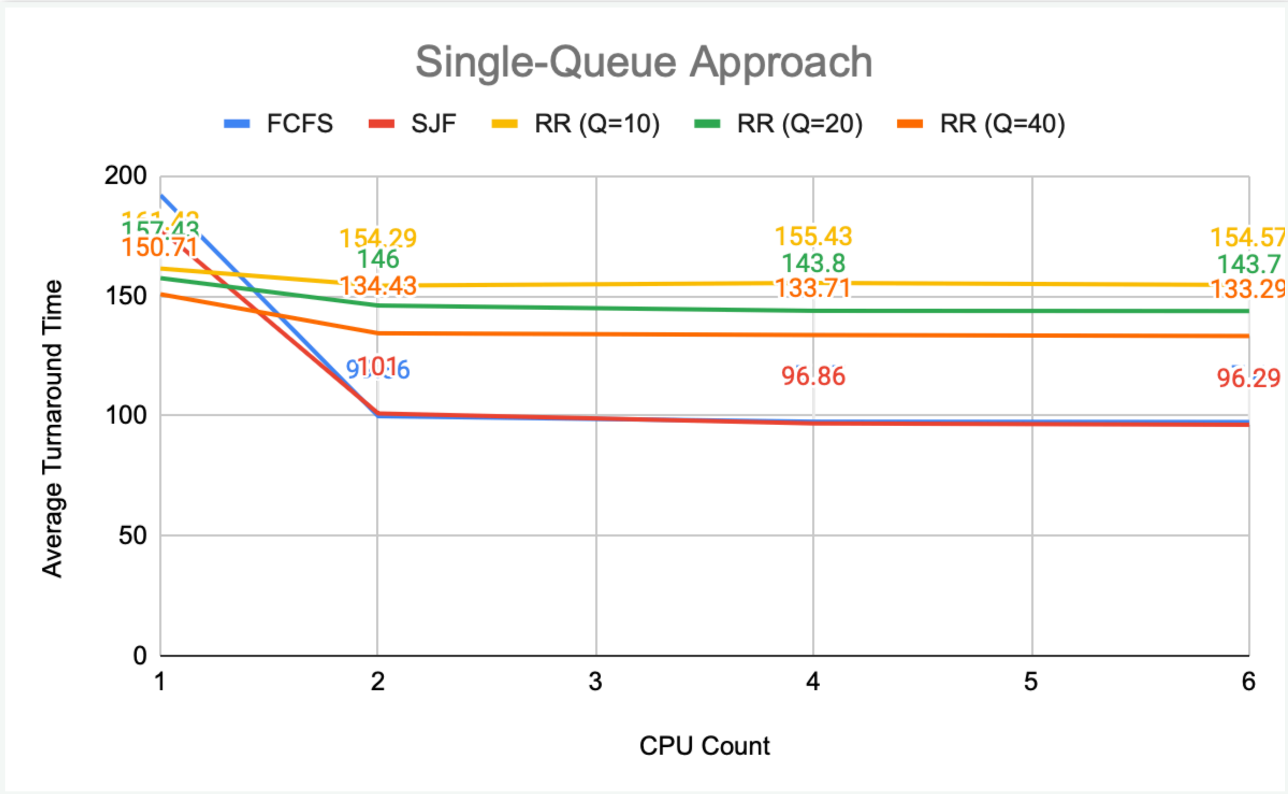

Figure 1: Comparison among scheduling algorithms with single queue approach

- *Figure 1* is the plot of the comparison among 5 different scheduling algorithms in terms of average turnaround times for different CPU counts in a single queue approach.
- It can be seen that, when there is only a single CPU for FCFS and SJF approaches with a single common queue the average turnaround time is pretty high which is not desirable. The turnaround times are high for those algorithms because even if the new bursts arrive to the single common queue, there is only 1 CPU that can execute the bursts so the newly arriving bursts need to wait until that CPU is released. So even if the first burst would have a very low waiting and turnaround time, the following bursts will start having more and more turnaround times and the average would be high in FCFS and SJF algorithms.

- After we increase the CPU count to 2, we can see a good amount of performance increase in average turnaround time in FCFS and SJF since now while a CPU is busy, the other one can start executing the waiting bursts in the queue. If we further increase the CPU count we see that we do not have much improvement after CPU count 2 because 2 CPU's would be enough to finish the execution of the burst before the new ones reach to the ready queue because new bursts are coming after an inter-arrival time.
- In terms of Round Robin algorithm, we see that when there is a single CPU, the average turnaround time is less than FCFS and SJF algorithms. It is because there is a preemption by a time quantum and no burst can use the CPU as they want. When a burst exceeds its time quantum, it is preempted and a new one arrives. If we increase the CPU count from 1 to 2, we do not see a rapid performance boost in terms of average turnaround time as we did in FCFS and SJF because the performance of RR was already better compared to the other algorithms when CPU count is 1. increasing the CPU count from 2 to 6 gradually does not show a remarkable increase again because of the same reasons that were applicable for FCFS and SJF algorithms.
- Also the overall performance of RR algorithm with higher time quantum (Q) is better in terms of average turnaround time because smaller time quantum causes higher overheads in our cases because of the small burst lengths. These overheads that are caused by small time quantum increases the average turnaround time so RR (Q=40) is better then RR (Q=20) which is better than RR (Q=10) in general in our cases.

Case 2: Comparisons among 5 Different Scheduling Algorithms in a Multi Queue Approach (RM) with Different CPU Counts

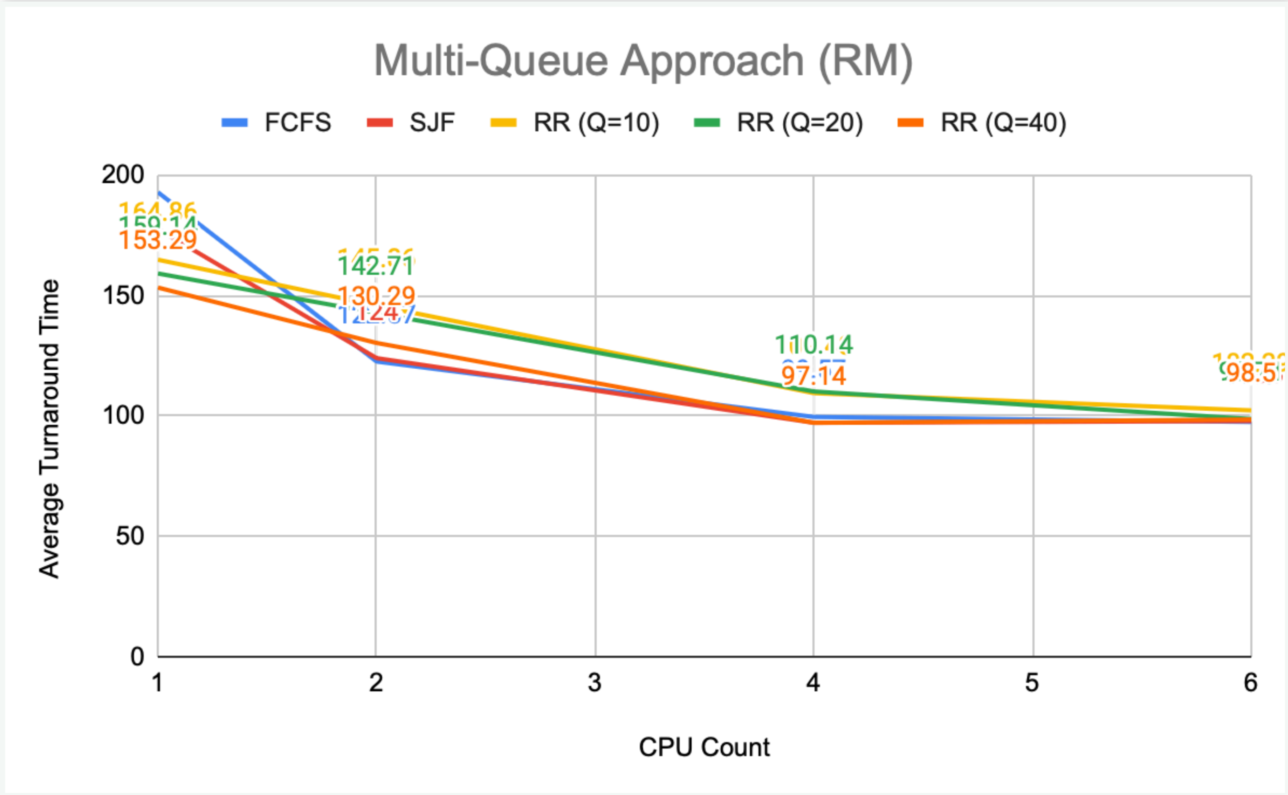

Figure 2: Comparison among scheduling algorithms with multi queue approach (RM)

- *Figure 2* is the plot of the comparison among 5 different scheduling algorithms in terms of average turnaround times for different CPU counts in a multi queue approach (RM).
- It can be seen that, when there is only a single CPU for FCFS and SJF approaches the result are similar to the ones in the single queue approach for a case when there is only a single CPU. The turnaround times are high for those algorithms in that case too because even if the new bursts arrive to the queue of the CPU, that CPU can execute only that bursts so the newly arriving bursts need to wait until that CPU is released.

- After we increase the CPU count to 2, we can see a good amount of performance increase in average turnaround time for FCFS and SJF since now all the new arriving bursts are evenly distributed to the 2 CPUs. If we further increase the CPU count this time (unlike the single common queue approach), we see that the performance in terms of average turnaround time keeps increasing. The thing that guarantees the constant increase is that all the bursts are evenly distributed among the CPUs because of the Round Robin method that is used in multi queue approach.
- In terms of Round Robin algorithm, we see that when there is a single CPU, the average turnaround time is less than FCFS and SJF algorithms again as it was in single queue approach. It is because there is a preemption by a time quantum and no burst can use the CPU as they want. When a burst exceeds its time quantum, it is preempted and a new one gets the CPU. If we increase the CPU count from 1 to 2, we again do not see a rapid performance boost in terms of average turnaround time as we did in FCFS and SJF because the performance of RR was already better compared to the other algorithms when CPU count is 1. Further increasing the CPU count from 2 to 6 provides us a further performance increase which was not the case in the single queue approach. Using their time quantum in separate queues for separate CPUs, the bursts are not making the queue busy as they did in the single queue approach and that makes the average turnaround time smaller together with the uniform distribution of the burst to the queues bu round robin method of multi queue approach.
- Overall performance of RR algorithm with higher time quantum (Q) is again better than the smaller time quantum because smaller time quantum causes higher overheads which causes the average turnaround time to be higher that is not desirable for us.
- At the 6 CPU, we got similar average turnaround times for each of the 5 scheduling algorithms this time with multi queue approach with round robin method. that was not the case with single queue approach.

Case 3: Comparisons among 5 Different Scheduling Algorithms in a Multi Queue Approach (LM) with Different CPU Counts

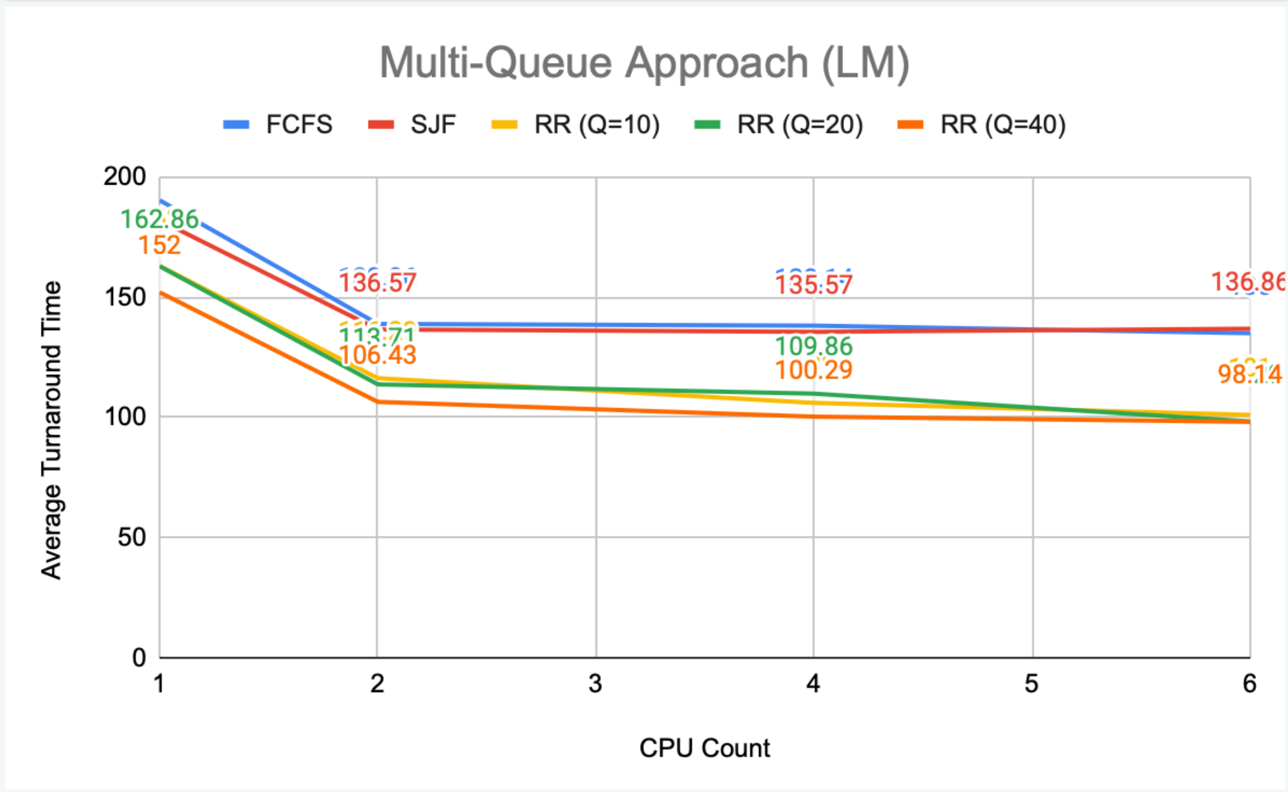

Figure 3: Comparison among scheduling algorithms with multi queue approach (LM)

- *Figure 3* is the plot of the comparison among 5 different scheduling algorithms in terms of average turnaround times for different CPU counts in a multi queue approach (LM).
- It can be seen that, when there is only a single CPU for FCFS and SJF approaches the result are similar to the ones in the single queue approach for a case when there is only a single CPU. The turnaround times are high for those algorithms in that case too because even if the new bursts arrive to the queue of the CPU, that CPU can execute only that bursts so the newly arriving bursts need to wait until the CPU is released.

- After we increase the CPU count to 2, we can see a good amount of performance increase in average turnaround time for FCFS and SJF. However we do not see the increases as much as the one with multi queue approach (RM). In RM case all the bursts were perfectly distributed to the CPU's but in this case load balancing might not guarantee the perfect distribution. If the burst lengths would vary greatly, then load balancing option would be preferable since balancing according to the load would make more sense in that case. But in this case, the burst lengths of the bursts are so similar and the load does not differ much. If we further increase the CPU count this time (unlike the multi queue approach with RM), we do not see any considerable further performance increase. It is because , just like in single common queue approach, the bursts are not much longer than the inter arrival times of the bursts so that while a burst is executed in first 2 queues, also the inter-arrival time is spent and while the new burst is come after the inter-arrival time, the executed burst is done so there is usually no need to extra CPUs after we increase the count from 2 to 6. therefore we do not see much increase in the average turnaround time.
- In terms of Round Robin algorithm, we see that when there is a single CPU, the average turnaround time is less than FCFS and SJF algorithms again as it was in single queue approach. It is because there is a preemption by a time quantum and no burst can use the CPU as they want. When a burst exceeds its time quantum, it is preempted and a new one gets the CPU. If we increase the CPU count from 1 to 2, this time we see a better improvement against FCFS and SJF algorithms which was not the case in single queue and multi queue with RM approaches. Further increasing the CPU count from 2 to 6 in RR algorithm does not provide us a considerable performance boost this time but the performance is still better than the FCFS and SJF algorithms in the increased CPU counts. It is because since the different bursts are placed in different queues and no burst can use the CPU as much as they can because of the time quantum, the average turnaround time is less than FCFS and SJF algorithms.
- Overall performance of RR algorithm with higher time quantum (Q) is again better than the smaller time quantum because smaller time quantum causes higher overheads which causes the average turnaround time to be higher that is not desirable for us.
- at the CPU count 6, this time RR scheduling algorithm and FCFS, SJF algorithms diverges in a way that RR algorithm (with all Q values) dominates the other two algorithms in the maximum amount of CPU count.