CS342 Operating Systems

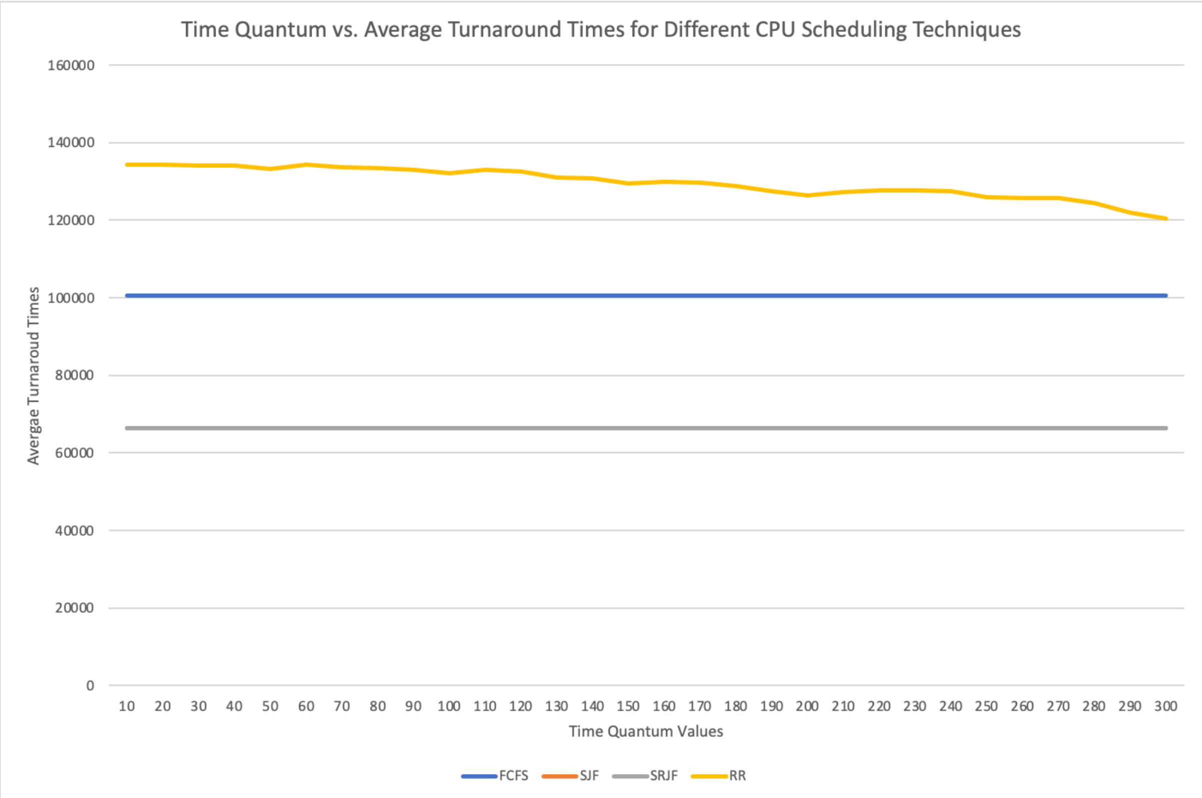
Project 2: CPU Scheduling

Öykü Irmak Hatipoğlu 21802791
Cansu Moran 21803665

In our experiments, we used an input file which included 1000 bursts. In the file, each input arrives in 5 second intervals and the length of the bursts are randomly generated between 5 ms and 400 ms.

We experimented with different time quantum values to see how it affects the average turnaround time of RR scheduling mechanism. Since the other scheduling mechanisms (FCFS, SJF, SRJF) only depended on the input file and since we performed our experiments on a single input file, in every experiment, their average turnaround times remained the same.

We experimented with 30 different time quantum values. These values start from 10 ms (the minimum value of time quantum) and incremented by 10 each time until 300 ms (maximum value of time quantum). The average turnaround times of different CPU scheduling techniques with varying time quantum values can be seen in Graph 1.



Graph 1: The average turnaround times of different CPU scheduling techniques with different time quantum values

Since it is hard to distinguish individual average turnaround times from the graph we also listed some exact times in milliseconds below:
FCFS: 100479 ms
SJF: 66292 ms
SRJF: 66284 ms
RR: 134327 ms (time quantum = 10 ms)
RR: 120412 ms (time quantum = 300 ms)

With increasing time quantum values, the average turnaround time of RR became closer to FCFS, which was expected. The largest time quantum we had was 300 ms, but if we were to choose a time quantum that was the equal to or larger than the maximum burst length, we would essentially obtain the same average turnaround times in RR as FCFS. In our experiments, a smaller time quantum resulted in higher average turnaround time in RR. In addition to higher turnaround time, in a real life scenario, smaller time quantum in RR could also cause overhead due to context switching. On the other hand, while a small time quantum value might not achieve optimal results in RR, it would make sure every burst gets equal timing and hence ensure a good CPU sharing.

We expected the best time in SRJF or SJF since they are designed to give optimal turnaround times and our expectations were satisfied with the achieved results from the experiments. Although the average turnaround times achieved from these algorithms were the smallest among the CPU scheduling algorithms we experimented with, these algorithms can cause starvation, which is a problem. Every time during scheduling the shortest burst/the burst with shortest remaining time is chosen in these algorithms, which means that a longer burst might not get the CPU for a long time since it will wait for all the shorter bursts to end. Although we obtained very similar results in SRJF and SJF, SRJF performed slightly better. However, in a real life CPU scheduling scenario, choosing SJF might be a more optimal solution due to overhead, which was not reflected in our experiments. In SRJF, we had to check whether a burst with a shorter remaining time was present every second. This comparison, in addition to constant context switching, can cause too much overhead compared to SJF which checks for shortest burst only when the previous one is complete. However, overall, SRJF achieves better turnaround times.

It should be noted that the average turnaround time of CPU scheduling algorithms depends on the input bursts. For example, if the same number of bursts with the same lengths are given in a different sequence, the results might change. However, overall, the observations above are aimed to be inclusive of all input types. Decreasing the turnaround time is usually not the only factor we need to think about while CPU scheduling. Other factors such as context switch overhead and starvation should also be considered in a real life scenario. In addition, we might favor one factor over the other. For example, for interactive programs, choosing RR might be the most optimal solution, since we wouldn’t want such a program to wait for too long and have a long response time, although we have observed that RR had higher turnaround times.