Michael Trinh

CS 4310

Professor Atanasio

Assignment 1: Scheduler Simulator

Introduction

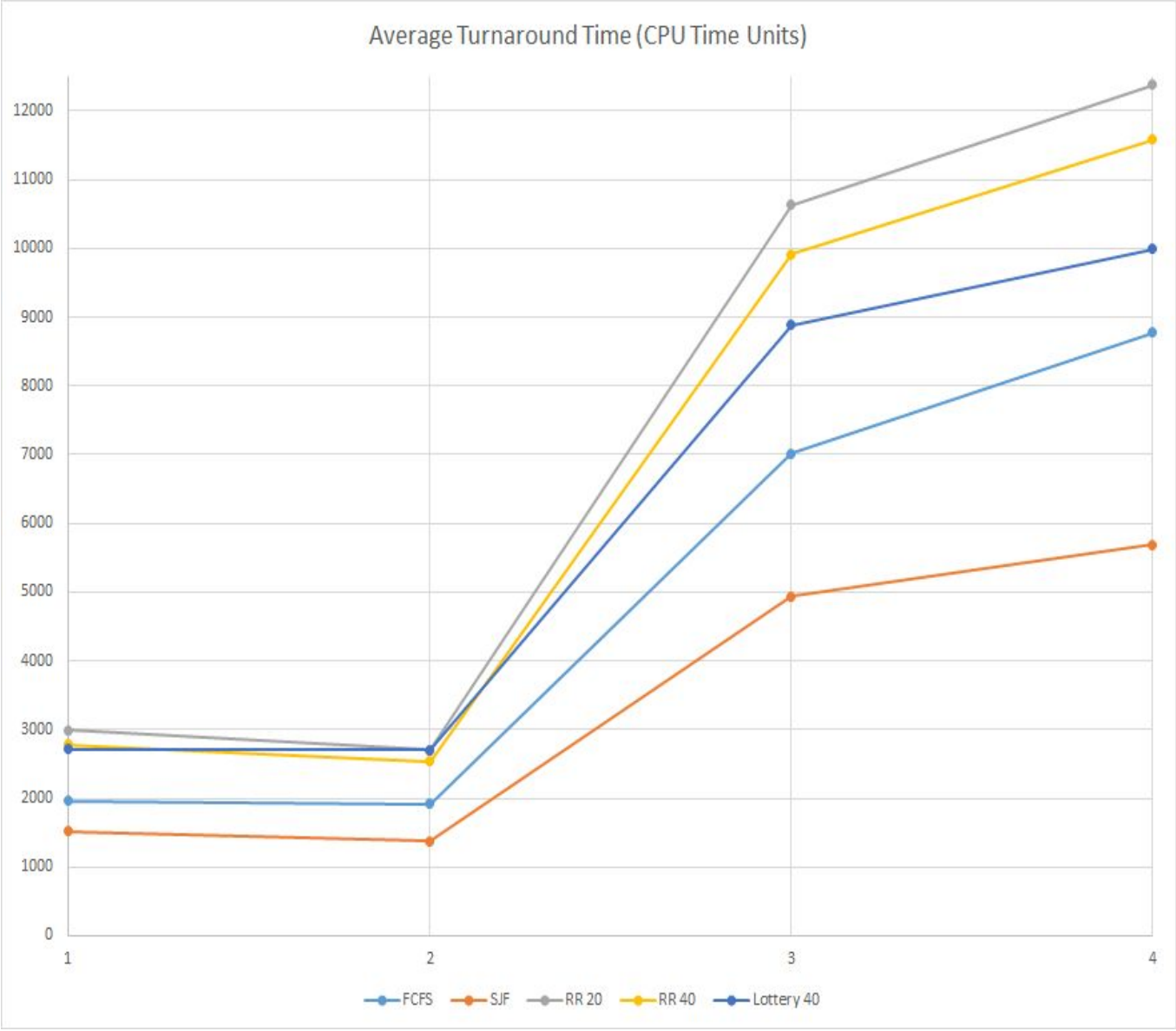
In Assignment 1, I created a program to simulate a process scheduler in an operating system. The program gets its data from user input of a text file, where each line is the process ID, the process's burst time, then the process's priority for each process. During the program's runtime, each scheduling algorithm is performed. The program will print the formatted input data and each scheduler as well as the total turnaround time, calculated by the sum of the completion times for the scheduler, and the average turnaround time, calculated by the total turnaround time divided by the number of completion times. The program will also ask if the user wants to save the scheduler to a file output. When a process switches to another process, a switch cost of 3 CPU time units is applied to the CPU.

There are 5 scheduling algorithms used for this particular assignment: 1) First Come First Served (FCFS), which runs processes from first to last occurred, 2) Shortest Job First (SJF), which runs processes from shortest to longest burst time, 3) 4) Round Robin (RR), which runs the processes from first to last occurred for a certain time quantum then adds the process to the end of the queue of processes, and 5) Lottery, which allocates a certain amount of "lottery tickets" based on the process's priority and selects a random ticket from the total amount of tickets given to be chosen to be added to the process queue.

For the report, the program was tested on four testdata.txt input files. Testdata1 & 2 files both have 9 processes, testdata 3 has 19 processes, and testdata 4 has 27 processes. The more processes the scheduler will have to take, the longer the total and average turnaround time it will take to complete the processes. The amount of burst time a process has will also influence the

turnaround time, where turnaround time is longer as burst time is longer. Average turnaround time is important in a scheduler because it is the average completion time to complete a process.

Analysis



Average Turnaround Times of schedulers tested on each of the given testdata.txt file inputs (1-4).

FCFS and SJF are the two lowest average turnaround times out of the scheduling algorithms for each of the testdata files. While both SJF and FCFS algorithms are implemented in batch systems, average turnaround time for SJF will always be less than or equal to the average turnaround time for FCFS. This is because both algorithms run processes until they are finished and that SJF will run the shortest lasting processes first, making the completion and CPU times the lowest possible, influencing average turnaround time. SJF is equal to FCFS when the processes in FCFS are ordered by shortest burst time.

RR20 and RR40 are the two highest average turnaround times. RR40 has a lower turnaround time than RR20 because RR40 switches process less than RR20 reducing the amount of times the switch cost is applied for processes with burst times over 20. Surprisingly, average turnaround time for RR40 isn't half as much as RR20. This could be due to a handful of processes in the testdata having burst times of less than or equal to 20 CPU time units. Average turnaround times for RR are longer than the batch scheduling algorithms because RR has to switch processes more often and that some processes are not completed after a time quantum, delaying the completion time for that process.

RR40 and Lottery40 are close to each other when the amount of processes are low, but as the amount of processes increase, the average turnaround time for Lottery40 is less than RR40. Lottery40's average completion time is lower than RR40's for high amounts of processes because processes in Lottery40 finish earlier than RR40. This is due to high priority processes being chosen more often than lower priority processes from the randomization method. This will make the Lottery40 be more like a batch system scheduling algorithm due to the high probability of a process being completed within a certain number of process switches, compared to RR which

will run each incomplete process once per circular iteration, where higher amount of processes can extend a process's completion time. The reason why average completion time for RR40 and Lottery40 are close when the number of processes are low is because for RR, the ratio of average completion time increases as the number of processes increases at a much higher rate than Lottery, in which the ratio of average completion time also increases as the number of processes increases. For testdata 2, Lottery has a higher average completion time than RR possibly due to randomization choosing low priority processes in between high priority processes, adding unnecessary CPU time if the process has a higher burst time than time quantum.

Conclusion

There are many factors that go into the calculation of completion time, influencing average completion time. One of the biggest factors is how frequent the scheduler switches processes due to the switch cost being added to CPU time each time. This means having no time quantum is ideal and having a high time quantum will yield a lower average turnaround time than a low time quantum. Another large factor is how early the scheduler completes processes, in which completion time is calculated only when a process is completed. This means that the scheduling algorithm that yields the highest average completion time is one with time quantum = 1 and has all of its processes run until they all have a burst time of 1, then run again. Because of these two factors, SJF is the fastest scheduling algorithm with FCFS, Lottery40, RR40, and RR20 being the next fastest in that order.