

Shiva Modugu
Professor Ozbirn
CS4348

Project Summary

Project

The project involves simulating a job scheduler that manages a set of jobs. These jobs are represented by various attributes, including their names, start times, and durations. Users can apply different scheduling algorithms to these input jobs. Two specific algorithms are considered, First-Come, First Served: FCFS, and Round Robin: RR. The FCFS algorithm processes jobs in the order they arrive. Jobs are executed sequentially, without preemption. The first job to arrive is the first to be executed. Once a job starts, it runs until completion. FCFS ensures fairness but may not be optimal in terms of turnaround time. Unlike the FCFS the RR is a little different. In this algorithm, each job is given a fixed time slice for execution. Jobs are processed in a circular queue. If a job doesn't complete within its quantum, it's moved to the end of the queue. RR provides better responsiveness but may lead to higher overhead due to context switching. Overall the goal was to understand how different scheduling algorithms impact job execution and visualize their effects.

Difficulties

I personally had a lot of trouble with the RR algorithm. I got all the jobs to print but I had trouble with the timing and making them print with the correct amount of spaces in between. I think it was due to my print method. I wanted to print my output sideways and couldn't wrap my head around how to do it for the RR algorithm.

Personal Experience

I honestly liked this project. This project for me honestly was harder than project 2 due to the RR algorithm, but I still enjoyed it. I hopefully will get a good enough grade to justify the amount of work I put in for the RR. The FCFS was simple enough due to the simplicity of the algorithm so

it made coding it enjoyable. Overall I had a good time with this project even though I couldn't figure out the RR algorithm. When compiling my program I would use `javac *.java` but, you are able to run it how it is asked in the project description.