Kevin Chen

nkc160130

Due: 30 November 2019

SE 4348

Prof. Ozbirn

**Project 3**

Out of all the projects that I have done for this class so far, this has to be one of the most exciting and fun one for me. The premise of this project is as such; it simulates a scheduler scheduling a set of jobs, and outputs a graph showing the jobs and their time of running. It allows the user to choose among six algorithms from the textbook. RR and FB uses a quantum of one, and FB uses three queues. According to the professor, either a text-based or a JavaFX-based graph would be acceptable for this specific project. As I wanted to challenge myself, I opted for the JavaFX route. I had used Java Swing before, but not JavaFX. Luckily, JavaFX is not much different from Java Swing. Unfortunately, JavaFX elements can be very cumbersome to deal with, especially since none of the graphs that JavaFX could render fit for the scheduling scenario I had. So, I opted to render everything in HTML, CSS and JavaScript, utilizing Java mainly for the backend and passing a 2-D array to JavaScript code for the HTML page to render. JQuery was utilized to make scripting in JavaScript easier, and the Bulma framework was used to make the UI look pretty. Though it was tedious, it all worked out in the end. For the algorithms, many of the operations done within them were very similar, and I grouped all of them into one superclass, called SchedulingAlg, that had a 2-D array of lists of job tuples, as well as job details. One interesting design decision that I think I did was using multithreading for FB, as I found it easier to have a separate thread inserting new nodes into the top queue. The jobs are designed to run to completion or until another job is waiting, depending on the algorithm. My biggest difficulty in this project was devising a design for the FB algorithm, as well as debugging and tweaking the front-end user interface. The most important thing I learned was a deeper understanding of all the scheduling algorithms. Before, I had only gone over FCFS, FB and RR, but with this project I have filled in the gaps with the other algorithms. I also learned how to get Java, with JavaFX, to communicate with the back-end, which I thought would be a major hurdle, and using Apache Ant to automate building my project. Overall, this project has strengthened my knowledge of web languages like HTML, CSS and JavaScript, has deepened my knowledge of the scheduling algorithms, and facilitating back-end to front-end communication.