Project Summary

The goal of this project is for the students to understand and demonstrate the ability to construct algorithm that will schedule job in an approach with operating concepts. The scheduler algorithm has six total and they are: first in first out, shortest process next, shortest remaining time, feedback, round robin and highest response ratio next. The entire algorithm have the pro and cons and through this project, the students should be able to understand the strength and weaknesses in the entire algorithm.

The approach to the project following a concept that scheduler should only held the responsibility of select job and call the job’s run method to execute itself. Because there are no processors implemented in the project, the job should execute itself and generate the result. The scheduler main responsibility is to select jobs, and switch between jobs according to specific concepts in certain algorithms. My approach to complete the project is done with using different classes and separate only unique characteristics to each class of the algorithm. I build an abstract class called scheduler that has all of the basic data structure and variable for all of the other specific scheduler class to inherit and use. Then I create each of the scheduler algorithms like FIFO, Highest Response Rate Next, FeedBack, etc… with its own method uniquely for that class. This approach save codes and that it use existing code from the parent class to the child class without having to rewrite all of the same code in another class.

During the process of working on the project, the most challenging algorithm which I have to implement is the Round Robin algorithm. The Round Robin algorithm alternate between jobs using a time slice and the arriving job go in the queue before the current job. In concept, this is not much challenging however, when working in code, it is really difficult to put the current running job back in the queue after a new arriving job because it require a lot of case testing and condition checking in order to do this. After the Round Robin Algorithm, the second most difficult algorithm that I had problem with is the Shortest Remaining Time algorithm. This algorithm switch project when a new one arrives. The difficult thing about this algorithm is that if the new job arrive have less execution time than the current job, the current job will get to switch out and the new job get to run. However, again, in code, this kind of procedure requires lots of cases and condition checking in order not to run the wrong project. Also the program has to keep track of the future job on when it arriving and how much time requires running the current process. Beside these two algorithms that I talked about above, the other four are fairly straight forward and simple. Generating the output is another challenging task, however I make use of the hash map and store rows that corresponding to the job’s name so that make the output much simpler.

In summary, I thought that this project is very interesting and I learn how these algorithms really work in code. It’s always good to know how things work inside the computer. This project in the beginning requires some planning and some object oriented design concept. From what I’ve experience, writing clean code and apply object oriented design concept well specifically for this project will help reduce the complexity of the overall project. Code will also can be reuse and share among other class if the module or functions in the class have a good design concepts. The professor did say that the feedback algorithm will be the most challenging one to implements, however surprisingly, this algorithm took the least amount of time compare to all the other to implement in my case.