Nicholas Belvin

**Scheduler Implementation**

My followed every requirement given in project 5 in the following ways. Firstly, the scheduler separated the scheduling mechanism from the scheduling policies by creating separate functions for the scheduling policies and simply calling on these policies for my unique use case. Next, my three scheduling algorithms were implemented in standalone functions following the techniques given to us in class for first come first serve, Round Robin, and Shortest time first. The wait times were calculated by taking the turnaround time – Burst time, the response time was taken by start time – arrival time, and turnaround time was calculated by Exit time – Arrival time. The command line parser was implemented by taking in argv’s from the main function, the argv was treated as an array of either size 2 or 3 depending on if the algorithm running was round robin or not. My solution is general as other than the simulator the functions implemented could be used in other projects. It is so general in fact a new scheduling policy could be added by simply implementing the new scheduling function and adding it’s call to main. My program offers the basic input error checking by making sure something was inputted and it was of the right type. My code was also organized in such a way that it is easy and orderly to read. While my solution wasn’t very innovative it was very straight forward. Finally, all outside sources were documented as I relied on my notes to develop my solution.

**Example outputs**



