**Analysis**

1. **Of the four simulated algorithms, which algorithm is the “best” algorithm for CPU-bound processes? Which algorithm is best-suited for I/O-bound processes?**

FCFS SJF SRT RR

Simulation 2: 9788 9788 9788 9788

Simulation 3: 16274 16274 16274 16330

Simulation 4: 78352 76258 77150 77579

Simulation 5: 78352 76258 77150 77579

1. **For the SJF and SRT algorithms, what value of α produced the “best” results?**

For SJF and SRT the value of α that produces the best results would be when α is approximately 0.5.

1. **For the SJF and SRT algorithms, how does changing from a non-preemptive**

**algorithm to a preemptive algorithm impact your results?**

Changing from a non-preemptive algorithm to a preemptive algorithm can significantly decrease the average waiting time among all processes. So, processes with a higher priority of running CPU bursts can always reduce the total waiting time for all the processes. This only has an impact on the results when there are multiple processes. If there are only one process in the system it doesn’t make any difference because there will be no preemptions.

1. **Describe at least three limitations of your simulation, in particular how the project specifications could be expanded to better model a real-world operating system.**

One limitation of our simulation is that we don’t kill the processes or close any processes. It would better model a real-world operating system if processes got cancelled. Secondly, we don’t consider system usage that will block the CPU. Finally, our simulation does not consider process priority

1. Describe a priority scheduling algorithm of your own design (i.e., how could you calculate priority?). What are its advantages and disadvantages?