

### **Analysis Summary**

In writing this program, I found that the logic for the priority, non-preemptive scheduler and the SJF, non-preemptive scheduler are very similar. Simply put, my solutions for both are nearly identical with the exception that instead of using priority in SJF, the total burst time of each process is used instead. However, despite the similarities in the logic of both algorithms, the results based on my data set are very different.

As expected, the processes are run in an entirely different order depending on the algorithm. The priority algorithm schedules processes that arrive in time by choosing the highest priority. Whereas the SJF algorithm schedules processes that arrive in time by choosing the shortest process. This is accomplished in my solution, by tracking the total running time of processes scheduled so far, and comparing that number with the arrival time of remaining processes. Then the next process is scheduled based on priority or job length of the processes that have arrived within the tracked total running time, and have not already been scheduled.

In terms of running statistics, there are some similarities and some differences. As expected, the total running time and the throughput of both algorithms is the same based on my data set, and should remain the same regardless of the data set (Assuming throughput is calculated using the total running time). However, based on my data set, the average wait time and the average turnaround time are much lower using SJF scheduling. This makes sense, as with SJF shorter processes are going to be scheduled and finished as soon as possible, promoting shorter wait times for other processes in general. Whereas, with priority scheduling it's reasonable to expect the possibility that some longer, high priority processes may be scheduled in front of several shorter, low priority processes; which causes those shorter processes to wait through the entire running time of the longer process.

I conclude then that priority scheduling is useful, when it is truly important that some process be finished sooner because they are more critical than others. However, if priority of processes is not a worry, then it would be better to use SJF, as it seems to be more efficient in simple terms of wait time and turnaround time. This is assuming we have a way to know how long processes are before hand.