#1: Compare the test files and their output. What can be said about the RR scheduling algorithm? What can you say about the effect of different quantum sizes?

The round robin scheduling algorithm allows all “ready” processes to advance toward completion. This algorithm works very well for this project because all processes have the same priority and none of them are forced to wait on each other, only to wait on independent events.

The different quantum sizes had a huge effect on the running of the simulation. Changing quantum sizes allowed processes to have different parents and different orders in both the ready and wait queues. The effect of changing the quantum sizes was more drastic than we had initially anticipated.

#2: What aspect of process management did you find most difficult to implement?

By far the most difficult part of this project to implement was the Delete operation. Using a vector STL allowed us to do this more simply as vectors allow you to delete from the middle of the queue. If we had used the queue STL then we would have had to create a temporary queue and cycle the data around to find the one to delete. Thus a queue STL is much more processor intensive and inefficient. Deleting the targeted process was simple, but deleting its entire hierarchy of children was difficult. For this we chose to implement a recursive delete function, something that both of us were not confident that we could do. After many “core dumps” and compilation errors, we were finally able to make the delete function work, only to find out that we had to rewrite the same function to check the wait queue in addition to the ready queue.

#3: What aspect of process management did you find easiest to implement?

The easiest part of the project to implement was the Interrupt operation, only because there was not much going on in this operation. This was the only operation that was actually implemented correctly on the first try! If we were making an actual process management function we imagine this would be a little more difficult seeing as how the interrupt function would call the ISR at this point and cache the processes in both ReadyQ and WaitQ.

#4: What, if anything, would you change in your current design?

Frankly, we don’t think that we would change too much. This is mainly due to the fact that we implemented about 50% of the project and then were forced to start over in order to complete all of the operations. Although this was a very frustrating
circumstance, it allowed us to reassess our approach to the project and to correct most of our design errors. In summary, we are satisfied with our current implementation and probably would not significantly change it if we had to start over for a second time.

#5: What, if anything, did you find interesting or surprising about the process management that you did not know before doing this project?

What we found most surprising about the round robin scheduling algorithm as a result of this project is the effect of the different quantum sizes. We were repeatedly amazed at the drastic changes in our output caused by increasing or decreasing the quantum size by only one.