In addition to the code that was in my original xv6 programs and code, I added the “schedTest.c” file. I also added the system calls forkBQ and debug which are written in the “proc.c” but are referenced elsewhere as well. I changed the scheduler to run a Binary-Queue rather than Round-Robin scheduling system instead. Lastly, I changed the number of CPUs used in the Makefile from 3 to 1.

When running “schedTest.c” with the Round-Robin scheduler, all of the child processes finish around the same time at the end of the program. They all run for the same amount of time then switching to the next child process, regardless of the runLength that they were given.

When running “schedTest.c” with the Binary-Queue scheduler, the child processes end at different times and are more spread out through the debug output for the program. The child processes that were created with higher runLength finished earlier than the child processes that had lower runLength or 0 runLength.

The Round-Robin scheduler consists of mainly one for-loop that goes through all of the processes and gives them each an equal amount of time before moving to the next process. The Binary-Queue scheduler does keep that for-loop (with the only addition making sure that the processes in that for-loop are low priority) but has the addition of a second for-loop before it that runs the high priority processes until the runLength of each high priority process is 0.