Clay Blankenship

CPSC 3220 Scheduling Project Write-Up

April 9, 2018

Before starting the project, I had a pretty good understanding of the scheduling concepts that were going to be implemented such as FIFO, Shortest Job First and Round Robin. The two that I understood completely were FIFO and Round Robin because I knew that the tasks for FIFO just ran one after the other in the order that they arrived, and for Round Robin, the next task in the queue would start after the given time slice had expired and so on until all of the tasks in the queue were complete.

For Shortest Job First, I had trouble understanding when certain jobs would start and when they would not start. For example, if the first job was 30 units in length and the second job was 15 units and arrived after 15 units had passed, what would be the completion time? I had trouble understanding if the second job would start after the second time slice (two would pass because the second job arrives after the first time slice finishes) or if it would start once it arrives. I later figures out that it would be the former before I started implementing the algorithm. One other thing that I did not quite understand when it came to Shortest Job First was if a time slice would be cut off or stopped when the running task completed. For example, if a task had 7 more units to run for and the time slice was 10 units long, would the time slice in that instance be 7 units, would it not change and nothing be done for 3 units since the current job had completed or would the next job run immediately after the previous job was finished? After some research and looking through the textbook, I understood that the next job would immediately run after the 7 units in the particular situation.

After I understood how each of the scheduling algorithms worked, I started implementing the concepts. I wrote the program in C++ and used Microsoft Visual Studio in order to debug the code more easily, but I ran into some issues when running the program on Linux as opposed to the terminal in Visual Studio. The issue pertained to a memcpy() that I used in order to move all of the jobs down the queue after the first job had been loaded onto the CPU. The third parameter in memcpy() requires the size of the type being passed in, and since I was trying to move multiple processes at once (i.e. memcpy(work_queue[0], work_queue[processes_left], sizeof(process) * processes_left)), Linux complained, so I proceeded to do this operation by using a for loop.