Patrick Hamod

Dr. Rose

CSCE 311-001

October 16, 2014

Shortest Job First

In this assignment I changed my program 2 from round robin scheduling to shortest job remaining. At first I thought I would change my ready queue to a sorted list based on the shortest estimated burst time. However, I decided against this because I wanted to get the program done and would reconsider based on the time of the execution of the simulator. In do\_dispatch method I compared the ruining thread, if the MMU had a page table, to see if there was a thread with a shorter burst time. If the ready queue had no shorter threads or the current running thread had only 2 milliseconds left to run the most efficient thing to do was to let that thread finish running. If a thread from the ready queue was to be dispatched I calculated the time it was dispatched in the variable lastDispatch. Also, whenever a thread was preempted I updated the estimated burst time with the formula:

thread.estimatedBurstTime = (int) (.75\*thread.lastCpuBurst + .25\*thread.estimatedBurstTime);

This mean that the estimation relied more on the last cpu burst then the last guess at the burst time. Lastly for the variable I felt it was bad coding practice to leave them to be accessed without accessors and mutators. Overall this was an easy programming assignment if the person doing it completed the first assignment.