jbobotek_lab2report

Question 1 - Multilevel feedback queue scheduler:

After reading and rereading the description for the assignment, the algorithm for making a MFQ Scheduler for this assignment turned out to be pretty easy.

I started by creating 3 queue vectors compared to the original file's one. I then modified the constructors to initialize these additional Vectors, as well as to modify timeSlice to fit the quantums allowed for each level queue. Note that these additional queue Vectors also mandated changing findMyTcb to account for the 3 queues it would have to search instead of just 1.

Next I modified the run method to, again, account for the 3 different queues. In my algorithm, queue0 is highest priority and all new Threads start out in queue0, and are allowed to run for timeSlice/2 quantum before they are suspended, and then removed from queue0 and placed into queue1.

For Threads in queue1, the same pattern occurs, but rather than just moving the Thread down after timeSlice/2, it checks if queue0 is not empty, and suspends the Thread to take care of the higher priority queue0. If nothing is found in queue0, it lets the Thread execute for another timeSlice/2 quantum before checking queue0 a second time. After the second time, if the Thread has not completed, it is suspended and moved to queue2, a lower priority queue.

For Threads in queue2, they again resume or start the first Thread in their queue and allow it to execute for timeSlice/2, and check both queue0 and queue1 for higher priority Threads, suspending as needed. Threads in queue2 check 4 times before the Scheduler suspends the Thread and places it back in the queue2.

Question 2 - Test Results - round robin vs MFQS:

In all cases, response, turnaround, and execution times were all improved by using MFQS instead of round robin. Note that because I also built a MFQS using setPriority instead of suspend and resume, that produced magnitudes faster response, turnaround, and execution times than the suspend and resume versions.

Question 3 - FCFS theoretical question:

So what's interesting about First Come First Serve is that it does have some advantages. If the Scheduler was implemented as FCFS, execution time would go down, as would turnaround time overall, but response time would shoot through the roof, meaning any sort of real time response would be drastically reduced.