CMPUT 379
OPERATING SYSTEM CONCEPTS

Assignment #4
A Multithreaded System Simulator

Due: Friday, December 7, 2018, 09:00 PM

By Hee Soo Park (1389532)

Objective:

To implement a multithreaded system Simulator using pthreads, with deadlock prevention and synchronization.

This assignment was emphasizing in learning of using multithreads, as well as deadlock prevention and synchronization, using pthreads, and locks.

Also to monitor the thread synchronization in terms of managing racing for a resources. Also to help understand the concurrency with the concept of Dining Philosophers Problem.

Acknowledgements:

Professor Ehab Elmallah lecture slides and examples.

TA’s help during lab sessions

https://www.thegeekstuff.com/2012/05/c-mutex-examples/?refcom

https://www.geeksforgeeks.org/multithreading-c-2/

https://stackoverflow.com/questions/122616/how-do-i-trim-leading-trailing-whitespace-in-a-standard-way

Design Overview:

The design of my a4tasks is followed by the description of Assignment page, following all the required outputs and

correctly display the monitor and termination information.

Project Status:

a4tasks have been finished with all the required functions, they are stable and correct. I have implemented all the required functions as modular as possible, separating into multiple files to follow modular instructions. Monitor thread and task thread work. I have also built the program using deadlock prevention, synchronization and concurrency. I am satisfied with my program and their outputs and GUI.

I had hard time with figuring out the deadlock prevention. Also managing multithreads was very challenging for me, as I have to manage the threads racing to its required resources. Also, since the output will not always be the same and may differ, I had hard time to make sure if my program is executing correctly or not, since I can not compare to to the sample output on eClass, since my output may be different even though my program is correct.

Previously I had difficulty coding in C since I had to manage dynamic allocated memory and had trouble debugging, but I feel like I have overcame that challenge from this assignment. I also used “gdb” debugger to debug my errors.

Testing and Results:

I have tested a4tasks by

Running functions by, running example from eClass, and executing program and examining the output.

The output of monitor thread and task thread seems correct as well as the termination output.