

Name: Drew Masters

Date: 9/7/2016

Class: COSC 462

### **Homework 1**

The problem as I understood it was there were a given amount of philosophers and the same amount of forks. The philosophers are around a table. The forks are placed between two philosophers. This arrangement means that each fork is used by the philosophers on each side but can only be used by one at a time.

The philosophers start out thinking. In order to eat, the philosopher must have both forks that are adjacent to him, one on the "left" of the philosopher and the other on the "right" of the philosopher. He can eat as much as he wants and there is a unlimited amount of food. Once he's finished eating he puts both forks back on the table and goes back to thinking.

The way I approached this problem was to use mutexs as the synchronization between the threads. I chose mutexs because they work well as the "forks" in the problem. Threads represent the philosophers. The threads execute a function that continually tries to get the forks adjacent to the philosopher to eat.

The status of the fork is determined by if the mutex is locked or unlocked. If the mutex is unlocked this means in the context of the problem no philosopher has that fork and it can be taken by one of the two philosophers it is between. If it is locked then that means the fork is taken by one of the philosophers.

After the philosopher/thread acquires both mutexs, it eats. In the program the philosophers/threads hold the mutexs for a random amount of time between 0 and 3 seconds. Once they are finished eating they release both mutexs. They think for a random amount of time between 0 and 3 seconds then try to eat again.

### **Steps to run code**

1. Run make. Makefile is included in the tar file turned in. Can run make clean to delete executable
2. Executable generated by makefile is dining\_philosophers. It takes one command line argument and that is the number of philosophers you want to use in that instance.
  - a. `./dining_philosophers number_of_philosophers`
3. In order to kill the executable you hit control c