MAT 450 — Operating Systems

Homework #4, Due on Friday, April 22, 2005.

In this homework, you will write pseudocode to solve the Dining Philosopher's problem without deadlock. For extra credit, solve this problem without starvation.

1. There are five philosphers who spend their lives thinking and eating. The philosophers share a circular table. Each philospher has a chopstick to her left and her right. The chopsticks are shared among philosphers so that there are only five chopsticks. (see page 208 Figure 6.14).

2. To solve this problem without deadlock, you will allow a philosopher to pick up her chopsticks only if both chopsticks are available. There are five philosophers denoted P_i where $i \in 0..4$.

3. Use the following shared variables in your solution:

   (a) **int chopsticks[5]** - Each element of this array is initialize to -1. When **chopsticks[j] == i** it means that philospher P_i is holding **chopsticks[j]**. **chopsticks[i]** is to the left of P_i. **chopsticks[(i+1)%5]** is to the right of P_i.

   (b) **int state[5]** - This is the state of each philosopher. Valid states are **EATING**, **HUNGRY**, and **THINKING**. As an example, when philosopher P_i wants to pick up the chopsticks, **state[i] == HUNGRY**.

   (c) **semaphore mutex = 1** - This is a semaphore shared among the philosophers to ensure mutual exclusion in critical sections.

4. You will be writing pseudocode for some generic philosopher P_i. You may assume that you know the value of i so that you may obtain appropriate chopsticks and set the correct state.

5. Use a do while(true) loop around the philosopher, since she will eat and think for eternity.

6. You may assume your philosopher starts out in the **HUNGRY** state. However, it is your duty to change that state to **EATING** and **THINKING** when appropriate. Be sure to set the state to **HUNGRY** at either the beginning or the end of your do while(true) loop.