Washington State University Vancouver

Systems Programming - CS 360

Assignment 6

Instructor:
Ben McCamish

Overall Assignment - 100 points

Write a program (in C) called `assignment6.c` targeted at the Linux platform. Re-implement the dining philosophers problem using five POSIX threads spawned from the parent thread. This new implementation will not use any IPC facilities (shared memory, semaphores, signals, etc), nor will it `fork()`. You will need to make either one or five instances of `pthread_mutex` semaphores, depending on the approach to take. Refer to Assignment 5 for details. Example compilation statement:

```
gcc -o assignment6 assignment6.c  -lm -lpthread
```

Specifications and Restrictions

- (80 points) Program must work on the lab machines, including the specifications here and assignment 5.
- (20 points) Must be robust, including error catching. You must catch errors and print out an appropriate error message containing the `errno` and the message produced by that error. This means you will need to use `errno.h` and `string.h`, libraries at least.
- You should just copy and paste the random function I provided for assignment 5 into your code.
- (Required) As a small addition to this assignment, philos may not go back to thinking if they cannot get the chopsticks. They must continue to wait until they get chopsticks.
- (Note) To create and manipulate threads, use `pthread_t`, `pthread_create()`, `pthread_join()`, and `pthread_mutex_init()`.
- (Required) Clean up any memory you allocate.

What to turn in (on Autolab):

- assignment6.c