Taimoor Hafeez CISC 3320 ER6

Assignment 3

I added the java classes I used in the first two assignments into the file for this program so those classes remained the same. I created my own mutex lock class that uses the Semaphore class in java by creating a private Semaphore object that would act as my lock. The object is allowed 1 permit as seen in the initialization.

The acquire() method first checks to see if a permit is available for this semaphore. If it is available, a permit is acquired for the pid to be allocated and a proper message is also printed to the console. If a permit is not available, then an error message is printed to console.

The release() method first checks if the number of available permits is 0. If the number of available permits is 0, the permit is released back to the semaphore object making the permit available. Messages are printing to console stating the permit is released and available, and the number of permits available which in our case is only 1.

The main method is pretty similar to the main method that was in my assignment 2 submission. The only difference is that the method checks to see if a permit is available and the thread is finished before allocating a different pid.