# CS4315 Operating Systems

## Lab Exercise – Chapter 8 Basic Synchronization

Name _____ ID _____

Write a C program on Gator to solve the following problems (Read textbook p332 – 334 to learn how to use pthreads). Turn in the source code of the program that is the solution to Problem 6.

1. Write a program to simulate deposit/withdraw activities on a banking account: Initialize the beginning balance to 1 million, withdraw 600 thousands, and then deposit 500 thousands. Finally print out the ending balance. What is the output?

2. Write two functions, one for withdraw, the other for deposit. Declare the balance as a global variable. Both withdraw and deposit functions have one parameter, which represent the amount to withdraw or deposit. Then call the two functions from main().

3. Modify the withdraw and the deposit functions to make them deduct the balance and add to the balance one dollar at a time, respectively. Therefore, to withdraw 600 thousands, for example, the withdraw function has to execute a loop with 600 thousand iterations. The deposit function works in the same way, in the reverse direction.

4. Create two Posix threads in main(), which call the withdraw and the deposit function respectively. You can create these two threads in any order. However, before you create the second thread, use pthread_join() to wait for the first thread to terminate. What is the ending balance?

5. Move the calls to pthread_join() function after the creation of both pthreads. Run the program several times. Do you see different result?

6. Use pthread-mutex_lock() and pthread-mutex_unlock() functions to ensure mutual exclusion between the two pthreads. Is the ending balance right now?