Clayton Brutus
CS470
Dr. Hwang
Thread Sync Project
2/20/17

Program has a modular design and is written in C++. The program parses the command line arguments into integers and creates/starts the threads based on those numbers. A mutex lock is used around every cout command since it is not atomic. Two semaphores full and empty, keep track of the number of items in the buffer and the number of empty spaces, respectively. The producer(s) wait on the empty semaphore before inserting an item and signals the full semaphore afterwards so that it only inserts an item when there is space. The consumer(s) wait on the full semaphore before removing an item and signal the empty semaphore afterwards so that it only removes an item when there is an item in the buffer to remove.

1. What aspect of thread manipulation did you find most difficult to understand?
    a. Passing parameters to threads
2. What aspect of thread manipulation did you find least difficult to understand?
    a. Creating threads
3. What aspect of thread synchronization did you find most difficult to understand?
    a. Where to place semaphore wait and signals so that the threads sync properly
4. What aspect of thread synchronization did you find least difficult to understand?
    a. How to use mutex locks
5. What, if anything, would you change in your current design?
    a. The datatype of the buffer, since it is currently an array but would make more sense for it to be a queue
6. What, if anything, did you find interesting or surprising about thread manipulation or thread synchronization that you did not know before doing this project?
    a. I was surprised about how easy and fast I was able to create and manipulate Pthreads