Project 1

CSCE 4600

Members: Robert Martinez, Isaac Thomas, Thomas Truong, Clint Wyatt

**Project 1 implementation**

Pseudo notes….

Thread 1 must have at least 1 node in the free list at all times, otherwise thread 2 will cause a segmentation fault. Thread 1, known as the producer, is controlled by 2 semaphores, 1 for mutual exclusion, and 1 representing the number of nodes (blocks) -1.

All threads use a while loop to cause the program to run indefinitely to allow the threads to execute multiple times in any order. The first semaphore in a thread is a counting semaphore on one of the lists to ensure that there is a node to unlink followed by a binary semaphore shared by all threads for mutual exclusion.

For thread 1, the counting semaphore uses the free list as the goal of thread 1 is to move a block from the free list to list 1. Thread 2 uses list 1 as its counting semaphore since the number of blocks from list 1 reduces by 1. While thread 2 unlinks a block from the free list, the number of blocks does not change, so it does not need to use a counting semaphore for the free list and the binary semaphore ensures that no other thread modifies it. Thread 3 uses list 2 for the counting semaphore as is the similar to thread 1 just with different lists.