Georgia Ma

V00849447

CSC 360

Assignment 2 - Deliverable 1

1. How many threads are you going to use? Specify the task that you intend each thread to perform.

I will use one thread for every customer and clerk, and have a main thread to set up and take down.

2. Do the threads work independently? Or, is there an overall “controller” thread?

They will work independently.

3. How many mutexes are you going to use? Specify the operation that each mutex will guard.

I will use two mutex locks to guard operations to each of the customer queues. They will be shared between all four clerks.

4. Will the main thread be idle? If not, what will it be doing?

The main thread will likely stay idle when waiting for the clerk threads to finish clearing the customer queues.

5. How are you going to represent customers? what type of data structure will you use?

The customers will be represented as structs inside 2 queues (Business and Economy). Each customer will also be represented by a thread to simulate waiting for an open clerk.

6. How are you going to ensure that data structures in your program will not be modified concurrently?

Every operation on the customer queues, including read operations, will be protected by mutex lock.

7. How many convars are you going to use? For each convar:
(a) Describe the condition that the convar will represent.
(b) Which mutex is associated with the convar? Why?
(c) What operation should be performed once pthread cond wait() has been unblocked and re-acquired the mutex?

Convar 1: Business Class

(a) If a clerk is ready for a new Business class customer, wake up the customers and lock the mutex so that it can access the queue.
(b) This is associated with the business queue-locking mutex to protect the queue from being edited concurrently.
(c) Lock the queue so the open clerk can receive the next customer.

Convar 2: Economy Class

(d) If a clerk is ready for a new Economy class customer, wake up the customers and lock the mutex so that it can access the queue.
(e) This is associated with the economy queue-locking mutex to protect the queue from being edited concurrently.
(f) Lock the queue so the open clerk can receive the next customer.

8. Briefly sketch the overall algorithm you will use. You may use sentences such as:
If clerk i finishes service, release clerkImutex.

The customer queues and threads will be set up fully in the main thread before 4 clerk threads will be created. Each clerk will take turns taking customers (Business then Economy) and serving them. Once a clerk finishes serving a customer, it will broadcast to the Business queue, check if the mutex lock is unlocked, and then it will check if the queue is empty. If Business is empty, it will do the same for Economy. If the queue is not empty, it will take the next customer in line to serve, otherwise it will exit.

To serve a customer each clerk thread will lock the mutex to check if the queue is empty, and keep it locked as it removes the customer from the queue. After, it will unlock and sleep for the alloted time duration of each customer before broadcasting once more.