CSC 360: Operating Systems Nikita Malhotra
Assignment #2: Design Document V00790483

1. The number of threads to be created will be based on the number of customers. And there will be a controller thread. So, if there are three customers, then there will be four threads in total.

2. There has to be a controller thread as this thread will be responsible for maintaining the priority queue and scheduling the next thread.

3. There are going to be two mutexes. One mutex is going to guard the clerk so that only one customer (thread) gets executed at a time. The other mutex is will guard the queue.

4. Yes, the main thread will be idle and it will wait until the clerk has serviced all the customers.

5. The customers will be represented by a structure.

```
typedef struct {
        int customer_id;
        int arrival_time;
        int priority_time;
        int service_time;
        pthread_condition_cond;
        pthread_mutex_t mutex;
} customer;
```

6. By locking the mutex before performing operations on shared data between the threads.

7. The number of condition variables (convars) will be the total number of customers plus one.
(a) The clerk conditional variable (convars) will represent whether or not there is a customer to be served. The other conditional variables (convars) for the customers will represent the customers be waiting.
(b) The controller thread mutex is associated with the condition variable (convar).
(c) One pthread_cond_wait() has been unblocked, the customer that was waiting will start to get serviced.

8. The overall algorithm that will be used is as below:

- If there are three customers, the total number of threads that will be created is four. Each of the customers will have their own thread and there will be one controller thread. There will be four conditional variables and two mutexes – one for the controller thread and the other for the queue in which the customers will be enqueued/dequeued based on their priority.

- Each time a customer is added to the queue, it will signal the controller thread (clerk) to announce that the customer is here. Once the customer (thread) has been serviced it will again signal the controller thread (clerk) stating that the thread has been served and will also signal the controller thread to pick up the next customer as the mutex is unlocked.