Jack Dewey
CSC360
Nov 6th, 2022
Assignment 2 Design Document

1. How many threads are you going to use?
    a. I am using at most 35 threads. There are up to 30 possible customers, each using a thread. We will always be using 5 clerks, so total 35 threads.
2. Do the threads work independently?
    a. The threads are independent, but receive status/convar updates from each other to signal when to wait or start.
3. How many mutexes are you going to use?
    a. I will be using 8 mutexes and 2 arrays of mutexes. Each array is responsible for protecting either all the clerks threads, or all the customer threads. I will be using 2 mutexes to protect the queue of both Economy and Business class. I will be using 1 mutex to protect the enqueuing process. I will use 2 mutexes to each protect the process of creating the clerk/customer threads (which is different than the array protecting the individual thread). I will use 1 mutex to protect the counter of number of customers served. I will use 2 mutexes to protect both the counts of total waiting time for both the business class and economy class.
4. Will the main thread be idle?
    a. The main thread will be idle while the other processes complete. The main thread creates, joins, destroys threads.
5. How are you going to represent customers?
    a. Customers are represented as a struct containing their id, class_type, arrival_time, and service_time. Also then stores the calculated waiting time.
    b. Struct customer_info{
        i.
```
int user_id;
```
        ii.
```
int class_type;
```
        iii.
```
int service_time;
```
        iv.
```
int arrival_time;
```
        v.
```
double start;
```

vi.
```
        double end;
```
vii.
```
        double service_end;
```
viii.

6. How are you going to ensure that data structures in your program will not be modified concurrently?
    a. Each data structure input is guarded by a mutex explained in 3.
7. How many convars are you going to use?
    a. I am using 2 arrays of conditional variables. I use arrays because I assigned a convar for each thread. 1 array of convars is for the clerk threads, telling them when they need to wait and when they can continue forward with a new customer. I use the second array of convar to wake a sleeping customer thread to notify them that a clerk is available.
8. Briefly sketch the overall algorithm you will use.
    a. Queue all customers as they appear.
        i. Each new customer gets a mutex
        ii. Their respect queue (business or econ) gets locked
        iii. We enqueue
        iv. Then unlock
        v. Then mutex wait each customer to wait for a convar telling them they can visit the next clerk
    b. If num_served == total customers
        i. Thread exit
    c. If business queue has anyone
        i. Prioritize business queue
        ii. Mutex protect num_served
        iii. Increment num_served
        iv. Un-protect num_served
        v. Mutex protect business queue
        vi. Decrement business queue
        vii. Unprotect business queue
        viii. Mutex lock clerk
        ix. Wait for signal a customer thread they wake up
        x. Wait for wake up
        xi. Unlock clerk
    d. Otherwise take someone from economy queue
        i. Mutex protect numserved
        ii. Increment num served

iii. Un-protect
iv. Mutex protect econ queue
v. Decrement economy queue
vi. Un-protect
vii. Mutex lock clerk
viii. Wait for signal a customer thread they wake up
ix. Wait for wake up
x. Unlock clerk