Design Document

Zheng Yin

V00915261

CSC360

Nov 17, 2020

1. In my design, there should be a main thread, clerk threads, 1 customer thread and a time thread. Main thread identifies the input, create the thread and send the data to the thread. Customer thread unpacks the data from main thread and processing the customer timeline like when they join the queue and get set the queue to clerk threads. The clerk threads sperate the customer to 4 different clerks to do the multiprocessing. The time thread gives the time data to customer thread.
2. There should be a controller to control the thread going, which can be the main thread that control most the signal.
3. At lest 8 mutexes that 4 of them control the 4 clerks. 2 mutexes is used on separate the customer to queues, and 2 of them should work on the queue to clerk.
4. The main thread may not be idle when send data to customer thread.
5. I think I will use double linked list which is stable on keeping the data.
6. The condition variable can keep most of the data structures stable.
7. I think there should be about the same number of the condition variable as the mutex.
8. using 2 condition variables to connect the queue between the big list of customer and queue, which make sure the custom goes into queue by order. 4 condition variables to stop the customer meet clerk which clerk is working with another customer, and 1 condition variable make sure the first in the queue go to clerk thread.
9. The four mutex work with clerk should be associated with the convar. Which mutex make sure customer do not go to two different clerks in the same time, the convar make sure the customer only goes when clerk is free.
10. Before pthread\_cond\_wait, the mutex should be locked, and do not forget using a while loop get keep wait until get the signal.
11. Using strtok and atoi to get all data and store them in a customer\_info array. By setting two queues for business customer and economy customer as global variables. that can implement data from main thread to clerk thread and customer thread. In customer thread, using class to separate the date. Using a mutex lock when a set of data implement into the queue and using condition variable to wait the signal from clerk thread send the read signal. The customer can go into the clerk process. Import the time function in the beginning of the customer join the queue to make sure the time in different step, after the service time, send a signal to the clerk the customer in to let the clerk free. In these steps, program can get the time information like time waiting in the queue.

In the clerk thread, if thread is free, broadcast the condition variable to customer that need customer, and unlock the mutex to let customer in. Then wait the signal from customer thread which check-in is done and free again.