Loren Howard, Yujian Zhang, Ken Wakaba

Professor Ahmed Ezzat

COEN 383

29 October 2017

Multi-threaded ticket sellers simulation Report

There are 10 sellers with different prices selling ticket of a concert. Each seller is represented by a thread. There are three kinds of seller, high-price seller, medium-price seller and low-price seller, each of them would have a different selling time and different seat section to sell. These threads are running in real time, and 1 sec in reality represents 1 minute in our simulation scenario. Every seller thread has its own customer queue. And we would decide the N by inputting our desired number in the console.

The routine for each seller is as follows: 1. Wait until a new customer (serve them in a FIFO policy); 2. Find an available seat according the price of the seller; 3. If there is no seat available for this kind of price, the customer would leave right away and the seller would terminate; if there is an available seat, seller would take a while to do the serving.

The whole seats in the concert are represented by an array of each individual seat. The whole seat array is protected by a single mutex. We think the looking up procedure would be executed very fast, there is no need to assign each seat with its own lock.

We have a total of three kinds of event, customer arrival, customer gets informed whether he gets a ticket or not, customer leaves. We have a linked list to store all the events and maintain their order. There is no special mechanism to maintain the events ordering when there is only 1 entry to the event storing, so we try to make the access to this event storage as quick as possible, which would maintain a low possibility of disordering. We have an additional printer thread to handle all the printing in real time without increase the disordering possibility.

The simulation lasts at most 60s, or until all seats are sold out. The sample output is as follow, each line represents a single event, and follows by the order of happening.