**CS5348**

**Operating Systems Concepts**

**Project 2**

**Threads**

**Design and Pseudocode**

**Name:** Madhusudan Pranav Venugopal

**UTD ID:** 2021163441

**netid:** mxv130430

***List of semaphores used:***

semaphore taq= {0} // For enqueuing for ticket

semaphore customer[330]= {0} // For synchronizing between customer i and other threads

semaphore cagent[330]= {10}// For synchronizing between box office agent 0/1 and customer i

semaphore ttqueue={1} // for enqueuing in queue for ticket taker

semaphore csqueue={1} // for enqueuing in queue for concessio stand

semaphore tq={1} // for dequeueing from queue by box office agent

semaphore ttqueue1= {1} // dequeuing by tickettaker

semaphore csd={1} // dequeuing by concession stand

semaphore cttready={0} // signal that customer has entered queue for ticket taker

semaphore cscready={0}// signal that customer has entered queue for concession stand

semaphore moviestatus={0} // to signal that movie structure is used by another box office agent

***Pseudo code:***

*Customer thread:*

semWait(taq);

enqueue() // Queue for buying tickets;

semSignal(taq)

semWait(customer[i])

// Generate movie name to be seen by customer

movie name= movie name associated with rand() % ( no of movies)

semSignal(cagent[i]);

semWait(customer[i]);

// Check seat sold

if(seat sold == N)

// goto last part of the thread function

else

semWait(ttqueue);

enqueue() // queue for seeing ticket taker

semsignal(cttready); // signal that customer entered the queue for ticket taker

semSignal(ttqueue);

semWait(customer[i]);

// Decide if want to go to concession stand

concessionstanddecide = rand() %2;

if(concessionstanddecide =0)

// goto to last part of the function ( does not want to enter the concession stand)

else

// Create concession stand selection choice

concessionstandchoice= rand()%3;

if(concessionstandchoice==0)

choice= popcorn;

if(concessionstandchoice==1)

choice = soda;

if(concessionstandchoice==2)

choice = popcorn and soda;

semWait(csqueue);

enqueue() ;// Enqueue for concession stand

semSignal(cscready); // Signal that customer has entered the queue for concession stand

semSignal(csqueue);

semWait(customer[i]);

//Enter the theatre

// Join thread in main function

*Box office agent thread:*

b:

semWait(tq);

dequeue(); // dequeue from ticket queue

semSignal(tq);

semSignal(customer[i]);

// Select movie structure for the movie selected by customer

semWait(moviestatus);

// Check no of seats available

if( no of seats available > 0)

noofseats--;

seat sold= y;

else

seat sold = N;

sleep(1.5);  
semSignal(moviestatus);

semSignal(customer[i]);

if (done!=1)

goto b;

*Ticket taker thread:*

tt:

semWait(cttready); // Wait for signal that customer as entered the queue (queue not empty)

semWait(ttqueue);

dequeue(); // dequeue from ticket taker queue

semSignal(ttqueue);

// print ticket taken from customer i

semSignal(ttqueue)

sleep(0.25);

semSignal(customer[i]);

if(done !=1)

goto tt;

*Concession stand thread:*

csa:

semWait(cscready);// Wait signal from customer that it has entered the queue (queue not empty)

semWait(csd);

dequeue() // dequeue from queue for concession stand

semSignal(csd);

// print order taken from customer i

sleep(3);

// print order given to customer i

semSignal(customer[i]);

if(done!=1)

goto csa;

*main function:*

// Receive filename as command line argument

moviescount(filename) // Function to count the number of movies

databasecreation(filename) // Function to count the number of movies

parbegin(customert,...300 times...customert,boxofficet,boxfficet,tickettakert,concessionstandt);

*moviescount function*

//Receive filename

// Open the file

// Read line by line

noofmovies++;

*databasecreation function*

//Receive filename

// Open the file

// Read line by line

/\*separate the movie name and no of seats available using strtok function and store in respective movie structure\*/

m[i].moviename= movieame;

m[i].noofseatsavaialable= seatsavailable;

// initialize moviestatus semaphore to 1

: