Tuğcan Barbin 25168 CS307 homework1 report

At the beginning I declared the global variables like turn, numberOfSeatsLeft and the matrix as seats[2][50].

In main;

I used srand(time(NULL)) for creating different seeds for each execution so that the reservation result will become different each time.

Then I declared two different phtread_t variables and after that I created 2 thread. One for TravelAgency1 and one for TravelAgency2 with my functions that are Agency1 and Agency2.

I used pthread join so that main will wait until the two child threads are done at this point.

In my thread functions. Firstly, there is a Boolean expression that controls the while loop. Actually in homework busy waiting reminders while loop which is declared as while(true){}; however, I tried pthread_exit code like slide examples and recitation examples but it did not work. And in visual studio 2012 I could not include the pthread.h and debug it. Then I tried return method and this time visual studio had too many errors about void * and return I could not handle. So I decided to do it with a Boolean. (cont)

Then, in while loop again there is a Boolean variable which controls if plane is full or not. (full)

I take the seat number as a (rand()%100 +1) so that it has become between 1 and 100.

Took row as seatNo/51 which allows me to decide which row the seat is, and if seat number is greater than 51 I took column as seatNo-51 otherwise seatNo-1 this allowed me to take index of column of that seat number.

Then the busy waiting part while(){} allows me to block that thread if the other thread is working on.

When I created threads turn was 0. So, for my code Agency 1 will be the first thread that enters the critical point.

In critical region I first checked the if the plane is full or not. If full then set the full as true. if not check the current seat is reserved or not. If it is do not reserve. If not reserved the seat, reserved, marked the seat and decremented the available seat number.

Then, exit the critical region. Set turn as 1 so that the other thread can continue.

Check if plain is full or not. If it is full set cont as false.

When the plain is full then firstly the full variable will be true and then cont variable will be false which is the condition of my outer most while loop. So thread will be done.

And because of the number of seats is global variable, other thread will do the exact same thing and terminate.

Then my program warns that no seats left and plane is full and

Prints the whole matrix.