CS 307 // HW1 // AIRLINE RESERVATION SYSTEM

-Berna Yıldıran-

1-main()

- “TravelAgency1” and “TravelAgency2” threads were created.
- These two threads will run while “flagofthreads” bool variable is “true”. If “countSeat (initially equal to 100)” became 0 after executions (because when a thread books a seat, “countSeat” will be decreased by 1), “flagofthreads” variable will change to “false” and this will stop the reservation_sys1 and reservation_sys2.
- When “countSeat” equals to 0, this means that all seats in the plane is full and while(flagofthreads) loop will end.
- After that, “No seats left” message will be printed and “TravelAgency1” and “TravelAgency2” will join to the main thread.
- Lastly, global matrix M which was representing the seats in the plane will be printed.

2.reservation_sys1() and reservation_sys2()

---both function works in the same manner, everything is same except the value of the “turn”---

- While “flagofthreads” did not changed inside the main thread by the “countSeat”, TravelAgency1” and “TravelAgency2” threads use these two functions to reserve seats in the plane. (TravelAgency1 → reservation_sys1(); TravelAgency2 → reservation_sys2())
- While “flagofthreads” is true both threads will synchronize with each other through “Busy Waiting”. There is a global “turn” variable which is initially equal to 0. While “turn” is not equal to 0, AgencyThread1 will do “Busy Waiting”, AgencyThread2 will run visa versa. This way two threads will work with “Strict Alternation” principles.
- The thread whose turn is come will exit from “Busy Waiting” and print “Agency X Entered Critical Region.” message depending on its agency number.
- bool “isEmptySeat” variable which is initially “false” will be used for exiting from the while loop which defined inside.
- Before thread enters the critical region which it accessed the shared variable 2D array “M”, a random “seatnumber” will be generated. If that “seatnumber”s location (functions which used for determining the location of that seat number in the matrix will be explained in the 3rd part) is empty, thread will go inside the while loop if isEmptySeat” and “flagsofthreads” are both “true”.
- Inside that while loop, thread will reserve that seat by changing the value of that location to 1 or 2 (depending on their agency number). At the same time “isEmptySeat” bool variable’s value will change to “true” and thread will exit from this while loop after printing “Seat Number X is reserved by Agency Y” message and decreasing “countSeat” by 1 (because 1 seat is reserved and 99 seats left).
- When thread is exit from this critical region, function will print “Agency X Exit Critical Region.” And change the value of “turn” variable. (Agency1 change it to 1, Agency2 change it to 0).

3.seat_loc0() and seat_loc1()

- Functions for finding the “seatnumber”s location in the matrix. seat_loc0() finds the which row it is in and seat_loc1() finds which column it is in.
- If “seatnumber” is smaller or equal to 50, seat_loc0() will return 0. If “seatnumber” is bigger than 50, seat_loc0() will return 1.
- If “seatnumber” is smaller or equal to 50, seat_loc1() will find its column by substraction 1 from “seatnumber” because indexing is starting from 0 inside the matrix (i.e. seatnumber 50 will be in the 49th column). If “seatnumber” is bigger than 50, seat_loc1() will substract 51 from that “seatnumber” (i.e if seatnumber is 100 it should be in the 49th colum).