

CS 307 – HW1 – REPORT

In the main function, I have created two pthreads as thread1 and thread2. Assigned these threads to their own function called func1 for the thread1 and func2 for the thread2 by pthread_create method. Then used pthread_join method so that main thread waits until thread1 and thread2 complete their work.

The busy waiting algorithm is integrated into func1 and func2 in order to prevent threads from accessing the critical region at the same time and reserving the same seat. While a thread is doing busy waiting, it creates a random number between 1 and 100. Then, by the time critical region is free to enter, thread goes into the critical region. First checks whether the plane is full. If plane is not full, they check the random integer seat and if it is not taken before (it is represented by 0), turn the 0's to their special number which is 1 for thread1 and 2 for thread2. If they reserve a seat, global remaining seat variable will be decremented by one. If the seat is already taken, thread exits from the critical region without making any reservation. The entire loop continues until the plane becomes full.

By the time, plane is full which means there is no remaining seat, both threads complete their assigned functions. Then main thread continues to execution by printing the global matrix variable.

Görkem Güzeler – 26841