Duygu Tümer

24842

REPORT PA3

In my PA3, first, I create my threads and semaphores in my main thread. I create 3 semaphores (one for team A, one for team B and the last one is for choosing the driver), 2 mutex (one for checking conditions and the other one for printing) and two arrays for my threads for team A and team B which the number of size is given by the user. After I create my threads, they go to given function. For array A, threads go to function “void *teamA()”. For array B, threads go to function “void *teamB()”.

In function “void *teamA()”, first I create random id for threads and then I create a variable “driver” for choosing the driver. For my “mutex” first I have to lock it because I want to check conditions thread by thread. I have a global value which name is “con”. “con” It helps me to print the desired outputs in sequence. Also, I have two global variables (“inA”, “inB”) which help me to count the number of people who wait for the bus.

First, we have to increment number of people who are waiting. After then, we need to check total number of people who are waiting. If number of people is four in teamA or teamB, then there can be a driver and also, they can find a spot. Else if the number of people is again four (but two from teamA and 2 from teamB) then there can be a driver and also, they can find a spot. So, I must decrement inA or both inA and inB since they don’t need to wait anymore. I use necessary number of sem_post statement to free the threads. Then I use another semaphore to make the driver wait for everyone to print. After then I make the last print statement for driver. Same system is working for also function “void *teamB()”. And at the end, since I don’t need them anymore, I must destroy my threads and semaphores.

Pseudocode for function “void *teamA()”:

Lock mutex

while(con != 0) {} //if condition is not 0 wait here (this is for printing one by one)

Lock mutex2

print: I am looking for a car

Unlock mutex2

inA += 1 // increment the number of people waiting for the bus

Check for:

if inA >= 4 //if there are 4 people who wait the bus in teamA

send them inside the bus using “sem_post(&waitA);” for 4 times

driver = 1 //we are looking for a driver (now there must be a driver)

con = 1

inA -= 4 //decrease the number of people who are waiting for the bus

else if inA >= 2 && inB >= 2 //if there are 4 people (2 from A – 2 from B) who wait the bus

send them inside the bus using 2 times “sem_post(&waitA);” and 2 times “sem_post(&waitB);”

driver = 1 //we are looking for a driver (now there must be a driver)

con = 1

inA -= 2 //decrease the number of Team A people who are waiting for the bus

inB -= 2 //decrease the number of Team B people who are waiting for the bus

Unlock mutex

sem_wait(&waitA) // it is like a barrier it will cross if num of people will be >= 4

lock mutex2

print I have found a spot in a car

Unlock mutex2

sem_post(&driverSem) // do this for 4 people

if driver == 1 //now choose the driver

//since the driver should print at the end of four people use “sem_wait(&driverSem);” 4 times

Lock mutex2

Print I am the captain and driving the car

Unlock mutex2

con = 0; // make con 0 so we will wait after these 4 people will be gone

Now, do this function for also teamB.