Mert Coşkuner-29120

At first, I used a single thread function to implement my logic in this assignment.

Data structures:
Struct FAN: this data structure is used to hold the team name and the representing thread related to that fan.
Vector FAN: this data structure is used to hold all the fans in a vector.

Main Function:
Starts with getting the input from the command line and that initializes the 2 fan semaphore with the value of 3 and the general semaphore, which is for the car, to 4. After that, initialize the 2 barriers with the value of 4. Then the standard conditions like the arguments must be the multiple of 2 and the total of 4 is checked. If these conditions are not met the program terminates directly. Moreover, with 2 loops, I am creating a specified number of A fans and a specified number of B fans. After that with another loop, I iterate through the vector getting the thread variables and team names, and create my threads. In the main, lastly, I am joining all the threads and then terminating the main.

Thread Function:
It starts with getting the team from the argument part of the thread function. After acquiring the mutex lock a thread will first say that it is looking for a car. Previously the information of the team was obtained from the argument. So, the program decreases the value of the related fan semaphore. But in this part, there is a need to be careful because if a thread goes through this part and it is the last value of the semaphore it can sleep and cause a deadlock therefore I included a condition that checks this problem. Furthermore, now there is a need to check the car semaphore to find a place for all of the passengers. Here, the values of x, y, and z will indicate the value of fan semaphore A, fan semaphore B, and general semaphore respectively. The fan sempahores values are initialized to 3 therefore it is easy to check the condition of the car in every 3 value. There are 4 conditions in this case; 3 of them can be B and the program can finish the share of A and wait for one more B, 3 of them can be A, and the program can finish the share of B and wait for one more A, 2 of them can be A and 1 of them can be B the program will decrement the values of the fan semaphores. Lastly, 1 of them can be A, and 2 of them can be B again the program will decrease the value of the fan semaphores by one. After

that, there is a barrier which is previously initialized to 4 and waits for 4 threads. After 4 correct threads, the passengers are ready and can say that they have found a spot in the car. After that, there is the selection process for the captain, and releasing all the semaphores gets all of them back to their initial position.