Assignment 4
Concurrency
Computer Systems Engineering 1
Monsoon 2019

Ober Cab Services:

Ober is a new cab service, which needs your help to implement the simple system. The requirements of the system are as follows given N cabs, M riders and K payment servers you need to implement a working system which ensures correctness and idempotency. Each cab has one driver. In Ober, payments by the riders should be done in Payment Servers only. Ober provides two types of cab services namely pool and premier ride. In pool ride maximum of two riders can share a cab whereas in premier ride only one rider. There are four states for a cab namely **waitState** (no rider in cab), **onRidePremier** , **onRidePoolFul**l (pool ride with two riders), **onRidePoolOne** (pool ride with only one rider).

As a part of this system you need to implement the following functionalities.

1. Riders :

a. **BookCab**: This function takes three inputs namely cabType , maxWaitTime , RideTime . If the rider doesn't find any cab (all cabs are in usage) until maxWaitTime he exits from the system with TimeOut message.
b. **MakePayment:** This function should be called by rider only after the end of the ride. If all the K payment servers are busy, then the rider should wait for the payment servers to get free. After payment is done rider can exit from the system.

2. Drivers :

a. AcceptRide: If ride is premier cab in waitState should accept ride and change its state to onRidePremier . If the ride is pool and there is a cab with state onRidePoolOne then that cab should accept the ride and changes its state to onRidePoolFull . If the ride is pool and there is no cab with the state onRidePoolOne then cab in wait state should accept ride and change its state to onRidePoolOne . Assume that there is no time gap between accept rider and pickup rider, i.e., ride starts

immediately after accepting ride. If there are multiple cabs available with required state take any random cab.

b. OnRide: When pool cab is on ride and state is onRidePoolOne then driver can accept another pool ride. No new rider is accepted when cab is on premier ride.

c. EndRide: The driver ends the ride after completion of the RideTime of the rider. If the ride is a premier it goes to waitState (ready to accept new rides) once ride is done. If cab's state is onRidePoolOne then driver ends the ride and goes to waitState . If cab's state is onRidePoolFull then it goes to onRidePoolOne state (Notably, it can accept another pool rider). Driver or cab need not wait for the payment to be done by the rider.

3. Payment Servers:

a. Accept payment: Payment servers accepts a payment and assume the payment time is constant for all the riders (assume 2sec for this assignment).

Instructions :

1. Your code must not result in busy-waiting (deadlocks).
2. Use appropriate times for RideTime and the maxWaitTime.3. You need not follow above mentioned functions you can create functions appropriately.
4. Use Semaphores and mutex accordingly.
5. Prepare a detailed report explaining your implementation and assumptions. Note that the report will be graded.
6. Use separate threads for each of the drivers, riders and payment servers.
7. Use only four states of Cab as mentioned above.
8. Print appropriate statements for every update.
9. Write a proper report or Readme explaining the code.