CSE 120
PA3 Discussion – Part 2

driveRoad() Implementation

- Cars are on the road and they are running in the same direction.

  At this point, you are about to enter the next position by calling **ProceedRoad()**. But, what if there is a car right ahead of you? This means that your car should be blocked.

  Hint: Define a semaphore for each position of the road.

  int road[NUMPOS]

driveRoad() Implementation

- Before you enter:

  1) Wait for the first position of the road using a semaphore **‘queue’** (one for each direction).

driveRoad() Implementation

- But can I always enter the road even if the first position is available?

  No. Why? Some cars might be travelling in the opposite direction on the road.

  Say, you are a car(4) from the west, and there are three cars(1,2,3) running on the road from east to west. When the very first car, car(1) entered the road from east , it should lock a semaphore **'door'** (one for each direction) so that cars cannot enter the road from the opposite direction.

driveRoad() Implementation

When should you signal **‘door’**?

When you are the last car travelling on the road in a given direction and you are about to leave the road, you should signal **‘door’** so that cars in the opposite direction can now enter the road.

driveRoad() Implementation

3) Is that all?

In the above case, say another car(5) now comes from east.

a) The first position is available.

b) There is no car travelling on the road in the opposite direction because all the cars on the road are travelling from east to west.

Can car(5) enter the road?

driveRoad() Implementation

No. Why?

According to the rules, car(4) should wait only for car(1,2,3) which were already on the road when car(4)arrived.

car(4) should not wait for car(5) because when car(4) arrived, car(5) was not present on the road.

driveRoad() Implementation

So you need another semaphore **'ticket'** to prevent car(5) from entering the road. Cars should get the **'ticket'** semaphore first even before they wait on the **'door'** semaphore.

Hint: You need only ONE **'ticket'** semaphore used by both the directions.

driveRoad() Implementation

1. Wait for the first position (**west_queue/east_queue**)
2. Wait for the **'ticket'**
3. Wait for the **'door'** to open
4. Check if you are the first car to enter the road in a given direction. If yes, then lock the opposite **'door'**
5. Signal **'ticket'**
6. Signal **'west_queue/east_queue'** when moving to the second position from the first position.
7. Check if you are the last car to leave the road in a given direction. If yes, then unlock the opposite **'door'.**

driveRoad() Implementation

Hint: The first position has two semaphores:

west_queue and road[0]

or

east_queue and road[9]

Why?

- Say road[0] is occupied by car A, travelling from east to west. Now say, car B arrives from west (opposite direction of A) and car C arrives from east (same direction as A). If car B waits for its first position using road[0] semaphore, then it will be blocked on road[0] semaphore since car A is in position 0. Car C would wait for its first position using road[9] semaphore. road[9] is available. So car C would enter the road before car B (since **'ticket'** and **'door'** semaphores also would be available for car C) . But as per the rules, car B should enter the road before car C because when car B came ,only car A was present on the road. So you will need another semaphore for the first positions.