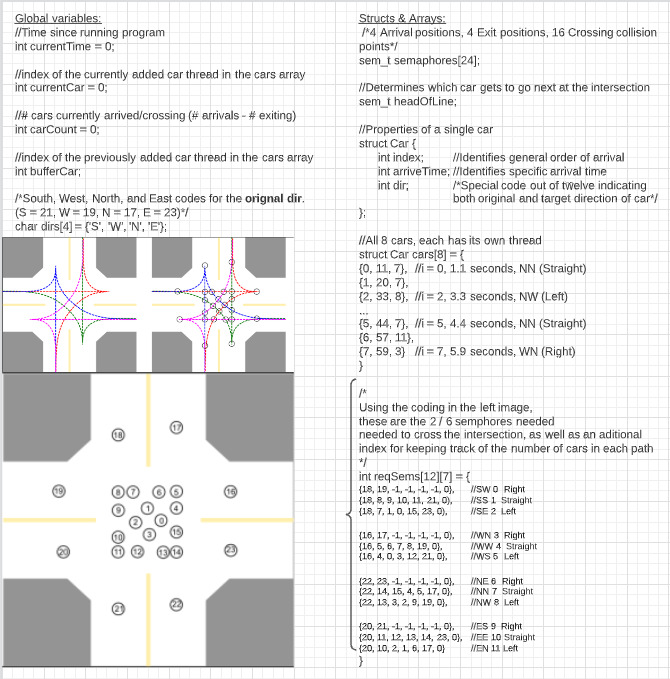
Programming 3 (P3): Four-way Intersection.

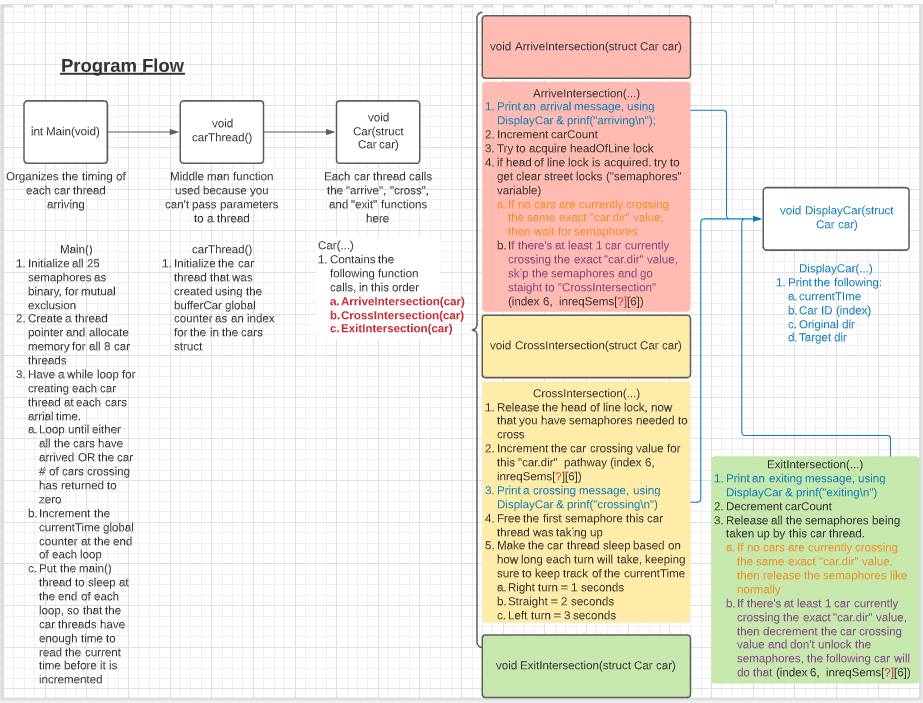
**Section 1: Program implementation details**

Variables and Functions used





General Workflow



Section 2: Working out the details

When my partner and I made our initial draft for P3, we only used 24 semaphores for each of the possible locations the car could be when it crosses the street. To simulate car arrivals from 4 directions (North, West, East, and West) and towards three different roads (Left, Right, and Straight from current directions), we programmed 12 preset paths across the 24 semaphores to simulate how cars would need certain spaces on intersection to safely cross the road.

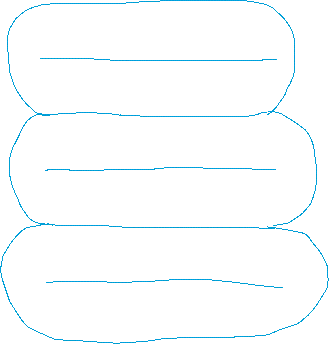
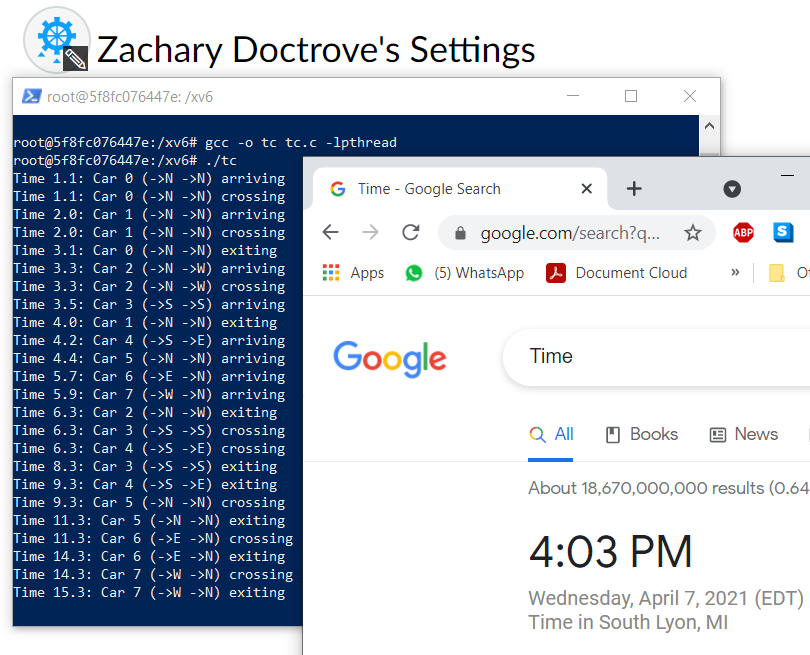
However, our output’s time with only 24 semaphores was still off and cars were showing up out of order. We quickly realized that we forgot an additional semaphore to simulate the turn taking system of intersection stop signs, which was called headOfLine. This fixed the exit order issue we were having, but the time of the last car’s exit was still off somehow.

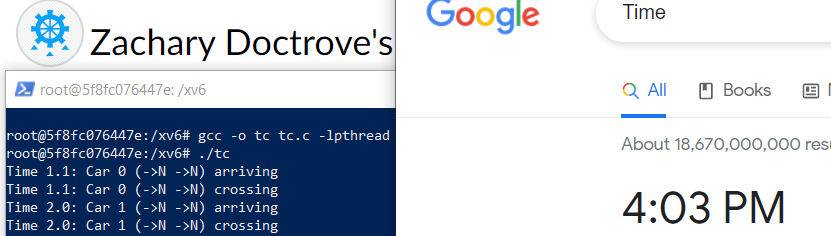
The final fix to this was the addition of semaphore skipping under the circumstance that cars are not in any real danger by ignoring them. In this case, this was only true when cars were driving in the same exact direction, as it was impossible for them to crash into each other without someone deciding to pull a break check.

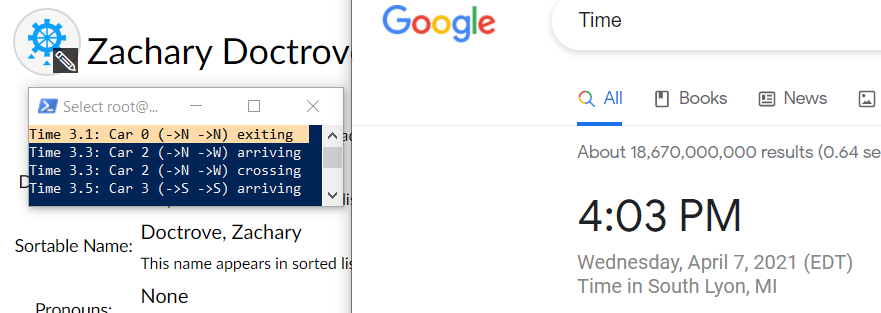
This semaphore skip function was tightly integrated into the pathing system discussed at the beginning of this section. By adding a counter that kept track of how many cars were crossing at a specific original/target direction, we were able to make it so cars that happen to come right after each other under such circumstance now can keep a chain of cars going in this one way if enough cars arrived on this one side before any other cars showed up.

**Section 3: Testing Results and Explanation**

Complete Output, broken down into four parts



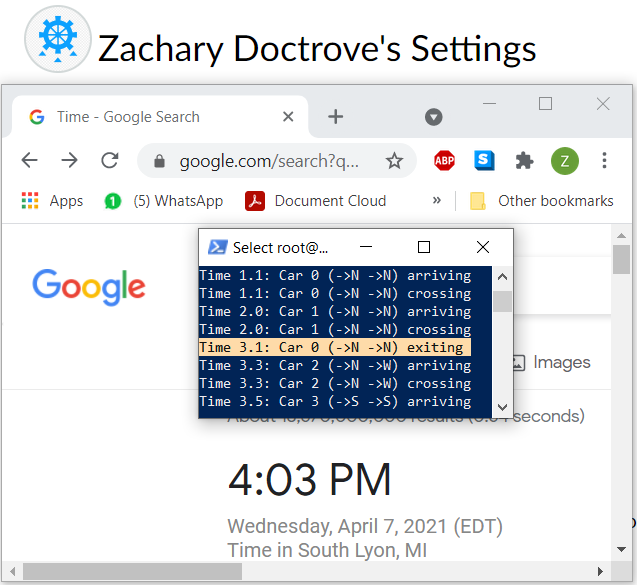
**1st four interactions** 

**2nd four interactions** 

**1st eight interactions:**

**Car 0 is the 1st and only car to arrive at the intersection at time 1.1, so it immediately crosses after it makes a stop. Car 1 arrives 0.9 seconds later while car0 is still crossing. But because the car1 is following the same exact path as car0, it doesn’t need to worry about hitting car0 since its driving away from car1. With no risk of hitting car0, car1 immediately goes as soon as it arrives. Since car0’s path is straight, it only took 2 seconds for it to cross. 2 seconds after arriving at the time 1.1, car0 is the first car to exit the intersection.**

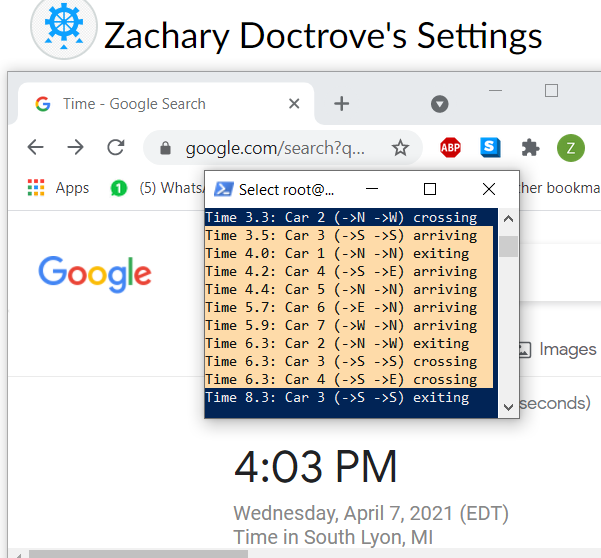
**At times 3.3 and 3.5, car2 and car3 arrive at the intersection. Unlike cars 0 & 1, the cars that come after all vary in their original and target directions frequently enough so that no car can cross the road at the same time (except cars3 and 4). Therefore, when car 3 arrives at time 3.5 and car 2 is already crossing, car3 is forced to wait at the stop sign.**



**2nd eight interactions**

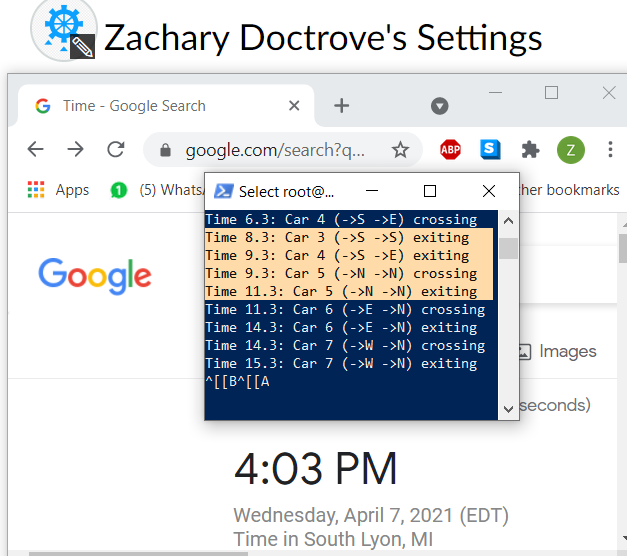
**After the previously mentioned car3 arrives and is stuck waiting at the stop sign, car1 finally completes its straight cross and exits the intersection. Soon after, a rush of four more cars arrives from all directions on the intersection. Cars 4, 5, 6, and finally 7 arrive at times 4.2, 4.4, 5.7, and 5.9. By this point, all the cars included in this demo have now arrived at the intersection.**

**Once car2 makes its 3-second-long left turn, it also exits the intersection. Cars 3 & 4 both begin crossing at the same time, due to their placement toward the same side, but they do not have the same target directions. The reason why they still leave at the same time is since they only must compete for the headOfLine semaphore and the 1st semaphore in their paths. Once car3 releases both earlier than the rest of the semaphores, car4 is free to cross as well. 2 seconds later, car3 exits straight through.**



**Last eight interactions**

**After car4 makes a lengthy left turn out of the intersection, now the target directions and original directions of each car truly never align. Because of this, the last three cars are forced to wait until the previous car has fully left the intersection and released all semaphores needed to cross. Finally, at 15.4 seconds, the final car in the demo has exited the intersection.**



**Section 4: Contributions made by each member.**

Bryan Pierce:

* Drew the graph containing the 24 street crossing semaphores, as well as the order in which they are ordered in the array.
* Implemented the “Arrive”, “Cross”, and “Exit” infrastructure of the code

Zachary Doctrove:

* Added the headOfLine lock to fix the order in which cars arrived at the intersection.
* Fixed bugs in the code