TungHo Lin(txl429)

Rongjie Zeng(rxz189)

EECS338 Final Project – Final Version

Design Document

Included Files:

-busfinal.c

-makefile

-pseudocode.txt

Description:

busfinal.c is the main class that simulates an operating bus route systems.

makefile can be used by to compile busfinal.c by calling “make”

pseudocode.txt is a pseudocode of our final version

Major Data Structures:

busfinal.c defines a couple of variables so that it would be easier to reference in the code:

|  |  |
| --- | --- |
| Variable Name | Value |
| RED | 0 |
| GREEN | 1 |
| BUS1 | 2 |
| BUS2 | 3 |

Other than the variables, a few data structures are defined:

|  |  |  |
| --- | --- | --- |
| Structure Name | Type | Usage |
| grid | 5x5 2D array | Simulate a 5x5 grid on which the bus will run |
| intersections | 1x2 2D array | Used to store the x-coordinate and y-coordinate of the intersections on the grid |
| done | int flag | Indicate whether all buses have reached their destinations |
| routeOne | 9x2 2D array | Contains the x and y coordinates of all the stops in route one. |
| routeTwo | 9x2 2D array | Contains the x and y coordinates of all the stops in route two. |
| horizontal\_light | Semaphore | This semaphore controls the horizontal traffic light. |
| vertical\_light | Semaphore | This semaphore controls the vertical traffic light. |
| mutex | Semaphore | This semaphore ensures that only one bus will be moved on the grid at a time. |
| next | Semaphore | This semaphore ensures bus2 moves after bus1 and that bus1 will wait until bus2 finishes its move. |

Difference between the final version and the beta version:

In the final version, we replaced the traffic light int flags with semaphores. It allows easier control and better readability. We also use a different route that does not only go in one direction; and because of that, we rewrote the implementation of how the buses will retrieve its next stop and move there.

TrafficLight() also has to be changed because we no longer use traffic light flags.

Things we didn’t implement from the Proposal:

We used a 5x5 grid instead of a 10x10 for the beta and we are still going to use the same grid size because we are only going to implement one intersection so there is really no point in doing the 10x10. We also didn’t let the user input the speed and the frequency the traffic lights are going to change because since we are doing a 5x5 grid, letting the user input values don’t really have a meaning. A thing that I regret is to implement the project in C. I thought since we learned everything in C, we should use C for it; but then I think using Java will allow for more creativity since my partner and I were planning to implement a complex GUI to actually show the bus moving across the grid. However, due to heavy work load from other classes, we decided to go with C.

Program Design: see pseudocode.txt

It contains 4 methods.

int main() will initialize the threads, assign the task of each thread and let them do their job. A total of 3 threads will be initialized for this beta.

void route1() will simulate a bus1 running a route. The route has been set in the routeOne array.

void route2() will simulate a bus2 running another route. The route has been set in the routeTwo array.

void trafficLights() will simulate the traffic lights at the intersection, which will change every 2 seconds.

Sample Output:

Bus1 at (0, 0)

Bus2 at (4, 0)

Bus1 at (1, 0)

Bus2 at (4, 1)

Bus2 at (4, 2)

Bus1 at (2, 0)

Bus2 at (3, 2)

Bus1 at (2, 1)

Bus1 at intersection(2, 2)

Bus1 at (2, 3)

Bus2 at intersection(2, 2)

Bus2 at (1, 2)

Bus2 at (0, 2)

Bus1 at (2, 4)

Bus2 at (0, 3)

Bus1 at (3, 4)

Bus2 at (0, 4)

Bus1 at (4, 4)

Bus2 has arrived at its destination

Bus1 has arrived at its destination