Traffic Light

written by Nik Tsonev

Intro

Like in real life, there is an intersection with a traffic light. Cars can come from South East North and West. The task is to make the cars go when its green for them and prevent any crashes.

Note that cars from opposing direction can intersect t the same time

Solution

have a traffic light for each direction and have each care wait until the semaphore for that direction is signaled.

the traffic light thread first signals that south and north or east and west can go, then wait for the respective semaphore to be signalled from the car thread, signalling that the car has crossed the intersection.

then signal that the other direction (so if first it was N S, then signal E W) that they can go and wait for their semaphore signal from the cars that crossed from those directions

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>
#include <semaphore.h>
#include <unistd.h>

#define DIRECTIONS 4

sem_t traffic_lights(DIRECTIONS);

void* car(void *arg) {
        int dir = *(int*) arg;
```

```
        sem_wait(&traffic_lights[dir]);

        printf("waiting\n");
        sleep(1); //cars passing by

        printf("car from %d crossed \n", dir);
        sem_post(&traffic_lights[dir]);

        return NULL;
}

void* controller(void *arg) {
        while (1) {
                sem_post(&traffic_lights[0]); // north
                sem_post(&traffic_lights[2]); // south

                printf("south and north can go");
                sleep(1); // cars passing
                sem_wait(&traffic_lights[0]);
                sem_wait(&traffic_lights[2]);


                sem_post(&traffic_lights[1]); // east
                sem_post(&traffic_lights[3]); // west

                printf("east and west can go");
                sleep(1);

                sem_wait(&traffic_lights[1]);
                sem_wait(&traffic_lights[3]);
        }
        return NULL;
}
```