Relatório - Simulação de Trânsito com mutex

Inicialmente, foram importadas algumas bibliotecas essenciais para o funcionamento do código. Dentre elas, tem-se a stdio.h e a stdlib.h, utilizadas por padrão para entradas e saídas na linguagem C. Além disso, tem-se pthread.h, unistd.h, termios.h e stdbool.h, utilizadas para recursos como threads e mutexs.

No método principal do código, inicialmente é realizado a limpeza do buffer do console, por meio do "system("clear")". Posteriormente, são feitas as declarações do mutex(pthread_mutex_init(&mutex, NULL)) e as threads(pthread_t c1, c2), que irão representar a movimentação de cada avenida. Após isso, um loop é iniciado para ficar sempre realizando a criação das threads declaradas, chamando seus respectivos métodos (transito1 e transito2). Por fim, é realizado o join nas threads, e o mutex é destruído(pthread_mutex_destroy(&mutex)).

Nos métodos das threads (transito 1 e transito 2), especificamente no transito1, por exemplo, inicialmente uma váriavel global booleana(t1) indica que a thread1 foi acionada, e em seguida, o mutex realiza o lock, para evitar que a outra thread possa ser acionada, evitando conflitos. Primeiramente, é checado se a thread 2 foi acionada anteriormente(t2), e se sim, o semáforo do transito 2 fica amarelo. Após ficar amarelo, ele entre no loop, que deixa o transito 2 com semáforo vermelho, e faz com que o carro do trânsito 1 se movimente. Esse loop é interrompido quando i é igual a 6, ou seja, o carro volta a sua posição de origem. Por fim, é realizado o unlock no mutex, e exit na thread. De forma análoga, é realizado no trânsito 2.