

Project 3 – Traffic Control Simulation

Course: Operating Systems

Student: Abdultawwab Tello

1. Overview

This project implements a multi-threaded traffic-control simulation using POSIX threads (pthread).

Each car is represented by a thread and follows these steps:

1. ArriveIntersection()
2. CrossIntersection()
3. ExitIntersection()

Synchronization is enforced using:

- A **head-of-line lock** per direction (only front car may proceed)
- **Quadrant locks** (modeled with a direction-sharing readers/writers design)
- A **global state lock** for coordination of waiting rules
- A **print lock** to serialize console output

The simulation prints timestamps for:

- Arrival
- Crossing
- Exiting

2. Traffic Rules Implemented

Stop Sign Rule

Each car must:

- Stop for **2 seconds**
- Wait until it reaches the front of its lane
- Yield to any earlier-arriving car from other directions that is still waiting

Quadrant Rule

GITHUB: <https://github.com/AbduTello/450-PA3/blob/main/450%20-%20PA3.pdf>

Each car acquires the required intersection quadrants based on its direction and turn:

- Right turn -> 1 quadrant
- Straight -> 2 quadrants
- Left turn -> 3 quadrants

Cars from the **same direction** may share quadrants (readers-writers style), which improves throughput.

No Deadlock

Quadrants are always acquired in fixed order (0 -> 1 -> 2 -> 3).

3. Notes

- Timing is simulated using `gettimeofday()` and `usleep()`.
- Shared state updates use mutex + condition variable.
- Printing is synchronized to avoid interleaving.
- Each car runs in its own thread and follows realistic timing constraints.

GITHUB: <https://github.com/AbduTello/450-PA3/blob/main/450%20-%20PA3.pdf>

The screenshot shows the Docker Desktop application window. The left sidebar contains navigation options: Ask Gordon (BETA), Containers, Images, Volumes, Builds, Models, MCP Toolkit (BETA), Docker Hub, Docker Scout, and Extensions. The main area displays the 'magical_ritchie' container, which is running on the '5511cf0ccf19' image. The 'Exec' tab is active, showing a terminal window with the following output:

```
Setting up libheif-plugin-aomdec:arm64 (1.17.6-1ubuntu4.1) ...
Setting up libheif:arm64 (1.17.6-1ubuntu4.1) ...
Setting up libgd3:arm64 (2.3.3-9ubuntu5) ...
Setting up libc-devtools (2.39-0ubuntu8.6) ...
Setting up libheif-plugin-libde265:arm64 (1.17.6-1ubuntu4.1) ...
Setting up libheif-plugin-aomenc:arm64 (1.17.6-1ubuntu4.1) ...
Processing triggers for libc-bin (2.39-0ubuntu8.6) ...
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
# nano tc.c
# gcc -o tc tc.c -lpthread -Wall
#
# ./tc
Traffic Control Simulation Started
=====
Time 1.1: Car 1 (^ ^) arriving
Time 2.2: Car 2 (^ ^) arriving
Time 3.1: Car 1 (^ ^) crossing
Time 3.3: Car 3 (^ <) arriving
Time 4.2: Car 2 (^ ^) crossing
Time 4.4: Car 4 (v v) arriving
Time 5.3: Car 3 (^ <) crossing
Time 5.5: Car 5 (v >) arriving
Time 6.6: Car 6 (^ ^) arriving
Time 7.1: Car 1 (^ ^) exiting
Time 7.7: Car 7 (> ^) arriving
Time 8.2: Car 2 (^ ^) exiting
Time 8.8: Car 8 (< ^) arriving
Time 10.3: Car 3 (^ <) exiting
Time 10.3: Car 4 (v v) crossing
Time 10.3: Car 5 (v >) crossing
Time 14.3: Car 4 (v v) exiting
Time 15.3: Car 5 (v >) exiting
Time 15.3: Car 6 (^ ^) crossing
Time 19.3: Car 6 (^ ^) exiting
Time 19.3: Car 7 (> ^) crossing
Time 24.3: Car 7 (> ^) exiting
Time 24.3: Car 8 (< ^) crossing
Time 27.3: Car 8 (< ^) exiting
=====
Simulation Complete
```

The bottom status bar indicates the engine is running, with RAM usage at 1.20 GB, CPU at 0.00%, and disk usage at 9.50 GB (limit 58.37 GB). A 'New version available' notification is also present.