

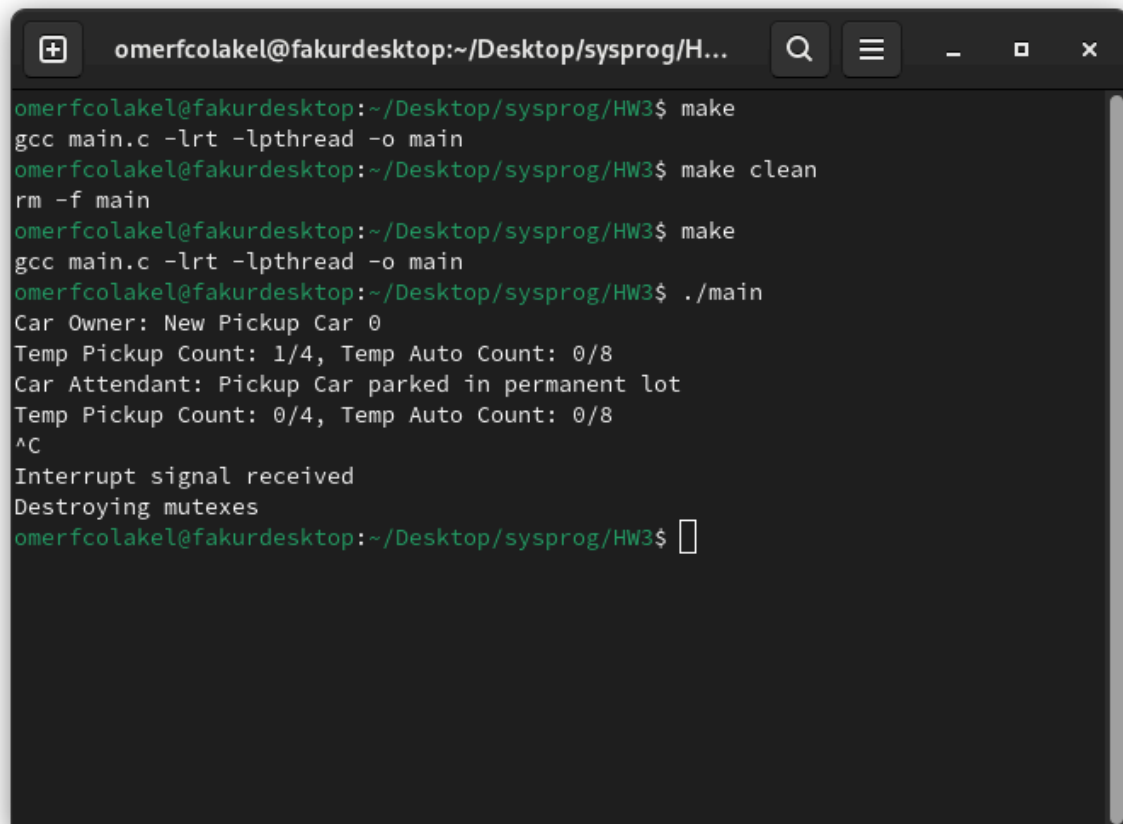
# **CSE344 Systems Programming**

## **Homework 3 Report**

Ömer Faruk Çolakel

200104004043

# How to Compile and Run



```
omerfcolakel@fakurdesktop:~/Desktop/sysprog/HW3$ make
gcc main.c -lrt -lpthread -o main
omerfcolakel@fakurdesktop:~/Desktop/sysprog/HW3$ make clean
rm -f main
omerfcolakel@fakurdesktop:~/Desktop/sysprog/HW3$ make
gcc main.c -lrt -lpthread -o main
omerfcolakel@fakurdesktop:~/Desktop/sysprog/HW3$ ./main
Car Owner: New Pickup Car 0
Temp Pickup Count: 1/4, Temp Auto Count: 0/8
Car Attendant: Pickup Car parked in permanent lot
Temp Pickup Count: 0/4, Temp Auto Count: 0/8
^C
Interrupt signal received
Destroying mutexes
omerfcolakel@fakurdesktop:~/Desktop/sysprog/HW3$
```

- make: Compiles the main.c
- make clean: Removes the object file
- ./main: Runs the compiled program

## How the Program Works

- There are an infinite number of carOwner threads and only two carAttendant threads
- The main function creates an infinite number of carOwners
- The main sleeps for at most three seconds after creating a carOwner thread. I did this so the output is readable. I removed it while testing to see if there is any problem. There was not.
- These carOwners creates car objects and determines their types randomly
- There are four mutexes: newPickup (used while adding a new pickup to the temp parking lot), newAuto (self explanatory), inChargePickup (used while transferring a pickup from temp. parking lot to parking lot) and inChargeAuto (self explanatory).
- Both thread types utilize these mutexes for synchronization
- The mutexes are destroyed after the user used the interrupt signal

# Functions, Their Purposes and How They Work

1. `void showParkingStatus() // Display current parking status`

Prints the occupancy rate of the temporary parking space.

2. `void handleExit() // Clean up mutexes`

Cleans all four mutexes.

3. `void sigINTHandler(int sig) // Signal handler for SIGINT`

The program works in a loop to create an infinite number of cars. To stop it, you can hit Ctrl + C.

4. `void *carAttendant(void *arg)`

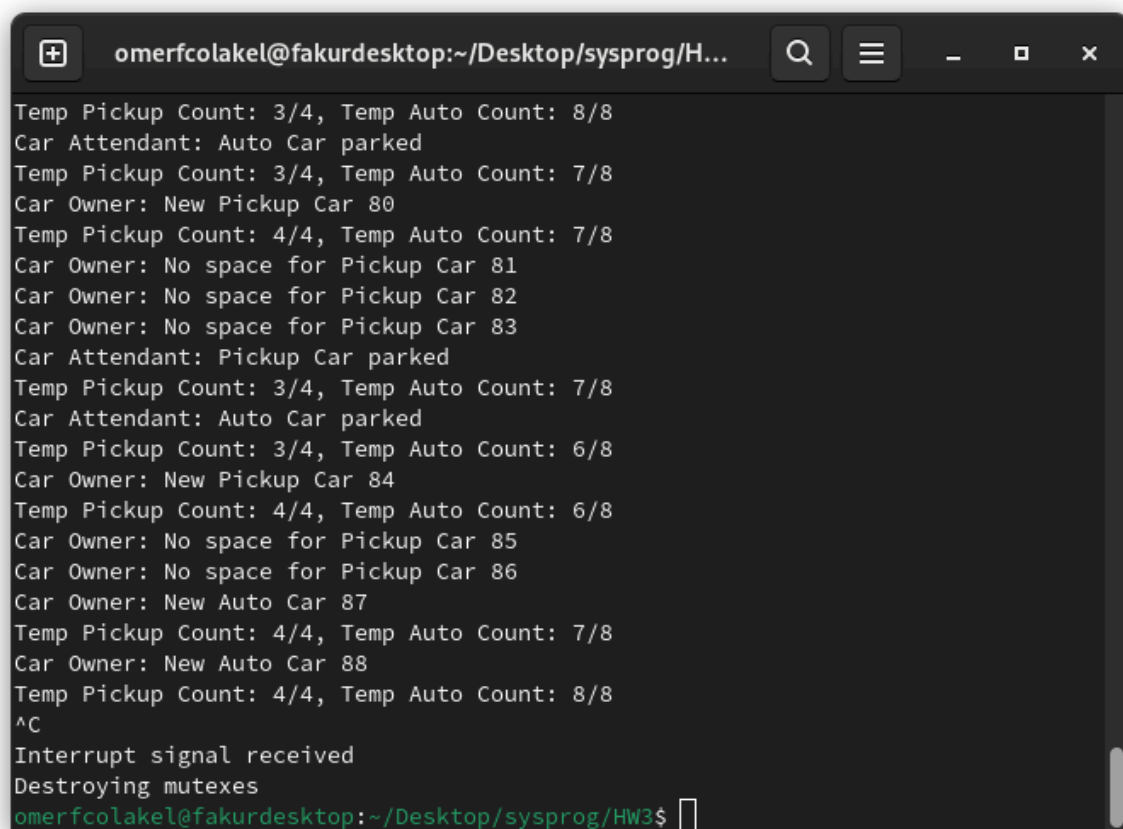
According to the type argument that it gets, it handles either the autos or pickups. Locks mutexes before using important resources and unlocks them after using.

5. `void *carOwner(void *arg)`

Every carOwner first locks the necessary mutexes according to the car type that they determined randomly. Increments either the autoCount or pickupCount after checking if the temp. the parking lot has space.

# Example Input and Outputs

- Total created car count is displayed
- Occupancy rates of both car and pickup parking spaces in the temporary parking lot
- A car parked in the temp. parking lot
- A car removed from the temp. parking lot and parked to the parking lot
- Car owner not being able to park due to not having space
- New car owners can park after there is space
- Exited successfully

A terminal window with a dark background and light gray text. The window title bar shows the user 'omerfcolakel' and the path '~/Desktop/sysprog/H...'. The output text shows the state of a parking simulation, including counts for pickup and auto cars, and messages for car owners and attendants. The program ends with an interrupt signal and the destruction of mutexes.

```
omerfcolakel@fakurdesktop:~/Desktop/sysprog/H...
Temp Pickup Count: 3/4, Temp Auto Count: 8/8
Car Attendant: Auto Car parked
Temp Pickup Count: 3/4, Temp Auto Count: 7/8
Car Owner: New Pickup Car 80
Temp Pickup Count: 4/4, Temp Auto Count: 7/8
Car Owner: No space for Pickup Car 81
Car Owner: No space for Pickup Car 82
Car Owner: No space for Pickup Car 83
Car Attendant: Pickup Car parked
Temp Pickup Count: 3/4, Temp Auto Count: 7/8
Car Attendant: Auto Car parked
Temp Pickup Count: 3/4, Temp Auto Count: 6/8
Car Owner: New Pickup Car 84
Temp Pickup Count: 4/4, Temp Auto Count: 6/8
Car Owner: No space for Pickup Car 85
Car Owner: No space for Pickup Car 86
Car Owner: New Auto Car 87
Temp Pickup Count: 4/4, Temp Auto Count: 7/8
Car Owner: New Auto Car 88
Temp Pickup Count: 4/4, Temp Auto Count: 8/8
^C
Interrupt signal received
Destroying mutexes
omerfcolakel@fakurdesktop:~/Desktop/sysprog/HW3$
```