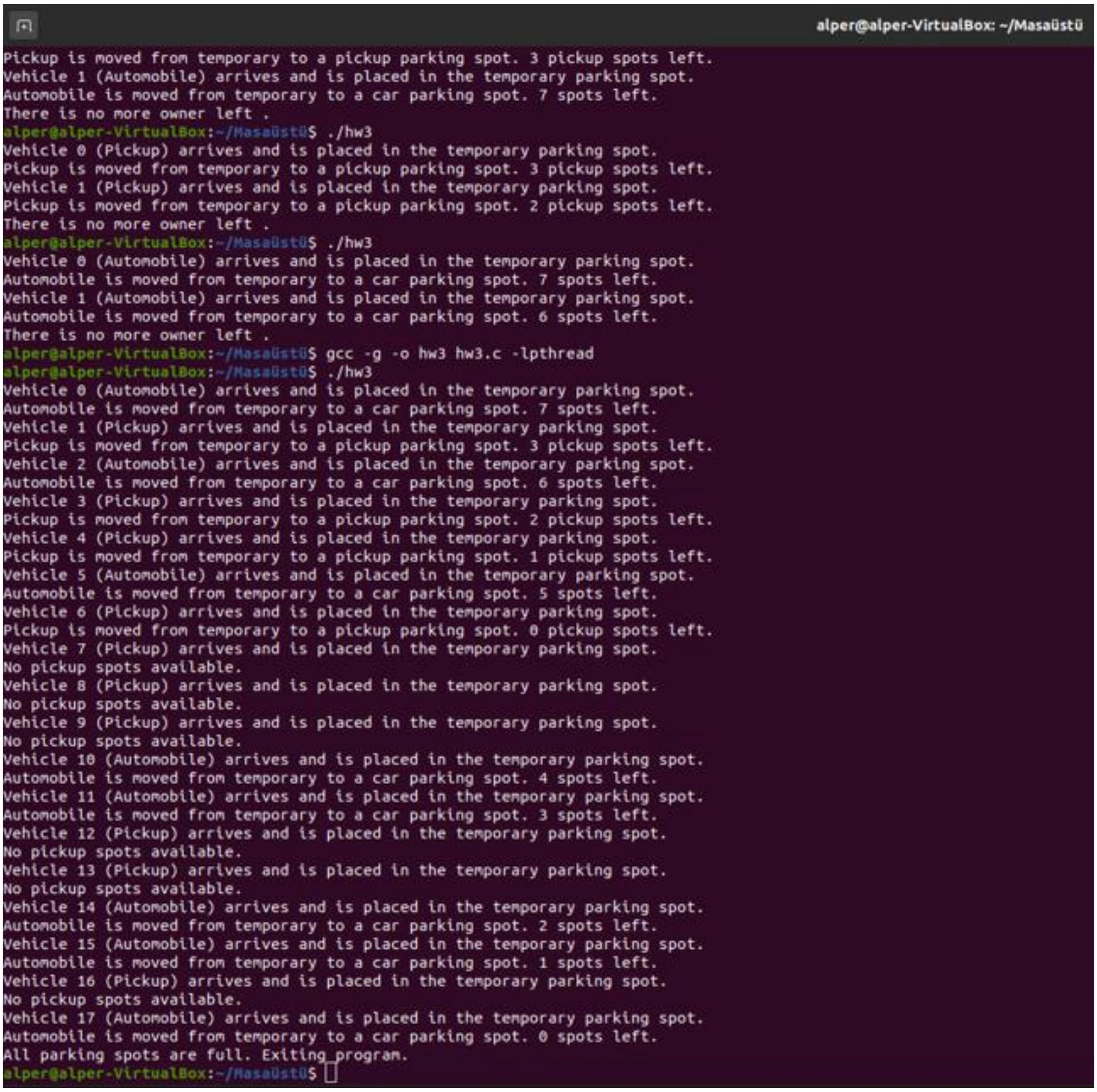
# Gebze Technical University
# Department of Computer Engineering
# CSE344 - Spring 2024
# Homework 3
# AHMET ALPER UZUNTEPE
# 1901042669



```
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
There is no more owner left .
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 2 pickup spots left.
There is no more owner left .
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
Vehicle 1 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 6 spots left.
There is no more owner left .
alper@alper-VirtualBox:~/Masaüstü$ gcc -g -o hw3 hw3.c -lpthread
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
Vehicle 1 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 2 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 6 spots left.
Vehicle 3 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 2 pickup spots left.
Vehicle 4 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 1 pickup spots left.
Vehicle 5 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 5 spots left.
Vehicle 6 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 0 pickup spots left.
Vehicle 7 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 8 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 9 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 10 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 4 spots left.
Vehicle 11 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 3 spots left.
Vehicle 12 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 13 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 14 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 2 spots left.
Vehicle 15 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 1 spots left.
Vehicle 16 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 17 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 0 spots left.
All parking spots are full. Exiting program.
alper@alper-VirtualBox:~/Masaüstü$
```

The parking management system two types of vehicles: automobiles and pickups. It manages available parking spots for each vehicle type and provides temporary spots to coordinate the parking process.

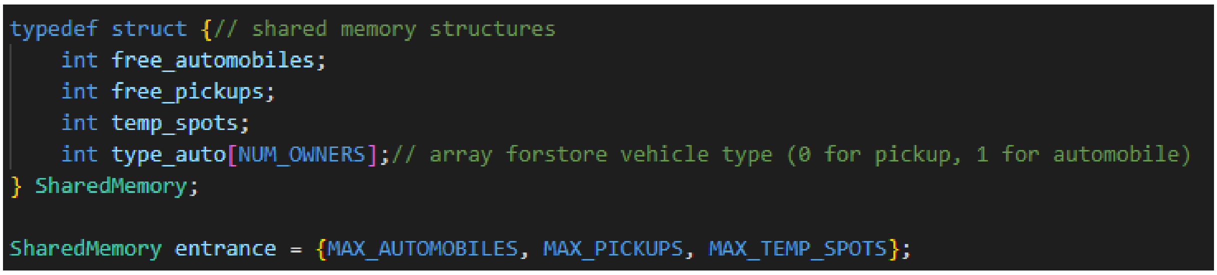
I used semaphore , shared memory and threads for synchronize and coordinate the activities of car owners and attendants.

- Separate parking areas for automobiles and pickups.
- Temporary spots for vehicles waiting to be parked.
- Semaphores to manage access to parking spots and synchronization between threads.
- Threads for car owners and attendants to simulate the parking process.

**HOW I DİD ?**

- First thing ı need to explain. I understand the template park spot like a valet spot , and ı use it like that .I take the car owner to template and ı give the car to attendants.
- It is like that Vehicle→Temp Spot →Main Spot(If there is a spot for pickups or automobiles.If not clear the temp and continue with next car) .
- When it is full →Exit.
- I used 4 threads 2 for car owners ,2 for car attendants. I initilaze them at the begining.
- I didn’t use mutex.

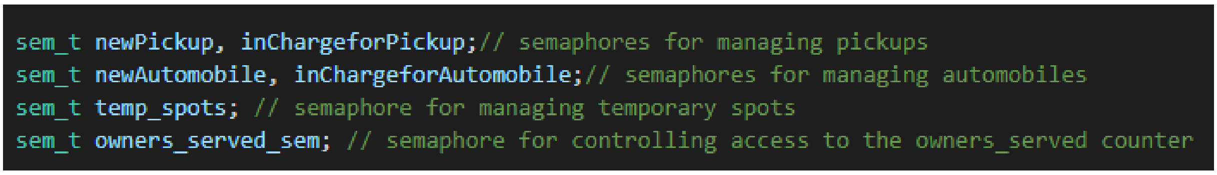
## System Design



```
typedef struct {// shared memory structures
    int free_automobiles;
    int free_pickups;
    int temp_spots;
    int type_auto[NUM_OWNERS];// array forstore vehicle type (0 for pickup, 1 for automobile)
} SharedMemory;

SharedMemory entrance = {MAX_AUTOMOBILES, MAX_PICKUPS, MAX_TEMP_SPOTS};
```

### 1. Shared Memory Structures

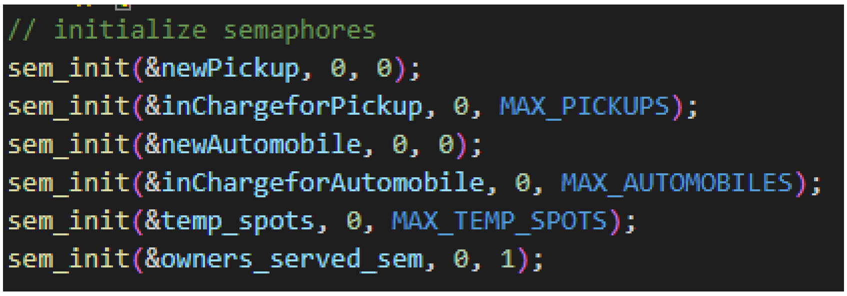
- free_automobiles: Number of available automobile spots
- free_pickups: Number of available pickup spots
- temp_spots: Number of available temporary spots
- type_auto: Array to store vehicle types (0 for pickup, 1 for automobile)



```
sem_t newPickup, inChargeforPickup;// semaphores for managing pickups
sem_t newAutomobile, inChargeforAutomobile;// semaphores for managing automobiles
sem_t temp_spots; // semaphore for managing temporary spots
sem_t owners_served_sem; // semaphore for controlling access to the owners_served counter
```

### 2. Semaphores

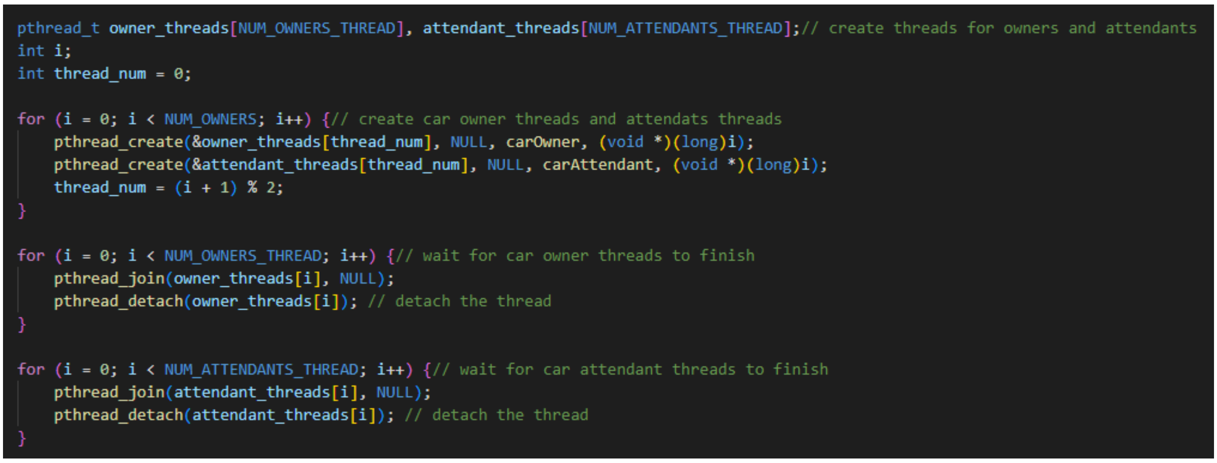
- newPickup, inChargeforPickup: For managing pickups
- newAutomobile, inChargeforAutomobile: For managing automobiles
- temp_spots: For managing temporary spots
- owners_served_sem: For controlling access to the owners_served counter



```
// initialize semaphores
sem_init(&newPickup, 0, 0);
sem_init(&inChargeforPickup, 0, MAX_PICKUPS);
sem_init(&newAutomobile, 0, 0);
sem_init(&inChargeforAutomobile, 0, MAX_AUTOMOBILES);
sem_init(&temp_spots, 0, MAX_TEMP_SPOTS);
sem_init(&owners_served_sem, 0, 1);
```

- sem_init(&newPickup, 0, 0) : I used to signal when a new pickup has arrived.
- sem_init(&inChargeforPickup, 0, MAX_PICKUPS) : Controls the availability of pickup parking spots.
- sem_init(&newAutomobile, 0, 0) : I used to to signal when a new automobile has arrived.

- sem_init(&inChargeforAutomobile, 0, MAX_AUTOMOBILES) : Controls the availability of automobile parking spots.
- sem_init(&temp_spots, 0, MAX_TEMP_SPOTS) : Manages the temporary spots for vehicles waiting to be parked.
- sem_init(&owners_served_sem, 0, 1) : Ensures mutual exclusion when updating the owners_served counter.
- Sem_init for initilaziation.
- I used sem_destroy for clean them.
- I used sem_post for unblocking.



```
pthread_t owner_threads[NUM_OWNERS_THREAD], attendant_threads[NUM_ATTENDANTS_THREAD];// create threads for owners and attendants
int i;
int thread_num = 0;

for (i = 0; i < NUM_OWNERS; i++) {// create car owner threads and attendats threads
    pthread_create(&owner_threads[thread_num], NULL, carOwner, (void *)(long)i);
    pthread_create(&attendant_threads[thread_num], NULL, carAttendant, (void *)(long)i);
    thread_num = (i + 1) % 2;
}

for (i = 0; i < NUM_OWNERS_THREAD; i++) {// wait for car owner threads to finish
    pthread_join(owner_threads[i], NULL);
    pthread_detach(owner_threads[i]); // detach the thread
}

for (i = 0; i < NUM_ATTENDANTS_THREAD; i++) {// wait for car attendant threads to finish
    pthread_join(attendant_threads[i], NULL);
    pthread_detach(attendant_threads[i]); // detach the thread
}
```

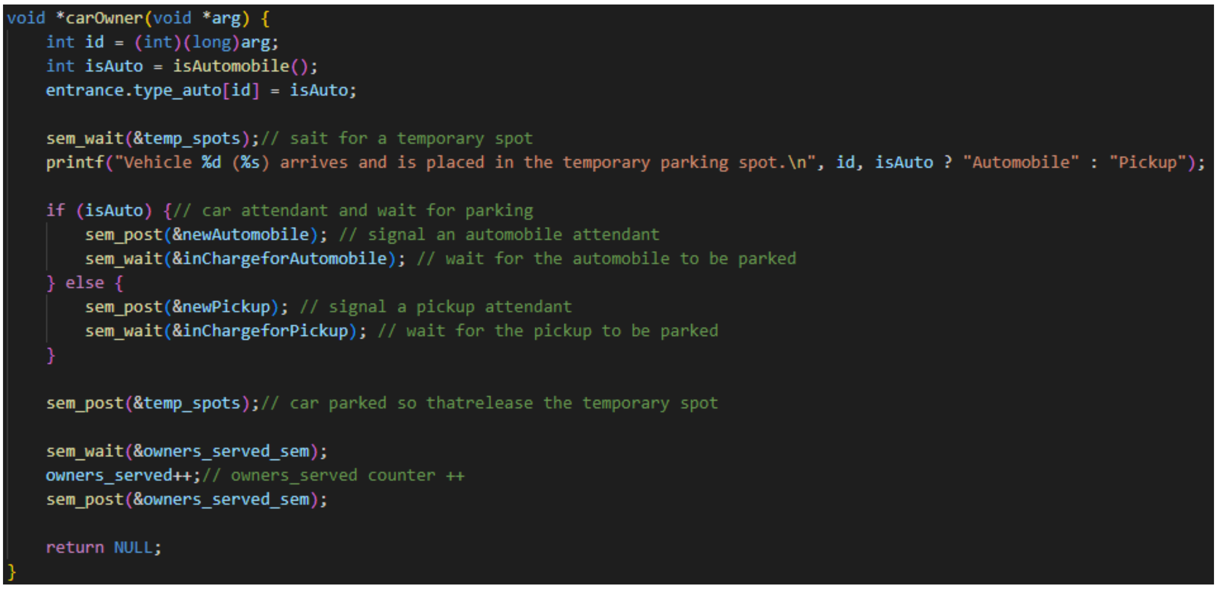
## 3. Threads

- owner_threads: Threads representing car owners
- attendant_threads: Threads representing parking attendants.There should be 2 attendant and ı did it like even odd numbers .
- pthread_t owner_threads[NUM_OWNERS_THREAD], attendant_threads[NUM_ATTENDANTS_THREAD] : Declares arrays of thread identifiers for owner threads and attendant threads.
- int thread_num = 0 : Initializes thread_num to 0, which is used to alternate between thread arrays.
- for (i = 0; i < NUM_OWNERS; i++) {: Loops NUM_OWNERS times to create threads.
- pthread_create(&owner_threads[thread_num], NULL, carOwner, (void *)(long)i) : Creates an owner thread that runs the carOwner function. The thread index i is passed as an argument.
- pthread_create(&attendant_threads[thread_num], NULL, carAttendant, (void *)(long)i) : Creates an attendant thread that runs the carAttendant function. The thread index i is passed as an argument.
- thread_num = (i + 1) % 2 : Alternates thread_num between 0 and 1 to alternate between the two thread arrays.

- for (i = 0; i < (Tread num); i++) {: Loops to wait for attendant threads to finish.
- pthread_join(attendant_threads[i], NULL) : Waits for the i-th attendant thread to terminate.
- pthread_detach(attendant_threads[i]) : Detaches the i-th attendant thread, indicating that the thread's resources can be reclaimed when it terminates.

## Car Owner Function

The carOwner function organize the arrival of a car owner. The vehicle type is randomly determined, and the vehicle is placed in a temporary spot before being moved to a designated parking spot by an attendant.

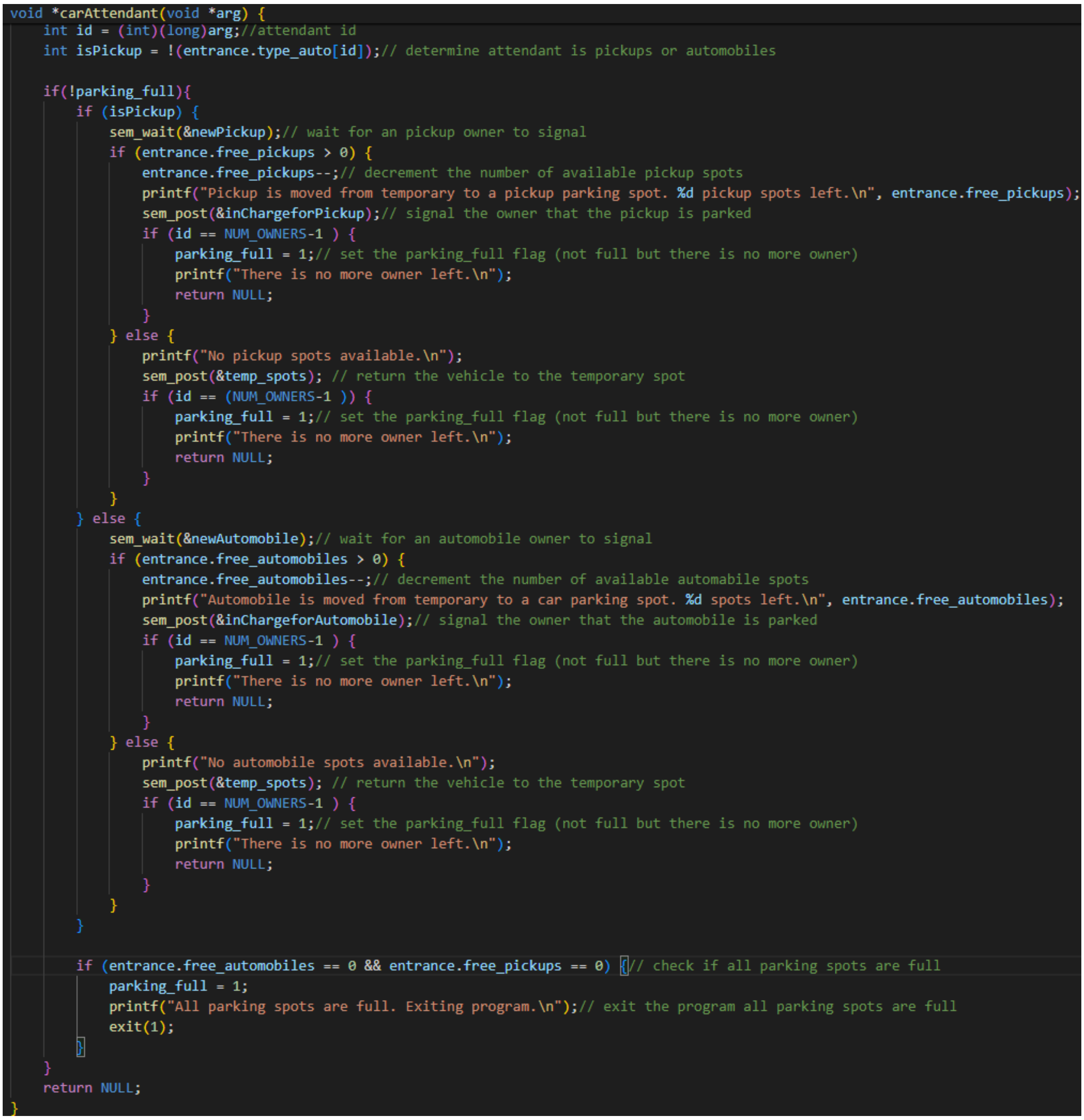


```c
void *carOwner(void *arg) {
    int id = (int)(long)arg;
    int isAuto = isAutomobile();
    entrance.type_auto[id] = isAuto;

    sem_wait(&temp_spots);// sait for a temporary spot
    printf("Vehicle %d (%s) arrives and is placed in the temporary parking spot.\n", id, isAuto ? "Automobile" : "Pickup");

    if (isAuto) {// car attendant and wait for parking
        sem_post(&newAutomobile); // signal an automobile attendant
        sem_wait(&inChargeforAutomobile); // wait for the automobile to be parked
    } else {
        sem_post(&newPickup); // signal a pickup attendant
        sem_wait(&inChargeforPickup); // wait for the pickup to be parked
    }

    sem_post(&temp_spots);// car parked so thatrelease the temporary spot

    sem_wait(&owners_served_sem);
    owners_served++;// owners_served counter ++
    sem_post(&owners_served_sem);

    return NULL;
}
```

- sem_wait(&temp_spots) : Decrements the temp_spots semaphore. If no temporary spots are available, the thread will block until one is freed.
- if (isAuto) {: Checks if the vehicle is an automobile.
- sem_post(&newAutomobile) : Increments the newAutomobile semaphore to signal an automobile attendant that a new automobile needs to be parked.
- sem_wait(&inChargeforAutomobile) : Decrements the inChargeforAutomobile semaphore, blocking the owner thread until the automobile is parked.
- else {: Executes if the vehicle is a pickup.
- sem_post(&newPickup) : Increments the newPickup semaphore to signal a pickup attendant that a new pickup needs to be parked.
- sem_wait(&inChargeforPickup) : Decrements the inChargeforPickup semaphore, blocking the owner thread until the pickup is parked.
- sem_post(&temp_spots) : Increments the temp_spots semaphore to release the temporary spot, making it available for another vehicle.
- sem_wait(&owners_served_sem) : Decrements the owners_served_sem semaphore to gain exclusive access to the owners_served counter.
- owners_served++ : Increments the owners_served counter to record that one more owner has been served.
- sem_post(&owners_served_sem) : Increments the owners_served_sem semaphore to release exclusive access to the owners_served counter.

## The carAttendant function

Organize the actions of an attendant. The attendant waits for a vehicle signal and then attempts to park the vehicle in the appropriate spot. If no spots are available, the vehicle is returned to a temporary spot.



```c
void *carAttendant(void *arg) {
    int id = (int)(long)arg;//attendant id
    int isPickup = !(entrance.type_auto[id]);// determine attendant is pickups or automobiles

    if(!parking_full){
        if (isPickup) {
            sem_wait(&newPickup);// wait for an pickup owner to signal
            if (entrance.free_pickups > 0) {
                entrance.free_pickups--;// decrement the number of available pickup spots
                printf("Pickup is moved from temporary to a pickup parking spot. %d pickup spots left.\n", entrance.free_pickups);
                sem_post(&inChargeforPickup);// signal the owner that the pickup is parked
                if (id == NUM_OWNERS-1 ) {
                    parking_full = 1;// set the parking_full flag (not full but there is no more owner)
                    printf("There is no more owner left.\n");
                    return NULL;
                }
            } else {
                printf("No pickup spots available.\n");
                sem_post(&temp_spots); // return the vehicle to the temporary spot
                if (id == (NUM_OWNERS-1 )) {
                    parking_full = 1;// set the parking_full flag (not full but there is no more owner)
                    printf("There is no more owner left.\n");
                    return NULL;
                }
            }
        } else {
            sem_wait(&newAutomobile);// wait for an automobile owner to signal
            if (entrance.free_automobiles > 0) {
                entrance.free_automobiles--;// decrement the number of available automabile spots
                printf("Automobile is moved from temporary to a car parking spot. %d spots left.\n", entrance.free_automobiles);
                sem_post(&inChargeforAutomobile);// signal the owner that the automobile is parked
                if (id == NUM_OWNERS-1 ) {
                    parking_full = 1;// set the parking_full flag (not full but there is no more owner)
                    printf("There is no more owner left.\n");
                    return NULL;
                }
            } else {
                printf("No automobile spots available.\n");
                sem_post(&temp_spots); // return the vehicle to the temporary spot
                if (id == NUM_OWNERS-1 ) {
                    parking_full = 1;// set the parking_full flag (not full but there is no more owner)
                    printf("There is no more owner left.\n");
                    return NULL;
                }
            }
        }

        if (entrance.free_automobiles == 0 && entrance.free_pickups == 0) {// check if all parking spots are full
            parking_full = 1;
            printf("All parking spots are full. Exiting program.\n");// exit the program all parking spots are full
            exit(1);
        }
    }
    return NULL;
}
```

# Process of Pickups

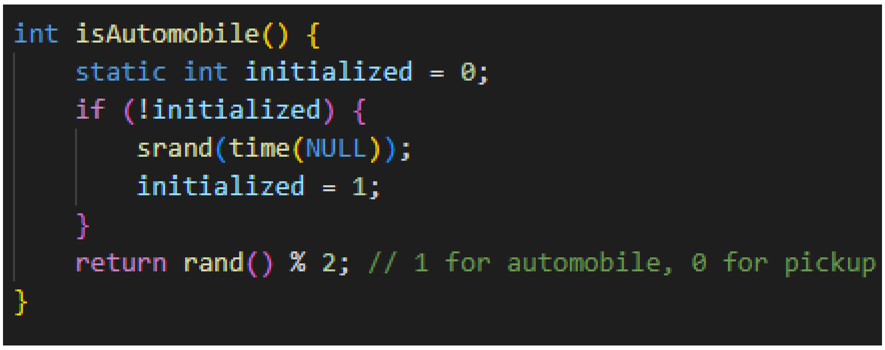
- if (!parking_full): Proceeds only if the parking is not full.
- sem_wait(&newPickup) : Waits for a signal from a pickup owner.
- if (entrance.free_pickups > 0) : Checks if there are available pickup spots.
- entrance.free_pickups-- : Decrements the count of available pickup spots.
- sem_post(&inChargeforPickup) : Signals the owner that the pickup is parked.
- if (id == NUM_OWNERS-1): Checks if this is the last owner.
- parking_full = 1 : Sets the parking full flag.
- printf("There is no more owner left.\n") : Prints a message indicating no more owners are left.
- return NULL : Exits the function.
- printf("No pickup spots available.\n") : Prints a message if no pickup spots are available.
- sem_post(&temp_spots) : Returns the vehicle to the temporary spot if no parking is available.

# Process Of Automobiles

- sem_wait(&newAutomobile) : Waits for a signal from an automobile owner.
- if (entrance.free_automobiles > 0): Checks if there are available automobile spots.
- entrance.free_automobiles-- : Decrements the count of available automobile spots.
- sem_post(&inChargeforAutomobile) : Signals the owner that the automobile is parked.
- if (id == NUM_OWNERS-1): Checks if this is the last owner.
- parking_full = 1 : Sets the parking full flag.
- printf("There is no more owner left.\n") : Prints a message indicating no more owners are left.
- return NULL : Exits the function.
- printf("No automobile spots available.\n") : Prints a message if no automobile spots are available.
- sem_post(&temp_spots) : Returns the vehicle to the temporary spot if no parking is available.

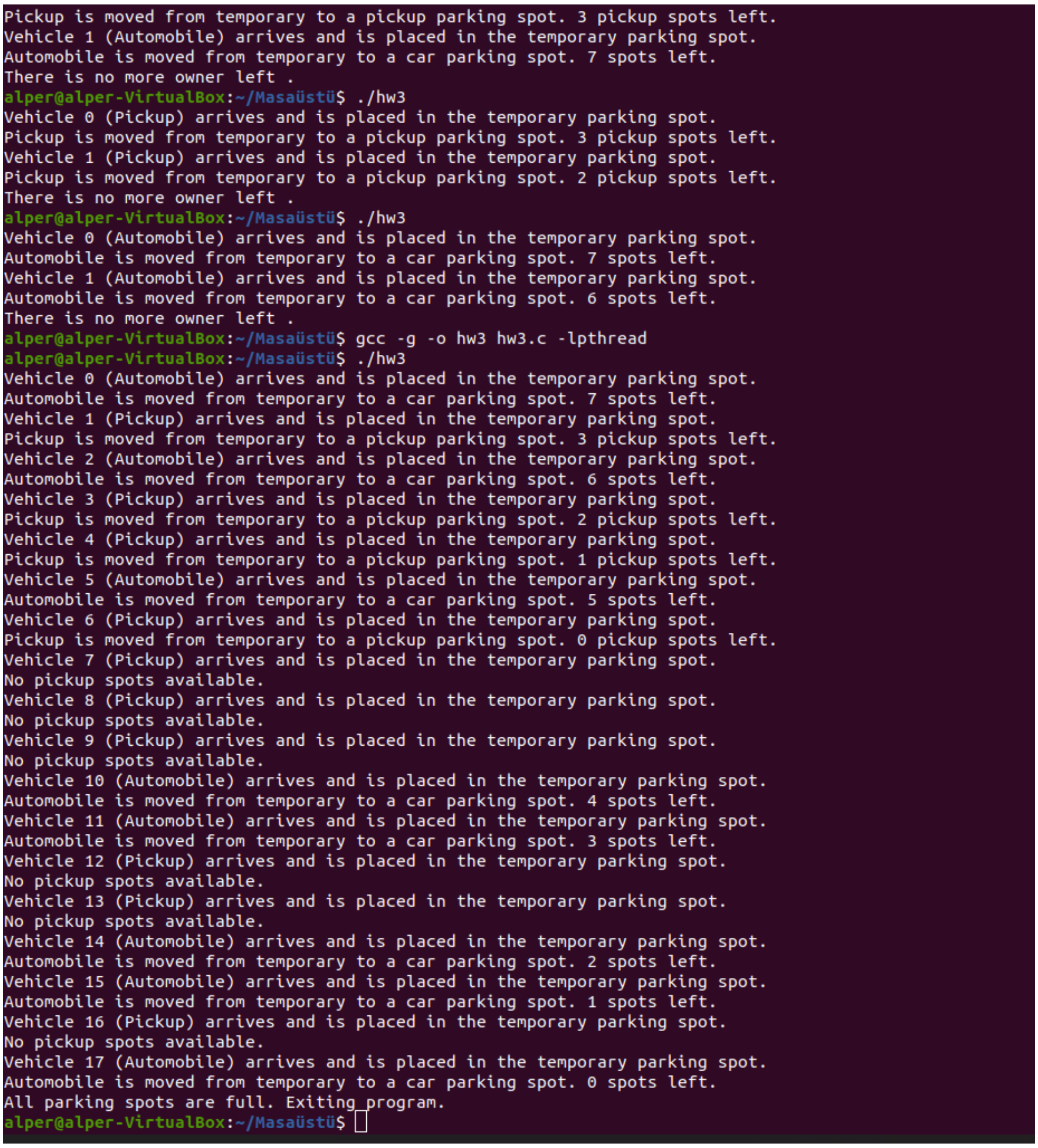
## Random Vehicle Type

The isAutomobile function randomly determines whether a vehicle is an automobile or a pickup.

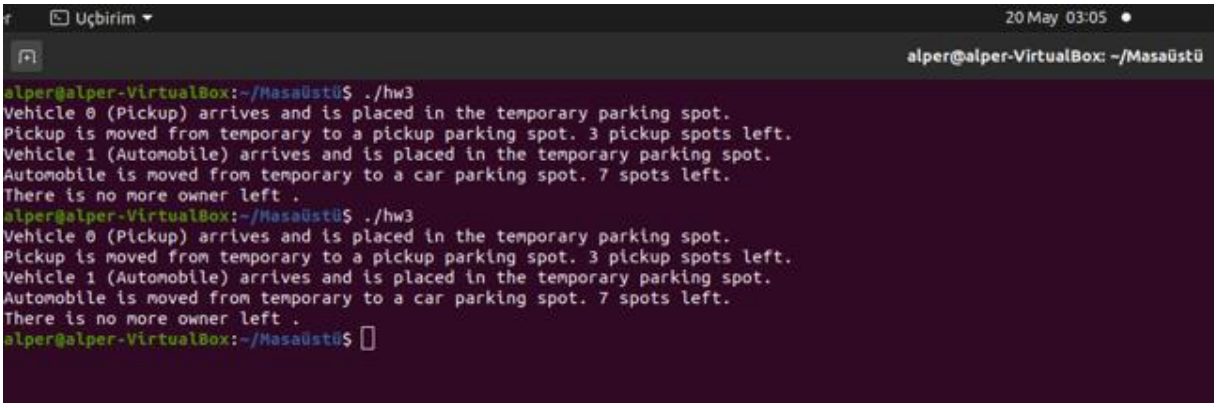


```c
int isAutomobile() {
    static int initialized = 0;
    if (!initialized) {
        srand(time(NULL));
        initialized = 1;
    }
    return rand() % 2; // 1 for automobile, 0 for pickup
}
```

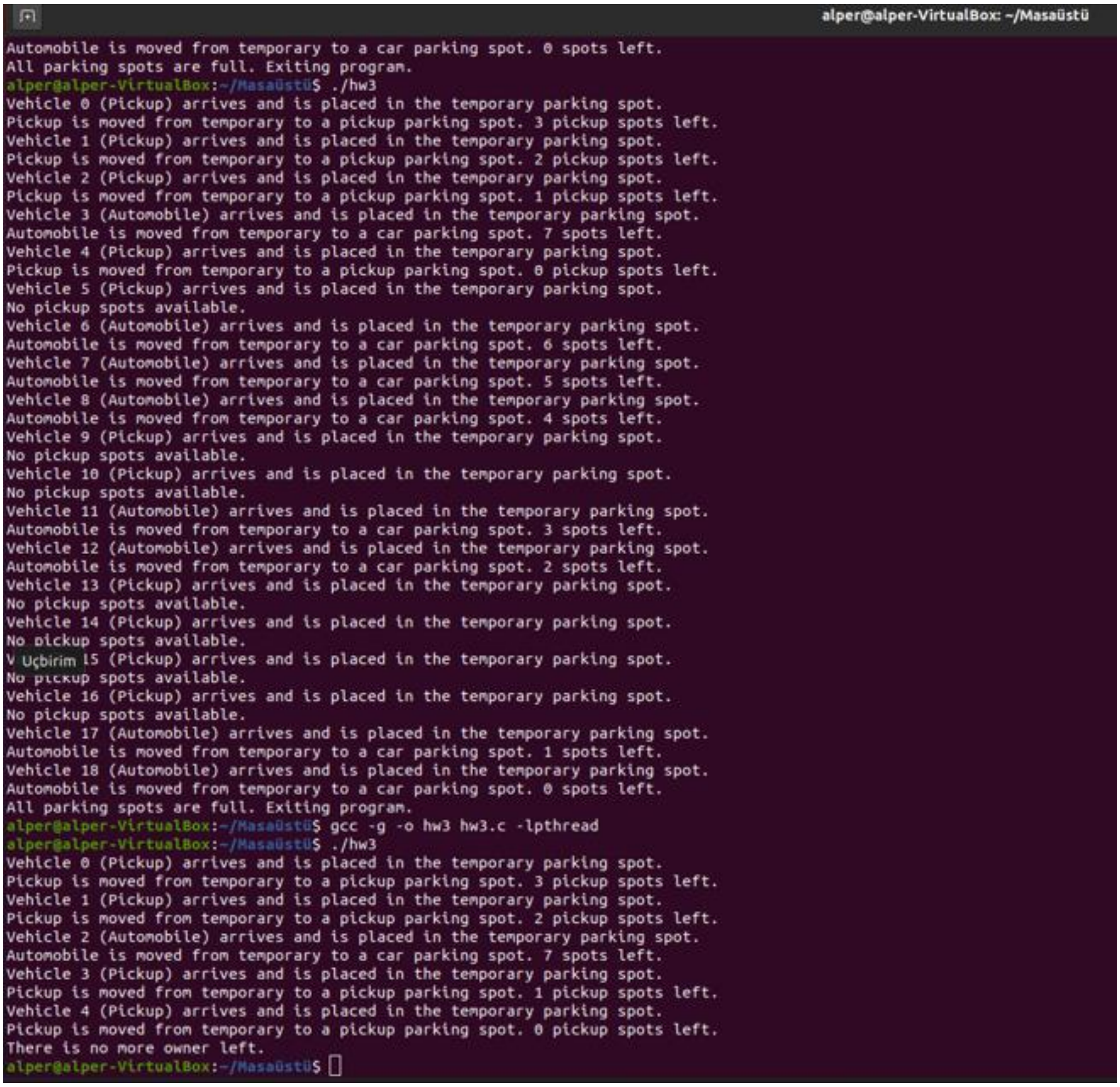
## OUTPUTS



```
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
There is no more owner left .
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 2 pickup spots left.
There is no more owner left .
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
Vehicle 1 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 6 spots left.
There is no more owner left .
alper@alper-VirtualBox:~/Masaüstü$ gcc -g -o hw3 hw3.c -lpthread
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
Vehicle 1 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 2 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 6 spots left.
Vehicle 3 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 2 pickup spots left.
Vehicle 4 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 1 pickup spots left.
Vehicle 5 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 5 spots left.
Vehicle 6 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 0 pickup spots left.
Vehicle 7 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 8 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 9 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 10 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 4 spots left.
Vehicle 11 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 3 spots left.
Vehicle 12 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 13 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 14 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 2 spots left.
Vehicle 15 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 1 spots left.
Vehicle 16 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 17 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 0 spots left.
All parking spots are full. Exiting program.
alper@alper-VirtualBox:~/Masaüstü$
```

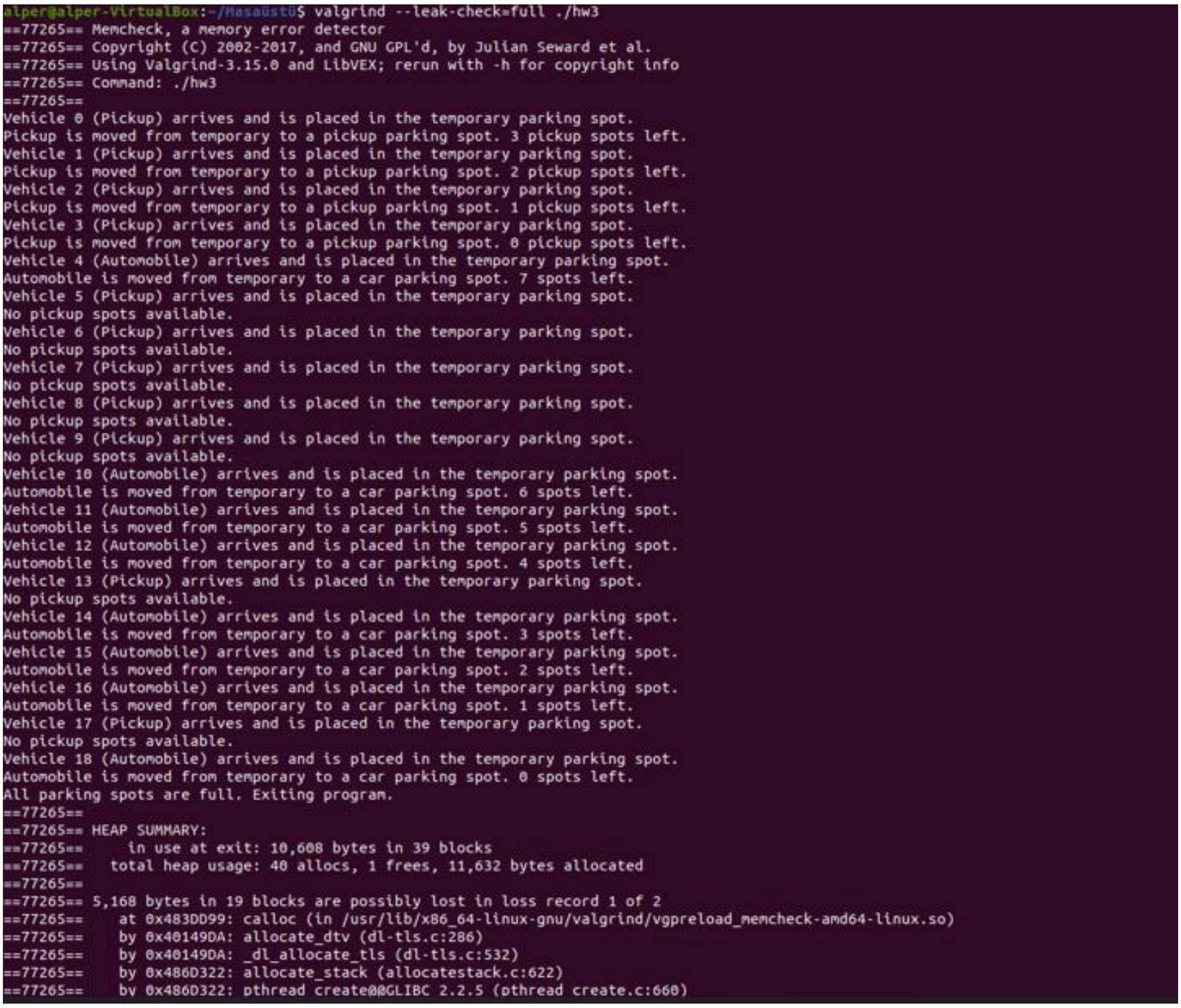


```
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
There is no more owner left .
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
There is no more owner left .
alper@alper-VirtualBox:~/Masaüstü$ 
```

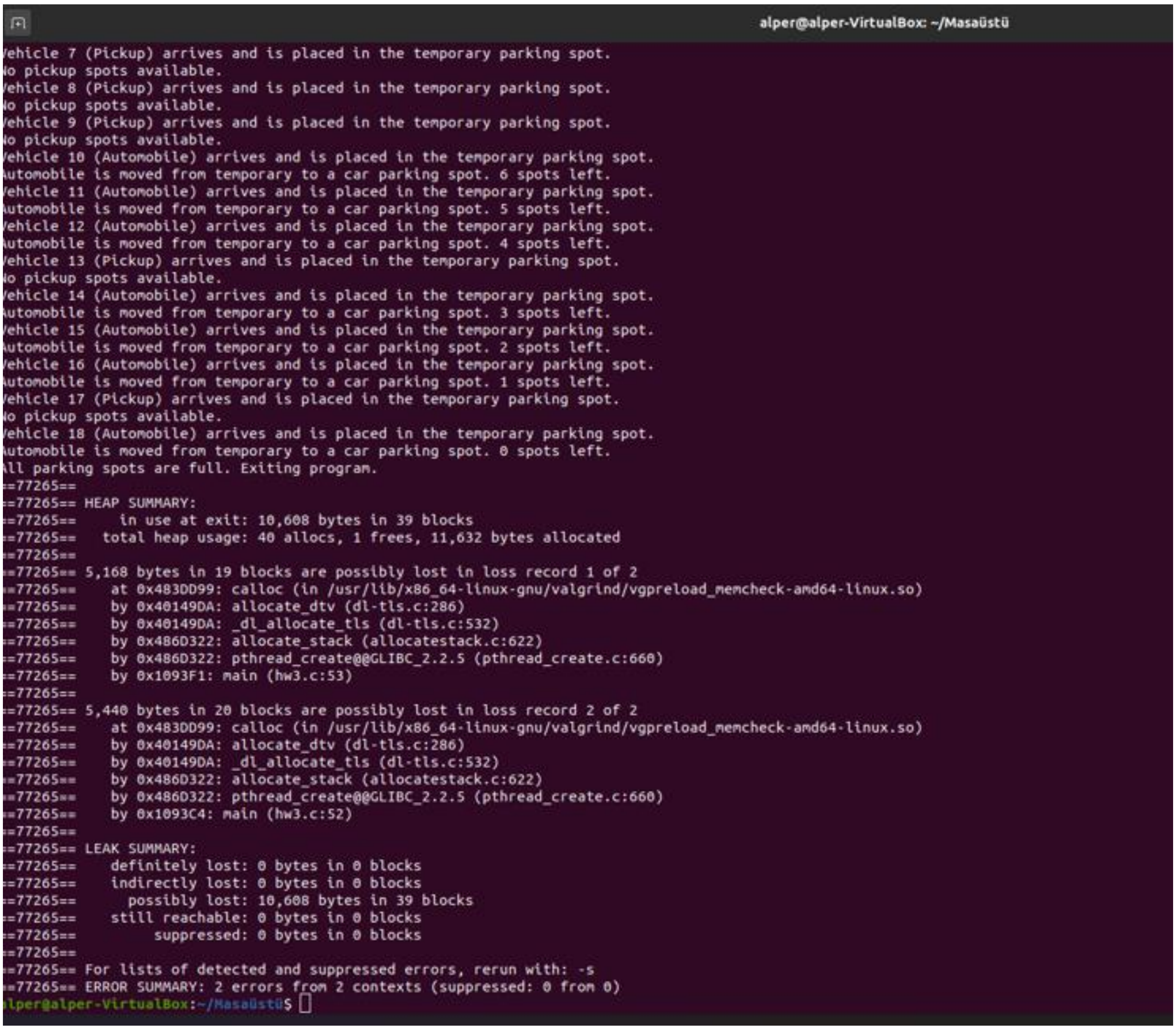


```
Automobile is moved from temporary to a car parking spot. 0 spots left.
All parking spots are full. Exiting program.
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 2 pickup spots left.
Vehicle 2 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 1 pickup spots left.
Vehicle 3 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
Vehicle 4 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 0 pickup spots left.
Vehicle 5 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 6 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 6 spots left.
Vehicle 7 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 5 spots left.
Vehicle 8 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 4 spots left.
Vehicle 9 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 10 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 11 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 3 spots left.
Vehicle 12 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 2 spots left.
Vehicle 13 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 14 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
V 15 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 16 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 17 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 1 spots left.
Vehicle 18 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 0 spots left.
All parking spots are full. Exiting program.
alper@alper-VirtualBox:~/Masaüstü$ gcc -g -o hw3 hw3.c -lpthread
alper@alper-VirtualBox:~/Masaüstü$ ./hw3
Vehicle 0 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 2 pickup spots left.
Vehicle 2 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
Vehicle 3 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 1 pickup spots left.
Vehicle 4 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 0 pickup spots left.
There is no more owner left.
alper@alper-VirtualBox:~/Masaüstü$ 
```

# MEMORY LEAK

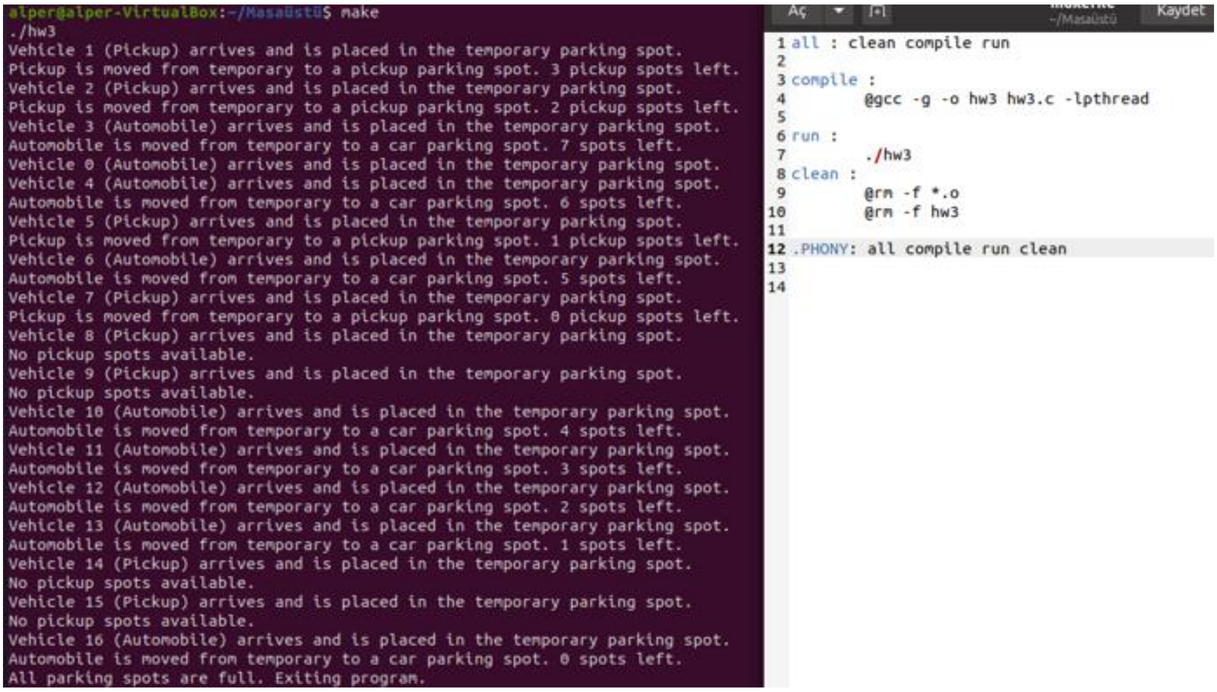


```
alper@alper-VirtualBox:~/Masaüstü$ valgrind --leak-check=full ./hw3
==77265== Memcheck, a memory error detector
==77265== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==77265== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==77265== Command: ./hw3
==77265==
Vehicle 0 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 1 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 2 pickup spots left.
Vehicle 2 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 1 pickup spots left.
Vehicle 3 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 0 pickup spots left.
Vehicle 4 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
Vehicle 5 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 6 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 7 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 8 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 9 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 10 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 6 spots left.
Vehicle 11 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 5 spots left.
Vehicle 12 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 4 spots left.
Vehicle 13 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 14 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 3 spots left.
Vehicle 15 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 2 spots left.
Vehicle 16 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 1 spots left.
Vehicle 17 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 18 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 0 spots left.
All parking spots are full. Exiting program.
==77265==
==77265== HEAP SUMMARY:
==77265==     in use at exit: 10,608 bytes in 39 blocks
==77265==   total heap usage: 40 allocs, 1 frees, 11,632 bytes allocated
==77265==
==77265== 5,168 bytes in 19 blocks are possibly lost in loss record 1 of 2
==77265==    at 0x483DD99: calloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==77265==    by 0x40149DA: allocate_dtv (dl-tls.c:286)
==77265==    by 0x40149DA: _dl_allocate_tls (dl-tls.c:532)
==77265==    by 0x486D322: allocate_stack (allocatestack.c:622)
==77265==    by 0x486D322: pthread create@@GLIBC 2.2.5 (pthread create.c:660)
```



```
ehicle 7 (Pickup) arrives and is placed in the temporary parking spot.
o pickup spots available.
ehicle 8 (Pickup) arrives and is placed in the temporary parking spot.
o pickup spots available.
ehicle 9 (Pickup) arrives and is placed in the temporary parking spot.
o pickup spots available.
ehicle 10 (Automobile) arrives and is placed in the temporary parking spot.
utomobile is moved from temporary to a car parking spot. 6 spots left.
ehicle 11 (Automobile) arrives and is placed in the temporary parking spot.
utomobile is moved from temporary to a car parking spot. 5 spots left.
ehicle 12 (Automobile) arrives and is placed in the temporary parking spot.
utomobile is moved from temporary to a car parking spot. 4 spots left.
ehicle 13 (Pickup) arrives and is placed in the temporary parking spot.
o pickup spots available.
ehicle 14 (Automobile) arrives and is placed in the temporary parking spot.
utomobile is moved from temporary to a car parking spot. 3 spots left.
ehicle 15 (Automobile) arrives and is placed in the temporary parking spot.
utomobile is moved from temporary to a car parking spot. 2 spots left.
ehicle 16 (Automobile) arrives and is placed in the temporary parking spot.
utomobile is moved from temporary to a car parking spot. 1 spots left.
ehicle 17 (Pickup) arrives and is placed in the temporary parking spot.
o pickup spots available.
ehicle 18 (Automobile) arrives and is placed in the temporary parking spot.
utomobile is moved from temporary to a car parking spot. 0 spots left.
ll parking spots are full. Exiting program.
=77265==
=77265== HEAP SUMMARY:
=77265==     in use at exit: 10,608 bytes in 39 blocks
=77265==   total heap usage: 40 allocs, 1 frees, 11,632 bytes allocated
=77265==
=77265== 5,168 bytes in 19 blocks are possibly lost in loss record 1 of 2
=77265==    at 0x483DD99: calloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
=77265==    by 0x40149DA: allocate_dtv (dl-tls.c:286)
=77265==    by 0x40149DA: _dl_allocate_tls (dl-tls.c:532)
=77265==    by 0x486D322: allocate_stack (allocatestack.c:622)
=77265==    by 0x486D322: pthread_create@@GLIBC_2.2.5 (pthread_create.c:660)
=77265==    by 0x1093F1: main (hw3.c:53)
=77265==
=77265== 5,440 bytes in 20 blocks are possibly lost in loss record 2 of 2
=77265==    at 0x483DD99: calloc (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
=77265==    by 0x40149DA: allocate_dtv (dl-tls.c:286)
=77265==    by 0x40149DA: _dl_allocate_tls (dl-tls.c:532)
=77265==    by 0x486D322: allocate_stack (allocatestack.c:622)
=77265==    by 0x486D322: pthread_create@@GLIBC_2.2.5 (pthread_create.c:660)
=77265==    by 0x1093C4: main (hw3.c:52)
=77265==
=77265== LEAK SUMMARY:
=77265==    definitely lost: 0 bytes in 0 blocks
=77265==    indirectly lost: 0 bytes in 0 blocks
=77265==      possibly lost: 10,608 bytes in 39 blocks
=77265==    still reachable: 0 bytes in 0 blocks
=77265==         suppressed: 0 bytes in 0 blocks
=77265==
=77265== For lists of detected and suppressed errors, rerun with: -s
=77265== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
lper@alper-VirtualBox:~/Masaüstü$
```

# MAKEFILE



```
alper@alper-VirtualBox:~/Masaüstü$ make
./hw3
Vehicle 1 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 3 pickup spots left.
Vehicle 2 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 2 pickup spots left.
Vehicle 3 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 7 spots left.
Vehicle 0 (Automobile) arrives and is placed in the temporary parking spot.
Vehicle 4 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 6 spots left.
Vehicle 5 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 1 pickup spots left.
Vehicle 6 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 5 spots left.
Vehicle 7 (Pickup) arrives and is placed in the temporary parking spot.
Pickup is moved from temporary to a pickup parking spot. 0 pickup spots left.
Vehicle 8 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 9 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 10 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 4 spots left.
Vehicle 11 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 3 spots left.
Vehicle 12 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 2 spots left.
Vehicle 13 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 1 spots left.
Vehicle 14 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 15 (Pickup) arrives and is placed in the temporary parking spot.
No pickup spots available.
Vehicle 16 (Automobile) arrives and is placed in the temporary parking spot.
Automobile is moved from temporary to a car parking spot. 0 spots left.
All parking spots are full. Exiting program.
```

```
all : clean compile run

compile :
	@gcc -g -o hw3 hw3.c -lpthread

run :
	./hw3
clean :
	@rm -f *.o
	@rm -f hw3

.PHONY: all compile run clean
```