

GEBZE TECHNICAL UNIVERSITY

CSE 344

SYSTEM PROGRAMMING

Homework #3 Report

BARIŞ YURDAKUL

1801042620

1.Problem Definition

The project involves the simulation of a parking lot system managed by two parking attendants and consisting of separate spaces for pickups and automobiles. The parking lot has a capacity of four spaces for pickups and eight spaces for automobiles. The primary objectives and challenges in this simulation are as follows:

Vehicle Arrival and Classification:

- Vehicles arriving at the parking lot entrance can be either pickups or automobiles. The type of vehicle is determined randomly.
- Each vehicle type needs to be parked in its designated area within the parking lot.

Synchronization of Parking Process:

- Only one vehicle can enter the parking system at a time.
- The system must ensure that there are available temporary parking spots before a vehicle is parked.
- Each type of vehicle (pickup or automobile) is managed by a specific parking attendant.
- The synchronization of vehicle owners and parking attendants needs to be handled efficiently to avoid conflicts and ensure smooth operation.

Temporary Parking and Final Parking:

- Upon arrival, vehicle owners first park their vehicles in a temporary parking area if there is space available.
- Parking attendants are responsible for moving vehicles from the temporary parking area to the main parking lot.
- The system needs to manage concurrent access to shared resources such as temporary parking spaces.
- Proper synchronization mechanisms must be implemented to ensure data consistency and prevent race conditions.

2.Implementation

To implement this parking lot management system, we utilize multithreading in C, leveraging semaphores and mutexes for synchronization. The solution involves creating two types of threads: carOwner threads, representing the vehicle owners, and carAttendant threads, representing the parking attendants. Additionally, two auxiliary threads are used to periodically free up spots in the main parking lot.

Key Components and Their Roles:

- Semaphores:
 - newPickup and inChargeforPickup: Used to signal when a new pickup arrives and when the pickup attendant processes it.
 - newAutomobile and inChargeforAutomobile: Used to signal when a new automobile arrives and when the automobile attendant processes it.
- Mutexes:
 - mutex_automobile and mutex_pickup: Protect access to the temporary parking spot counters for automobiles and pickups, respectively.
 - main_lot_mutex: Protects access to the main parking lot counters for both vehicle types.
- Counters:
 - tempFree_automobile and tempFree_pickup: Track the availability of temporary parking spots for automobiles and pickups.
 - mainFree_automobile and mainFree_pickup: Track the availability of main parking lot spots for automobiles and pickups.

Process Flow:

- Car Owner Threads:
 - Each carOwner thread simulates a vehicle owner arriving at the parking lot.
 - The vehicle type is randomly determined (0 for automobile, 1 for pickup).
 - The thread checks if a temporary parking spot is available for the vehicle type. If available, the vehicle is parked temporarily, and the corresponding semaphore (newAutomobile or newPickup) is signaled.

- If no temporary spot is available, the vehicle owner leaves, and an appropriate message is printed.
- *Car Attendant Threads:*
 - The pickupAttendant and automobileAttendant threads wait for signals from the newPickup and newAutomobile semaphores, respectively.
 - Upon receiving a signal, the attendant processes the vehicle, updates the temporary parking spot counter, and signals the inChargeforPickup or inChargeforAutomobile semaphore.
 - The attendant then attempts to park the vehicle in the main parking lot, updating the main parking lot counter. If the main parking lot is full, a message indicating the full status is printed.
- *Main Lot Freeing Threads:*
 - Two additional threads, mainLotFreePickup and mainLotFreeAutomobile, periodically free up spots in the main parking lot to simulate vehicles leaving the lot.
 - These threads run in an infinite loop with a sleep interval, ensuring that the main parking lot is not perpetually full and new vehicles can be accommodated over time.

Main Function:

The main function sets up signal handling, initializes semaphores and mutexes, and creates the parking attendant thread. It then simulates the arrival of 20 vehicles (Scenario 1) by creating a thread for each vehicle owner and randomizing the time between vehicle arrivals. The main function waits for the attendant thread to finish (though it runs indefinitely in this simulation).

3.Results and Test Cases

- Scenario 1

- VehicleType Randomly Generated with 20 vehicle loop

```
for (int i = 0; i < 20; i++) {  
    int* vehicleType = malloc( size: sizeof(int));  
    *vehicleType = rand() % 2; // Randomly generate vehicle type (0 for automobile, 1 for pickup)  
    pthread_t ownerThread;  
    pthread_create( newthread: &ownerThread, attr: NULL, start_routine: carOwner, arg: vehicleType);  
    usleep( useconds: 500000); // Simulate time delay between vehicle arrivals  
}
```

```
-Pickup OWNER ARRIVED, temporary pickup spots left: 0  
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1  
Pickup PARKED in the main lot, main pickup spots left: 3  
-Pickup OWNER ARRIVED, temporary pickup spots left: 0  
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1  
Pickup PARKED in the main lot, main pickup spots left: 2  
-Automobile OWNER ARRIVED, temporary automobile spots left: 0  
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1  
Automobile PARKED in the main lot, main automobile spots left: 7  
-Pickup OWNER ARRIVED, temporary pickup spots left: 0  
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1  
Pickup PARKED in the main lot, main pickup spots left: 1  
Freed up a spot in main lot for automobiles, main automobile spots left: 8  
-Automobile OWNER ARRIVED, temporary automobile spots left: 0  
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1  
Automobile PARKED in the main lot, main automobile spots left: 7  
Freed up a spot in main lot for pickups, main pickup spots left: 2  
-Automobile OWNER ARRIVED, temporary automobile spots left: 0  
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1  
Automobile PARKED in the main lot, main automobile spots left: 6  
-Automobile OWNER ARRIVED, temporary automobile spots left: 0  
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1  
Automobile PARKED in the main lot, main automobile spots left: 5  
-Pickup OWNER ARRIVED, temporary pickup spots left: 0  
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1  
Pickup PARKED in the main lot, main pickup spots left: 1  
-Automobile OWNER ARRIVED, temporary automobile spots left: 0  
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1  
Automobile PARKED in the main lot, main automobile spots left: 4  
Freed up a spot in main lot for pickups, main pickup spots left: 2
```

First Example of Scenario 1

```
-Automobile OWNER ARRIVED, temporary automobile spots left: 0  
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1  
Automobile PARKED in the main lot, main automobile spots left: 2  
Freed up a spot in main lot for automobiles, main automobile spots left: 3  
-Pickup OWNER ARRIVED, temporary pickup spots left: 0  
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1  
Pickup PARKED in the main lot, main pickup spots left: 1  
Freed up a spot in main lot for pickups, main pickup spots left: 1  
-Pickup OWNER ARRIVED, temporary pickup spots left: 0  
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1  
Pickup PARKED in the main lot, main pickup spots left: 0  
Freed up a spot in main lot for automobiles, main automobile spots left: 4  
-Pickup OWNER ARRIVED, temporary pickup spots left: 0  
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1  
Main parking lot for pickups is full. Pickup cannot be parked.  
-Automobile OWNER ARRIVED, temporary automobile spots left: 0  
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1  
Automobile PARKED in the main lot, main automobile spots left: 3  
-Pickup OWNER ARRIVED, temporary pickup spots left: 0  
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1  
Main parking lot for pickups is full. Pickup cannot be parked.  
Freed up a spot in main lot for pickups, main pickup spots left: 1  
Freed up a spot in main lot for automobiles, main automobile spots left: 4  
Freed up a spot in main lot for pickups, main pickup spots left: 2  
Freed up a spot in main lot for automobiles, main automobile spots left: 5  
Freed up a spot in main lot for pickups, main pickup spots left: 6  
Freed up a spot in main lot for pickups, main pickup spots left: 3  
Freed up a spot in main lot for pickups, main pickup spots left: 4  
Freed up a spot in main lot for automobiles, main automobile spots left: 7  
Freed up a spot in main lot for automobiles, main automobile spots left: 8
```

Second Part of Same Example

Freeing up occurs randomly depending on time.

- Scenario 2
- Only Automobiles arrived

```
for (int i = 0; i < 20; i++) {  
    int* vehicleType = malloc( size: sizeof(int));  
    *vehicleType = 0; // Only automobiles  
    pthread_t ownerThread;  
    pthread_create( newthread: &ownerThread, attr: NULL, start_routine: carOwner, arg: vehicleType);  
    usleep( useconds: 500000); // Simulate time delay between vehicle arrivals  
}
```

```

-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 3
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 2
Freed up a spot in main lot for automobiles, main automobile spots left: 3
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 2
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 1
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 0
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Main parking lot for automobiles is full. Automobile cannot be parked.
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Main parking lot for automobiles is full. Automobile cannot be parked.
Freed up a spot in main lot for automobiles, main automobile spots left: 1

```

Some Part of The Output of Scenario 2

- Scenario 3
 - Only Pickups arrived

```

for (int i = 0; i < 20; i++) {
    int* vehicleType = malloc(sizeof(int));
    *vehicleType = 1; // Only pickups
    pthread_t ownerThread;
    pthread_create(&ownerThread, attr: NULL, start_routine: carOwner, arg: vehicleType);
    usleep(useconds: 500000); // Simulate time delay between vehicle arrivals
}

```

```

-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Main parking lot for pickups is full. Pickup cannot be parked.
Freed up a spot in main lot for pickups, main pickup spots left: 1
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 0
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Main parking lot for pickups is full. Pickup cannot be parked.
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Main parking lot for pickups is full. Pickup cannot be parked.
Freed up a spot in main lot for pickups, main pickup spots left: 1
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 0
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Main parking lot for pickups is full. Pickup cannot be parked.
Freed up a spot in main lot for pickups, main pickup spots left: 1
Freed up a spot in main lot for pickups, main pickup spots left: 2
Freed up a spot in main lot for pickups, main pickup spots left: 3
Freed up a spot in main lot for pickups, main pickup spots left: 4

```

Some Last Part of Scenario 3

– Scenario 4

- If we generate time randomly, lets check is there any problem :

```

for (int i = 0; i < 20; i++) {
    int* vehicleType = malloc( size: sizeof(int));
    *vehicleType = rand() % 2; // Randomly generate vehicle type (0 for automobile, 1 for pickup)
    pthread_t ownerThread;
    pthread_create( newthread: &ownerThread, attr: NULL, start_routine: carOwner, arg: vehicleType);
    usleep( useconds: (rand() % 5 + 1) * 200000); // Simulate time delay between vehicle arrivals
}

```

```

-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 7
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 3
Freed up a spot in main lot for automobiles, main automobile spots left: 8
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 7
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 2
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 1
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 6
Freed up a spot in main lot for pickups, main pickup spots left: 2
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 1
Freed up a spot in main lot for automobiles, main automobile spots left: 7
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 6

```

Works fine !

- Signal Handler (CTRL+C)

```

-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 5
Freed up a spot in main lot for automobiles, main automobile spots left: 6
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 3
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 2
Freed up a spot in main lot for pickups, main pickup spots left: 3
Freed up a spot in main lot for automobiles, main automobile spots left: 7
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 6
-Automobile OWNER ARRIVED, temporary automobile spots left: 0
Automobile ATTENDANT MOVED an automobile, temporary automobile spots left: 1
Automobile PARKED in the main lot, main automobile spots left: 5
-Pickup OWNER ARRIVED, temporary pickup spots left: 0
Pickup ATTENDANT MOVED a pickup, temporary pickup spots left: 1
Pickup PARKED in the main lot, main pickup spots left: 2
^C
Exiting the Program!!!

```


UPDATED : Added cancel and join(to ensure) methods to handler function.Result is good because there is no memory leaks right now.Before that,there were 4 error.

```
void handle_sigint(int sig) {
    printf(format: "\nExiting the Program!!!\n");

    pthread_cancel(th: pickupAttendantThread);
    pthread_cancel(th: automobileAttendantThread);
    pthread_cancel(th: freePickupThread);
    pthread_cancel(th: freeAutomobileThread);

    pthread_join(th: pickupAttendantThread, thread_return: NULL);
    pthread_join(th: automobileAttendantThread, thread_return: NULL);
    pthread_join(th: freePickupThread, thread_return: NULL);
    pthread_join(th: freeAutomobileThread, thread_return: NULL);
}

Main parking lot for pickups is full. Pickup cannot be parked.
Freed up a spot in main lot for automobiles, main automobile spots left: 7
Freed up a spot in main lot for pickups, main pickup spots left: 1
Freed up a spot in main lot for automobiles, main automobile spots left: 8
Freed up a spot in main lot for pickups, main pickup spots left: 2
Freed up a spot in main lot for pickups, main pickup spots left: 3
Freed up a spot in main lot for pickups, main pickup spots left: 4
^C
Exiting the Program!!!
==28252==
==28252== HEAP SUMMARY:
==28252==    in use at exit: 0 bytes in 0 blocks
==28252==   total heap usage: 52 allocs, 52 frees, 11,734 bytes allocated
==28252==
==28252== All heap blocks were freed -- no leaks are possible
==28252==
==28252== For lists of detected and suppressed errors, rerun with: -s
==28252== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
baris@baris-Ubuntu:~/Desktop/system_hw3$
```