

Gebze Technical University
CSE 344
SYSTEM PROGRAMMING
HOMEWORK #3

200104004066 - Ahmet Yiğit

- Introduction and Aim
 - **Introduction:**
 - This program presents a multi-threaded parking management system designed to allocate parking spaces for automobiles and pickups efficiently. There are two attendants to get the car/pickup from temporary to real parking area. Also random customers get his cars to a temporary park area.
 - **Aim:**
 - The aim of this project is to simulate a parking management system using multi-threading in C programming. The system manages both real and temporary parking spaces for automobiles and pickups.
 1. **Resource Allocation:** Utilize semaphores to manage access to parking spaces, ensuring that vehicles are parked only when there are available spots.
 2. **Thread Synchronization:** Employ pthreads to create threads representing car owners and parking attendants. Synchronization mechanisms prevent race conditions and ensure orderly parking.
 3. **Dynamic Parking Management:** Dynamically allocate parking spaces between real and temporary areas based on availability, optimizing resource utilization and accommodating fluctuations in demand.
- Problem Solving Approach
 - First of all, a random customer comes to our temporary parking space and when he is parking his car I lock the global variable so nobody can change or prevent data race conditions.
 - After parking if there is available, a specific vendor comes here and takes an automobile if he is an automobile vendor and there is a space to park in the real park area.
 - This situation goes on until all parking areas temp and real are full.

Detailed view of Homework:

- **Global Variables:**

```

1
2  int mFree_automobile = 0;
3  int mFree_pickup = 0;
4  int temp_automobiles = 0;
5  int temp_pickups = 0;
6
7  sem_t inChargeForAutomobile;
8  sem_t newAutomobile;
9  sem_t inChargeForPickup;
10 sem_t newPickup;
11 sem_t printSemaphore;
12

```

- **mFree_automobile** and **mFree_pickup** keep track of the available parking spots for automobiles and pickups, respectively, in the real parking area.
- **temp_automobiles** and **temp_pickups** count the vehicles temporarily parked by random selected customers.
- Semaphores like **inChargeForAutomobile** and **inChargeForPickup** control access to critical sections of code, ensuring that parking spots are managed safely and efficiently, without conflicts between threads.

- Initialization of Threads and semaphores

```

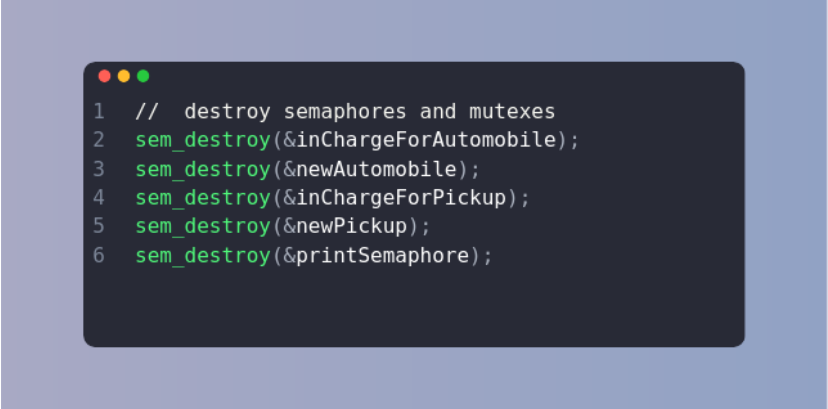
1 // init semaphores
2 sem_init(&inChargeForAutomobile, 0, MAX_REAL_AUTOMOBILES);
3 sem_init(&newAutomobile, 0, 1);
4 sem_init(&inChargeForPickup, 0, MAX_REAL_PICKUPS);
5 sem_init(&newPickup, 0, 1);
6 sem_init(&printSemaphore, 0, 1);
7
8 // creating threads
9 pthread_t owner_thread, attendant_thread_automobile, attendant_thread_pickup;
10 pthread_create(&owner_thread, NULL, carOwner, NULL);
11 pthread_create(&attendant_thread_automobile, NULL, carAttendantAutomobile, NULL);
12 pthread_create(&attendant_thread_pickup, NULL, carAttendantPickup, NULL);
13
14 // waiting for threads
15 pthread_join(attendant_thread_automobile, NULL);
16 pthread_join(attendant_thread_pickup, NULL);
17 pthread_join(owner_thread, NULL);

```

- **Semaphore Initialization:** This section initializes five semaphores:
 - **inChargeForAutomobile** and **inChargeForPickup** ensure exclusive access to the real parking areas for automobiles and pickups, respectively.
 - **newAutomobile** and **newPickup** control access to a critical section of code where parking decisions are made, preventing multiple threads from attempting to park simultaneously.
 - **printSemaphore** regulates access to the printing function **printDetail**, ensuring that output is synchronized and avoids interleaved messages.
- **Thread Creation:** Three threads are created using `pthread_create`:
 - **owner_thread** represents the car owners who attempt to park their vehicles.
 - **attendant_thread_automobile** and **attendant_thread_pickup** act as parking attendants responsible for managing the parking areas for automobiles and pickups, respectively.
- **Thread Synchronization:** The main thread waits for the parking attendant threads and the car owner thread to finish their tasks using `pthread_join`. This

ensures that the program waits for all threads to complete before proceeding with further execution.

- Destroying Semaphores :



```
1 // destroy semaphores and mutexes
2 sem_destroy(&inChargeForAutomobile);
3 sem_destroy(&newAutomobile);
4 sem_destroy(&inChargeForPickup);
5 sem_destroy(&newPickup);
6 sem_destroy(&printSemaphore);
```

-
- In this part of the code, the semaphores are destroyed after their use:
 - **sem_destroy(&inChargeForAutomobile):** Destroys the semaphore responsible for controlling access to the real parking area for automobiles.
 - **sem_destroy(&newAutomobile):** Destroys the semaphore regulating access to the critical section where decisions about parking automobiles are made.
 - **sem_destroy(&inChargeForPickup):** Destroys the semaphore controlling access to the real parking area for pickups.
 - **sem_destroy(&newPickup):** Destroys the semaphore regulating access to the critical section where decisions about parking pickups are made.
 - **sem_destroy(&printSemaphore):** Destroys the semaphore used for synchronizing access to the printing function printDetail.
- By destroying the semaphores, system resources allocated to them are released, ensuring proper cleanup and preventing resource leaks. This is a good practice to ensure efficient memory usage and avoid potential issues in long-running programs.

- Car Owner Function | Thread

```
1 void *carOwner(void *arg) {
2 while (1) {
3     sem_wait(&newPickup);
4     sem_wait(&newAutomobile);
5     if (
6         (temp_pickups < MAX_TEMP_PICKUPS)
7         || (temp_automobiles < MAX_TEMP_AUTOMOBILES)
8         || (mFree_automobile < MAX_REAL_AUTOMOBILES)
9         || (mFree_pickup < MAX_REAL_PICKUPS)
10    ){
11        // Temporary parking space available, park the vehicle
12        int vehicleType = rand() % 2;
13        // 0 for pickup, 1 for automobile
14        if (vehicleType == 0) {
15            if(temp_pickups == MAX_TEMP_PICKUPS){
16                printf("A pickup comes to park but there ...");
17            }else{
18                sem_wait(&printSemaphore);
19                temp_pickups++;
20                printf("A pickup is parked in the tempor...");
21                printDetail(100000);
22            }
23        } else {
24            if(temp_automobiles == MAX_TEMP_AUTOMOBILES)
25                printf("A automobile comes to park but...");
26            else{
27                sem_wait(&printSemaphore);
28                temp_automobiles++;
29                printf("An automobile is parked in ...");
30                printDetail(100000);
31            }
32        }
33        sem_post(&newPickup);
34        sem_post(&newAutomobile);
35    } else{
36        printf("DOLDUK ABI\n");
37        sem_post(&newPickup);
38        sem_post(&newAutomobile);
39        // No parking space available, exit the loop
40        break;
41    }
42 }
43 return NULL;
44 }
```

-
- This function carOwner represents the behavior of a car owner attempting to park their vehicle in the parking system. Let's break down its key components:

- **Loop:** The function runs in an infinite loop (while (1)) to continuously attempt parking until all parking spaces are occupied or the program is terminated externally.
 - **Semaphore Waits:** It first waits on both newPickup and newAutomobile semaphores. These waits ensure that the car owner waits until it's their turn to make a parking decision, preventing multiple owners from trying to park simultaneously.
 - **Parking Decision:** The function checks if there is space available in either the temporary or real parking areas for either pickups or automobiles. If space is available, it randomly chooses between parking a pickup or an automobile.
 - **Parking Process:** If space is available for the chosen vehicle type, the function updates the respective count of vehicles in the temporary parking area, prints a message indicating the successful parking, and then waits for a brief period (defined by printDetail) to simulate the parking process.
 - **No Space Available:** If no space is available for parking, it prints a message indicating the unavailability of parking spaces and exits the loop, effectively ending the parking attempt.
 - **Semaphore Posts:** After parking or deciding not to park due to unavailability of space, the function releases the newPickup and newAutomobile semaphores to allow other threads to access the critical section.
 - **Return:** The function returns NULL when it completes its task.
- This function encapsulates the behavior of car owners in the parking system, ensuring that they attempt to park their vehicles in an orderly and synchronized manner.
 - **Car Attendant Automobile Function | Thread**

```

1
2 void *carAttendantAutomobile(void *arg) {
3     while (1) {
4         sem_wait(&newAutomobile);
5
6         // Check if all parking spaces are full
7         if (
8             (mFree_automobile == MAX_REAL_AUTOMOBILES) &&
9             (temp_automobiles == MAX_TEMP_AUTOMOBILES))
10        {
11            // exit
12            sem_post(&newAutomobile);
13            break;
14        }
15        // Check if there are vehicles in temporary parking area
16        if (
17            (temp_automobiles > 0) &&
18            ((mFree_automobile < MAX_REAL_AUTOMOBILES)))
19        {
20            sem_wait(&printSemaphore);
21            temp_automobiles--;
22            mFree_automobile++;
23            printf("An automobile is moved from ....\n");
24            printDetail(1000000);
25            sem_post(&newAutomobile);
26        } else {
27            sem_post(&newAutomobile);
28            continue;
29        }
30    }
31    return NULL;
32 }
33

```

-
- This function `carAttendantAutomobile` represents the behavior of a parking attendant responsible for managing parking spaces for automobiles in the parking system.
 - **Loop:** The function runs in an infinite loop (`while (1)`) to continuously monitor the parking spaces for automobiles until all spaces are occupied in both the real and temporary parking areas or the program is terminated externally.
 - **Semaphore Wait:** It waits on the `newAutomobile` semaphore, ensuring that the attendant waits until it's their turn to check for parking or relocation of automobiles, preventing multiple attendants from attempting to access the parking spaces simultaneously.
 - **Parking Status Check:** The function checks if all parking spaces for automobiles are occupied in both the real and temporary parking areas. If all spaces are full, it exits the loop, effectively ending the attendant's task for managing automobile parking.
 - **Parking Space Allocation:** If there are automobiles parked in the temporary parking area and empty spaces available in the real parking area, the function moves an automobile from the temporary parking to the real parking area. It updates the counts of automobiles in both areas accordingly.
 - **Printing Details:** Gives users to information.
 - **Semaphore Post:** After completing its task of managing parking spaces for automobiles, the function releases the `newAutomobile` semaphore to allow other threads to access the critical section.
 - **Return:** The function returns `NULL` when it completes its task.

Car Attendant Pickup Function | Thread

```
1 void *carAttendantPickup(void *arg){
2
3     while (1) {
4         sem_wait(&newPickup);
5
6         // Check if all parking spaces are full
7         if (
8             (mFree_pickup == MAX_REAL_PICKUPS) &&
9             (temp_pickups == MAX_TEMP_PICKUPS))
10        {
11            // exit
12            sem_post(&newPickup);
13            break;
14        }
15
16        if (
17            (temp_pickups > 0) &&
18            (mFree_pickup < MAX_REAL_PICKUPS))
19        {
20            sem_wait(&printSemaphore);
21            temp_pickups--;
22            mFree_pickup++;
23            printf("A pickup is moved from temporary ..");
24            printDetail(1000000);
25            sem_post(&newPickup);
26        } else {
27            sem_post(&newPickup);
28            continue;
29        }
30    }
31
32    return NULL;
33 }
```

- This function carAttendantPickup serves as the behavior for a parking attendant responsible for managing parking spaces for pickups in the parking system.
- **Loop:** Similar to other attendant functions, this function runs in an infinite loop (while (1)) to continuously monitor the parking spaces for pickups until all spaces are occupied in both the real and temporary parking areas or the program is terminated externally.
- **Semaphore Wait:** It waits on the newPickup semaphore, ensuring that the attendant waits until it's their turn to check for parking or relocation of pickups, preventing multiple attendants from attempting to access the parking spaces simultaneously.
- **Parking Status Check:** The function checks if all parking spaces for pickups are occupied in both the real and temporary parking areas. If all spaces are full, it exits the loop, effectively ending the attendant's task for managing pickup parking.
- **Parking Space Allocation:** If there are pickups parked in the temporary parking area and empty spaces available in the real parking area, the function moves a pickup from the temporary parking to the real parking area. It updates the counts of pickups in both areas accordingly.
- **Printing Details:** Gives users to information.

- **Semaphore Post:** After completing its task of managing parking spaces for pickups, the function releases the newPickup semaphore to allow other threads to access the critical section.

- Print Detail Function

- This function **printDetail** is responsible for printing the current status of parking spaces, including the number of automobiles and pickups parked in both the real and temporary parking areas.

• Test Cases

- Real Automobile Park Area = RA
- Real Pickup Park Area = RP
- Temp Automobile Park Area = TA
- Temp Pickup Park Area = TP

• RA=8, RP=4, TA=8, TP=4

```

6
7 #define MAX_REAL_AUTOMOBILES 8
8 #define MAX_REAL_PICKUPS 4
9 #define MAX_TEMP_AUTOMOBILES 8
10 #define MAX_TEMP_PICKUPS 4
11
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Automobiles in Real Parking 0/8 Pickups in Real Parking 0/4 Automobiles in Temp Parking 4/8 Pickups in Temp Parking 1/4
An automobile is moved from temporary parking to real parking area. Total automobiles in real: 1
Automobiles in Real Parking 1/8 Pickups in Real Parking 0/4 Automobiles in Temp Parking 3/8 Pickups in Temp Parking 1/4
An automobile is moved from temporary parking to real parking area. Total automobiles in real: 2
Automobiles in Real Parking 2/8 Pickups in Real Parking 0/4 Automobiles in Temp Parking 2/8 Pickups in Temp Parking 1/4
An automobile is moved from temporary parking to real parking area. Total automobiles in real: 3
Automobiles in Real Parking 3/8 Pickups in Real Parking 0/4 Automobiles in Temp Parking 1/8 Pickups in Temp Parking 1/4
An automobile is moved from temporary parking to real parking area. Total automobiles in real: 4
Automobiles in Real Parking 4/8 Pickups in Real Parking 0/4 Automobiles in Temp Parking 0/8 Pickups in Temp Parking 1/4
A pickup comes to park but there is no parking area...
A pickup comes to park but there is no parking area...
A pickup comes to park but there is no parking area...
An automobile is parked in the temporary parking area. Total automobiles in temp: 4
Automobiles in Real Parking 8/8 Pickups in Real Parking 4/4 Automobiles in Temp Parking 4/8 Pickups in Temp Parking 4/4
An automobile is parked in the temporary parking area. Total automobiles in temp: 5
Automobiles in Real Parking 8/8 Pickups in Real Parking 4/4 Automobiles in Temp Parking 5/8 Pickups in Temp Parking 4/4
An automobile is parked in the temporary parking area. Total automobiles in temp: 6
Automobiles in Real Parking 8/8 Pickups in Real Parking 4/4 Automobiles in Temp Parking 6/8 Pickups in Temp Parking 4/4
An automobile is parked in the temporary parking area. Total automobiles in temp: 7
Automobiles in Real Parking 8/8 Pickups in Real Parking 4/4 Automobiles in Temp Parking 7/8 Pickups in Temp Parking 4/4
A pickup comes to park but there is no parking area...
A pickup comes to park but there is no parking area...
A pickup comes to park but there is no parking area...
An automobile is parked in the temporary parking area. Total automobiles in temp: 8
Automobiles in Real Parking 8/8 Pickups in Real Parking 4/4 Automobiles in Temp Parking 8/8 Pickups in Temp Parking 4/4
NO PARKING SPACE AVAILABLE
All threads are finished.
mens1s@mens1s: ~/Desktop/hw3System$

```

- RA=24, RP=17, TA=6, TP=5

```

7 | #define MAX_REAL_AUTOMOBILES 24
8 | #define MAX_REAL_PICKUPS 17
9 | #define MAX_TEMP_AUTOMOBILES 6
10 | #define MAX_TEMP_PICKUPS 5
11 |
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

|Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 17/24 | 15/17 | 4/6 | 3/5 |
|-----|-----|-----|-----|
An automobile is parked in the temporary parking area. Total automobiles in temp: 5

|Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 17/24 | 15/17 | 5/6 | 3/5 |
|-----|-----|-----|-----|
A pickup is parked in the temporary parking area. Total pickups in temp: 4

|Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 17/24 | 15/17 | 5/6 | 4/5 |
|-----|-----|-----|-----|
A pickup is parked in the temporary parking area. Total pickups in temp: 5

```

```

An automobile is parked in the temporary parking area. Total automobiles in temp: 6

|Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 23/24 | 17/17 | 6/6 | 5/5 |
|-----|-----|-----|-----|
A pickup comes to park but there is no parking area...
A pickup comes to park but there is no parking area...
A pickup comes to park but there is no parking area...
A automobile comes to park but there is no parking area...
A automobile comes to park but there is no parking area...
A automobile comes to park but there is no parking area...
A pickup comes to park but there is no parking area...
An automobile is moved from temporary parking to real parking area. Total automobiles in real: 24

|Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 24/24 | 17/17 | 5/6 | 5/5 |
|-----|-----|-----|-----|
An automobile is parked in the temporary parking area. Total automobiles in temp: 6

|Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 24/24 | 17/17 | 6/6 | 5/5 |
|-----|-----|-----|-----|
NO PARKING SPACE AVAILABLE
All threads are finished.
mens1s@mens1s:~/Desktop/hw3System$

```

- Valgrind Control

```

|Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 8/8 | 4/4 | 5/8 | 4/4 |
|-----|-----|-----|-----|
is An automobile is parked in the temporary parking area. Total automobiles in temp: 6
is
is
temp |Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 8/8 | 4/4 | 6/8 | 4/4 |
|-----|-----|-----|-----|
is An automobile is parked in the temporary parking area. Total automobiles in temp: 7
is
is
is |Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
temp | 8/8 | 4/4 | 7/8 | 4/4 |
|-----|-----|-----|-----|
An automobile is parked in the temporary parking area. Total automobiles in temp: 8

temp |Automobiles in Real Parking | Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
| 8/8 | 4/4 | 8/8 | 4/4 |
|-----|-----|-----|-----|
NO PARKING SPACE AVAILABLE
All threads are finished.
is ==18704==
is
is ==18704== HEAP SUMMARY:
per ==18704== in use at exit: 0 bytes in 0 blocks
per ==18704== total heap usage: 4 allocs, 4 frees, 1,840 bytes allocated
is ==18704==
ra ==18704== All heap blocks were freed -- no leaks are possible
==18704==
==18704== For lists of detected and suppressed errors, rerun with: -s
temp ==18704== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
mens1s@mens1s:~/Desktop/hw3System$
Pickups in Real Parking | Automobiles in Temp Parking | Pickups in Temp Parking|
17/17 | 6/6 | 5/5 |

```

