Hotel Simulation Project Documentation

# I. Project Organization

## Implementation (40 points)

The project implements a multi-threaded application in C/C++ using POSIX threads and semaphores to simulate a hotel environment. This simulation manages guests checking into the hotel, participating in activities, and then checking out.

The code is structured into multiple functions, each handling a different aspect of the simulation, such as guest routines, check-in, and check-out processes. It is formatted for readability, with proper indentation, clear naming conventions, and logical structure.

Comments throughout the code explain the purpose of functions and the logic behind significant sections. The use of POSIX threads and semaphores for concurrency and synchronization is particularly emphasized.

### Semaphores Usage

|  |  |  |
| --- | --- | --- |
| Semaphore | Function | Initial Value |
| room\_sem | Controls access to individual hotel rooms. | 1 per room |
| check\_in\_queue\_sem | Synchronizes access to the check-in queue. | 1 |
| check\_out\_queue\_sem | Synchronizes access to the check-out queue. | 1 |
| check\_in\_process\_sem | Manages check-in process for each reservationist. | 1 per reservationist |
| check\_out\_process\_sem | Manages check-out process for each reservationist. | 1 per reservationist |
| checked\_in\_sem | Signals when a guest has checked in. | 0 per guest |
| checked\_out\_sem | Signals when a guest has checked out. | 0 per guest |
| room\_available\_sem | Indicates when a room becomes available. | 0 |

## Output (10 points)

The program’s output is assessed by executing the simulation, showcasing the seamless operation of the multi-threaded hotel environment. The output validates the proper functioning of guest handling, from their entry to the hotel through to their departure. The effective use of semaphores is evident in managing synchronization across various processes like check-in, activity selection, and check-out.

```  
CSCI 474’ public\_html

c.c.robinson@lab20:~$ cd CSCI\ 474/

c.c.robinson@lab20:~/CSCI 474$ cd Project\ 2/

c.c.robinson@lab20:~/CSCI 474/Project 2$ gcc -o hotel\_simulation main.c -pthread

c.c.robinson@lab20:~/CSCI 474/Project 2$ ./hotel\_simulation

Guest 0 enters the hotel

Guest 3 enters the hotel

Guest 4 enters the hotel

Guest 5 enters the hotel

Check-in reservationist 1 greets Guest 0 and assigns Room 0

Check-in reservationist 1 greets Guest 3 and assigns Room 1

Check-in reservationist 1 greets Guest 4 and assigns Room 2

Check-in reservationist 1 greets Guest 5 and assigns Room 3

Guest 3 goes to the business center

Guest 2 enters the hotel

Guest 1 enters the hotel

Guest 5 goes to the restaurant

Check-in reservationist 0 greets Guest 2 and assigns Room 4

Guest 4 goes to the business center

Guest 0 goes to the restaurant

Guest 6 enters the hotel

Guest 2 goes to the restaurant

Guest 7 enters the hotel

Guest 8 enters the hotel

Guest 9 enters the hotel

Check-out reservationist 0 greets Guest 0

Check-out reservationist 0 printed the receipt for Guest 0

Check-in reservationist 0 greets Guest 1 and assigns Room 0

Guest 1 goes to the fitness center

Check-out reservationist 0 greets Guest 3

Check-out reservationist 1 greets Guest 5

Check-out reservationist 1 printed the receipt for Guest 5

Check-out reservationist 0 printed the receipt for Guest 3

Check-out reservationist 1 greets Guest 2

Check-out reservationist 1 printed the receipt for Guest 2

Check-out reservationist 1 greets Guest 4

Check-out reservationist 1 printed the receipt for Guest 4

Check-in reservationist 0 greets Guest 7 and assigns Room 1

Check-in reservationist 0 greets Guest 8 and assigns Room 2

Check-in reservationist 0 greets Guest 9 and assigns Room 3

Check-in reservationist 1 greets Guest 6 and assigns Room 0

Guest 9 goes to the fitness center

Guest 8 goes to the fitness center

Guest 7 goes to the swimming pool

Guest 6 goes to the business center

Check-out reservationist 1 greets Guest 1

Check-out reservationist 1 printed the receipt for Guest 1

Check-out reservationist 0 greets Guest 8

Check-out reservationist 0 printed the receipt for Guest 8

Check-out reservationist 1 greets Guest 7

Check-out reservationist 1 printed the receipt for Guest 7

Check-out reservationist 0 greets Guest 6

Check-out reservationist 0 printed the receipt for Guest 6

Check-out reservationist 0 greets Guest 9

Check-out reservationist 0 printed the receipt for Guest 9

Total Guests: 10

Pool: 1

Restaurant: 3

Fitness Center: 3

Business Center: 3

c.c.robinson@lab20:~/CSCI 474/Project 2$  
```

This output reflects the logical flow of the hotel simulation. Guests are checked in and assigned rooms, they engage in various activities, and eventually check out, receiving their receipts. The synchronization of these activities, facilitated by the reservationists and the semaphore mechanism, demonstrates the successful implementation of threading and process control in the simulation.

# Conclusion

This project provided practical experience in multi-threaded programming using POSIX threads and semaphores in C/C++. The challenges of ensuring mutual exclusion and synchronization in a concurrent environment were successfully addressed and overcome.