

Operating Systems Concepts

CSci 474

Project #2: Threads and semaphores

Due Date: 11:55PM, November 27 2023

Submission: Submit your source code and word document to the blackboard. Late submission will NOT be accepted. Email submission will NOT be accepted. If you cannot complete the project before the deadline due to Covid-19, you must contact the instructor BEFORE the deadline.

I. Project Organization

This project will implement a multi-threaded application using the POSIX threads and semaphores with the programming language of c/c++. Your program will be tested on a machine (lab01.cs.ndsu.nodak.edu -- lab20.cs.ndsu.nodak.edu) in the computer science Linux lab QBB 244.

You should do the following pieces to complete your project. Each piece is explained below:

- Implementation 40 points
- Output 10 points

Implementation

Your code should be nicely formatted with plenty of comments. The code should be easy to read, properly indented, employ good naming standards, good structure, and should correctly implement the design. **If you use any online resource, you must cite the source in your program.**

In addition to the source code, you must submit a word document, which completes the following table. In the table, you should list all semaphores used in the program, the function and initial value for each semaphore.

Semaphore	Function	Initial Value

Output

Output will be graded by running your program.

II. Project Description

Hotel Simulation

You must use POSIX threads to complete this project.

This project creates a system to simulate a hotel. The following rules apply:

Guest:

- 1) 5 guests visit the hotel (1 thread per guest created at start of simulation). Guests have a unique index from 0 to 4.
- 2) Hotel only has 3 rooms – room numbers ranging from 0 to 2. If a room is available, print the corresponding activity, such as “Guest 0 enters the hotel”. Otherwise, the guest is blocked until a room is available.
- 3) Guest – Check in

- a) A guest is waiting in a line (i.e., blocked) if the check-in reservationist is serving another customer. If the check-in reservationist is available, the guest goes to the check-in reservationist. Print the corresponding activity, such as “Guest 0 goes to the check-in reservationist”
 - b) The guest gets a room number from the check-in reservationist. Print the corresponding activity, such as “Guest 1 receives Room 2 and completes check-in.”
- 4) Guest – Hotel Activity
 - a) After check in, a guest does one of those activities, i.e., (swimming pool, restaurant, fitness center or business center). Print the corresponding activity, such as “Guest 1 goes to the swimming pool”. Then randomly sleep for 1-3 seconds.
- 5) Guest - Check out.
 - a) A guest is waiting in a line if the check-out reservationist is serving another customer. If the check-out reservationist is available, the guest goes to the check-out reservationist. Print the corresponding activity, such as “Guest 1 goes to the check-out reservationist and returns room 2”
 - b) The guest receives the receipt. Print the corresponding activity, such as “Guest 1 receives the receipt”.

Check-in reservationist – one thread created at start of simulation

- 1) If no guest, the check-in reservationist is blocked. If a guest goes to the check-in reservationist, print the corresponding activity, such as “The check-in reservationist greets Guest 1”
- 2) The check-in reservationist assigns an empty room to the guest. Print the corresponding activity, such as “Check-in reservationist assigns room 2 to Guest 1”

Check-out reservationist - one thread created at start of simulation

- 1) If no guest, the check-out reservationist is blocked. If a guest goes to the check-out reservationist, print the corresponding activity, such as “The check-out reservationist greets Guest 1 and receives the key from room 2”
- 2) The check-out reservationist prints the receipt. Print the corresponding balance, such as “The receipt was printed.”

Your program must clearly print out the activity of each entity based on the above description.

At the end of the simulation, give an accounting as follows:

Number of Customers

Total Guests:
 Pool:
 Restaurant:
 Fitness Center:
 Business Center:

III. Project Guidelines

Submitting

Submit your project on Blackboard. Include in your submission the following files:

- 1) A Word document for the written pieces of the project
- 2) Your source files

Cheating

All work must be your own. If cheating is detected, all parties involved will be given a zero for the project and the penalty will be documented on a form that you must sign. You may be referred to the Dean’s office for further discussion. If you refer to any online resource, you must cite the source in your program.

Grading

The written portions will be graded subjectively based on completeness and quality. The code will be graded based on points allocated for each key part of the processing as determined by the instructor. The output will be graded based on expected results for the runs.