

Assignment 8

Adv C Programming Comprehensive Assignment

Title: Mastering Advanced C Programming
Concepts - Semaphores 2

Objective: The objective of this assignment is to reinforce your understanding of advanced C programming concepts.

Assignment Task: Solving the sleeping barber problem using semaphores.

The sleeping barber problem is a well-known challenge in inter-process communication and synchronization, highlighting the complexities that arise in systems with multiple operating system processes. This problem was initially introduced by computer science pioneer Edsger Dijkstra in 1965, using it to emphasize the potential redundancy of general semaphores (https://en.wikipedia.org/wiki/Sleeping_barber_problem)

The scenario involves a hypothetical barbershop with one barber, a single barber chair, and a waiting room containing n chairs (where n may be 0) for waiting customers. The rules governing this scenario are as follows:

- If no customers are present, the barber falls asleep in the chair.
- A customer must wake the barber if they find him asleep.

- If a customer arrives while the barber is working, the customer leaves if all chairs are occupied; otherwise, they sit in an empty chair if available.
- When the barber completes a haircut, they check the waiting room for any waiting customers and fall asleep if none are found.

Two main complications arise in this scenario. Firstly, there is a risk of a race condition, where the barber may sleep while a customer waits for a haircut, leading to potential issues during the various actions involved. Secondly, problems may occur when two customers arrive simultaneously, both attempting to sit in the single available chair.

In the case of the multiple sleeping barbers problem, additional complexity arises in coordinating multiple barbers among waiting customers.

Various solutions exist for the sleeping barber problem, all of which require the use of a mutex to ensure that only one participant can change state at a time. The barber acquires the room status mutex before checking for customers and releases it when transitioning to sleep or haircut. Customers acquire the mutex before entering the shop, release it upon sitting in a waiting room or barber chair, and acquire it again when leaving the shop due to a lack of available seats. Additionally, semaphores are needed to indicate the system's state, such as storing the number of people in the waiting room.

The implementation of a solution provided (assignment8.c) may lead to the potential starvation of a customer or a deadlock problem.

Instructions for the implementation

1. Download assignment8.c
2. Make your own Makefile to compile
3. Fix any starvation or deadlock problems
4. Any enhancement is allowed.
5. Properly handle any error cases and print error messages in the error cases.
6. Add comments on any of your modifications. Describe why you added them in details.

Documentation

- Document your code thoroughly with comments explaining each section.
- Prepare a report PDF document summarizing everything such as each step or functions that you implemented, challenges faced, and what you've learned during this assignment.

Submission Guidelines:

- Submit the consolidated source code (a zip file) for the entire assignment.
- Include a README file with instructions for compiling and running your program.
- Ensure your code is well-documented with comments.

Assessment Criteria:

Your assignment will be assessed based on the following criteria:

- Submit your own work (80% in points) even though it doesn't execute as expected.
- Proper use of C data types, pointers, arrays, structures, and dynamic memory allocation.
- Efficiency and readability of the code.
- Effective use of memory, functions, and function pointers.
- Proper error handling and validation.
- Utilization of macros for code optimization.
- Utilization of C standard library functions.
- Quality and completeness of documentation.

Important Note: Plagiarism will not be tolerated, and students are expected to produce their work independently. Please perform this assignment by yourself and don't copy any solutions from any Generative AI applications like ChatGPT or Bard, your friends or online. If I find any things that imply plagiarism, you will lose the whole points and be reported.