Assignment 3 - Part 2: Still all about synchronization of threads

Fall 2024

Due: Saturday, Oct 12, 11:59 pm

Coding Task 1: Choose a synchronization technique (mutex, condition variables, barrier, or any combination) to correct the problem of the "Neuromancer Cyber Heist Game". Use neuromancer.c as your starter code.

Hint 1: The currentPlayer tracker is a global variable and the players are not waiting for their turns to attempt a hack.

Hint 2: You'll need to use two synchronization techniques to make this a fair game.

Coding Task 2: Choose a synchronization technique (mutex, condition variables, or barrier) to correct the problem of the finding the average of an array. Use average.c as your starter code.

Hint: This is a phased computation. The average is being calculated before the summing process is fully complete. Ensure that all threads finish summing before proceeding with the average calculation.

NOTE: Ensure proper creation, initialization, management, and destroy of chosen synchronization technique.

D2L Discussion Task 3: Discuss how the resource hierarchy solution effectively addresses the Dining Philosopher's Problem. Explain how this approach manages concurrency, prevents starvation, and avoids deadlock. (Reply to at least 2 peer posts)

Resource Hierarchy Solution – treats the forks as resources and numbers them 1 through 5. Each philosopher must first pick up the lower numbered fork before the higher numbered one to prevent circular wait.

Reading references:

- https://eng.libretexts.org/Courses/Delta_College/Operating_System%3A_The_Basics/06%3A_Deadlock/6.4%3A_Dining_Philosopher_Problem#:~:text=This%20solution%20to%20the%20problem,not%20completely%20known%20in%20advance.
- https://gist.github.com/davidjpfeiffer/2bda83fbba570592e4d46d5e47853d32?permalink_comment_id=4013641

Deliverables:

- A zipped folder named A3_part2 submitted to D2L dropbox that contains:
    - Your source code files for both coding tasks 1 and 2.
    - A pdf file for screenshots showing your codes fixed the issue and work as expected.
    - A discussion post for Task 3 under D2L discuss – Processes and Threads.
    - A README.txt file
- Available on GitHub inside the folder where you added me as a collaborator.

Total Points (100)

- Code runs and works as expected – Task 1: 50 points
- Code runs and works as expected – Task 2: 30 points
- Clear, detailed explanation & replied to at least 2 peer posts – Task 3: 10 points
- Proper commenting and screenshot of code output: 5 points
- Available on GitHub: 5 points