

Synchronization

Overview

The practise questions are designed to complement and enhance your knowledge of topics covered in the lectures. Not all answers will be readily found on the lecture notes and slides, and you may be required to engage in some self-learning to complete the questions.

Suggested answers to the practise questions shall be provided at a later date. It is strongly advised that you attempt these questions on your own, and discuss them with your peers before consulting the suggested answers.

Practise Questions

1. Although useful for demonstration purposes, Peterson's Solution is not guaranteed to work on modern architectures. Propose an implementation of Peterson's Solution that utilizes memory barriers to address this and handle out-of-order execution.
2. Consider a banking system with three methods - `read_balance()`, `deposit(amount)`, and `withdraw(amount)`, where `amount` is the amount to be deposited or withdrawn from the bank account. Explain, with code examples, how race condition is possible in such a system and what can be done to prevent them from occurring.
3. Explain why implementing synchronization primitives by disabling interrupts is not appropriate in a single-processor system if the synchronization primitives are to be used in user-level programs.
4. Propose a solution using semaphores or mutexes to the producer-consumer problem. Your solution should support multiple producers and multiple consumers concurrently. Clearly explain the semaphores or mutexes used.

5. Refer to the concurrent processes P1 and P2 with their respective code segments c1 to c8 executing in the orders shown below. Using semaphores with their `wait()` and `signal()` functions, design a solution to ensure the fulfilment of both conditions where c2 must execute before c6, and c8 must execute before c4. Clearly define the initial value of the semaphores.

```
P1 { c1; c2; c3; c4; }  
P2 { c5; c6; c7; c8; }
```

Is it possible to design a solution with just a single semaphore? If no, explain the reason. If yes, show the code.

END OF DOCUMENT