Operating Systems

List 4

Exercise 1

Give an example of adverse competition of two processes with respect to shared resource access. Show that the result achieved through resource modification may depend on arbitrary decisions taken by short-term scheduler. Define following terms: race condition and a deadlock.

Exercise 2

Define critical section problem and give a set of conditions, that have to be satisfied in order to solve the problem. List additional assumptions about timing issues, that have to be taken into consideration.

Exercise 3

How computer hardware can support thread synchronization? Why, in this case, is it important to distinguish between uniprocessor and multiprocessor machines? Explain atomicity term. Using atomic instruction like `XCHG` (exchange) or `CAS` (compare-and-swap) implement `semWait` and `semSignal` function for binary semaphores.

Exercise 4

Another hardware mechanism of supporting thread or process synchronization is `LL-SC` (`load-link` / `store-conditional`). It can also be used to implement transactional memory Please define this term. Explain how does `LL-SC` mechanism work and show its advantages over atomic instructions. Describe ABA problem.

Exercise 5

What is a semaphore and what they are used for? Define following terms: binary, counting, weak, strong semaphore and mutex. Explain key differences between them. Given binary semaphores, please implement (in pseudocode) a counting semaphore.

Exercises 6

A process that waits for a semaphore to be signalled, can either relinquish CPU or perfom so called busy-waiting. What are the advantages and disadvantages of both solutions? Describe hybrid approach called adaptive mutex. Explain spinlock term.

Exercise 7

List four conditions necessary for a deadlock to occur and explain how one can prevent it. Describe priority inversion phenomenon and how a scheduler has to be modified in order to avoid it. Give a real-life example of a project that was severly threatened with failure as a consequence of programming mistake related to priority inversion.

Exercise 8

One of the high-level programming language constructs used for synchronization is a monitor. Define this term and give an example of real-life application. How does a monitor make use of conditional variables? Do monitors solve critical section problem? What is a key difference between Hoare’s and Mesa monitors?

Exercise 9

Interprocess communication by message passing requires at least two functions to be implemented: `send(dest, msg)` and `receive(src, msg)`. Describe possible message format. How can the addresses be interpreted? List possible semantics for send and receive operations (w.r.t. message buffering, addressing, time ordering and so on)?

Exercise 10

What is a barrier and what they are used for? (Don’t mix up a barrier with memory barrier!) Given binary semaphores, please implement reusable barrier in pseudocode.