This course has been archived (Saturday, 17 December 2022, 00:00).

This course has already ended.
The latest instance of the course can be found at: Concurrent Programming: 2023

Assignment description My submissions (1/10)

Spinlock -- Low-level mechanisms for mutual exclusion and concurrency control.

Spinlock

One way of enforcing limits on access to a shared resource in a multi-threaded execution environment is using locks. A lock is an important low-level synchronization mechanism used to enforce mutual exclusion and concurrency control policy.

A spinlock is a lock which causes a thread trying to acquire it to simply wait in a loop ("spin") while repeatedly checking if the lock is available. In other words, when the calling thread requests a spin lock that is already held by another thread, the second thread spins in a loop to test if the lock has become available. Note that when the lock is obtained, it should be held only for a short time, as the spinning wastes processor cycles.

A spinlock can be considered a busy-wait solution for the Mutual Exclusion Problem which requires that no two threads enter their critical section at the same time.

Code

Download the assignment template here

Task

In this exercise, we implement a simple spinlock with lock and unlock methods. We will assume we have an implementation of a simple atomic integer with a compare set method and base our spinlock implementation on it. The compare and set method is defined as `compareAndSet(expect: Int, update: Int): Boolean`. The Full interface definition of `SimpleAtomicInteger` is as follows:

```
class SimpleAtomicInteger(initValue: Int) {
    protected var value:Int
    def compareAndSet(expect: Int, update: Int): Boolean
    def get: Int
    def set(newValue: Int): Unit
}
```

Hint

Base your implemenation on `compareAndSet(expect: Int, update: Int): Boolean` method.

Spinlock.scala

Choose File No file chosen

Submit

Earned points

25 / 25

Exercise info

Assignment category
Programming exercises

Your submissions
1 / 10

Deadline
Tuesday, 16 November 2021, 14:00

Late submission deadline
Tuesday, 23 November 2021, 14:00 (-30%)

Total number of submitters
55