Course

CS-E4110

Course materials

Your points

MyCourses

Zulip Chat

This course has been archived (Saturday, 17 December 2022, 00:00).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

« 2 Spinlock -- Low-level mechanisms for mutual exclusion and concurrenc... Course materials 4 Lock-free Stack -- Implementing lock-free concurrent algorithms. »

CS-E4110 / Round 2 - Scala concurrency - Part 2 / 3 Countdown latch -- A versatile synchronization tool implemented using low-level synchronization primitives.

Assignment description My submissions (1/10)

Earned points

25 / 25

Countdown latch -- A versatile synchronization tool implemented using low-level synchronization primitives.

Countdown latch

Many languages including Scala and Java provide a low-level mechanism that effectively allows one thread to signal another. Without such a mechanism, various high-level constructs would be difficult to implement. To put it simply, Java and Scala provide `wait()`, `notify()` and `notifyAll()` methods to facilitate signaling between threads.

One simple example of a high-level construct that can be implemented using such a signaling mechanism is a countdown latch. A countdown latch is a counter that triggers an event when a count reaches zero from an initially set value. A countdown latch allows one or more threads to wait until a set of operations being performed in other threads complete.

Code

Download the assignment template here

Task

In this exercise, we implement a countdown latch using Scala's (Java's) synchronization and signaling mechanisms.

Hint

You may use the synchronization and signaling mechanisms provided by Java's runtime. You could get some more hints from here.

SimpleCountDownLatch.scala

Choose File No file chosen

Submit

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
52

« 2 Spinlock -- Low-level mechanisms for mutual exclusion and concurrenc... Course materials 4 Lock-free Stack -- Implementing lock-free concurrent algorithms. »