Java.util.concurrent同步器(Synchronizer)框架

摘要

在J2SE1.5的java.util.concurrent包里，很多同步器(locks,barriers等)的实现都用到了一个小的框架，他们建立在AbstractQueuedSynchronizer类之上。AQS框架提供了一些通用的机制，像自动管理同步状态，线程阻塞，线程解锁，线程排队。这篇论文描述了AQS框架的理论基础，设计，实现，用法和性能

类别和主题

并发编程设计，并行编程设计

一般用语

算法(Algorithms)，测量(Measurement)，性能(Performance)，设计(Design)

关键字

同步(Synchronization)，Java

1. 简介

Java的J2SE的1.5版引入了java.util.concurrent包。它是一些中级并发支持的类的集合遵循JCP(Java Community Process)JSP166。这些组件是一套同步器---抽象数据类型(ADT)类。它实现了一个内部的同步状态(例如标志一个锁是锁住或者解锁)，更新和检查状态的操作，以及在状态需要时至少一个会引起线程阻塞的方法，当一些其他线程改变同步状态允许它重新恢复的方法。并发包里包括各种形式的排他锁(exclusion locks)，读写锁(read-write locks)，信号量(semaphores)，栅栏(barriers)，futures，甚至指示器(indicators)及可相互切换的队列(handoff queues)。

众所周知，几乎所有的同步器都可以实现其他的组建。例如，可以使用重入锁(reentrant locks)建立一个信号量(semaphores)，反之也行。然而这样做通常足够复杂，经常不灵活。是一个二流的工程观点。

1. 要求
2. 设计和实现
3. 用例
4. 性能
5. 结论
6. 致谢
7. 参考