

(http://www.csdn.net?ref=toolbar)

Marshall

19 原创 1 转载

1 译文 0 评论

1341 访问



Java锁机制 自旋锁（spinlock）剖析 (/Marshall_One/article/details/52530185)

分类：Java多线程

👁 (182) 💬 (0)

Spinlock 介绍

线程通过 busy-wait-loop 方式来获取锁，任何时刻只有一个线程能够获得锁，其它线程忙等待知道获得锁。

应用场景

spinlock 不会有线程状态切换，所以响应更快。

使用spinlock时，临界区要尽量短，不要有显示或隐式的系统调用。如读写文件等操作。

临界区：指的是一个访问共用资源（例如：共用设备或是共用存储器）的程序片段，当有线程进入临界区段时，其他线程或是进程必须等待

当存在大量线程，竞争激烈时，不适合使用自旋锁，因为每个线程都在在执行，占用CPU时间。

Java 实现自旋锁

本例子使用了CAS原子操作，非公平锁，即获得锁的先后顺序，不会按照进入lock的先后顺序进行。

```
1 package concurrent.lock;
2
3 import
4 java.util.concurrent.atomic.AtomicReference;
5
6
```

📡 (http://blog.cs

(http://www.csdn.net?ref=toolbar)

Marshall

19 原创 1 转载

1 译文 0 评论

1341 访问

京 ICP 证 070598 号

Copyright © 1999-2016, CSDN.NET, All Rights Reserved

7 /** 8 * 白旋锁

登录 (https://passport.csdn.net/a ref=toolbar) 注 册 (http://passport.csdn.net/ac ref=toolbar&action=mobile (http://i



-
-
-
-
-
-

(http://blog.cs