



博客

登录 | 注册

Q

≡

shihuacai的专栏

目录视图

摘要视图

RSS 订阅

个人资料



shihuacai

+ 加关注

发私信

访问：303168次

积分：2708

等级：**BLOG > 5**

排名：千里之外

原创：20篇

转载：87篇

译文：0篇

评论：23条

文章搜索

Q

文章分类

Java技术 (36)

Android技术 (1)

IT技术概念 (5)

Linux (12)

MyEclipse (2)

Lucene (2)

Solr (7)

SQL (30)

JS (1)

GIS (2)

Python (2)

MongoDB (4)

SSH (3)

Redis (1)

Python (0)

文章存档

2015年01月 (9)

2014年07月 (2)

2014年06月 (1)

2014年04月 (2)

2014年03月 (1)

展开

阅读排行

Java之CountDownLatch (32295)

Java之HashMap get方法 (26674)

评论送书 | 7月书讯：众多畅销书升级！

征文 | 你会为 AI 转型么？

赠书 | AI专场（AI圣经！《深度学习》中文版）

转 Java之Semaphore

标签：Java 多线程 semaphore

2013-04-26 21:28 19903人阅读 评论 快速回复 报

分类：Java技术 (35)

目录(?)

我要收藏

返回顶部

一个计数信号量。从概念上讲，信号量维护了一个许可集。如有必要，在许可可用前会阻塞每一个 `acquire()`，然后再获取该许可。每个 `release()` 添加一个许可，从而可能释放一个正在阻塞的获取者。但是，不使用实际的许可对象，`Semaphore` 只对可用许可的号码进行计数，并采取相应的行动。拿到信号量的线程可以进入代码，否则就等待。通过 `acquire()` 和 `release()` 获取和释放访问许可。

### 相关方法：

**acquire**

```
public void acquire()
    throws InterruptedException
```

从此信号量获取一个许可，在提供一个许可前一直将线程阻塞，否则线程被中断。获取一个许可（如果提供了一个）并立即返回，将可用的许可数减 1。

如果没有可用的许可，则在发生以下两种情况之一前，禁止将当前线程用于线程安排目的并使其处于休眠状态：

- 某些其他线程调用此信号量的 `release()` 方法，并且当前线程是下一个要被分配许可的线程；或者
- 其他某些线程中断当前线程。

如果当前线程：

- 被此方法将其已中断状态设置为 `on`；或者
- 在等待许可时被中断。

则抛出 `InterruptedException`，并且清除当前线程的已中断状态。

抛出：

`InterruptedException` - 如果当前线程被中断

**release**

```
public void release()
```

释放一个许可，将其返回给信号量。释放一个许可，将可用的许可数增加 1。如果任意线程试图获取许可，则选中一个线程并将刚刚释放的许可给予它。然后针对线程安排目的启用（或再启用）该线程。

不要求释放许可的线程必须通过调用 `acquire()` 来获取许可。通过应用程序中的编程约定来建立信号量的正确用法。

### 相关例子：

下面的例子只允许5个线程同时进入执行 `acquire()` 和 `release()` 之间的代码：

```
public class SemaphoreTest {

    public static void main(String[] args) {
        // 线程池
        ExecutorService exec = Executors.newCachedThreadPool();
        // 只能5个线程同时访问
```

关闭

http://blog.csdn.net/shihuacai/article/details/8856526

1/3

Spring批量更新JdbcTem	
Java之CyclicBarrier使用	(26195)
Java之Semaphore	(21878)
Java下出现异常Unabl	(19899)
ubuntu下如何用gedit编	(11140)
修改了Ubuntu下的/usr	(8642)
修改了Ubuntu下的/usr	(8140)
Java之volatile, synchron	(6369)
solr下配置IKAnalyzer分	(6309)

评论排行	
Java之CountDownLatch	(3)
Java之CyclicBarrier使用	(3)
Java之volatile, synchron	(3)
Java 调用Oracle的PLSC	(2)
Java之Semaphore	(2)
PostGIS之Java入门	(2)
Spring批量更新JdbcTem	(2)
spring bean id重复覆盖	(1)
solr下配置IKAnalyzer同	(1)
solr下配置IKAnalyzer分	(1)

推荐文章	
* CSDN日报20170725——《新的开始，从研究生到入职亚马逊》	
* 深入剖析基于并发AQS的重入锁(ReentrantLock)及其Condition实现原理	
* Android版本的"Wannacry"文件加密病毒样本分析(附带锁机)	
* 工作与生活真的可以平衡吗？	
* 《Real-Time Rendering 3rd》提炼总结——高级着色：BRDF及相关技术	
* 《三体》读后思考-泰勒展开/维度打击/黑暗森林	

最新评论	
Java之CyclicBarrier使用	QH_JAVA: CountDownLatch 也可以阻塞子线程的吧，就看你在哪里调用吧？
spring bean id重复覆盖的问题解	吴名霄: 博主spring ioc容器在加载bean的过程中会去判断beanName 是否有重复，如果发现重复...
Java之Semaphore	neifan: 最好将release放到finally中，不然线程终端后 信号量是不会释放的
Java之CyclicBarrier使用	JohnnyJay: @chenchaofuck1:场景不一样啊，CountDownLatch阻塞的是主线程，而Cycli...
Java之CountDownLatch使用	xdn541: qweqweqw
Spring批量更新JdbcTemplate.b:	as64544556: @hua00shao:List有一个 LinkedList(有序的),只需要实例化这个就是正常的顺...
Spring批量更新JdbcTemplate.b:	hua00shao: spring JdbcTemplate 的batchUpdate方法批量插入数据库的顺序是乱的，不是...
ubuntu下如何用gedit编辑etc, u	yht12121212: 你好，我像问一下，我操作时显示说gedit找不到命令，这应该如何解决？

```
final Semaphore semp = new Semaphore(5);
// 模拟20个客户端访问
for (int index = 0; index < 20; index++) {
    final int NO = index;
    Runnable run = new Runnable() {
        public void run() {
            try {
                // 获取许可
                semp.acquire();
                System.out.println("Accessing: " + NO);
                Thread.sleep((long) (Math.random() * 10000));
                // 访问完后，释放，如果屏蔽下面的语句，则在控制台只能打印5条记录，之后线程一直阻塞

                semp.release();
            } catch (InterruptedException e) {
            }
        }
    };
    exec.execute(run);
}
// 退出线程池
exec.shutdown();
}
```

快速回复

我要收藏

返回顶部



顶9

踩0

上一篇

Java之CompletionService

下一篇

Java之Final

相关文章推荐	
• SQLite之大数据量批量入库	• 信号量Semaphore初探
• 理解Semaphore及其用法详解	• 深入理解Semaphore
• Java之Final	• Semaphore 的理解
• Java并发之Semaphore详解	• Semaphore的介绍和使用
• Java并发编程中Semaphore的用法	• 内核同步机制-信号量（ semaphore ）

猜你在找

扫我入群

CSDN组队！掌握AI开发不累！

即刻加入【深度学习30天系统

查看评论

2楼	neifan	2017-02-18 17:13发表	最好将release放到finally中，不然线程终端后 信号量是不会释放的
1楼	你猜我猜不猜你猜不猜	2016-06-02 17:21发表	

Java之CyclicBarrier使用  
记忆力不好: 这不和  
CountDownLatch一样吗? 只不  
过不用手动count减1而已

Java之Semaphore  
你猜我猜不猜你猜不猜: 还不错,  
收藏待用



您还没有登录,请[登录](#)或[注册](#)

\* 以上用户言论只代表其个人观点, 不代表CSDN网站的观点或立场

- 快速回复
- 我要收藏
- 返回顶部