fuyanqing03

永久域名 http://fuyanqing03.javaeye.com

fuyanqing03

浏览: 4492 次

性别:

来自: 北京

我现在离线

详细资料 留言簿

替换字符串中的小括号 | 通过 Jersey 客户端 API 调用 REST 风格的 ...

2009-10-12

java并发实践笔记

关键字: java并发实践笔记

1, 保证线程安全的三种方法 :
a, 不要跨线程访问共享变量
b, 使共享变量是 final类型的
c, 将共享变量的操作加上同步

2, 一开始就将类设计成线程安全的 , 比在后期重新修复它 ,更容易 .

3, 编写多线程程序 , 首先保证它是正确的 , 其次再考虑性能 .

4, 无状态或只读对象永远是线程安全的 .

5, 不要将一个共享变量裸露在多线程环境下 (无同步或不可变性保护)

6, 多线程环境下的延迟加载需要同步的保护 , 因为延迟加载会造成对象重复实例化

7, 对于 volatile 声明的数值类型变量进行运算 , 往往是不安全的 (volatile 只能保证可见性 , 不能保证原子性).
详见 volatile 原理与技巧中 , 脏数据问题讨论 .

8, 当一个线程请求获得它自己占有的锁时 (同一把锁的嵌套使用), 我们称该锁为可重入锁 .
在 jdk1.5 并发包中 , 提供了可重入锁的 java 实现 -ReentrantLock.

9, 每个共享变量 , 都应该由一个唯一确定的锁保护 .
创建与变量相同数目的 ReentrantLock, 使他们负责每个变量的线程安全 .

10,虽然缩小同步块的范围 , 可以提升系统性能 .
但在保证原子性的情况下 , 不可将原子操作分解成多个 synchronized块 .

11, 在没有同步的情况下 , 编译器与处理器运行时的指令执行顺序可能完全出乎意料 .
原因是 , 编译器或处理器为了优化自身执行效率 , 而对指令进行了的重排序 (reordering).

- 全部博客 (14)

其他分类

- 我的收藏 (8)
- 我的论坛主题贴 (14)
- 我的所有论坛贴 (1)
- 我的精华良好贴 (0)

最近加入圈子

- jersey
- Hibernate
- struts2
- flex
- EXT

存档

- 2009-12 (1)
- 2009-10 (4)
- 2009-09 (3)
- 更多存档...

最新评论

- 通过 Jersey 客户端 API ...

  jersey真的是很不错。但是就是不知道其性能如何.如果一个产品只需要发布成JAX ...

  -- by 超级潜水艇
- 导出MYSQL数据库数据与表 ...

  导入数据库时： 常用source 命令 进入mysql数据库控制台，如m ...

  -- by fuyanqing03

12, 当一个线程在没有同步的情况下读取变量，它可能会得到一个过期值，但是至少它可以看到那个线程在当时设定的一个真实数值. 而不是凭空而来的值. 这种安全保证，称之为最低限的安全性 (out-of-thin-air safety)

在开发并发应用程序时，有时为了大幅度提高系统的吞吐量与性能，会采用这种无保障的做法.
但是针对，数值的运算，仍旧是被否决的.

13, volatile 变量，只能保证可见性，无法保证原子性.
详见 volatile原理与技巧

14, 某些耗时较长的网络操作或 IO, 确保执行时，不要占有锁.

15, 发布 (publish) 对象，指的是使它能够被当前范围之外的代码所使用.(引用传递)
对象逸出 (escape), 指的是一个对象在尚未准备好时将它发布.

原则：为防止逸出，对象必须要被完全构造完后，才可以被发布 (最好的解决方式是采用同步)

this 关键字引用对象逸出
例子：在构造函数中，开启线程，并将自身对象 this 传入线程，造成引用传递.
而此时，构造函数尚未执行完，就会发生对象逸出了.

16, 必要时，使用 ThreadLocal变量确保线程封闭性 (封闭线程往往是比较安全的，但一定程度上会造成性能损耗)
封闭对象的例子在实际使用过程中，比较常见，例如 hibernate openSessionInView机制，jdbc的 connection机制.

17, 单一不可变对象往往是线程安全的 (复杂不可变对象需要保证其内部成员变量也是不可变的)
良好的多线程编程习惯是：将所有的域都声明为 final, 除非它们是可变的

18, 保证共享变量的发布是安全的
  a, 通过静态初始化器初始化对象 (jls 12.4.2 叙述，jvm 会保证静态初始化变量是同步的)
  b, 将对象申明为 volatile 或使用 AtomicReference
  c, 保证对象是不可变的
  d, 将引用或可变操作都由锁来保护

19, 设计线程安全的类，应该包括的基本要素：
  a, 确定哪些是可变共享变量
  b, 确定哪些是不可变的变量
  c, 指定一个管理并发访问对象状态的策略

20, 将数据封装在对象内部，并保证对数据的访问是原子的.
建议采用 volatile javabean 模型或者构造同步的 getter,setter.

21, 线程限制性使构造线程安全的类变得更容易，因为类的状态被限制后，分析它的线程安全性时，就不必检查完整的程序.

- flex 中date的计算

  var flexDate:Date=new Date("2009/10/ ...

  -- by fuyanqing03
- flex+java 项目创建 和 ...

  你需要先把项目发布一下。然后在运行myflex.mxml。

  -- by fuyanqing03
- flex+java 项目创建 和 ...

  我按上面的例子做，怎么不行呢？当点击myflex.mxml运行，是404错啊，哥有 ...

  -- by slieer

评论排行榜

- flex+java 项目创建 和 例子
- 导出MYSQL数据库数据与表结构
- flex 中date的计算
- 通过 Jersey 客户端 API 调用 REST 风格的 ...
- java并发实践笔记

RSS
Add to Google
订阅到 鲜果
[什么是RSS?]

22, 编写并发程序，需要更全的注释，更完整的文档说明 .

23, 在需要细分锁的分配时，使用 java监视器模式好于使用自身对象的监视器锁 .
前者的灵活性更好 .

```
Object target = new Object();
// 这里使用外部对象来作为监视器，而非 this
synchronized(target) {
   // TODO
}
```

针对 java monitor pattern, 实际上 ReentrantLock的实现更易于并发编程 .
功能上，也更强大 .

24, 设计并发程序时，在保证伸缩性与性能折中的前提下，优先考虑将共享变量委托给线程安全的类 .
由它来控制全局的并发访问 .

25, 使用普通同步容器 (Vector, Hashtable) 的迭代器，需要外部锁来保证其原子性 .
原因是，普通同步容器产生的迭代器是非线程安全的 .

26, 在并发编程中，需要容器支持的时候，优先考虑使用 jdk 并发容器 (ConcurrentHashMap, ConcurrentLinkedQueue, CopyOnWriteArrayList...).

27, ConcurrentHashMap, CopyOnWriteArrayList
并发容器的迭代器，以及全范围的 size(), isEmpty() 都表现出弱一致性 .
他们只能标示容器当时的一个数据状态 . 无法完整响应容器之后的变化和修改 .

28, 使用有界队列，在队列充满或为空时，阻塞所有的读与写操作 . (实现生产 - 消费的良好方案)
BlockQueue 下的实现有 LinkedBlockingQueue 与 ArrayBlockingQueue, 前者为链表，可变操作频繁优先考虑，后者为数组，读取操作频繁优先考虑 .
PriorityBlockingQueue 是一个按优先级顺序排列的阻塞队列，它可以对所有置入的元素进行排序 (实现 Comparator 接口)

29, 当一个方法，能抛出 InterruptedException, 则意味着，这个方法是一个可阻塞的方法，如果它被中断，将提前结束阻塞状态 .
当你调用一个阻塞方法，也就意味着，本身也称为了一个阻塞方法，因为你必须等待阻塞方法返回 .

如果阻塞方法抛出了中断异常，我们需要做的是，将其往上层抛，除非当前已经是需要捕获异常的层次 .
如果当前方法，不能抛出 InterruptedException, 可以使用 Thread.currentThread.interrupt() 方法，手动进行中断 .

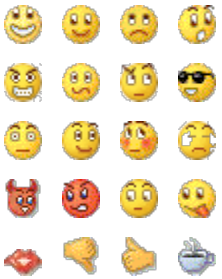
替换字符串中的小括号 | 通过 Jersey 客户端 API 调用 REST 风格的 ...

16:32 | 浏览 (90) | 评论 (0) | 相关推荐

评论

发表评论

表情图标



B I U Quote Code List Img URL Flash Table

字体颜色: 标准 字体大小: 标准 对齐: 标准

提示：选择您需要装饰的文字, 按上列按钮即可添加上相应的标签

您还没有登录，请登录后发表评论(快捷键 Alt+S / Ctrl+Enter) 提交