
Java并发性和多线程介绍

作者：Jakob Jenkov 译者：[Simon-SZ](#) 校对：方腾飞

<http://tutorials.jenkov.com/java-concurrency/index.html>

在过去单CPU时代，单任务在一个时间点只能执行单一程序。之后发展到多任务阶段，计算机能在同一时间点并行执行多任务或多进程。虽然并不是真正意义上的“同一时间点”，而是多个任务或进程共享一个CPU，并交由操作系统来完成多任务间对CPU的运行切换，以使得每个任务都有机会获得一定的时间片运行。

随着多任务对软件开发者带来的新挑战，程序不在能假设独占所有的CPU时间、所有的内存和其他计算机资源。一个好的程序榜样是在其不再使用这些资源时对其进行释放，以使得其他程序能有机会使用这些资源。

再后来发展到多线程技术，使得在一个程序内部能拥有多个线程并行执行。一个线程的执行可以被认为是一个CPU在执行该程序。当一个程序运行在多线程下，就好像有多个CPU在同时执行该程序。

多线程比多任务更加有挑战。多线程是在同一个程序内部并行执行，因此会对相同的内存空间进行并发读写操作。这可能是在单线程程序中从来不会遇到的问题。其中的一些错误也未必会在单CPU机器上出现，因为两个线程从来不会得到真正的并行执行。然而，更现代的计算机伴随着多核CPU的出现，也就意味着不同的线程能被不同的CPU核得到真正意义的并行执行。

如果一个线程在读一个内存时，另一个线程正向该内存进行写操作，那进行读操作的那个线程将获得什么结果呢？是写操作之前旧的值？还是写操作成功之后的新值？或是一半新一半旧的值？或者，如果是两个线程同时写同一个内存，在操作完成后将会是什么结果呢？是第一个线程写入的值？还是第二个线程写入的值？还是两个线程写入的一个混合值？因此如没有合适的预防措施，任何结果都是可能的。而且这种行为的发生甚至不能预测，所以结果也是不确定性的。

Java的多线程和并发性

Java是最先支持多线程的开发的语言之一，Java从一开始就支持了多线程能力，因此Java开发者能常遇到上面描述的问题场景。这也是我想为Java并发技术而写这篇系列的原因。作为对自己的笔记，和对其他Java开发的追随者都可获益的。

该系列主要关注Java多线程，但有些在多线程中出现的问题会和多任务以及分布式系统中出现的存在类似，因此该系列会将多任务和分布式系统方面作为参考，所以叫法上称为“并发性”，而不是“多线程”。

winner

2013/03/31 10:58上午

是否可以形象的理解为两个小孩打架的时候，互相拽着对方的头发不松手。

winner

2013/03/31 10:59上午

不好意思，前面那条评论我应该是发到死锁的那一篇文章的。

Arrow

2013/04/18 1:18下午

从这篇开始学习多线程

Ulric Qin

2013/05/18 7:58下午

这个系列的文章不错，真的应该好好充充电了

付政委

2013/07/25 4:38下午

喜欢ticmy的文章，喜欢他不博客。呵呵 我都快算是个粉丝了

方 腾飞

2013/07/25 11:18下午

喜欢就好，那我们的推荐才有价值。

伽达默尔

2013/08/07 9:56上午

博主，我想请教一个问题。java并发编程实践3.5.1之前有一个不安全发布的例子，某个类有一个属性非Volatile的Holder类型holder，initialize方法中就一行代码holder = new Holder(42)。书上说这是一个不安全的发布。我不太明白，可否解释一下。是不是因为这个赋值的结果其他线程可能看不到？

方 腾飞

2013/08/07 11:26上午

能把完整的代码贴出来吗？截图也行

伽达默尔

2013/08/08 8:54上午

```
class ClassA{  
  
    //不安全发布  
  
    public Holder holder;  
  
    public void initialize(){  
        holder = new Holer(42);  
    }  
  
}
```

代码是这样的，原文中没有写ClassA，只有holder和initialize。文中说这是一个不安全的发布。“由于可见性的问题，容器还是会在其他线程中被设置为一个不一致的状态，即使它的不变约束已经在构造函数中得以正确的创建。这种不正确的发布导致其他线程可以观察到‘局部创建对象（partially constructed object）’。”

方 腾飞

2013/08/09 3:45上午

我是这么理解不安全发布的，是指在并发场景下不一定能拿到自己想要的结果。

比如A线程执行initialize方法holder = new Holer(42);，如果B线程也在执行holder=new Holer(20);

那holder到底等于什么，这个结果就不一定了。

PS：原文翻译得羞涩难懂。

伽达默尔

2013/08/09 12:29下午

你说的对。还有，翻译确实有点晦涩。多谢解答。

hengzheCP

2014/11/23 12:28下午

像这种field未使用final、或者volatile修饰的，其它线程访问它的时候，由于重排序，可能导致其它线程在访问该变量的时候，它的值还没有被赋上，因此该线程读到了一个“错误”的变量。以下面的代码为例，holder = new Holer(42);其实是两个操作：

1、初始化Holder实例 2、将Holder实例赋值给引用变量holder。由于重排序，代码的实际执行顺序可能是先执行2,再执行1。此时，其它线程在访问holder的时候（由于它没有采用volatile、也没有用final修饰）而未加锁的话，就可能访问到一个未初始化完毕的Holder实例（仅仅是指向一块内存地址的引用而已，并无实质性对象）。

```
class ClassA{  
  
    //不安全发布  
  
    public Holder holder;  
  
    public void initialize(){  
        holder = new Holer(42);  
    }  
  
}
```

it_power

2015/04/13 5:41下午

这个不安全引用是因为，holder 是public 的，如果一个线程A，在初始化的时候，线程B获取hodler的值，就可能造成结果不一致，所以不安全。

bad man

2014/04/02 3:51下午

给力哦 开始理解多线程

yehong542983647

2015/12/02 4:00下午

看完一篇，就需要顶一下，谢谢你们的劳动

丁国航

2016/03/10 3:12下午

路过点赞

波波

2016/04/07 9:42上午

最近准备研究多线程呢，由于英文有限看java并发编程实战的译文有点难懂。手里正好买了一本方老师的。来到这个网站，希望和书一起学习，也能得到大神的指点

帝释天老邪

2016/08/09 10:00上午

衷心感谢翻译人员，你们的辛勤劳动，可以使我们这些新人踩在你们的肩膀上更好的前进

张秀才

2021/03/31 11:54上午

开始学习多线程