

# 史上最难的一道Java面试题 (分析篇)



匠心零度 (/u/3def157aab07) [+ 关注](#)

2017.10.15 23:07\* 字数 2076 阅读 8095 评论 44 喜欢 51 阅读 8095 评论 44 喜欢 51

(/u/3def157aab07)

简书 匠心零度 (<http://www.jianshu.com/u/3def157aab07>)  
转载请注明原创出处，谢谢！

无意中了解到如下题目，觉得蛮好。

## 题目如下：

```
public class TestSync2 implements Runnable {
    int b = 100;

    synchronized void m1() throws InterruptedException {
        b = 1000;
        Thread.sleep(500); //6
        System.out.println("b=" + b);
    }

    synchronized void m2() throws InterruptedException {
        Thread.sleep(250); //5
        b = 2000;
    }

    public static void main(String[] args) throws InterruptedException {
        TestSync2 tt = new TestSync2();
        Thread t = new Thread(tt); //1
        t.start(); //2

        tt.m2(); //3
        System.out.println("main thread b=" + tt.b); //4
    }

    @Override
    public void run() {
        try {
            m1();
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

该程序的输出结果？

## 程序输出结果

```
main thread b=2000
b=1000
或
main thread b=1000
b=1000
```

## 考察知识点

- synchronize实例锁。
- 并发下的内存可见性。

在java中，多线程的程序最难理解、调试，很多时候执行结果并不像我们想象的那样执行。所以在java多线程特别难，依稀记得大学的时候考c语言二级的时候，里面的题目是什么++和很多其他优先级的符号在一起问最后的输出结果，这类题目就想考一些运行符优先级和结合性问题。那个背背就行了，但是java多线程还是需要好好理解才行，靠背是不行的。

## 下面开始简单分析：

该题目涉及到2个线程（主线程main、子线程）、关键词涉及到synchronized、Thread.sleep。

synchronized关键词还是比较复杂的（可能有时候没有理解到位所以上面题目会有点误区），他的作用就是实现线程的同步（实现线程同步有很多方法，它只是一种，后续文章会说其他的，需要好好研究大神Doug Lea的一些实现），它的工作就是对需要同步的代码加锁，使得每一次只有一个线程可以进入同步块（其实是一种悲观策略）从而保证线程只记得安全性。

### 一般关键词synchronized的用法

- 指定加锁对象:对给定对象加锁，进入同步代码前需要活的给定对象的锁。
- 直接作用于实例方法:相当于对当前实例加锁，进入同步代码前要获得当前实例的锁。
- 直接作用于静态方法:相当于对当前类加锁，进入同步代码前要获得当前类的锁。

上面的代码，synchronized用法其实就 属于第二种情况。直接作用于实例方法:相当于对当前实例加锁，进入同步代码前要获得当前实例的锁。

### 可能存在的误区

1. 由于对synchronized理解的不到位，由于很多时候，我们多线程都是操作一个synchronized的方法，当2个线程调用2个不同synchronized的方法的时候，认为是没有关系的，这种想法是存在误区的。**直接作用于实例方法:相当于对当前实例加锁，进入同步代码前要获得当前实例的锁。**
2. 如果一个调用synchronized方法。另外一个调用普通方法是没有关系的，2个是不存在等待关系的。

这些对于后面的分析很有作用。

### Thread.sleep

使当前线程（即调用该方法的线程）暂停执行一段时间，让其他线程有机会继续执行，但它并不释放对象锁。**也就是说如果有synchronized同步块，其他线程仍然不能访问共享数据。**注意该方法要捕捉异常，对于后面的分析很有作用。一些细节可以参考我的系统学习java高并发系列二 (<http://www.jianshu.com/p/1098731f1e88>)。

### 分析流程

java 都是从main方法执行的，上面说了有2个线程，但是这里就算修改线程优先级也没用，优先级是在2个程序都还没有执行的时候才有先后，现在这个代码一执行，主线程main已经执行了。对于属性变量 int b =100由于使用了synchronized也不会存在可见性问题（也没有必要再使用volatile申明），当执行**1步骤**的时候（Thread t = new Thread(tt); //1）线程是new状态，还没有开始工作。当执行**2步骤**的时候（t.start(); //2）当调用start方法，这个线程才正真被启动，进入runnable状态，runnable状态表示可以执行，一切准备就绪了，但是并不表示一定在cpu上面执行，有没有真正执行取决服务cpu的调度。在这里当执行**3步骤**必定是先获得锁（由于start需要调用native方法，并且在用完成之后在一切准备就绪了，但是并不表示一定在cpu上面执行，有没有真正执行取决服务cpu的调度，之后才会调用run方法，执行m1方法）。这里其实2个synchronized方法里面的Thread.sleep其实要不要是无所所谓的，估计是就为混淆增加难

度。3步骤执行的时候其实很快子线程也准备好了，但是由于synchronized的存在，并且是作用同一对象，所以子线程就只有必须等待了。由于main方法里面执行顺序是顺序执行的，所以必须是步骤3执行完成之后才可以到**4步骤**，而由于3步骤执行完成，子线程就可以执行m1了。这里就存在一个多线程谁先获取到的问题，如果**4步骤**先获取那么**main thread b=2000**，如果子线程m1获取到可能就b已经赋值成1000或者还没有来得及赋值**4步骤**就输出了可能结果就是**main thread b=1000**或者**main thread b=2000**，在这里如果把**6步骤**去掉那么b=执行在前和main thread b=在前就不确定了。但是由于6步骤存在，所以不管怎么都是main thread b=在前面，那么等于1000还是2000看情况，之后b=1000是一定固定的了。

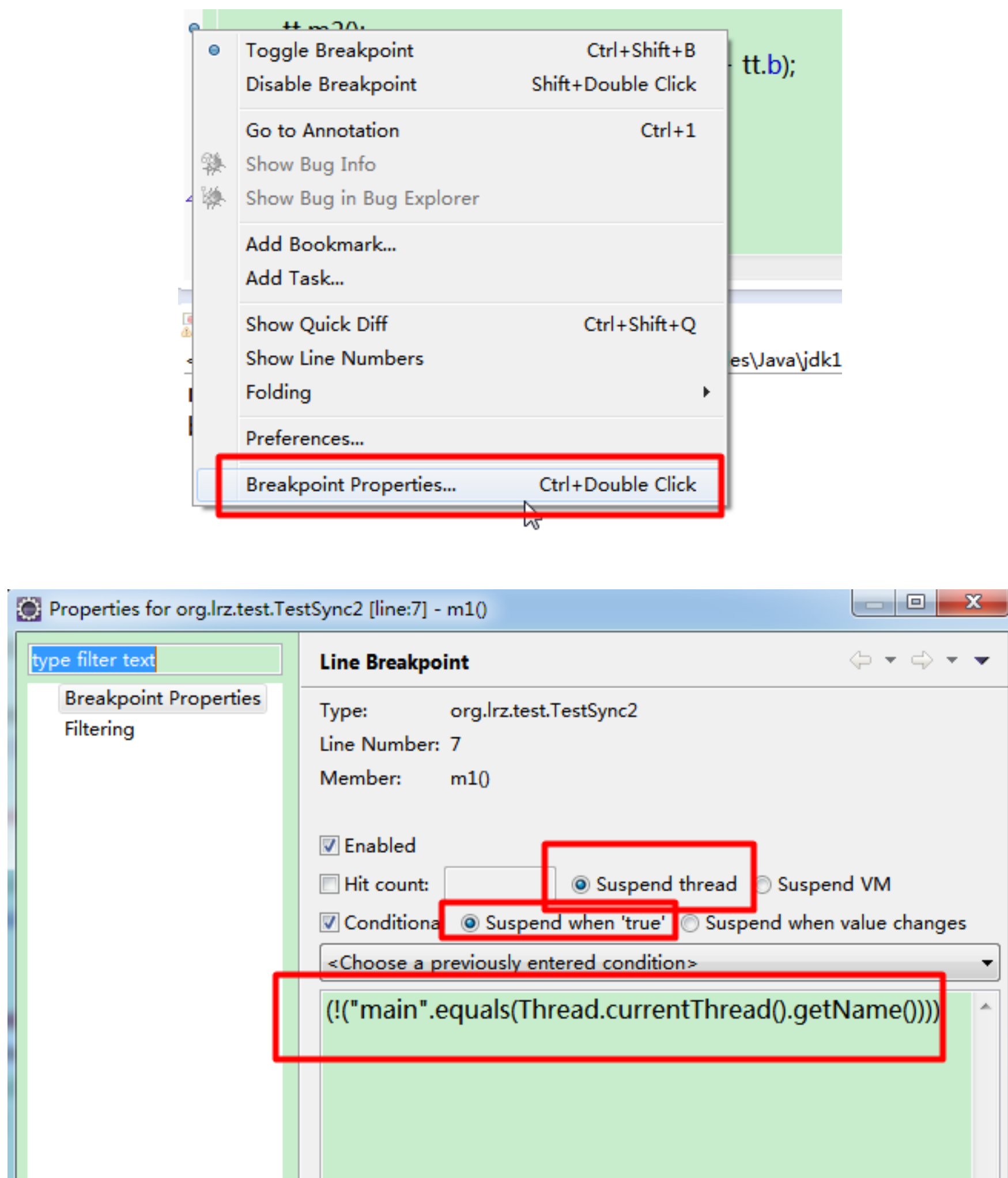
## 多线程一些建议

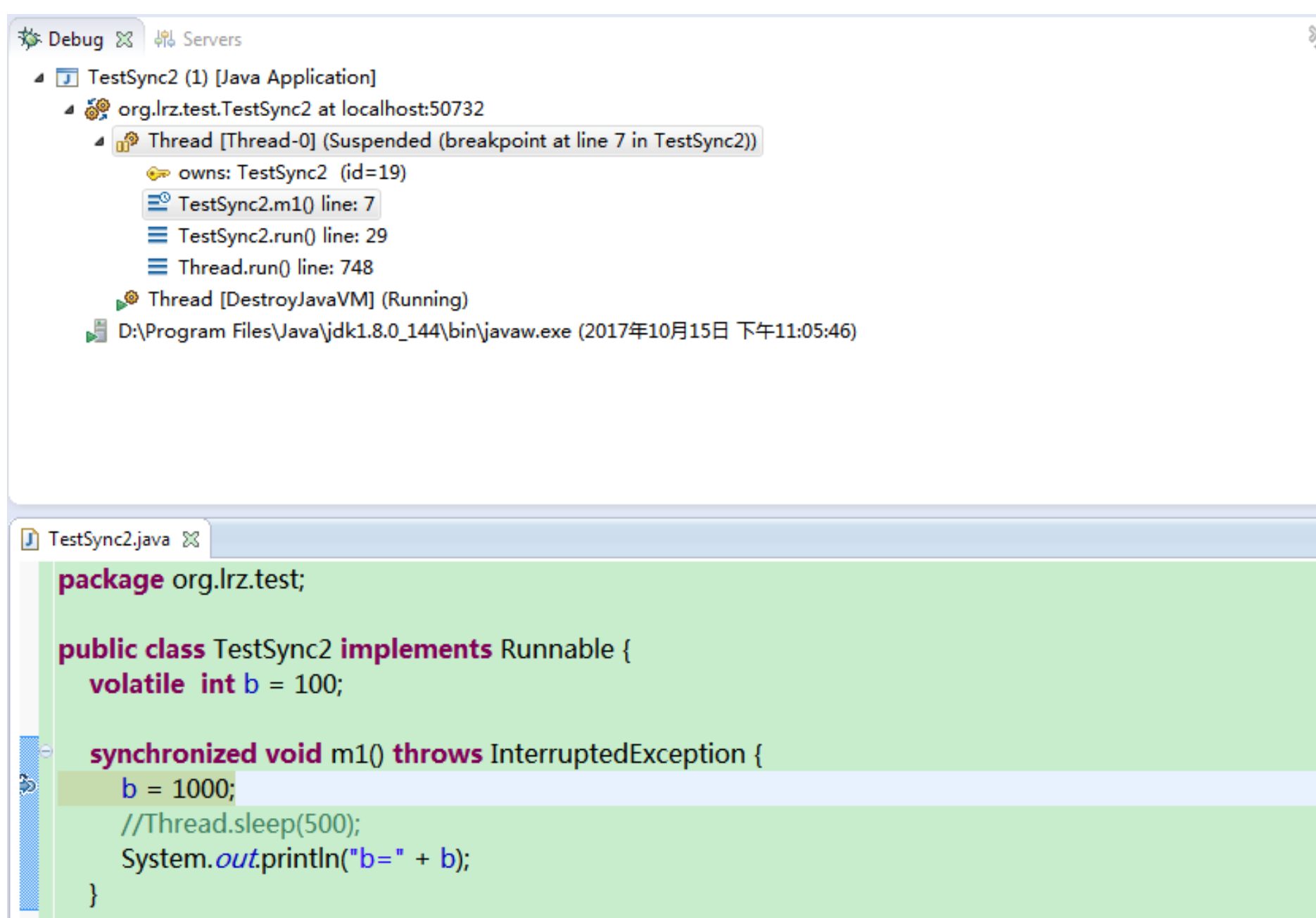
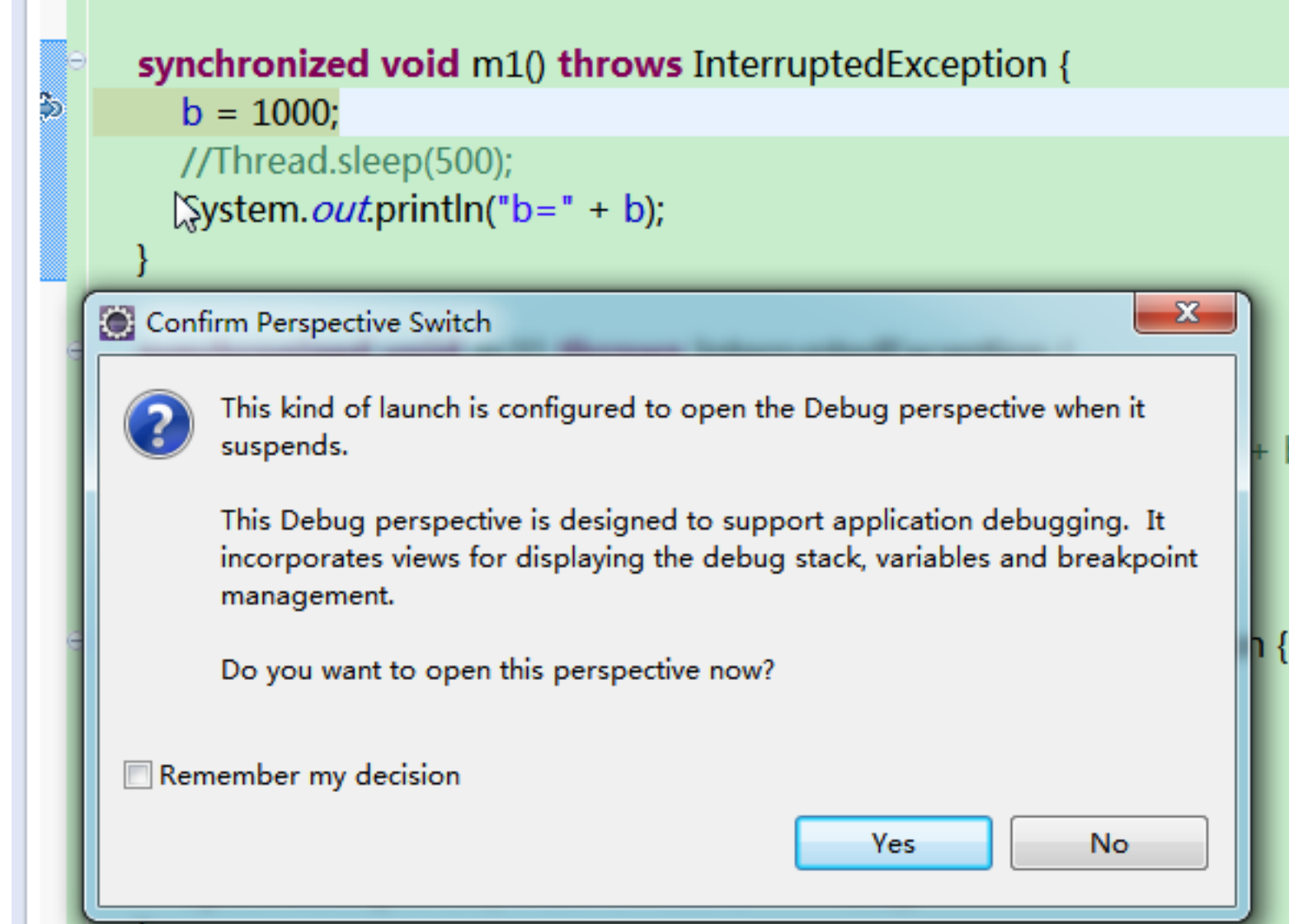
- 线程也很珍贵，所以建议使用**线程池**，线程池用的很多，后续准备分享下，特别重要，需要做到心中有数。
- 给线程起名字，当线上cpu高的时候，需要用到高级jstack，如果有名称就方便很多。
- 多线程特别需要注意线程安全问题，也需要了解jdk哪些是线程安全不安全，那样使用的时候不会出现莫名其妙的问题。

还有一些技巧后续文章分享在慢慢提，多线程特别重要，也特别难，希望大家也多多花心思在上面。

## 多线程的一些调试技巧

由于断点，所有线程经过断点的时候，都需要停下，导致这个点不停的断住，很难受，eclipse里面有条件断点，当满足条件的时候就可以停下来，那么这样就方便了。





关于线程dump分析以及后续线程内容会在后面继续分析分享。

## 本人其他JVM菜鸟进阶高手之路相关文章

- JVM菜鸟进阶高手之路十三（等你来战！！！！）([https://mp.weixin.qq.com/s?\\_\\_biz=MzU2NjlzNDk5NQ==&mid=2247483734&idx=1&sn=4ff0a25805ea3fa6af6f3a790b21c4d4&chksm=fcaedb4acbd9525cd3a72590fde64045b3321ac1b4a3b7e4fc3233cb0569d01c8bf211d9865a#rd](https://mp.weixin.qq.com/s?__biz=MzU2NjlzNDk5NQ==&mid=2247483734&idx=1&sn=4ff0a25805ea3fa6af6f3a790b21c4d4&chksm=fcaedb4acbd9525cd3a72590fde64045b3321ac1b4a3b7e4fc3233cb0569d01c8bf211d9865a#rd))
- JVM菜鸟进阶高手之路十二（jdk9、JVM方面变化， 蹭热度）([https://mp.weixin.qq.com/s?\\_\\_biz=MzU2NjlzNDk5NQ==&mid=2247483711&idx=1&sn=2e2e702f68d84d3c9dc9173b56de603c&scene=19#wechat\\_redirect](https://mp.weixin.qq.com/s?__biz=MzU2NjlzNDk5NQ==&mid=2247483711&idx=1&sn=2e2e702f68d84d3c9dc9173b56de603c&scene=19#wechat_redirect))
- JVM菜鸟进阶高手之路十一（eden survivor分配问题）([https://mp.weixin.qq.com/s?\\_\\_biz=MzU2NjlzNDk5NQ==&mid=2247483702&idx=1&sn=310d34a5a10f12959fd58581507e7b3f&scene=19#wechat\\_redirect](https://mp.weixin.qq.com/s?__biz=MzU2NjlzNDk5NQ==&mid=2247483702&idx=1&sn=310d34a5a10f12959fd58581507e7b3f&scene=19#wechat_redirect))
- JVM菜鸟进阶高手之路十（基础知识开场白）([https://mp.weixin.qq.com/s?\\_\\_biz=MzU2NjlzNDk5NQ==&mid=2247483703&idx=1&sn=d24bf3f2fe869e272e0](https://mp.weixin.qq.com/s?__biz=MzU2NjlzNDk5NQ==&mid=2247483703&idx=1&sn=d24bf3f2fe869e272e0))



- c1543cc8ebf42&scene=19#wechat\_redirect)
- JVM菜鸟进阶高手之路九（解惑）(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483673&idx=1&sn=bc11600868fd7d659bb6792101c1fd20&scene=19#wechat\_redirect)
  - JVM菜鸟进阶高手之路八（一些细节）(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483688&idx=1&sn=4d9023d7f556b308070915df999ce4a9&scene=19#wechat\_redirect)
  - JVM菜鸟进阶高手之路七（tomcat调优以及tomcat7、8性能对比）(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483695&idx=1&sn=3afc6e930ed602dad45bc1ae75afef7e&scene=19#wechat\_redirect)
  - JVM菜鸟进阶高手之路六（JVM每隔一小时执行一次Full GC）(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483691&idx=1&sn=61e77c56a654985eb44c3e8f91f088a9&scene=19#wechat\_redirect)
  - JVM菜鸟进阶高手之路五(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483691&idx=2&sn=e80b391d9bec94e24cad615dab156d46&scene=19#wechat\_redirect)
  - JVM菜鸟进阶高手之路四(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483680&idx=4&sn=d590b1dcb0533027b0e632fe6c1f4d21&scene=19#wechat\_redirect)
  - JVM菜鸟进阶高手之路三(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483680&idx=3&sn=0f0d3ca480c28470ef41686ad7d61a6c&scene=19#wechat\_redirect)
  - JVM菜鸟进阶高手之路二(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483680&idx=2&sn=c83333705f423ad344c61a8541457897&scene=19#wechat\_redirect)
  - JVM菜鸟进阶高手之路一(一次与笨神，阿飞近距离接触修改JVM)(https://mp.weixin.qq.com/s?\_\_biz=MzU2NjlzNDk5NQ==&mid=2247483680&idx=1&sn=079d801a792b03016704dfcc99158b6e&scene=19#wechat\_redirect)

如果读完觉得有收获的话，欢迎点赞加关注。



匠心零度 (/u/3def157aab07) ♂

写了 29557 字，被 423 人关注，获得了 266 个喜欢

(/u/3def157aab07) 写了 29557 字，被 423 人关注，获得了 266 个喜欢

+ 关注

欢迎点赞加关注！！！！

被以下专题收入，发现更多相似内容

- [互联网科技 \(/c/93d58e9169cb?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/93d58e9169cb?utm_source=desktop&utm_medium=notes-included-collection)
- [程序员 \(/c/NEt52a?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)
- [Java基础面试 \(/c/5bb29a2f4ae6?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/5bb29a2f4ae6?utm_source=desktop&utm_medium=notes-included-collection)
- [@IT·互联网 \(/c/V2CqjW?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/V2CqjW?utm_source=desktop&utm_medium=notes-included-collection)
- [java jvm \(/c/18966952a373?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/18966952a373?utm_source=desktop&utm_medium=notes-included-collection)
- [java进阶干货 \(/c/addfce4ca518?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/addfce4ca518?utm_source=desktop&utm_medium=notes-included-collection)
- [JAVA \(/c/6979101de49d?utm\\_source=desktop&utm\\_medium=notes-included-collection\)](/c/6979101de49d?utm_source=desktop&utm_medium=notes-included-collection)

展开更多

推荐阅读 更多精彩内容 > (/)

**JVM菜鸟进阶高手之路十（基础知识开场白） (/p/099bbaa5eacb0?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)**

转载请注明原创出处，谢谢！最近没有什么实战，准备把JVM知识梳理一遍，先以开发人员的交流来谈谈jvm这块的知识以及重要性，依稀记得2、3年前用

匠心零度 (/u/3def157aab07?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**JVM菜鸟进阶高手之路一(一次与笨神，阿飞近距离接... (/p/d32308a28cb1?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)**

转载请注明原创出处，谢谢！今天在JVMPocket群里面看见，阿牛发了一个gc截图，之后ak47截图了 Jstat是JDK自带的一个轻量级小工具。全称“Java

匠心零度 (/u/3def157aab07?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**200元我是如何在深圳生活一月的 (/p/3310b588a65a?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)**

我出生在农村，交通很烂，生活很艰苦。90年代才通电，村子里十几户人，在2000年之前，村子里连个上过高中的人都没人。也很少有人外出打工，大多数

青衣姑娘 (/u/5668063193f0?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**张充和：看见自己、爱上自己，世间不再颠沛流离（民... (/p/8d44e69048b0?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)**

我写过很多人物，要么是跌宕起伏、要么有很多噱头让人惊讶万分。唯有她，没有这些东西，我写不出跌宕、写不出冲突、写不出想看到的生死离别。因

蓝胖说说 (/u/604159f29174?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

**感谢简书——让我遇见我的程序猿男友 (/p/283ce035e641?utm\_campaign=maleskine&utm\_content=note&utm\_medium=pc\_all\_hots&utm\_source=recommendation)**

1024M=1G，1024是2的十次方，二进制计数的基本计量单位之一。今天是2017年10月24日。程序员就像是一个个1024，以最低调、踏实、核心的功能模

悠然小妖 (/u/9688e3f2e6df?utm\_campaign=maleskine&utm\_content=user&utm\_medium=pc\_all\_hots&utm\_source=recommendation)

