# 标题1

## 标题

## 标题

# 标题2

## 标题

### 标题

正文。

### 标题

#### 标题

正文。

# Java经典问题

## 双重检查加锁问题

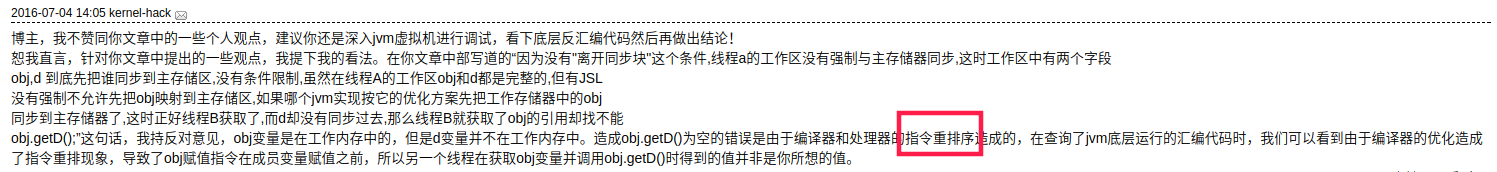
### 参考资料

<https://www.cnblogs.com/programerlrc/articles/5254675.html>

<https://blog.csdn.net/chenchaofuck1/article/details/51702129>

### 详细分析

#### 指令重排问题



### 解析方案

#### 内部私有静态类

public MyObject{

private static class instanceHoder{//内部私有的类，我特别用了小写开头。

static MyObject instance = new MyObject();

}

private static MyObect obj;

private Date d = new Data();

public Data getD(){return this.d;}

public static MyObect getInstance(){

return instanceHoder.instance;

}

}

private static class instanceHoder是类的定义并不会引起初始化。只有在首次调用getInstance时才会加载

instanceHoder类然后初始化instance实例。而静态初始化是由JVM来保证线程安全的，所以整个过程都不需要同

步参与，极大地提高了性能。

#### 线程安全的getInstance()方法

public static synchronized Singleton getInstance()

{

if (instance == null) //1

instance = new Singleton(); //2

return instance; //3

}

#### Volatile和synchronized组合方法

