Java 中的线程：对象锁 - II

切坦·库达尔卡

2021 年 6 月 14 日 警察

这是我之前关于“Java 中的线程：对象锁 - I”的文章的延续

本系列文章试图帮助读者了解 Java 中的锁概念。假设读者对 Java 和 Java 中的线程创建有一些先验知识。

介绍

这是我上一篇关于“ Java 中的线程处理：对象锁 - I ”的延续，其中我们从同一个处理器类对象“ `p1`”中创建了线程，并将它们与“ `this`”即“ `p1`”对象本身同步。对某些人来说，使用“ `this`”很难理解这个概念。因此，让我们对前面的程序稍加改动，以便更好地理解对象锁。

爪哇 缩小▲ 复制代码

```
locks.java
===========

class processor implements Runnable{
    Object objectLock;
    public processor(){
        objectLock = new Object();
    }
    public void run() {
        display();
    }
    public void display() {
        synchronized(objectLock) {
            for (int i = 0 ; i<= 5; i++) {
                System.out.println("i = " + i + " In thread" + Thread.currentThread());
            }
        }
    }
}

class locks {
    public static void main(String[] args) {
        processor p1 = new processor();
        Thread t1 = new Thread(p1, "t1");
        Thread t2 = new Thread(p1, "t2");
        t1.start();
        t2.start();
    }
}
```

```
Output
======

i = 0 In threadThread[t1,5,main]
i = 1 In threadThread[t1,5,main]
i = 2 In threadThread[t1,5,main]
i = 3 In threadThread[t1,5,main]
i = 4 In threadThread[t1,5,main]
i = 5 In threadThread[t1,5,main]
i = 0 In threadThread[t2,5,main]
i = 1 In threadThread[t2,5,main]
i = 2 In threadThread[t2,5,main]
i = 3 In threadThread[t2,5,main]
i = 4 In threadThread[t2,5,main]
i = 5 In threadThread[t2,5,main]
```

上述程序编译并成功运行以提供所需的输出。

我们保留了前面例子中的锁类。在"处理器"类的开头添加了一个细微的变化。我们已经创建了一个Object参考变量objLock。处理器类的默认构造函数通过创建Object类的对象来初始化它。objectLock线程在同步块中使用这个" "来锁定应该相互访问的关键区域。

在内部main()，当处理器对象" p1"被创建时，它的成员objectLock通过new Object(). 接下来两个线程的创建，t1以及t2基于同一对象上p1。"因此，这两个线程' t1'和' t2'看到一个共同的目标锁定这是' objLock'的那一刻，' t1'和' t2'的开始，双方开始争夺收购一个锁，以便进入同步块。让我们假设，" t1"获取" objectLock"并进入同步块。在这种情况下，" t2"不允许进入同步块，直到" t1"完成for循环的执行并退出同步块。t1objectLock" 然后由 " t2"获取，然后进入同步块并完成for循环的执行以产生所需的输出。

因此，两个线程" t1"和" t2"之间的同步发生是因为objectLock每个线程一次获取一个" "。请记住，" objectLock"属于" p1"，并且所有线程，即" t1"和" t2"都是从" p1"创建的。

兴趣点

因此"对象锁"确保所有依赖于同一对象的线程都是同步的。

历史

- 2021 年 6 月14日：初始版本

链接到第 1 部分

执照

分享

关于作者

切坦·库达尔卡
软件开发人员（高级）
印度

手表
该会员

我是一名软件工程师，拥有大约 7 年以上的经验。我的大部分经验是在存储技术方面。

评论和讨论

添加评论或问题 | 电子邮件提醒 | Search Comments

第一 页上一页 下一页

这篇文章不是必须的
Klaus Luedenscheidt 15-Jun-21 12:56

刷新 1

一般 新闻 建议 问题 错误 答案 笑话 赞美 咆哮 管理员

使用Ctrl+Left/Right 切换消息，Ctrl+Up/Down 切换主题，Ctrl+Shift+Left/Right 切换页面。