

真的有外星人吗?

假如这个世界上只剩下你一个人，当你正坐在屋子里的时候，这时突然响起了敲门声，那么会是谁呢?

C#线程系列讲座(5): 同步技术之Monitor

本文为原创，如需转载，请注明作者和出处，谢谢!

上一篇: C#线程系列讲座(4): 同步与死锁

在上一讲介绍了使用lock来实现线程之间的同步。实际上，这个lock是C#的一个障眼法，在C#编译器编译lock语句时，将其编译成了调用Monitor类。先看看下面的C#源代码:

```
public static void MyLock()  
{  
    lock (typeof(Program))  
    {  
    }  
}
```

上面的代码通过lock语句使MyLock同步，这个方法被编译成IL后，代码如图1所示。

| 2008年7月 | | | | | | |
|---------|----|----|----|----|----|----|
| < | 一 | 二 | 三 | 四 | 五 | > |
| 日 | | | | | | |
| 29 | 30 | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | 31 | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |

导航

博客园

首页

新随笔

联系

订阅 XML

管理

统计

随笔 - 85

文章 - 0

评论 - 650

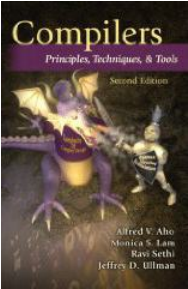
引用 - 29

公告

我的其他Blog

http://nokiaguy.blogjava.net
http://blog.csdn.net/nokiaguy

正在读的书



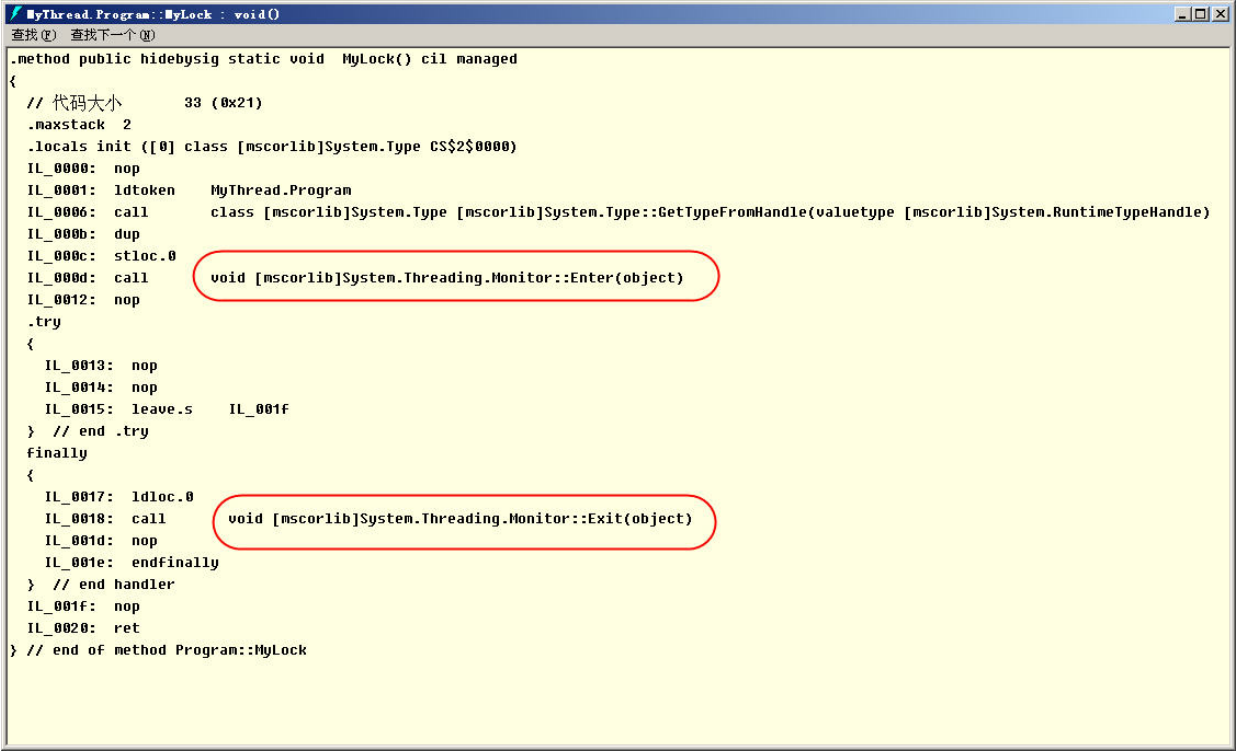


图1

从上图被标注的区域可以看到，一条lock语句被编译成了调用Monitor的Enter和Exit方法。Monitor在System.Threading命名空间中。lock的功能就相当于直接调用Monitor的Entry方法，所不同的是，lock方法在结束后，会自动解除锁定，当然，在IL中是调用了Monitor的Exit方法，但在C#程序中，看起来是自动解锁的，这类似于C#中的using语句，可以自动释放数据库等的资源。但如果直接在C#源程序中使用Monitor类，就必须调用Exit方法来显式地解除锁定。如下面的代码所示：

```
Monitor.Entry(lockObj);
try
{
    // lockObj的同布区
}
catch (Exception e)
{
    // 异常处理代码
}
finally
{
    Monitor.Exit(lockObj); // 解除锁定
}
```

Exit方法最后在finally里调用，这样无论在方法在发生异常、返回还是正常执行，都会执行到finally，并调用Exit方法解除锁定。

Monitor类不仅可以完全取代lock语句（如果只使用lock语句本身的功能，最好还是直接用lock语句吧），还可以使用TryEntry方法设置

我的最新闪存

今天真热

与我联系

发短消息

搜索

常用链接

我的随笔

我的空间

我的短信

我的评论

更多链接

留言簿

给我留言

查看留言

我参加的小组

创业交流

设计模式

AJAX

.NET 3.x

写书译书小组

博客园精华集出版小组

一个锁定超时，单位是毫秒。如下面的代码所示：

```
if(Monitor.TryEntry(lockObj, 1000))
{
    try
    {
    }
    finally
    {
        Monitor.Exit(lockObj);
    }
}
else
{
    // 超时后的处理代码
}
```

上面的代码设置了锁定超时时间为1秒，也就是说，在1秒中后，lockObj还未被解锁，TryEntry方法就会返回false，如果在1秒之内，lockObj被解锁，TryEntry返回true。我们可以使用这种方法来避免死锁，如下面的代码所示：

```
class Program
{
    private static Object objA = new Object();
    private static Object objB = new Object();
    public static void LockA()
    {
        if (Monitor.TryEnter(objA, 1000))
        {
            Thread.Sleep(1000);
            if (Monitor.TryEnter(objB, 2000))
            {
                Monitor.Exit(objB);
            }
            else
            {
                Console.WriteLine("LockB timeout");
            }
            Monitor.Exit(objA);
        }
        Console.WriteLine("LockA");
    }
    public static void LockB()
    {
        if (Monitor.TryEnter(objB, 2000))
        {
            Thread.Sleep(2000);
            if (Monitor.TryEnter(objA, 1000))
            {
                Monitor.Exit(objA);
            }
        }
    }
}
```

沈阳.NET俱乐部
《编译原理》

我参与的团队

北京.NET俱乐部(0/0)
沈阳.NET俱乐部(0/0)
CLR基础研究团队(0/0)

我的标签

C#(4)
SQL Server(4)
数据库(3)
web(2)
.net(2)
c++(2)
CTE(2)
公共表表达式(2)
SQL Server 2005(1)
递归(1)
更多

随笔分类(169)

获奖作品(2) (rss)
原创(47) (rss)
.net高级技术(7) (rss)
.net入门技术(1) (rss)
.net新特性(2) (rss)
ajax(4) (rss)
algorithm(14) (rss)
C#(13) (rss)
C/C++(9) (rss)
database(11) (rss)
delphi(2) (rss)
IE (6至8) (1) (rss)
javascript(5) (rss)
linux(1) (rss)
MSIL(2) (rss)
mysql(4) (rss)
open source(3) (rss)
SQL Server(11) (rss)
VBA(1) (rss)
web(8) (rss)
WPF(1) (rss)
wxWidgets(1) (rss)
安全(2) (rss)
进程、线程、并发(5) (rss)
设计模式(1) (rss)
网络营销(1) (rss)
宇宙探秘(1) (rss)
杂七杂八(9) (rss)

随笔档案(85)

2009年8月 (1)

```
        else
        {
            Console.WriteLine("LockA timeout");
        }
        Monitor.Exit(objB);
    }
    Console.WriteLine("LockB");
}
public static void Main()
{
    Thread threadA = new Thread(LockA);
    Thread threadB = new Thread(LockB);
    threadA.Start();
    threadB.Start();
    Thread.Sleep(4000);
    Console.WriteLine("线程结束");
}
```

上面的代码是在上一讲举的死锁的例子，但在这一讲将lock语句改成了TryEntry方法，而且设置了锁定超时间，由于在等待一定时间后，不管被锁定的对象是否被解锁，TryEntry方法都会返回，因此，上面的代码是不会死锁的。运行上面的代码的结果如图2所示。

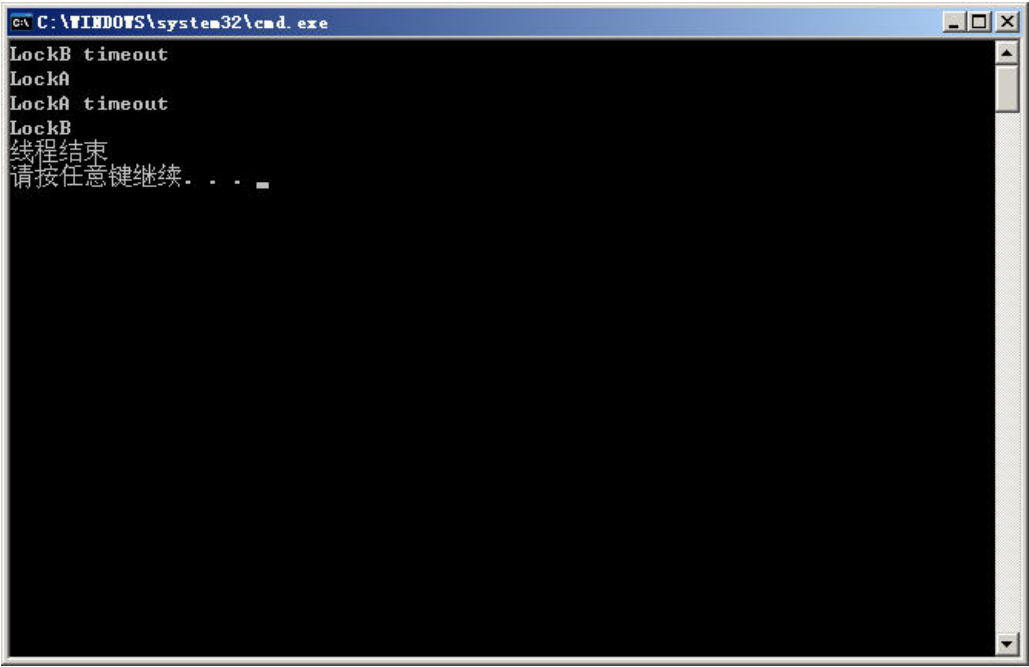


图2

如果TryEntry方法的超时时间为System.Threading.Timeout.Infinite，TryEntry方法就相当于Entry方法，如果超时时间为0，不管是否解锁，TryEntry方法都会立即返回。

2009年7月 (2)
2009年6月 (3)
2009年3月 (3)
2009年2月 (6)
2009年1月 (5)
2008年12月 (3)
2008年10月 (4)
2008年9月 (3)
2008年8月 (3)
2008年7月 (6)
2008年6月 (9)
2008年5月 (36)
2008年4月 (1)

相册

images

其他

微软认证

Blogs

anytao

leo zhang (职业规划师)

刘江 (图灵主编)

开源

微软开源网站

协议

MSN Messenger

积分与排名

积分 - 145137

排名 - 381

最新评论 XML

1. Re:.net framework3.5新特性2: var、初始化、匿名类和扩展方法
我看了文章和大家的说评语，楼主写的也不错，大家说也很好，让我对var又加深对var的了解。

--Dean123

2. Re:全排列算法原理和实现
回8楼和9楼的，在permut函数for循环中，调用完permut之后还要执行一次swap的，要把数组还原成未调用permut之前的状态，保证下一次循环或者上一级循环swap数组元素的正确。愚见愚见。...

--Little_Angel

3. Re:移动的MobileMarket个人终于可以上传软件了
还是即将上线，敬请期待~? ~?

--DAP

《银河系列原创教程》发布

00

(请您对文章做出评价)

posted on 2008-07-31 21:20 银河使者 阅读(2281) 评论(8) 编辑 收藏 网摘 所属分类: 进程、线程、并发, .net高级技术, 原创

评论

#1楼 2008-07-31 21:23 uwebs[未注册用户]
顶了再看 回复 引用

#2楼 2008-07-31 22:02 侯垒
看完了就顶. 回复 引用 查看

#3楼 2008-08-01 00:17 BZZ
占个位, 慢慢看
回复 引用 查看

#4楼 2008-08-02 20:56 陈晨
能介绍下Monitor.Wait方法和Monitor.Pulse方法吗?
一直没能完全理解这两个方法 回复 引用 查看

#5楼[楼主] 2008-08-13 23:11 银河使者
@ 陈晨

实际上, 在拥有某个对象锁的线程内调用Wait方法, 就相当于暂时将被锁定的对象释放, 然后将调用Wait方法的线程放到等待队列中. 这时其他等待刚才被释放的对象锁的线程就会获得这个对象. 如果对这个对象调用Pulse方法, 就会通知等待队列中曾经在这个对象上调用Wait方法的第一个线程(栈顶的线程)可以重新获得这对象锁, 但系统只是将这个线程从等待队列放到就绪队列中, 至于什么时候这个线程会重新获得对象锁, 就由系统调度算法决定了. 如下面的代码如示:

```
private Object obj = new Object();
private void method1()
{
    lock(obj)
    {
        Monitor.Wait(obj); // 将method1所在的线程放到等待队列中
    }
}

private void method2()
{
    lock(obj)
    {
        // 当在method1方法中调用Wait时, method2所在的线程可能会获得obj锁
    }
}

private void method3()
```

4. Re: 移动的MobileMarket个人终于可以上传软件了
哦, 好久没去关注这方面了, 去看看
--peterzb

5. Re: 想抢先体验Android操作系统的魅力吗?
那就使用Android LiveCD吧!
买不起, 只有这样体验下了~~~
--J.Motto

阅读排行榜

- 1. 全排列算法原理和实现(6165)
- 2. C# 线程系列讲座(1): BeginInvoke和EndInvoke方法(5552)
- 3. C# 线程系列讲座(2): Thread类的应用(4360)
- 4. 实现Web程序的自动登录(3762)
- 5. 用C# 2.0实现网络蜘蛛(WebSpider)(3704)

评论排行榜

- 1. 用VC实现洪水攻击程序(38)
- 2. C# 线程系列讲座(1): BeginInvoke和EndInvoke方法(34)
- 3. .net framework3.5新特性1: Lambda表达式(33)
- 4. .net framework3.5新特性2: var、初始化、匿名类和扩展方法(29)
- 5. 实现Web程序的自动登录(28)

```
{
Monitor.Pulse(obj);// 通知method1所在的线程可以获得obj锁了, 但这个程被放在了就绪队列中
}
```

[回复](#) [引用](#) [查看](#)

#6楼 2008-09-30 11:46 [追忆似水流年](#)

上面的代码设置了锁定超时时间为1秒, 也就是说, 在1秒中后, lockObj还未被解锁, TryEntry方法就会返回false, 如果在1秒之内, lockObj被解锁, TryEntry返回true 我个人觉得这段话这样表达比较好理解:上面的代码设置了锁定超时时间为1秒, 也就是说, 在1秒时间内去获取锁定对象lockObj,如果未获取,TryEntry方法就会返回false.不会去执行TryEntry方法体内的语句;反之,如果在1秒内获取了lockObj,就去执行TryEntry方法体内的语句. 还有这个例子中时间的设置,我个人觉得应该设置的区分度比较大才容易得出结果.我拷贝你的代码运行,得出的结果刚好与楼主的相反,可能跟系统电脑配置有一定关系.不是之处,希望大家探讨! [回复](#) [引用](#) [查看](#)

#7楼 2008-09-30 21:33 [microle\[未注册用户\]](#)

顶完再看~ [回复](#) [引用](#)

#8楼 2009-02-27 13:41 [haohaohuai\[未注册用户\]](#)

没顶,看了,没看懂! [回复](#) [引用](#)

[免费学习asp.net课程](#)
本市人员可享受50-100%政府补贴 合格颁发国家职业资格和微软双认证
[www.zili.cn](#)

[刷新评论列表](#) [刷新页面](#) [返回页首](#)

发表评论

昵称: [\[登录\]](#) [\[注册\]](#)

主页:

邮箱: (仅博主可见)

评论内容: [闪存](#) [个人主页](#)

[登录](#) [注册](#)

[使用Ctrl+Enter键快速提交评论]

[个人主页上线测试中](#)

今天你闪了吗?

2009博客园纪念T恤

[上海微软认证唯一授权考点](#)
微软认证讲师MCT全程授课 全国最低价考费，100%保过
[www.itwin.com](#)
[寻找18-28岁待业者](#)
权威:华浦ISEP国际软件工程师 高薪: 半年IT认证+高薪就业
[www.isen.com.cn](#)
[专业Web报表工具-博计报表](#)
支持SqlServer，不用ReportViewer控件 开发Web报表不用写SQL，不用GridView
[www.hongzhenreport.com](#)
[找个老外学英语](#)
迅速提高英语口语 口语就要这样练
[www.MarsEnglish.com](#)



China-pub 计算机图书网上专卖店! 6.5万品种 2-8折!
China-Pub 计算机绝版图书按需印刷服务
链接: 切换模板
导航: 网站首页 个人主页 社区 新闻 博问 闪存 网摘 招聘 找找看 Google搜索

最新**IT**新闻:
Delphi 2010初体验
谷歌经济学家: 搜索关键词表明美国经济正复苏
Facebook应吸取谷歌经验避免重蹈雅虎覆辙
唐骏传授成功秘笈: 创业要有自己的“杀手锏”
商业周刊: 企业用户不愿甲骨文壮大 称其店大欺客

相关链接:
博客园.NET频道, 专业.NET技术门户
ASP.NET MVC 专题, 从零开始学.NET技术
英文技术文章推荐
系列教程: C#多线程学习
ASP.NET技术内幕
碰到技术难题? 马上使用找找看!
技术成就梦想, 最鲜活的IT新闻资讯