

C#多线程学习(六) 互斥对象

如何控制好多个线程相互之间的联系，不产生冲突和重复，这需要用到互斥对象，即：**System.Threading** 命名空间中的 **Mutex** 类。

我们可以把**Mutex**看作一个出租车，乘客看作线程。乘客首先等车，然后上车，最后下车。当一个乘客在车上时，其他乘客就只有等他下车以后才可以上车。而线程与**Mutex**对象的关系也是如此，线程使用**Mutex.WaitOne()**方法等待**Mutex**对象被释放，如果它等待的**Mutex**对象被释放了，它就自动拥有这个对象，直到它调用**Mutex.ReleaseMutex()**方法释放这个对象，而在此期间，其他想要获取这个**Mutex**对象的线程都只有等待。

下面这个例子使用了**Mutex**对象来同步四个线程，主线程等待四个线程的结束，而这四个线程的运行又是与两个**Mutex**对象相关联的。

其中还用到**AutoResetEvent**类的对象，可以把它理解为一个信号灯。这里用它的有信号状态来表示一个线程的结束。

```
// AutoResetEvent.Set()方法设置它为有信号状态  
// AutoResetEvent.Reset()方法设置它为无信号状态
```

Mutex 类的程序示例：

```
using System;
```

```
using System.Threading;

namespace ThreadExample
{
    public class MutexSample
    {
        static Mutex gM1;
        static Mutex gM2;
        const int ITERS = 100;
        static AutoResetEvent Event1 = new AutoResetEvent(false);
        static AutoResetEvent Event2 = new AutoResetEvent(false);
        static AutoResetEvent Event3 = new AutoResetEvent(false);
        static AutoResetEvent Event4 = new AutoResetEvent(false);

        public static void Main(String[] args)
        {
            Console.WriteLine("Mutex Sample ...");
            //创建一个Mutex对象, 并且命名为MyMutex
            gM1 = new Mutex(true, "MyMutex");
            //创建一个未命名的Mutex 对象.
            gM2 = new Mutex(true);
            Console.WriteLine(" - Main Owns gM1 and gM2");

            AutoResetEvent[] evs = new AutoResetEvent[4];
            evs[0] = Event1; //为后面的线程t1,t2,t3,t4定义AutoResetEvent对象
            evs[1] = Event2;
            evs[2] = Event3;
```

```
    evs[3] = Event4;

    MutexSample tm = new MutexSample( );

    Thread t1 = new Thread(new ThreadStart(tm.t1Start));
    Thread t2 = new Thread(new ThreadStart(tm.t2Start));
    Thread t3 = new Thread(new ThreadStart(tm.t3Start));
    Thread t4 = new Thread(new ThreadStart(tm.t4Start));

    t1.Start( );// 使用Mutex.WaitAll()方法等待一个Mutex数组中的对象全部被释放
    t2.Start( );// 使用Mutex.WaitOne()方法等待gM1的释放
    t3.Start( );// 使用Mutex.WaitAny()方法等待一个Mutex数组中任意一个对象被释放
    t4.Start( );// 使用Mutex.WaitOne()方法等待gM2的释放

    Thread.Sleep(2000);

    Console.WriteLine(" - Main releases gM1");
    gM1.ReleaseMutex( ); //线程t2,t3结束条件满足

    Thread.Sleep(1000);

    Console.WriteLine(" - Main releases gM2");
    gM2.ReleaseMutex( ); //线程t1,t4结束条件满足

    //等待所有四个线程结束
    WaitHandle.WaitAll(evs);

    Console.WriteLine("… Mutex Sample");
    Console.ReadLine();
}

public void t1Start( )
```

```
{  
  
    Console.WriteLine("t1Start started, Mutex.WaitAll(Mutex[])");  
    Mutex[] gMs = new Mutex[2];  
    gMs[0] = gM1; //创建一个Mutex数组作为Mutex.WaitAll()方法的参数  
    gMs[1] = gM2;  
    Mutex.WaitAll(gMs); //等待gM1和gM2都被释放  
    Thread.Sleep(2000);  
    Console.WriteLine("t1Start finished, Mutex.WaitAll(Mutex[]) satisfied");  
    Event1.Set( ); //线程结束, 将Event1设置为有信号状态  
}  
  
public void t2Start( )  
{  
  
    Console.WriteLine("t2Start started, gM1.WaitOne( )");  
    gM1.WaitOne( ); //等待gM1的释放  
    Console.WriteLine("t2Start finished, gM1.WaitOne( ) satisfied");  
    Event2.Set( ); //线程结束, 将Event2设置为有信号状态  
}  
  
public void t3Start( )  
{  
  
    Console.WriteLine("t3Start started, Mutex.WaitAny(Mutex[])");  
    Mutex[] gMs = new Mutex[2];  
    gMs[0] = gM1; //创建一个Mutex数组作为Mutex.WaitAny()方法的参数  
    gMs[1] = gM2;  
    Mutex.WaitAny(gMs); //等待数组中任意一个Mutex对象被释放  
    Console.WriteLine("t3Start finished, Mutex.WaitAny(Mutex[])");  
    Event3.Set( ); //线程结束, 将Event3设置为有信号状态  
}
```

```
public void t4Start( )
{
    Console.WriteLine("t4Start started, gM2.WaitOne( );");
    gM2.WaitOne( );//等待gM2被释放
    Console.WriteLine("t4Start finished, gM2.WaitOne( );");
    Event4.Set( );//线程结束，将Event4设置为有信号状态
}
}
}
```

程序的输出结果：

```
Mutex Sample ...
- Main Owns gM1 and gM2
t1Start started, Mutex.WaitAll(Mutex[])
t2Start started, gM1.WaitOne( )
t3Start started, Mutex.WaitAny(Mutex[])
t4Start started, gM2.WaitOne( )
- Main releases gM1
t2Start finished, gM1.WaitOne( ) satisfied
t3Start finished, Mutex.WaitAny(Mutex[])
- Main releases gM2
t1Start finished, Mutex.WaitAll(Mutex[]) satisfied
t4Start finished, gM2.WaitOne( )
... Mutex Sample
```

从执行结果可以很清楚地看到，线程**t2,t3**的运行是以**gM1**的释放为条件的，而**t4**在**gM2**释放后开始执行，**t1**则在**gM1**和**gM2**都被释放了之后才执行。**Main()**函数最后，使用**WaitHandle**等待所有

的AutoResetEvent对象的信号，这些对象的信号代表相应线程的结束。

相关文章推荐：

使用 .NET 实现 Ajax 长连接 (Part 2 - Mutex Wait & Signal)

0

0

(请您对文章做出评价)

posted on 2008-03-23 16:06 钢钢 阅读(2140) 评论(3) 编辑 收藏 网摘 所属分类: C#

评论

#1楼 2009-05-15 10:36 高东

我是刚开始学习 C# 的多线程。
你的这一系列文章是我的启蒙。。
我运行互斥对象的代码时。
报了一个 由于出现被放弃的 mutex，等待过程结束。（AbandoneMutexException）

望请解决。。
[回复](#) [引用](#)

#2楼 2009-06-27 20:53 法巫师

结果还不一样

```
Mutex Sample
- Main Owns gM1 and gM2
t1Start started, Mutex.WaitAll(Mutex[])
t2Start started, gM1.WaitOne( )
t3Start started, Mutex.WaitAny(Mutex[])
t4Start started, gM2.WaitOne( )
- Main releases gM1
t3Start finished, Mutex.WaitAny(Mutex[])
t2Start finished, gM1.WaitOne( ) satisfied
- Main releases gM2
t4Start finished, gM2.WaitOne( )
t1Start finished, Mutex.WaitAll(Mutex[]) satisfied
```

Mutex Sample

[回复](#) [引用](#) [查看](#)

[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

[寻找18-28岁待业者](#)

权威:华浦ISEP国际软件工程师 免费:职场规划帮你找到好工作

www.isen.com.cn

[LabVIEW—引领多核技术时代](#)

自动多线程,充分利用多核提高系统性能 大大简化多核编程的复杂度

www.ni.com/multicore/zhs/

[ASP.NET报表高效开发--博计](#)

不用GridView, DataGridView做Web报表 无控件, 纯Web打印, 轻松设计复杂报表

www.bonzerreport.com

[Pitemidlist c#](#)

Explorer shell controls for .NET Written in 100% C#.

Free Trial!

www.iam-software.com/developer/



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:

系列教程: [C#多线程学习](#)

[08年博客园纪念T恤珍藏版](#)

[大学英语六级677分沪友谈英语学习](#)

[少儿英语学习卡片 \(学习用具\)](#)

ASP.NET MVC 专题，从零开始学.NET技术
博客园.net频道上线啦！
职场点睛：大学生应该如何学习英语

导航

博客园

首页

新随笔

联系

订阅 XML

管理

统计

随笔 - 166

文章 - 27

评论 - 579

引用 - 68

公告

穷则独善其身
达则兼济天下



新闻

- Delphi 2010初体验 2小时前
- 谷歌经济学家：搜索关键词表
明美国经济正复苏 2小时前
- Facebook应吸取谷歌经验避免
重蹈雅虎覆辙 4小时前
- 唐骏传授成功秘笈：创业要有
自己的杀手铜” 4小时前

在线词典

查词

我的最新闪存

链接

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

web标准设计

jQuery

读书(Books)

ASP.NET

AJAX

.Net+MySQL

博客园期刊团队

我参与的团队

[.NET 控件与组件开发\(0/0\)](#)
[湖南.NET俱乐部\(0/0\)](#)
[ASP.NET AJAX \(Atlas\)学习\(0/0\)](#)
[asp.net开发团队\(0/0\)](#)

随笔分类(124)

[AJAX\(4\) \(rss\)](#)
[C#\(22\) \(rss\)](#)
[DotNET\(27\) \(rss\)](#)
[GDI\(2\) \(rss\)](#)
[JavaScript\(20\) \(rss\)](#)
[MS_SQL\(4\) \(rss\)](#)
[MySQL\(1\) \(rss\)](#)
[Oracle\(6\) \(rss\)](#)
[编程规范\(6\) \(rss\)](#)
[设计模式\(2\) \(rss\)](#)
[我的项目\(9\) \(rss\)](#)
[相关知识\(21\) \(rss\)](#)

随笔档案(166)

[2008年12月 \(1\)](#)
[2008年11月 \(1\)](#)
[2008年10月 \(1\)](#)
[2008年9月 \(2\)](#)
[2008年8月 \(4\)](#)
[2008年7月 \(1\)](#)
[2008年6月 \(4\)](#)
[2008年5月 \(4\)](#)
[2008年4月 \(7\)](#)
[2008年3月 \(8\)](#)
[2008年2月 \(7\)](#)
[2008年1月 \(19\)](#)
[2007年12月 \(26\)](#)
[2007年11月 \(21\)](#)
[2007年10月 \(24\)](#)
[2007年9月 \(15\)](#)
[2007年8月 \(12\)](#)
[2007年7月 \(9\)](#)

文章分类(26)

Java学习(1) (rss)

读书笔记(16) (rss)

经典收藏(9) (rss)

收藏夹(199)

ASP.NET 技术(128) (rss)

Enterprise Library(5) (rss)

PetShop 4架构分析 (15) (rss)

网络教程 (36) (rss)

优秀Blog地址(15) (rss)

Blogs

Guushuuse

LoveCherry

Nicky

Q.yuhen

Scott Guthrie

怪怪

蝈蝈俊

黄忠成

李天平

孟岩

木子

萧寒

张子阳

Tools

Convert vbnet to C# (rss)

DotNet API (rss)

My Diagram

Text Translation (rss)

WorldLingo (rss)

Websites

AspAlliance (rss)

C# Corner (rss)

C#开源资源大全 (rss)

C#语言参考视频 (rss)

cnBeta (rss)	cnBeta (rss)
CodePlex (rss)	CodePlex (rss)
codeusing.com (rss)	codeusing.com (rss)
connectionstrings.com (rss)	connectionstrings.com (rss)
CSDN DotNet	CSDN DotNet
dotNET_程序小作坊 (rss)	dotNET_程序小作坊 (rss)
DotNetTools (rss)	DotNetTools (rss)
DotNet开源社区	DotNet开源社区
http://csharp-source.net (rss)	http://csharp-source.net (rss)
ItPub论坛 (rss)	ItPub论坛 (rss)
MSDN中国 (rss)	MSDN中国 (rss)
MSProject 开源技术 (rss)	MSProject 开源技术 (rss)
planet-source-code.com (rss)	planet-source-code.com (rss)
Sawin (rss)	Sawin (rss)
开源中国	开源中国
懒人图库 (rss)	懒人图库 (rss)
浪曦视频在线 (rss)	浪曦视频在线 (rss)
中国协议分析网 (rss)	中国协议分析网 (rss)

积分与排名	积分与排名
积分 - 216335	积分 - 216335
排名 - 224	排名 - 224

最新评论 XML	最新评论 XML
1. Re:程序员心理小测试：你是否患上抑郁症？ 天啊,我得了13分,我是不是应该去曙光医院看看啊! --purplesun	1. Re:程序员心理小测试：你是否患上抑郁症？ 天啊,我得了13分,我是不是应该去曙光医院看看啊! --purplesun
2. Re:Donald Knuth 简介 虽然学编程一年了,但对这行业总是迷茫,怎样才算会编程也不知道 --Greenhant	2. Re:Donald Knuth 简介 虽然学编程一年了,但对这行业总是迷茫,怎样才算会编程也不知道 --Greenhant
3. Re:一个C#的加锁解锁示例 没看懂什么意思呀 --王继	3. Re:一个C#的加锁解锁示例 没看懂什么意思呀 --王继
4. Re:将GridView导入到Excel和word(完全可实现) 我的加了还是倒不出来你呢	4. Re:将GridView导入到Excel和word(完全可实现) 我的加了还是倒不出来你呢

--ASP.NET爱妃

5. Re:程序员心理小测试：你是否患上抑郁症？

看来，我也得了抑郁怔拉----该怎么办呢----真的好烦啊-----

--伤心哥

- 阅读排行榜
-
1. C#多线程学习(一) 多线程的相关概念(9310)
 2. IIS日志分析方法及工具(9230)
 3. Javascript 刷新框架及页面的方法总集(9020)
 4. (新)中华人民共和国劳动合同法(8263)
 5. C#多线程学习(二) 如何操纵一个线程(7237)

- 评论排行榜
-
1. 论不使用ObjectDataSource之数据绑定控件纯代码实现是否更优化?(52)
 2. 程序员心理小测试：你是否患上抑郁症？ (51)
 3. 电视剧《奋斗》能叫奋斗吗？45)
 4. (新)中华人民共和国劳动合同法(42)
 5. Donald Knuth 简介(21)