

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

使用多线程创建高响应智能客户端应用程序

付仲恺
微软特邀开发专家

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

课程准备

- 具备使用VS.NET进行Windows操作系统之上托管应用程序开发的相关知识

Level 200



MSDN Webcasts

议题

- 介绍
- 原生Windows® 进程和线程
- 创建和终止线程
- 线程内更新UI控件
- 线程与线程池
- 线程安全与同步对象

介绍

多线程应用程序

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 具有同时有多条“执行线路”的能力
- 行为无法预知并且每次调试时行为不同
- 线程增加了程序的复杂性
- 多线程不但可以用于我们所关注的SmartClient当中，也可以用于任何应用程序当中

介绍

为什么要讨论多线程应用程序？

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 向用户提供及时响应
- 在运行时层面并行执行任务
- 获得更好的全面的应用程序性能
- 硬件资源得到全面利用

您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

demo

为什么需要多线程？

进程和线程

进程概览

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 进程是正在执行的程序的一个实例
- 进程只有在执行时才存在
- 进程拥有**4GB**地址空间以包含应用程序的代码和数据
- 每个进程至少有一个线程

进程和线程

线程概览

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 线程是“执行单元”，负责进程内所包含代码的执行
- 调度器识别并且只调度线程，以允许程序并行运行
- 线程使用处理器寄存器，并且当其它线程开始运行时栈和其它资源都会被保存起来
- 线程可以运行在不同的优先级上
- 一个进程能够拥有多个线程

进程和线程

基本概念

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 只有当确实需要时才使用处理器
- 不要使用轮询的方式来同步线程
- 使用同步对象
- 多线程增加了应用程序的复杂性

创建和终止线程

线程对象

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 托管对象构建在Windows基本线程模型之上
- `System.Threading.Thread`类封装了Windows线程功能
- `Thread`类提供了如下功能:
 - 创建线程
 - 启动线程
 - 设置线程优先级
 - 挂起 / 恢复线程执行
 - 终止线程

创建和终止线程

启动线程

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 启动线程非常容易，但是代价“昂贵”

```
Thread myThread= new Thread(new ThreadStart(MyWorkerThread));  
myThread.Start();  
  
private void MyWorkerThread()  
{  
    while (! workerThreadDone)  
    {  
        // simulate some processing  
        Thread.Sleep(1000);  
    }  
}
```

创建和终止线程

终止线程

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 当所有的线程都结束时，该应用程序才能够被终止
- 由Thread.Abort引发异常的方法，在通常环境中不是终止线程结束的最好方法
- 在应用程序终止前，所有前台线程都必须终止

终止线程

采用哪种方案？

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 可以采用不同方法来终止线程
- 使用 **Boolean**（实例）变量
- 使用 **Thread.Abort**
- 使用后台线程
- 使用 **Thread.Join** 来获知线程何时终止

您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

demo

创建和终止线程

更新UI控件

介绍Control.Invoke

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 控件应该只被创建它们的线程更新
- 使用Control.Invoke来从其他线程委托更新控件
- 同步和异步Invoke方法均可用
- 支持参数传递
- 当从其它线程更新UI控件时，会抛出异常

更新UI控件 忽略Control.Invoke

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 绝对不要直接更新UI控件

```
private void WorkerThread()  
{  
    statusBar1.Text = "WorkerThreadactive";  
  
    while (!workerThreadDone)  
    {  
        Thread.Sleep(1000);  
    }  
}
```

更新UI控件 正确方法

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 使用Control.(Begin)Invoke替代

```
private delegate void LabelUpdater (string labelText);

private void WorkerThread()
{
    LabelUpdater lblUpdater= new LabelUpdater(UpdateLabel);
    this.Invoke(lblUpdater, new object[] {"WorkerThreadactive"});
    while (!workerThreadDone)
    {
        Thread.Sleep(1000);
    }
}

private void UpdateLabel (string labelText)
{
    label1.Text = labelText;
}
```

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

demo

在工作线程中更新UI控件

线程 vs. 线程池

线程重用概念

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 采用通常方式来创建线程代价非常昂贵
- 对于生命周期相对短的线程使用线程池
- 线程池由公共语言运行时（CLR）应用，支持：
 - Threading.Timer对象
 - 异步操作
- ThreadPool性能
 - 当在线程池中沒有空闲线程可用时，其代价与CreateThread相同
 - 由于线程池线程可以被重用，因此当线程池中存在空闲线程时创建线程所带来的开销非常小

线程 vs. 线程池

线程池线程

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 线程池中存在的线程可以被重用
- 如果线程池中没有任何线程存在，则创建新的线程（当线程池内线程数小于25个线程时）
- 如果线程池中的线程空闲一段时间（通常是60秒），那么该线程将会从池中移除，释放其所占资源

您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

demo

线程与线程池

线程同步

线程间协调

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 非同步线程的运行时间不可预知
- 保护数据, 当多线程访问数据时避免出现不可知结果
- 协调线程使线程在达到某个状态时另一个线程可以改变其状态

线程同步

同步对象

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 线程通常来讲不能自治，需要开发者进行协调处理
- 我们可以使用下列同步对象：
 - Interlocked
 - AutoResetEvent
 - ManualResetEvent
 - Monitor
 - Mutex

线程同步

线程安全

您的潜力. 我们的动力

Microsoft
微软(中国)有限公司

- 多线程访问对象可能会引发问题
- 读取不一致数据
- 两个线程可能会同时更新数据, 导致产生出错误的结果
- 使用同步对象来创建线程安全类
- 只有需要并发访问的类才需要线程安全

您的潜力. 我们的动力

Microsoft®
微软(中国)有限公司

demo

线程同步

总结


- 多线程应用程序比单线程应用程序更加复杂
- 理解多线程将有助于你创建性能更好的应用程序
- 掌握线程创建和终止操作的正确方法
- 理解线程与线程池的特点
- 会被多线程并发访问的对象需要线程安全

获取更多MSDN资源

- **MSDN中文网站**
<http://www.microsoft.com/china/msdn>
- **MSDN中文网络广播**
<http://www.msdnwebcast.com.cn>
- **MSDN Flash**
<http://www.microsoft.com/china/newsletter/case/msdn.aspx>
- **MSDN开发中心**
<http://www.microsoft.com/china/msdn/DeveloperCenter/default.mspx>



Question & Answer

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

 **问题和解答 (无问题)** ▲ ×

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问

提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

Microsoft®

msdn


MSDN Webcasts