**[C#窗体 WinForm 进程，线程](https://www.cnblogs.com/bloodPhoenix/p/5844355.html)**

<https://www.cnblogs.com/bloodPhoenix/p/5844355.html>

# C# 之 System.Diagnostics.Process.Start的妙用

[**https://www.cnblogs.com/xinaixia/p/5520190.html**](https://www.cnblogs.com/xinaixia/p/5520190.html)

# C#中的System.Threading

[**https://www.jianshu.com/p/2b207971c2b1**](https://www.jianshu.com/p/2b207971c2b1)

# C#的四种Timer介绍

[**http://www.cnblogs.com/Khan-Sadas/p/4953446.html**](http://www.cnblogs.com/Khan-Sadas/p/4953446.html)

**一、进程**

进程是一个具有独立功能的程序关于某个数据集合的一次运行活动。

它可以申请和拥有系统资源，是一个动态的概念，是一个活动的实体。

**Process 类**，用来操作进程。

命名空间：**using System.Diagnostics;**

**Process.Start("calc");** 　　//打开计算器  
**Process.Start("mspaint");** 　　//打开画图  
**Process.Start("iexplore" , "http://www.baidu.com");**//打开浏览器并指定地址

**(一)通过一个进程，打开指定的文件：**

**1.创建进程对象**  
Process p = new Process();

**2.创建一个StartInfo对象，是指定带盘符的路径。**  
ProcessStartInfo psi = new ProcessStartInfo(@"C:\user\.....);

**3.进程指定及开始**  
p.StartInfo = psi; //指定路径  
p.Start(); //开始进程

**(二)通过文件选择框让用户自己选择所需要打开的程序并打开：**

[复制代码](javascript:void(0);)

复制代码

private void button1\_Click(object sender, EventArgs e)

{

//这是选择文件的类型

openFileDialog1.Filter = "应用程序|\*.exe";

//显示对话框并且判断用户有没有选中文件

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

//取文件路径

string path = openFileDialog1.FileName;

//创建一个新的进程

Process p = new Process();

//制造进程的启动信息

ProcessStartInfo psf = new ProcessStartInfo(path);

//设置该进程的执行信息

p.StartInfo = psf;

//启动进程

p.Start();

}

}

复制代码

[复制代码](javascript:void(0);)

**例子：注销**

**//这个path就是你要调用的exe程序的绝对路径**

string path = Application.StartupPath;

**//获取自身exe或dll的文件名路径**

string s = **System.Reflection.Assembly.GetExecutingAssembly().ManifestModule.FullyQualifiedName;**

[复制代码](javascript:void(0);)

复制代码

private void button2\_Click(object sender, EventArgs e)

{

//打开该程序

//取该程序文件的路径

//string path = Application.StartupPath;

string path = System.Reflection.Assembly.GetExecutingAssembly().ManifestModule.FullyQualifiedName;

//造一个进程

Process p = new Process();

//造一个进程的启动信息

ProcessStartInfo ps = new ProcessStartInfo(path);

//设置进程启动信息

p.StartInfo = ps;

//启动进程

p.Start();

//关闭程序

this.Close();

}

复制代码

[复制代码](javascript:void(0);)

**例：窗口实现鼠标拖动**

[复制代码](javascript:void(0);)

复制代码

private int x;

private int y;

private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)

{

//鼠标点击时的XY坐标

x = e.X;

y = e.Y;

}

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

//判断鼠标按下的是左键

if (e.Button == System.Windows.Forms.MouseButtons.Left)

{

//边距=当前的距离+移动的距离

this.Left = this.Left+(e.X - x);

this.Top = this.Top +(e.Y - y);

}

}

复制代码

[复制代码](javascript:void(0);)

**二、线程**

**线程**，有时被称为轻量级进程(Lightweight Process，LWP），是程序执行流的最小单元。

**区别：**进程有自己独立的资源，线程共用资源。

**Thread类**， 命名空间：**using System.Threading ;**

程序默认只有一个主线程，如果做复杂处理，则会出现假死状态，因为一条线程同时只能做一件事。

**多线程的作用**：同时做多件事情，节约时间，后台运行程序，提高程序的运行效率，也不会使主界面出现无响应的情况。

**创建线程：**

开启新的线程执行哪个函数

**Thread th = new Thread( 里面写线程要操作的方法，方法名不含括号 );**

标记这个线程准备就绪了，可以随时执行，CPU决定执行时间

**th.Start();**

**线程分为两种：**

**前台线程**  -  只有所有的前台线程都关闭才能完成程序关闭。

**后台线程**  -  只要所有的前台线程都结束，后台线程则会自动结束。

**两者的区别：**应用程序必须运行完所有的前台线程才可以退出；

　　　　　　而对于后台线程，应用程序则可以不考虑其是否已经运行完毕而直接退出，所有的后台线程在应用程序退出时都会自动结束。

**设置th线程为后台线程**

**th . IsBackground = true ;**

**控件由主线程创建，新线程想要访问主线程资源，程序默认是不允许跨线程访问的。**

**取消跨线程访问限制**  
**在界面Load事件里**

**Control.CheckForIllegalCrossThreadCalls = false;**

**FormClosing 事件中：**

　　//判断新线程是否为null，如果为null，那就自动关闭了  
　**if(th != null)**  
**{**  
　　　　//立即结束线程，但是结束后就不能再被重新Start了  
　**th.Abort();**  
**}**

**执行带参数的函数：**

**如果线程执行的方法需要参数**，**那么要求这个参数必须是object类型！**

　　Thread th = new Thread(Test);   //参数名不需要带括号  
　　th . IsBackground = true;  
　　th.Start("123"); 　　　　　　　　//在这里传入Test函数所需要的参数

**这样函数中就需要强转：**  
　　**private void Test (object s)**  
**{**  
**string ss = (string) s ;**  
**......**  
**}**