实验6 C#线程技术

实验类型：设计性实验 要求：必做 学时：2

**一、实验目的**

1、熟悉线程的创建及状态控制。

2、掌握多线程的应用。

**二、实验内容**

1、并发线程的使用。

2、用多线程技术实现向两个文本框中同时写入文本的效果。

**三、实验步骤**

1、使用两个并发线程（一个主线程和一个工作者线程）显示计数，要求工作者线程先于主线程执行。

假定主线程与工作者线程执行的核心方法都是DisplayNumbers()，该方法每次累加一个数字，并定期显示每次累加的结果。累加的数字取决于interval字段，其值是用户输入的。如果用户输入100，就累加到800并显示数字100，200，300，400，500，600，700和800；如果用户输入1000，就累加到8000，显示数字1000，2000，3000，4000，5000，6000，7000和8000；依此类推。可参考教材P200-203。

关键代码如下：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace Project1

{

internal class sy6\_1

{

private static int interval;

static void DisplayNumbers()

{

// 获取当前运行线程的Thread对象实例

Thread thisThread = Thread.CurrentThread;

Console.WriteLine("线程："+thisThread.Name+" 已经开始运行");

for (int i = 1; i <= 8\*interval; i++)

{

if (i%interval==0)

{

Console.WriteLine(thisThread.Name+": 当前计数为:"+i);

}

}

Console.WriteLine("线程："+thisThread.Name+"完成");

}

static void Main(string[] args)

{

Console.Write("请输入一个数字：");

interval = int.Parse(Console.ReadLine());

Thread thisThread = Thread.CurrentThread;

thisThread.Name = "Main Thread";

Thread workThread = new Thread(new ThreadStart(DisplayNumbers));

workThread.Name = "Worker Thread";

workThread.Start();

workThread.Suspend();

DisplayNumbers();

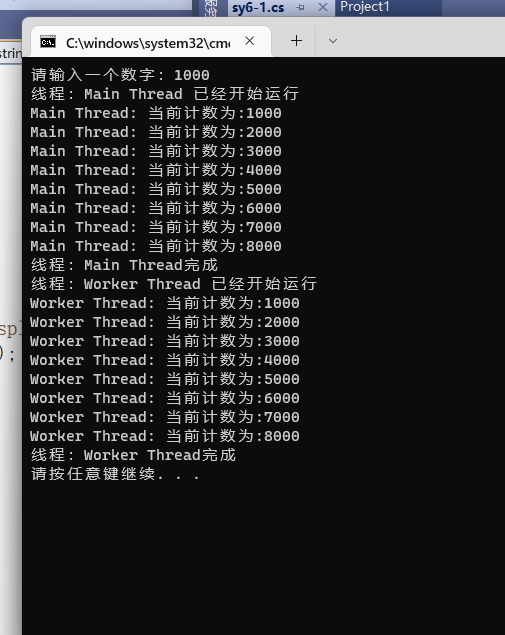
workThread.Resume();

}

}

}

运行结果截图如下：



2、用多线程技术实现向两个文本框中同时写入文本的效果。

可参考教材P217-219。

关键代码如下：

namespace sy6\_2

{

public partial class Form1 : Form

{

private delegate void WriteTextBox(char ch);

private WriteTextBox writeTextBox;

public Form1()

{

InitializeComponent();

// 允许跨线程调用

CheckForIllegalCrossThreadCalls = false;

}

private void DoTsk1()

{

if (checkBox1.Checked == true)

{

groupBox2.Text = "运行中……";

groupBox2.Refresh();

textBox1.Clear();

textBox1.Refresh();

writeTextBox = new WriteTextBox(WriTextBox1);

WriTxt(writeTextBox);

groupBox2.Text = "任务1";

textBox3.Focus();

textBox3.SelectAll();

}

}

private void DoTsk2()

{

if (checkBox2.Checked == true)

{

groupBox2.Refresh();

groupBox3.Text = "运行中……";

groupBox3.Refresh();

textBox2.Clear();

textBox2.Refresh();

//

writeTextBox = new WriteTextBox(WriTextBox2);

WriTxt(writeTextBox);

//

groupBox3.Text = "任务2";

textBox3.Focus();

textBox3.SelectAll();

}

}

private void WriTxt(WriteTextBox wMethod)

{

string strdata = textBox3.Text;

for (int i = 0; i < strdata.Length; i++)

{

wMethod(strdata[i]);

//间歇延时

DateTime now = DateTime.Now;

while (now.AddSeconds(1) > DateTime.Now) { }

}

}

private void WriTextBox1(char ch)

{

textBox1.AppendText(ch + "\r");

}

private void WriTextBox2(char ch)

{

textBox2.AppendText(ch + "\r");

}

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

{

ThreadStart doTask1 = new ThreadStart(DoTsk1);

Thread tsk1Thread = new Thread(doTask1);

tsk1Thread.Start();

ThreadStart doTask2 = new ThreadStart(DoTsk2);

Thread tsk2Thread = new Thread(doTask2);

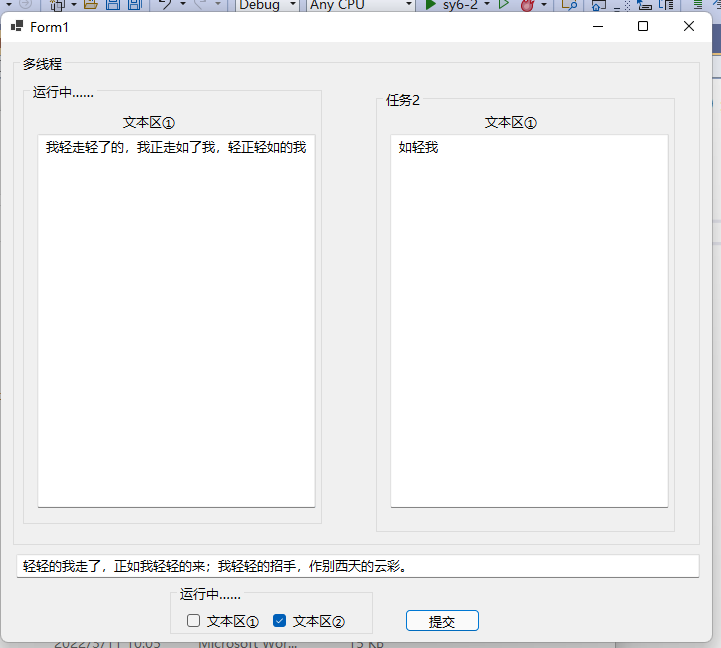
tsk2Thread.Start();

}

}

}

运行结果截图如下：



**四、实验小结**

通过本次实验我学会使用C#的多线程技术，以及使用Thread创建、启动和终止线程，静态方法Thread.Sleep()让主线程暂停1ms，thread.Suspend()挂起线程、thread..Resume()恢复线程；使用ThreadPriority设置线程优先级，使用lock关键字保证一段代码在同一时间只能被一个线程执行

**五、实验提交**

把本文档以学号姓名sy6为名(形如019301784160张三sy6)保存后提交到指定的ftp处。