**实验6.1 并行编程**

实验相关材料位于：<ftp://192.168.40.100/>下的“**--- 梁其洋**”目录下（可以通过搜索名字找到）

作业提交到：[ftp://192.168.40.14/梁其洋/网络编程\*\*班/实验\*](ftp://192.168.40.14/梁其洋/网络编程**班/实验*)

要求：1. 提交时**将本文档重新命名**，文档命名规则：学号+姓名+实验\*，例如，**517300614400 张三 实验\*.docx （一定要学号在前姓名在后，方便自动排序！）**

2. 你们没有FTP文件的删除权限，**如果需要提交新版本，在姓名后加序号后提交即可，例如，517300614400张三（1）实验\*.docx**

实验6-1 利用Parallel.For方法和普通for循环计算两个数组的和，并统计耗时。

实验步骤：

（1）新建一个WPF工程，在MainWindow.xaml中添加以下代码：

<DockPanel>

<Border DockPanel.Dock="Top" >

<TextBlock Text="简单Parallel.For方法基本用法" />

</Border>

<Border DockPanel.Dock="Bottom" >

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center" Height="20">

<Button Name="btnStart" Width="70" Content="执行" Click="btnStart\_Click"/>

</StackPanel>

</Border>

<ScrollViewer>

<StackPanel Background="White" TextBlock.LineHeight="20">

<TextBlock x:Name="textBlock1" Margin="5" TextWrapping="Wrap"/>

</StackPanel>

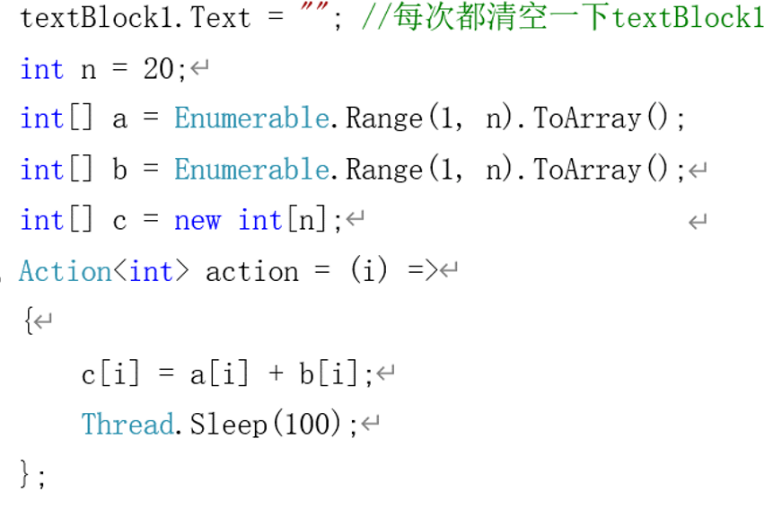
</ScrollViewer>

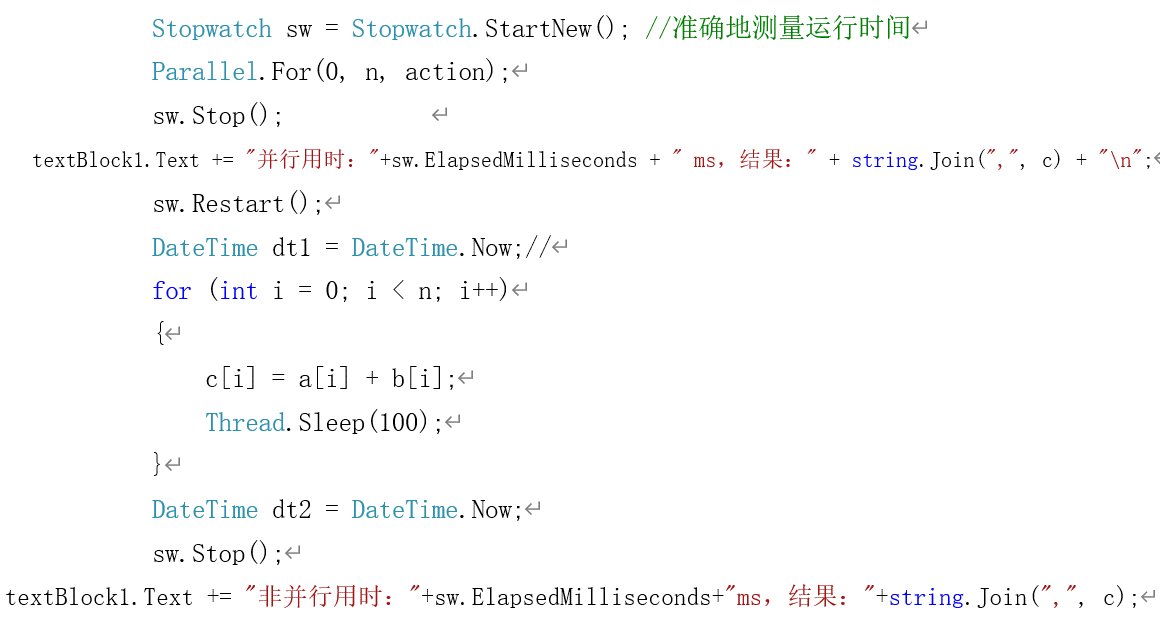
</DockPanel>

（2）实现btnStart的click事件：

private void btnStart\_Click(object sender, RoutedEventArgs e)

{





}

程序运行截图：

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace sy6\_1

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void btnStart\_Click(object sender, RoutedEventArgs e)

{

textBlock1.Text = "";

int n = 20;

int[] a = Enumerable.Range(1, n).ToArray();

int[] b = Enumerable.Range(1, n).ToArray();

int[] c = new int[n];

Action<int> action = (i) =>

{

c[i] = a[i] + b[i];

Thread.Sleep(100);

};

Stopwatch sw = Stopwatch.StartNew();

Parallel.For(0, n, action);

sw.Stop();

textBlock1.Text += "并行用时：" + sw.ElapsedMilliseconds + "ms,结果：" + string.Join(",", c) + "\n";

sw.Restart();

DateTime dt1 = DateTime.Now;

for (int i = 0; i < n; i++)

{

c[i] = a[i] + b[i];

Thread.Sleep(100);

}

DateTime dt2 = DateTime.Now;

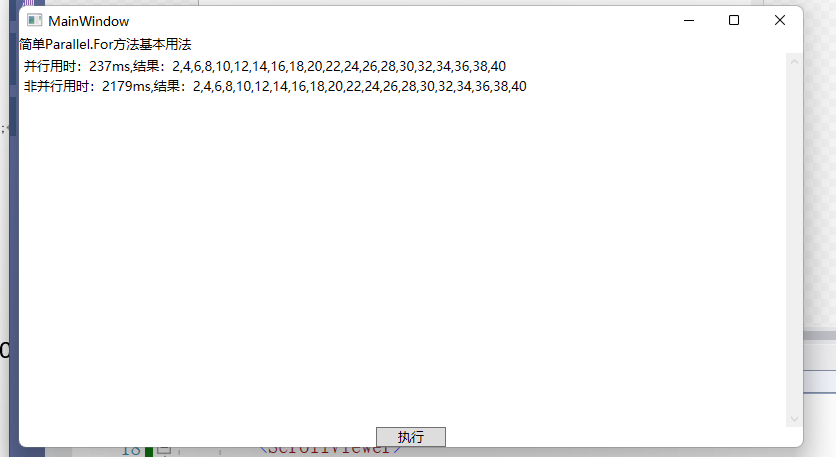
sw.Stop();

textBlock1.Text += "非并行用时：" + sw.ElapsedMilliseconds + "ms,结果：" + string.Join(",", c);

}

}

}



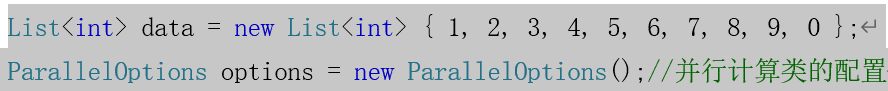
实验6-2 用普通for/foreach循环和并行for/foreach循环输出10个数，并分别统计耗时。

实验步骤：

（1）新建一个**控制台应用程序** ，在Program类中添加两行代码

class Program

{



*static void Main(string[] args)*

*{*

………………

*}*

}

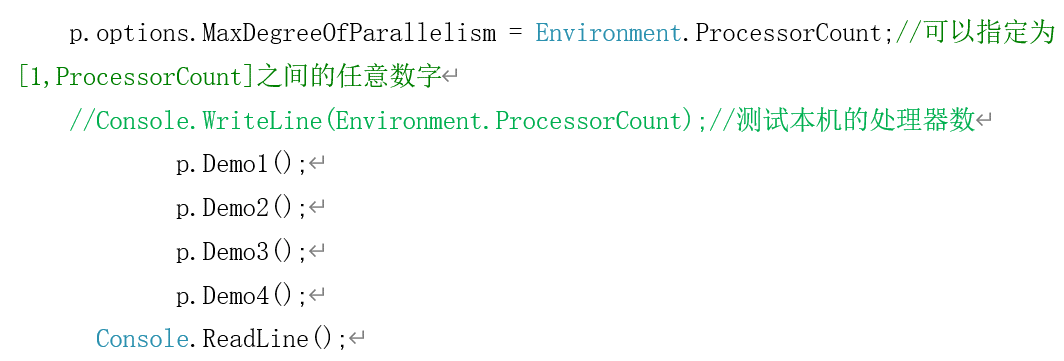
（2）在main方法中添加代码：

static void Main(string[] args)

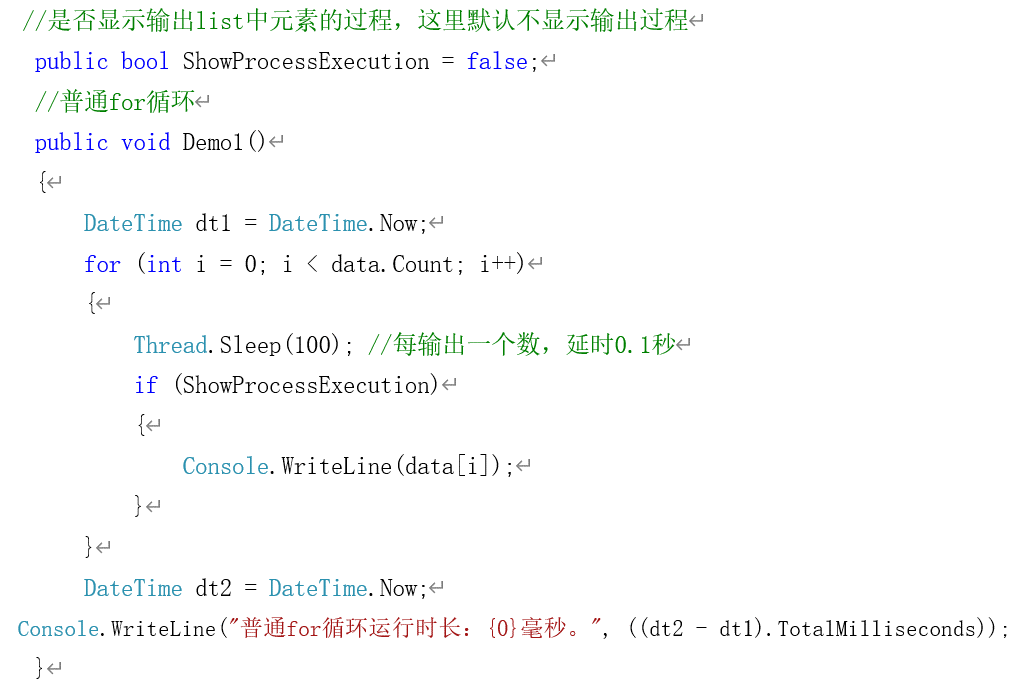
{

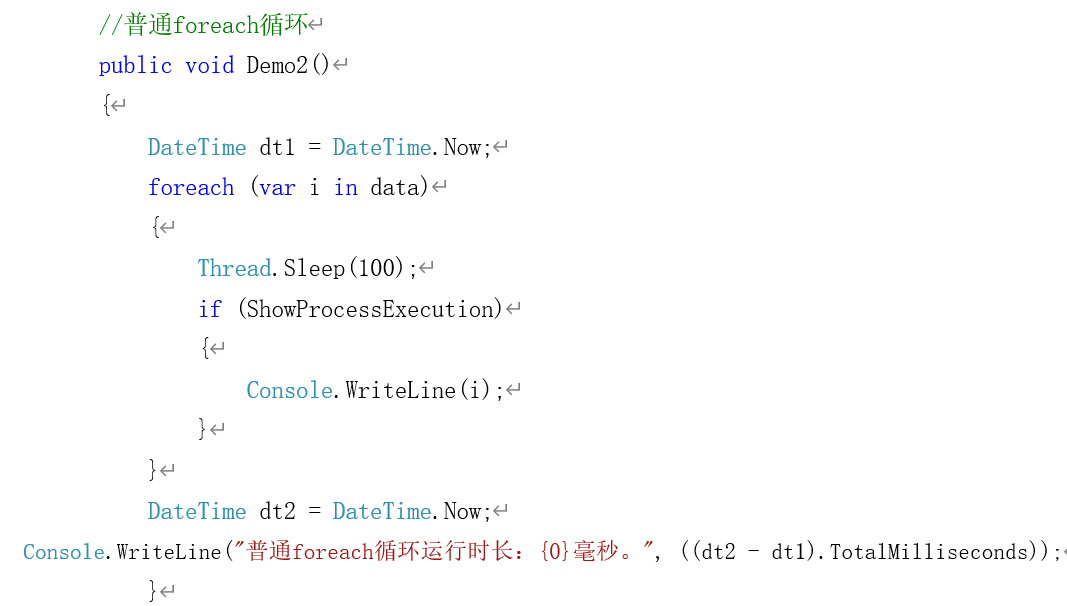
Program p= new Program();

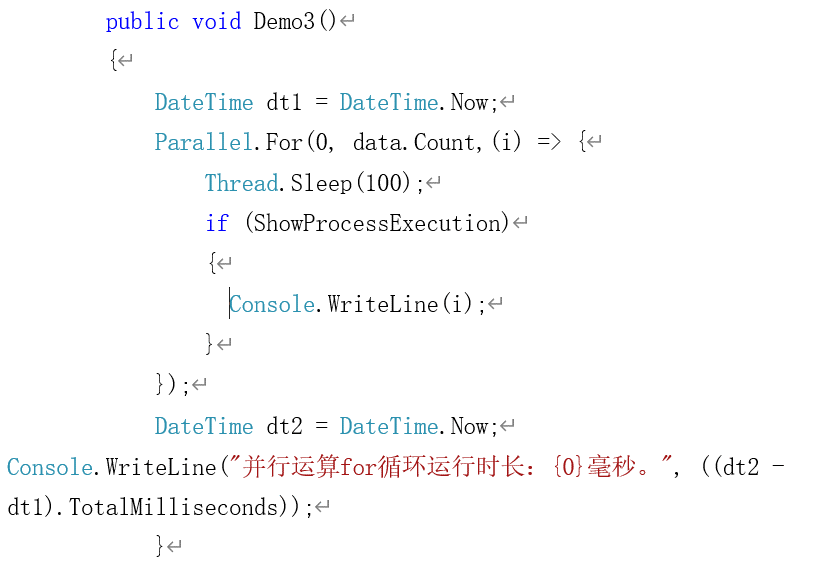
//并行度——最多同时执行计算的线程数，这里是本机的处理器数

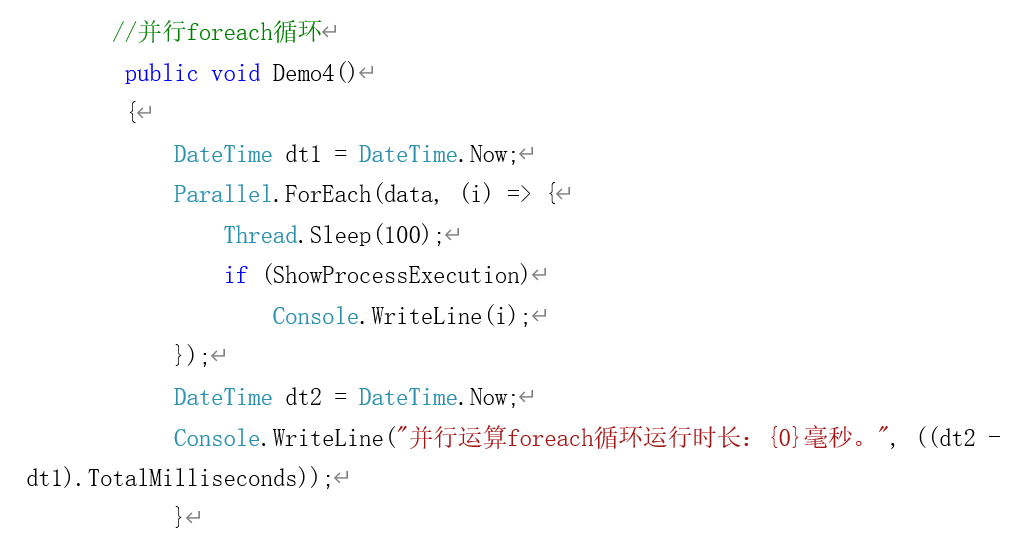


}

（3）在main方法外添加4个方法： 







程序运行截图：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace sy6\_2

{

internal class Program

{

private List<int> data = new List<int> {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};

private ParallelOptions options = new ParallelOptions();

static void Main(string[] args)

{

Program p = new Program();

p.options.MaxDegreeOfParallelism = Environment.ProcessorCount;

Console.WriteLine(Environment.ProcessorCount);

p.Demo01();

p.Demo02();

p.Demo03();

p.Demo04();

}

public bool ShowProgressExecution = false;

public void Demo01()

{

DateTime dt1 = DateTime.Now;

for (int i = 0; i < data.Count; i++)

{

Thread.Sleep(100);

if (ShowProgressExecution)

{

Console.WriteLine(data[i]);

}

}

DateTime dt2 = DateTime.Now;

Console.WriteLine("普通for循环运行时长，{0}毫秒", (dt2 - dt1).TotalMilliseconds);

}

public void Demo02()

{

DateTime dt1 = DateTime.Now;

foreach (var i in data)

{

Thread.Sleep(100);

if (ShowProgressExecution)

{

Console.WriteLine(i);

}

}

DateTime dt2 = DateTime.Now;

Console.WriteLine("普通foreach循环运行时长，{0}毫秒", (dt2 - dt1).TotalMilliseconds);

}

public void Demo03()

{

DateTime dt1 = DateTime.Now;

Parallel.For(0, data.Count, i =>

{

Thread.Sleep(100);

if (ShowProgressExecution)

{

Console.WriteLine(i);

}

});

DateTime dt2 = DateTime.Now;

Console.WriteLine("并行运算for循环运行时长，{0}毫秒", (dt2 - dt1).TotalMilliseconds);

}

public void Demo04()

{

DateTime dt1 = DateTime.Now;

Parallel.ForEach(data, i =>

{

Thread.Sleep(100);

if (ShowProgressExecution)

{

Console.WriteLine(i);

}

});

DateTime dt2 = DateTime.Now;

Console.WriteLine("并行运算foreach循环运行时长，{0}毫秒", (dt2 - dt1).TotalMilliseconds);

}

}

}

