实验3.1

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1，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**

**实验目的：**

掌握进程和线程的基础知识

**实验步骤：**

一、新建一个WPF工程，命名为example1（用其他工程名时注意修改xaml和cs代码中的命名空间）

二、将MainWindow.xaml中的内容替换为下面的内容

<Window x:Class="example1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

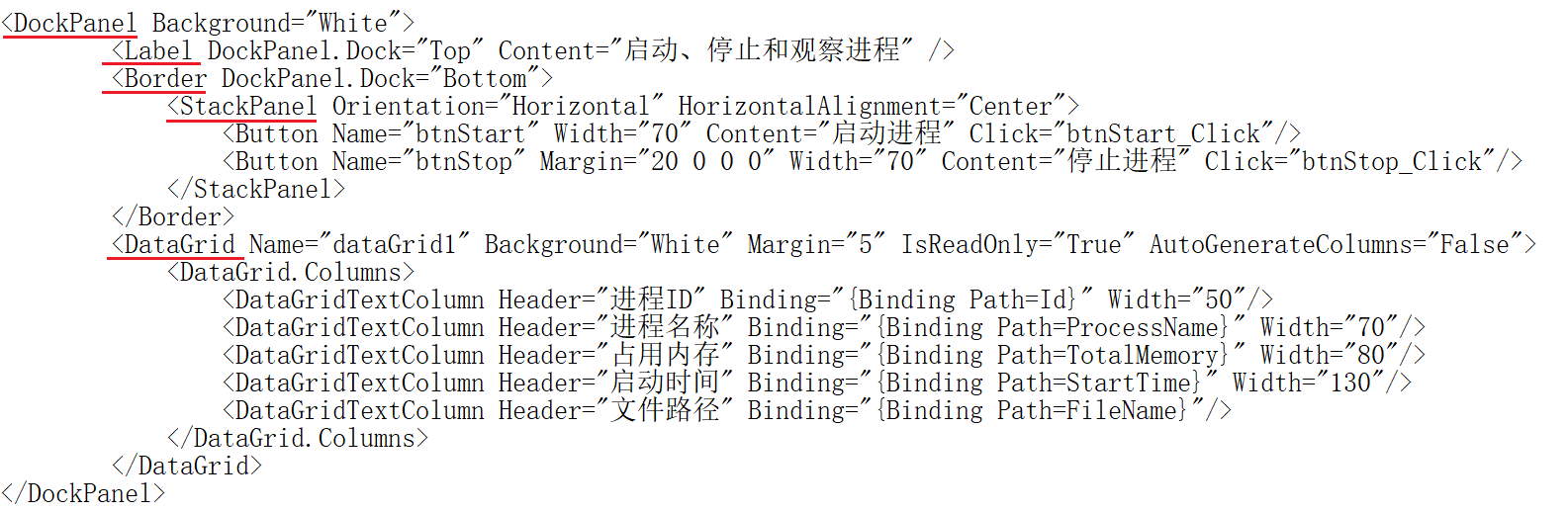
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

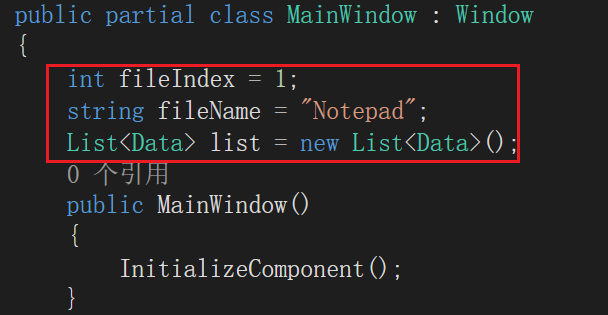
xmlns:local="clr-namespace:example1"

mc:Ignorable="d"

Title="启动、停止和观察进程" Height="350" Width="900" Background="#FFF0F9D8" > 

</Window>

**步骤三： 在主类public partial class MainWindow : Window 下添加如下定义：**



**步骤四：实现开始按钮的功能**

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

{

string argument = Environment.CurrentDirectory + "\\myfile" + (fileIndex++) + ".txt";

if (File.Exists(argument) == false)

{

File.CreateText(argument);

}

Process p = new Process();

p.StartInfo.FileName = fileName;

p.StartInfo.Arguments = argument;

p.StartInfo.UseShellExecute = false;

p.StartInfo.WindowStyle = ProcessWindowStyle.Normal;

p.Start();

p.WaitForInputIdle();

RefreshProcessInfo();

}

**步骤五：实现结束按钮的功能：**

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

{

this.Cursor = Cursors.Wait;

Process[] myprocesses;

myprocesses = Process.GetProcessesByName(fileName);

foreach (Process p in myprocesses)

{

using (p)

{

p.CloseMainWindow();

Thread.Sleep(1000);

p.WaitForExit();

}

}

fileIndex = 0;

RefreshProcessInfo();

this.Cursor = Cursors.Arrow;

}

步骤五：在主类public partial class MainWindow : Window中添加辅助方法 RefreshProcessInfo()：

private void RefreshProcessInfo()

{

dataGrid1.ItemsSource = null;

list.Clear();

Process[] processes = Process.GetProcessesByName(fileName);

foreach (Process p in processes)

{

list.Add(new Data()

{

Id = p.Id,

ProcessName = p.ProcessName,

TotalMemory = string.Format("{0,10:0} KB", p.WorkingSet64 / 1024d),

StartTime = p.StartTime.ToString("yyyy-M-d HH:mm:ss"),

FileName = p.MainModule.FileName

});

}

dataGrid1.ItemsSource = list;

}

**步骤六：在主类之外添加一个新类：**

public class Data

{

public int Id { get; set; }

public string ProcessName { get; set; }

public string TotalMemory { get; set; }

public string StartTime { get; set; }

public string FileName { get; set; }

}

步骤七：运行程序，查看结果：

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace sy3.\_1

{

internal class Data

{

public int Id { get; set; }

public string ProcessName { get; set; }

public string TotalMemory { get; set; }

public string StartTime { get; set; }

public string FileName { get; set; }

}

}

<Window x:Class="sy3.\_1.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:sy3.\_1"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<DockPanel>

<Label DockPanel.Dock="Top" Content="启动、停止和观察进程"/>

<Border DockPanel.Dock="Top">

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

<Button Name="btnStart" Width="70" Content="启动进程" Click="btnStart\_Click"/>

<Button Name="btnStop" Margin="20 0 0 0" Width="70" Content="停止进程" Click="btnStop\_Click"/>

</StackPanel>

</Border>

<DataGrid Name="dataGrid1" Background="White" Margin="5" IsReadOnly="True" AutoGenerateColumns="False">

<DataGrid.Columns>

<DataGridTextColumn Header="进程ID" Binding="{Binding Path=Id}" Width="50"></DataGridTextColumn>

<DataGridTextColumn Header="进程名称" Binding="{Binding Path=ProcessName}" Width="70"></DataGridTextColumn>

<DataGridTextColumn Header="占用内存" Binding="{Binding Path=TotalMemory}" Width="80"></DataGridTextColumn>

<DataGridTextColumn Header="启动时间" Binding="{Binding Path=StartTime}" Width="130"></DataGridTextColumn>

<DataGridTextColumn Header="文件路径" Binding="{Binding Path=FileName}" ></DataGridTextColumn>

</DataGrid.Columns>

</DataGrid>

</DockPanel>

</Window>

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.IO;

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 sy3.\_1

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

private int fileIndex = 1;

private string fileName = "Notepad";

List<Data> list = new List<Data>();

public MainWindow()

{

InitializeComponent();

}

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

{

string argument = Environment.CurrentDirectory + "\\myfile" + (fileIndex++) + ".txt";

if (File.Exists(argument) == false)

{

File.CreateText(argument);

}

Process p = new Process();

p.StartInfo.FileName = fileName;

p.StartInfo.Arguments = argument;

p.StartInfo.UseShellExecute = false;

p.StartInfo.WindowStyle = ProcessWindowStyle.Normal;

p.Start();

p.WaitForInputIdle();

RefreshProcessInfo();

}

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

{

this.Cursor = Cursors.Wait;

Process[] myprocesses;

myprocesses = Process.GetProcessesByName(fileName);

foreach (Process p in myprocesses)

{

using (p)

{

p.CloseMainWindow();

Thread.Sleep(1000);

p.WaitForExit();

}

}

fileIndex = 0;

RefreshProcessInfo();

this.Cursor = Cursors.Arrow;

}

private void RefreshProcessInfo()

{

dataGrid1.ItemsSource = null;

list.Clear();

Process[] processes = Process.GetProcessesByName(fileName);

foreach (Process p in processes)

{

list.Add(new Data()

{

Id = p.Id,

ProcessName = p.ProcessName,

TotalMemory = string.Format("{0,10:0} KB", p.WorkingSet64 / 1024d),

StartTime = p.StartTime.ToString("yyyy-M-d HH:mm:ss"),

FileName = p.MainModule.FileName

});

}

dataGrid1.ItemsSource = list;

}

}

}

