ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

"ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ"

Факультет КНТ

Кафедра ПИ

Лабораторная работа №3

по теме: «Изучение средств организации многопоточности в С#»

Выполнил:

ст. гр. ПИ-18б

Куркурин Н.Л.

Проверил:

Ногтев Е.А.

Сереженко О.А.

ДОНЕЦК – 2020

Задание:

|  |  |
| --- | --- |
| №вар. mod 5 | Средство реализации многопоточности |
| 4 | BackgroundWorker |

|  |  |
| --- | --- |
| №вар. mod 2 | Исследование алгоритма обработки |
| 1 | По результатам параллельной обработки 10 различных изображений построить график зависимости времени работы алгоритма от площади изображения. |

Листинг:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

using System.Windows.Forms.VisualStyles;

using System.Xml.Schema;

namespace third\_lab

{

public partial class Form1 : Form

{

String[] files;

int red;

int green;

int blue;

PictureBox[] array\_of\_pictureboxes;

TimeSpan[] dateTime = new TimeSpan[10];

SortedList<double, double> sortedLists = new SortedList<double, double>();

int size = 0;

public Form1()

{

InitializeComponent();

trackBar1.Visible = false;

trackBar2.Visible = false;

trackBar3.Visible = false;

button2.Visible = false;

chart1.Visible = false;

Axis ax = new Axis();

ax.Title = "Время";

chart1.ChartAreas[0].AxisX = ax;

Axis ay = new Axis();

ay.Title = "Размер";

chart1.ChartAreas[0].AxisY = ay;

chart1.ChartAreas[0].AxisY.Minimum = 27500;

chart1.ChartAreas[0].AxisY.Maximum = 29500;

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

// throw new System.NotImplementedException();

}

private void tableLayoutPanel1\_Paint(object sender, PaintEventArgs e)

{

// throw new System.NotImplementedException();

}

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

{

OpenFileDialog openDialog = new OpenFileDialog();

openDialog.InitialDirectory = Directory.GetCurrentDirectory().Substring(0, Directory.GetCurrentDirectory().IndexOf("bin", StringComparison.Ordinal)) + "For3Lab";

openDialog.Filter = @"Файлы изображений|\*.bmp;\*.png;\*.jpg";

openDialog.Multiselect = true;

while (openDialog.FileNames.Length != 10) openDialog.ShowDialog();

files = openDialog.FileNames;

pictureBox1.Image = Image.FromFile(files[0]);

pictureBox2.Image = Image.FromFile(files[1]);

pictureBox3.Image = Image.FromFile(files[2]);

pictureBox4.Image = Image.FromFile(files[3]);

pictureBox5.Image = Image.FromFile(files[4]);

pictureBox6.Image = Image.FromFile(files[5]);

pictureBox7.Image = Image.FromFile(files[6]);

pictureBox8.Image = Image.FromFile(files[7]);

pictureBox9.Image = Image.FromFile(files[8]);

pictureBox10.Image = Image.FromFile(files[9]);

pictureBox11.Image = Image.FromFile(files[0]);

pictureBox12.Image = Image.FromFile(files[1]);

pictureBox13.Image = Image.FromFile(files[2]);

pictureBox14.Image = Image.FromFile(files[3]);

pictureBox15.Image = Image.FromFile(files[4]);

pictureBox16.Image = Image.FromFile(files[5]);

pictureBox17.Image = Image.FromFile(files[6]);

pictureBox18.Image = Image.FromFile(files[7]);

pictureBox19.Image = Image.FromFile(files[8]);

pictureBox20.Image = Image.FromFile(files[9]);

trackBar1.Visible = true;

trackBar2.Visible = true;

trackBar3.Visible = true;

button2.Visible = true;

}

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

{

this.red = trackBar1.Value;

this.green = trackBar2.Value;

this.blue = trackBar3.Value;

array\_of\_pictureboxes = new PictureBox[] { pictureBox11, pictureBox12, pictureBox13, pictureBox14, pictureBox15, pictureBox16, pictureBox17, pictureBox18, pictureBox19, pictureBox20, };

BackgroundWorker[] bw = new BackgroundWorker[10];

size = 0;

bw[0] = new BackgroundWorker();

bw[0].DoWork += (obj, ea) => thread(1, 0);

bw[0].RunWorkerAsync();

bw[1] = new BackgroundWorker();

bw[1].DoWork += (obj, ea) => thread(1, 1);

bw[1].RunWorkerAsync();

bw[2] = new BackgroundWorker();

bw[2].DoWork += (obj, ea) => thread(1, 2);

bw[2].RunWorkerAsync();

bw[3] = new BackgroundWorker();

bw[3].DoWork += (obj, ea) => thread(1, 3);

bw[3].RunWorkerAsync();

bw[4] = new BackgroundWorker();

bw[4].DoWork += (obj, ea) => thread(1, 4);

bw[4].RunWorkerAsync();

bw[5] = new BackgroundWorker();

bw[5].DoWork += (obj, ea) => thread(1, 5);

bw[5].RunWorkerAsync();

bw[6] = new BackgroundWorker();

bw[6].DoWork += (obj, ea) => thread(1, 6);

bw[6].RunWorkerAsync();

bw[7] = new BackgroundWorker();

bw[7].DoWork += (obj, ea) => thread(1, 7);

bw[7].RunWorkerAsync();

bw[8] = new BackgroundWorker();

bw[8].DoWork += (obj, ea) => thread(1, 8);

bw[8].RunWorkerAsync();

bw[9] = new BackgroundWorker();

bw[9].DoWork += (obj, ea) => thread(1, 9);

bw[9].RunWorkerAsync();

}

private async void thread(int times, int i)

{

DateTime dt = DateTime.Now;

Image bmp = Image.FromFile(files[i]);

Graphics gr = Graphics.FromImage(bmp);

Random rnd = new Random();

double progress = 0;

for (int j = 0; j < bmp.Width; j++)

{

for (int k = 0; k < bmp.Height; k++)

{

SolidBrush solidBrush = new SolidBrush(Color.FromArgb(rnd.Next(1, 255), this.red, this.green, this.blue));

gr.FillRectangle(solidBrush, j, k, 1, 1);

progress += 0.0004;

}

}

gr.Save();

array\_of\_pictureboxes[i].Image = bmp;

TimeSpan temp = DateTime.Now.Subtract(dt);

sortedLists.Add(temp.TotalMilliseconds, temp.TotalMilliseconds);

size += 1;

}

private void button3\_Click(object sender, EventArgs e)

{

//if (dateTime[9].TotalMilliseconds > 0 && array\_of\_pictureboxes != null)

//{

chart1.Series[0].Points.Clear();

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

{

chart1.Series[0].Points.AddXY(sortedLists.Keys[i], array\_of\_pictureboxes[i].Size.Width \* array\_of\_pictureboxes[i].Size.Height);

chart1.Series[0].Points[i].MarkerSize = 10;

chart1.Series[0].Points[i].MarkerColor = Color.Red;

chart1.Series[0].Points[i].MarkerStyle = MarkerStyle.Circle;

}

chart1.Visible = true;

//}

}

}

}



