БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

ИИТ БГУИР

Программное обеспечение информационных технологий.

Дисциплина: «Сетевые технологии и администрирование операционных систем»

Отчет

По лабораторной работе №9

«СИНХРОНИЗАЦИЯ ВЗАИМОДЕЙСТВУЮЩИХ ПОТОКОВ В WIN32 API»

Выполнил студент группы 481064 Сорока А.А.

Проверил преподаватель: Калитеня И. Л.

Минск 2017 г

**Цель:**

1. Изучить возможности согласования работы взаимодействующих потоков с целью исключения гонок и обеспечения необходимой последовательности их выполнения, а так же используемые для этой цели системные средства.
2. Приобрести практические навыки синхронизации работы потоков и создания безопасных, с точки зрения многопоточности, приложений Windows.

**Задание:**

1. Изучить представленные в папке работы примеры многопоточных приложений и реализованные в них возможности синхронизации работы.
2. Разработать собственные безопасные с точки зрения многопоточности приложения в соответствии с индивидуальными заданиями.

**Скриншоты работающего приложения:**

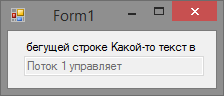


Рис. 1 Работа программы

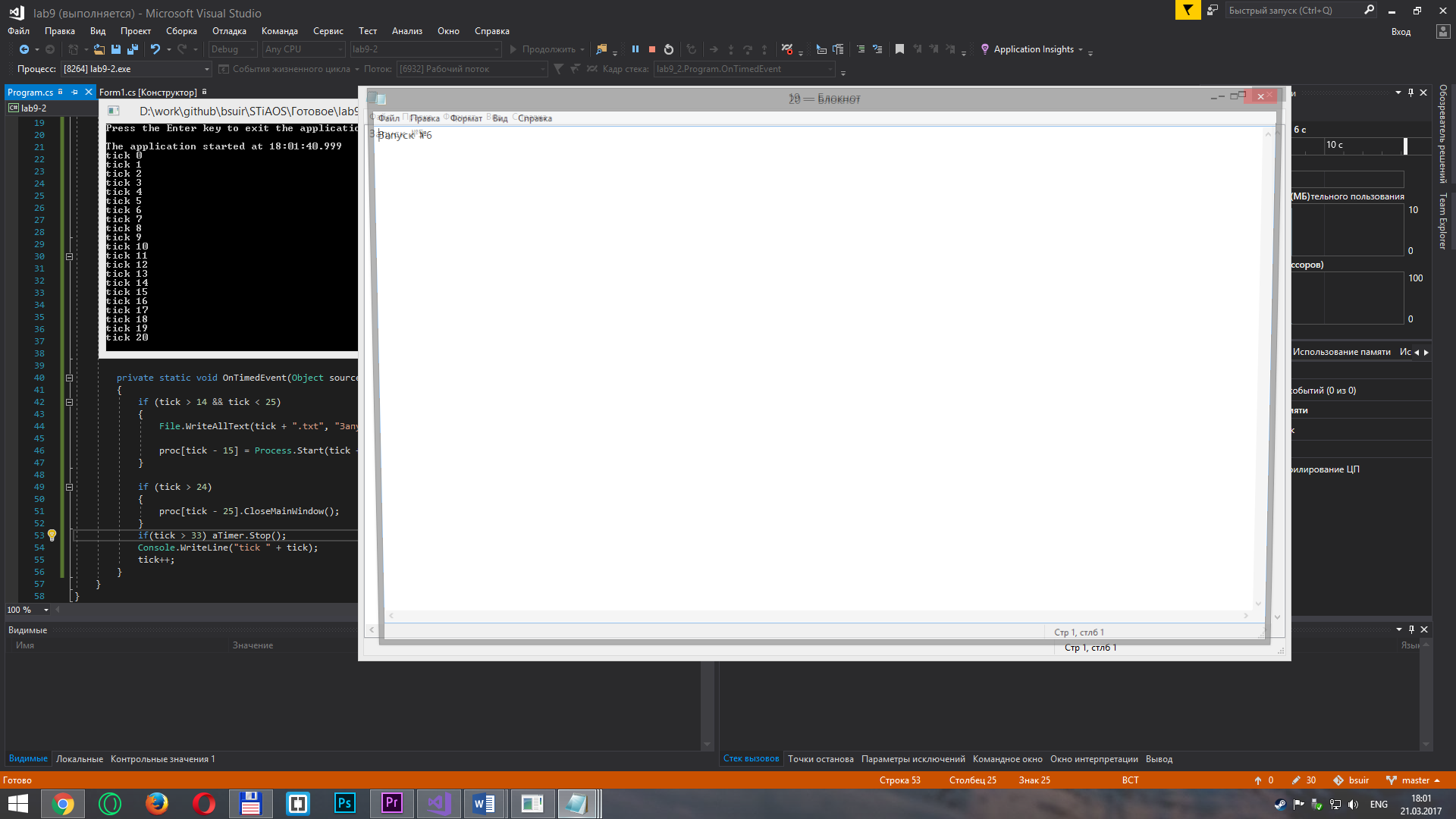


Рис. 2 Работа программы

**Код приложения:**

Задание 1:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading;

using System.Windows.Forms;

namespace lab9

{

public partial class Form1 : Form

{

private static Mutex mut = new Mutex();

public static bool th = false;

public static Thread th1, th2;

public static string txt = " Какой-то текст в бегущей строке";

public static void ThFunc()

{

mut.WaitOne();

th = !th;

Thread.Sleep(3000);

mut.ReleaseMutex();

}

public Form1()

{

th1 = new Thread(new ThreadStart(ThFunc));

th1.Start();

th2 = new Thread(new ThreadStart(ThFunc));

th2.Start();

InitializeComponent();

}

private void timer1\_Tick(object sender, EventArgs e)

{

textBox1.Text = (th ? "Поток 1 управляет" : "Поток 2 управляет");

string scrollText = label1.Text;

if (th)

{

scrollText = scrollText.Substring(1, (scrollText.Length - 1)) + scrollText.Substring(0, 1);

} else

{

scrollText = scrollText.Substring((scrollText.Length - 1), 1) + scrollText.Substring(0, (scrollText.Length - 1));

}

label1.Text = scrollText;

}

}

}

Задание 2:

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.IO;

using System.Linq;

using System.Text;

using System.Timers;

namespace lab9\_2

{

class Program

{

private static System.Timers.Timer aTimer;

private static Process[] proc = new Process[11];

private static int tick = 0;

public static void Main()

{

SetTimer();

Console.WriteLine("\nPress the Enter key to exit the application...\n");

Console.WriteLine("The application started at {0:HH:mm:ss.fff}", DateTime.Now);

Console.ReadLine();

aTimer.Stop();

aTimer.Dispose();

Console.WriteLine("Terminating the application...");

}

private static void SetTimer()

{

// Create a timer with a two second interval.

aTimer = new System.Timers.Timer(300);

// Hook up the Elapsed event for the timer.

aTimer.Elapsed += OnTimedEvent;

aTimer.AutoReset = true;

aTimer.Enabled = true;

}

private static void OnTimedEvent(Object source, ElapsedEventArgs e)

{

if (tick > 14 && tick < 25)

{

File.WriteAllText(tick + ".txt", "Запуск №" + (tick - 14));

proc[tick - 15] = Process.Start(tick + ".txt");

}

if (tick > 24)

{

proc[tick - 25].CloseMainWindow();

}

if(tick > 33) aTimer.Stop();

Console.WriteLine("tick " + tick);

tick++;

}

}

}