

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

## **Простой браузер и FTP клиент**

### **Отчет по лабораторной работе №3**

**По дисциплине**

**«Сети и телекоммуникации»**

Выполнил: студент гр.439-3

\_\_\_\_\_ Антонов А.А.

«\_\_\_»\_\_\_\_\_ 2022 г.

Выполнил: студент гр.439-4

\_\_\_\_\_ Фомин М.А.

«\_\_\_»\_\_\_\_\_ 2022 г.

Выполнил: студент гр.439-5

\_\_\_\_\_ Бабкина А.О.

Проверил: доцент каф. АСУ

\_\_\_\_\_ Суханов А.Я.

«\_\_\_»\_\_\_\_\_ 2022 г.

Томск 2022

## Цель лабораторной работы

Написать GUI приложение для ОС Windows или ОС Linux, представляющее собой простой Web – браузер и FTP – клиент, используя стандартные компоненты.

## Задание на лабораторную работу

Необходимо сделать простой браузер и ftp client. Желательно выполнять "сырые" запросы HTTP (HTTPS) или FTP с помощью сокетов, на крайний случай используя компоненты вида HTTPClient, HTTPrequest, реализуя даже простое скачивание контента в виде страниц, текстовый режим отображения или добавления скачанной страницы на форму в готовый веб-компонент.

3) Создать Браузер с тремя вкладками, возможностями навигации, вперед, назад.

## Ход работы

При выполнении данной лабораторной работы необходимо загрузить форму «Form1» в Visual Studio и расставить компоненты. В файле есть компоненты кнопок, с помощью которых происходит навигация по страницам, функции html парсинга и создания ftp соединения. В коде размещен объект tabControl, который содержит 5 вкладок. Первые 3 вкладки выделены под три компонента веб-браузера, четвертая - под вывод контента парсинга html страницы, пятая вкладка содержит textBox, выводящий информацию, полученную в ходе ftp соединения с конкретным сервером. Кроме вкладок браузер содержит textbox для считывания адресов сайтов.

Первые три вкладки предназначены для навигации. Необходимо проверить активную вкладку, чтобы в зависимости от нее происходила навигация конкретного объекта браузера.

HTML:

Через TCP-соединение защищенного SSL подключаемся к сайту ТУСУРа: tusur.ru. Создадим потоки и воспользуемся командой GET (строка 94, см. раздел «Листинг»). Далее с помощью данной команды пытаемся получить объект index.html. Затем необходимо запустить чтение потока, который построчно считывает html-документ. В таком потоке строк нужно найти последнее вхождение строки https, и в результате манипуляций над строкой, получаем адрес. Данный адрес вводим в четвертой вкладке через объект веб-браузера. Последним шагом является закрытие потоков.

FTP:

Для начала необходимо подключиться к серверу <ftp.iao.ru> с помощью обычного tcp-соединения по 21 порту. Далее выполняем авторизацию, используя USER anonymous (строка 125) и PASS pore (строка 130). Получаем ответ, затем используя команду PASV переходим в пассивный режим. В ответ получаем строку, содержащую числа, с помощью которых можно получить IP и порт соединения для передачи файлов. Затем используем команду LIST, которая делает запрос вывода содержимого каталога. Ответ предоставляется по второму соединению, к которому подключается tcp-соединение через IP и порт. Ответом является набор файлов. Последний шаг – закрытие потоков.

## Заключение

В ходе выполнения данной лабораторной работы нами был реализован браузер с тремя вкладками и возможностями навигации, а также FTP клиент на языке C#. Также мы научились создавать формы в Visual Studio.

## Листинг

```
1. using System;
2. using System.Collections.Generic;
3. using System.ComponentModel;
4. using System.Data;
5. using System.Drawing;
6. using System.Linq;
7. using System.Text;
8. using System.Threading.Tasks;
9. using System.Windows.Forms;
10.     using System.Net;
11.     using System.Net.Sockets;
12.     using System.Net.Security;
13.     using System.IO;
14.
15.     namespace TestWebBrowser
16.     {
17.         public partial class Form1 : Form
18.         {
19.             public Form1()
20.             {
21.                 InitializeComponent();
```

```
22.         }
23.
24.     private void Go_Click(object sender, EventArgs e)
25.     {
26.         switch(tabControl1.SelectedIndex)
27.         {
28.             case 0
29.                 webBrowser1.Navigate(textBox1.Text);
30.                 break;
31.             case 1
32.                 webBrowser2.Navigate(textBox1.Text);
33.                 break;
34.             case 2
35.                 webBrowser3.Navigate(textBox1.Text);
36.                 break;
37.             default
38.                 break;
39.         }
40.
41.     }
42.
43.     private void Forward_Click(object sender, EventArgs
    e)
44.     {
45.         switch (tabControl1.SelectedIndex)
46.         {
47.             case 0
48.                 webBrowser1.GoForward();
49.                 break;
50.             case 1
51.                 webBrowser2.GoForward();
52.                 break;
53.             case 2
54.                 webBrowser3.GoForward();
55.                 break;
56.             default
57.                 break;
58.         }
59.     }
60.
```

```

61.         private void Backward_Click(object sender, EventArgs
           e)
62.         {
63.             switch (tabControl1.SelectedIndex)
64.             {
65.                 case 0
66.                     webBrowser1.GoBack();
67.                     break;
68.                 case 1
69.                     webBrowser2.GoBack();
70.                     break;
71.                 case 2
72.                     webBrowser3.GoBack();
73.                     break;
74.                 default
75.                     break;
76.             }
77.         }
78.
79.         private void Exit_Click(object sender, EventArgs e)
80.         {
81.             this.Close();
82.         }
83.
84.         Ищу последнюю запись с подстрокой http, вывожу
           страницу на экран htmlTest
85.         private void httpButton_Click(object sender,
           EventArgs e)
86.         {
87.             string server = tusur.ru;
88.             TcpClient client = new TcpClient(server, 443);
89.             SslStream sslStream = new
           SslStream(client.GetStream());
90.             sslStream.AuthenticateAsClient(server);
91.             var reader = new StreamReader(sslStream);
92.             var writer = new StreamWriter(sslStream);
93.             string navigationString = index.html;
94.             writer.WriteLine(GET + navigationString
           + HTTP1.0 rnHost + server + rnConnection closernrn);
95.             writer.Flush();

```

```
96.         string line;
97.         int hrefIndex;
98.         while (!reader.EndOfStream)
99.         {
100.             line = reader.ReadLine();
101.             if ((hrefIndex = line.IndexOf(https)) != -1)
102.             {
103.                 string tmp = line.Substring(hrefIndex);
104.                 hrefIndex = tmp.IndexOf();
105.                 if(hrefIndex != -1) navigationString =
                    tmp.Remove(hrefIndex, tmp.Length - hrefIndex);
106.             }
107.         }
108.         writer.Close();
109.         reader.Close();
110.         client.Close();
111.         webBrowser4.Navigate(navigationString);
112.     }
113.
114.     private void ftpButton_Click(object sender,
        EventArgs e)
115.     {
116.         string server = ftp.iao.ru;
117.         TcpClient client = new TcpClient(server, 21);
118.         var reader = new
            StreamReader(client.GetStream());
119.         var writer = new
            StreamWriter(client.GetStream());
120.         string second, tmp;
121.
122.         textBox2.AppendText(reader.ReadLine());
123.         textBox2.AppendText(Environment.NewLine);
124.
125.         writer.WriteLine(USER anonymous);
126.         writer.Flush();
127.         textBox2.AppendText(reader.ReadLine());
128.         textBox2.AppendText(Environment.NewLine);
129.
130.         writer.WriteLine(PASS nope);
131.         writer.Flush();
```

```
132.         textBox2.AppendText(reader.ReadLine());
133.         textBox2.AppendText(Environment.NewLine);
134.         textBox2.AppendText(reader.ReadLine());
135.         textBox2.AppendText(Environment.NewLine);
136.         textBox2.AppendText(reader.ReadLine());
137.         textBox2.AppendText(Environment.NewLine);
138.         textBox2.AppendText(reader.ReadLine());
139.         textBox2.AppendText(Environment.NewLine);
140.
141.         writer.WriteLine(PASV);
142.         writer.Flush();
143.         tmp = reader.ReadLine();
144.         textBox2.AppendText(tmp);
145.         textBox2.AppendText(Environment.NewLine);
146.
147.         writer.WriteLine(LIST);
148.         writer.Flush();
149.
150.         string portNum1, portNum2;
151.         int indx;
152.         tmp = tmp.Substring(tmp.IndexOf(( ) + 1);
153.         second = tmp.Remove(tmp.IndexOf(( ), 2);
154.         indx = tmp.LastIndexOf(,);
155.         portNum1 = second.Substring(indx + 1);
156.         second = second.Remove(indx, second.Length -
            indx);
157.         indx = second.LastIndexOf(,);
158.         portNum2 = second.Substring(indx + 1);
159.         second = second.Remove(indx, second.Length -
            indx);
160.
161.         int port = Convert.ToInt32(portNum2) 256 +
            Convert.ToInt32(portNum1);
162.         second = second.Replace(',','.');
163.         TcpClient reciever = new TcpClient(second,
            port);
164.         var recieverRd = new
            StreamReader(reciever.GetStream());
165.         var recieverWr = new
            StreamWriter(reciever.GetStream());
```

```
166.         while(!recieverRd.EndOfStream)
167.         {
168.             textBox2.AppendText(recieverRd.ReadLine());
169.             textBox2.AppendText(Environment.NewLine);
170.         }
171.
172.         recieverRd.Close();
173.         recieverWr.Close();
174.         reciever.Close();
175.         reader.Close();
176.         writer.Close();
177.         client.Close();
178.     }
179. }
180. }
```