МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА»

ФАКУЛЬТЕТ ИНФОРМАТИКИ

КАФЕДРА ПРОГРАММНЫХ СИСТЕМ

Отчет к лабораторной работе № 2

«Исследование структуры FTP-сервера»

по курсу «Сети ЭВМ и телекоммуникации»

Выполнил:

Студент группы 6304-090301D

Алеев Ибрагим Ильясович

Проверил:

Пигусов Андрей Сергеевич

Цель работы: приобретение навыков работы с протоколом передачи файлов FTP и написания серверных приложений.

**Задание**

Разработать клиентское приложение, позволяющее пользователю получать следующую информацию о FTP-сервере:

* суммарный размер файлов, размещенных на сервере;
* суммарный размер размещенных на сервере файлов, сгруппированных по их типам (тип файлов определяется по его расширению);
* структуру каталогов FTP-сервера.

Адрес FTP-сервера, имя пользователя и пароль вводятся пользователем

**Листинг программы:**

using System;

using System.Collections.Generic;

using System.IO;

using System.Net;

using System.Text.RegularExpressions;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Network\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

List<string> files = new List<string>();

List<string> dirs = new List<string>();

string host, login, pass;

private void doReq()

{

FtpWebRequest req = (FtpWebRequest)WebRequest.Create("ftp://" + host);

req.Credentials = new NetworkCredential(login, pass);

req.Method = WebRequestMethods.Ftp.ListDirectoryDetails;

var root = new TreeNode(host);

var c = (new StreamReader(req.GetResponse().GetResponseStream())).ReadToEnd().Replace("\r", "").Split('\n');

Match m;

foreach (var i in c)

{

m = Regex.Match(i, @"^d[rwx-]{3,9}([\s\t]+\d+){4}\s\w{3}\s\d\d\s+[\d:]+\s(.\*)$");

if (m.Success)

{

var \_ = new TreeNode(m.Groups[2].Value);

\_.Tag = m.Groups[2].Value;

root.Nodes.Add(\_);

dirs.Add("/" + m.Groups[2].Value);

}

m = Regex.Match(i, @"^[rwx-]{3,9}([\s\t]+\d+){4}\s\w{3}\s\d\d\s+[\d:]+\s(.\*)$");

if (m.Success)

files.Add(m.Groups[2].Value);

}

string pwd;

for (int i = 0; i < dirs.Count - 1; i ++)

{

if (dirs[i].Length == 0)

{

dirs.Remove(dirs[i]);

i--;

continue;

}

pwd = dirs[i];

req = (FtpWebRequest)WebRequest.Create("ftp://" + host + pwd);

req.Credentials = new NetworkCredential(login, pass);

req.Method = WebRequestMethods.Ftp.ListDirectoryDetails;

var tmp =

(

new StreamReader(

req

.GetResponse()

.GetResponseStream()

)

)

.ReadToEnd()

.Replace("\r", "")

.Split('\n');

foreach (var item in tmp)

{

m = Regex.Match(item, @"^d[rwx-]{3,9}([\s\t]+\d+){4}\s\w{3}\s\d\d\s+[\d:]+\s(.\*)$");

if (m.Success)

{

dirs.Add(pwd + "/" + m.Groups[2].Value);

TreeNode n = root;

var s\_pwd = pwd.Substring(1).Split('/');

for (int j = 0; n != null && j < s\_pwd.Length; j++)

{

n = n.FirstNode;

while (n != null && n.Text != s\_pwd[j])

n = n.NextNode;

}

var \_ = new TreeNode(m.Groups[2].Value);

\_.Tag = m.Groups[2].Value;

n.Nodes.Add(\_);

}

m = Regex.Match(item, @"^-[rwx-]{3,9}([\s\t]+\d+){4}\s\w{3}\s\d\d\s+[\d:]+\s(.\*)$");

if (m.Success)

files.Add(pwd + "/" + m.Groups[2].Value);

}

}

this.Invoke((MethodInvoker)delegate

{

Dictionary<string, int> cnt = new Dictionary<string, int>();

dataGridView1.Rows.Clear();

int sum = 0;

foreach (string f in files)

{

req = (FtpWebRequest)WebRequest.Create("ftp://" + host + f);

req.Credentials = new NetworkCredential(login, pass);

req.Method = WebRequestMethods.Ftp.GetFileSize;

var f\_siz = req.GetResponse().ContentLength;

m = Regex.Match(f, @"^/(.+/)\*[\w\s.-]+(?<extt>(\.[a-z]+){1,2})$");

if (!m.Success)

break;

var ext = m.Groups["extt"].Value;

if (cnt.ContainsKey(ext))

cnt[ext]+=(int)f\_siz;

else

cnt.Add(ext, (int)f\_siz);

sum += (int)f\_siz;

dataGridView1.Rows.Add(f, ext, f\_siz);

}

dataGridView1.Rows.Add("", "", "");

foreach (KeyValuePair<string,int> pa in cnt)

{

dataGridView1.Rows.Add("\*", pa.Key, pa.Value);

}

dataGridView1.Rows.Add("\*", "\*", sum);

foreach (var item in root.Nodes)

treeView2.Nodes.Add(item as TreeNode);

listBox1.Items.AddRange(files.ToArray());

files.Clear();

dirs.Clear();

button1.Enabled = true;

});

}

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

{

host = textBox1.Text;

login = textBox2.Text;

pass = textBox3.Text;

Task t = new Task(doReq);

t.Start();

if (sender is Button)

(sender as Button).Enabled = false;

}

}

}

**Результат выполнения программы:**

