**武汉大学计算机学院**

**本科生实验报告**

**网络系统软件与应用软件开发**

专 业 名 称 ：软件工程

课 程 名 称 ：网络工程与编程实践

团 队 名 称 ：软工卓工班10组

指 导 教 师 一：吕慧 讲师

指 导 教 师 二：

团 队 成 员 一：王纯扬（2016302580216）

团 队 成 员二：

二○一九年四月

**郑 重 声 明**

本团队呈交的实验报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本实验报告不包含他人享有著作权的内容。对本实验报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本实验报告的知识产权归属于培养单位。



团队成员签名： 日期： 2019.4.1

摘□□要

网络系统软件与应用软件开发实验的实验目的是熟悉网络规划与设计的基本知识和方法、掌握网络系统软件与应用软件开发的方法。

实验设计主要遵循科学性原则。

实验内容主要包括：实现一个具有图形化界面、上传、下载功能，并且有断点续传功能的FTP客户端。

实验结论为成功按要求使用C#完成了FTP客户端的编写。

**关键词：**FTP客户端；断点续传；C#

**目□□录**

**1**□**实验目的和意义**

1.1□ 实验目的 ………………………………………………………………………1

1.2□ 实验意义………………………………………………1

(各章的名称黑体4号，其余宋体小4)

**……**

**……**

**……**

**2**□**实验设计**

2.1□概述……………………………………………………………………………… 2

2.2□实验原理………………………………………………………………………… 2

2.3□实验方案………………………………………………………………………… 2

**……**

**……**

**……**

**结论** …………………………………………………………………………………… 57

**参考文献 ………………………………………………………………………………** 59

**1□实验目的和意义**

**1.1**□**实验目的**

本实验是使学生熟悉网络规划与设计的基本知识和方法、掌握网络系统软件与应用软件开发的方法，能将所学的操作系统、数据库、软件工程、计算机网络等方面的知识集成到一起，规划、安装、调试实际网络系统、开发实际软件系统。

**1．1．1** □**目的一：面向系统的软件开发**

本实验是使学生掌握网络系统软件的开发方法、开发平台的使用、与实际数据库的集成方法。用JAVA/ VC++/C＃完成FTP客户端、SMTP客户端、POP3客户端三个系统程序。

**1.1.2 目的二：面向网络应用的软件开发**

利用软件工程的方法，设计一个小规模的应用系统，与具体数据库连接起来，从用户界面设计、数据库设计到处理流程设计，最终完成系统的编程。本实验将用网上书店作为例子，要求实现基于Web的远程功能（Web页面）和后端管理功能。

**2.实验意义**

该实验是理论知识和动手能力的综合体现。通过本实验，掌握网络系统软件、网络应用软件的开发方法、开发平台的使用、与实际数据库的集成方法。

**2 实验设计**

**2.1 概述**

本次实验我使用C#完成了FTP客户端的编写，该客户端拥有图形用户化界面，可以连接FTP服务器，上传下载文件，并且支持断点续传功能。通过本次实验，我对FTP网络协议和网络编程方面有了更深刻的认识。

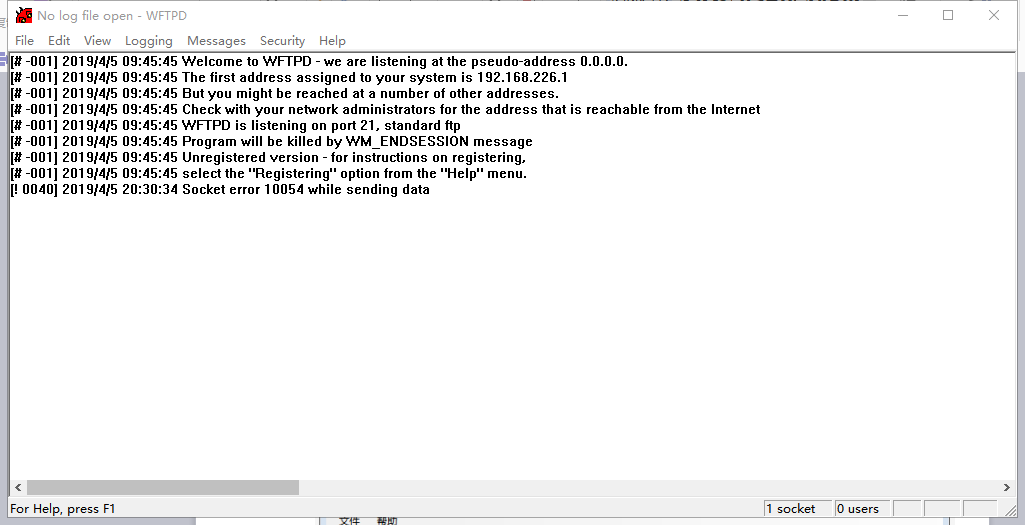
**2.2 实验原理**

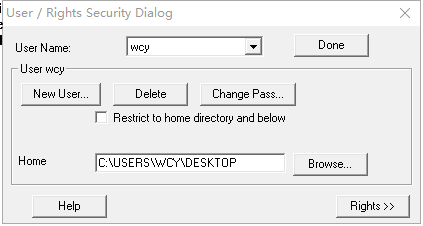
文件传输协议（英文：File Transfer Protocol，缩写：FTP）是用于在网络上进行文件传输的一套标准协议，使用客户/服务器模式。它属于网络传输协议的应用层。FTP的传输有两种方式：ASCII、二进制。ASCII传输方式：假定用户正在拷贝的文件包含的简单ASCII码文本，如果在远程机器上运行的不是UNIX，当文件传输时ftp通常会自动地调整文件的内容以便于把文件解释成另外那台计算机存储文本文件的格式。但是常常有这样的情况，用户正在传输的文件包含的不是文本文件，它们可能是程序，数据库，字处理文件或者压缩文件。在拷贝任何非文本文件之前，用binary 命令告诉ftp逐字拷贝。二进制传输模式：在二进制传输中，保存文件的位序，以便原始和拷贝的是逐位一一对应的。即使目的地机器上包含位序列的文件是没意义的。例如，macintosh以二进制方式传送可执行文件到Windows系统，在对方系统上，此文件不能执行。如在ASCII方式下传输二进制文件，即使不需要也仍会转译。这会损坏数据。（ASCII方式一般假设每一字符的第一有效位无意义，因为ASCII字符组合不使用它。如果传输二进制文件，所有的位都是重要的）。本次实验我采用的是二进制传输模式。FTP支持两种工作模式：Standard (PORT方式，主动方式)，Passive (PASV，被动方式)。Port模式：FTP 客户端首先和服务器的TCP 21端口建立连接，用来发送命令，客户端需要接收数据的时候在这个通道上发送PORT命令。PORT命令包含了客户端用什么端口接收数据。在传送数据的时候，服务器端通过自己的TCP 20端口连接至客户端的指定端口发送数据。FTP server必须和客户端建立一个新的连接用来传送数据。Passive模式：建立控制通道和Standard模式类似，但建立连接后发送Pasv命令。服务器收到Pasv命令后，打开一个临时端口（端口号大于1023小于65535）并且通知客户端在这个端口上传送数据的请求，客户端连接FTP服务器此端口，然后FTP服务器将通过这个端口传送数据。很多防火墙在设置的时候都是不允许接受外部发起的连接的，所以许多位于防火墙后或内网的FTP服务器不支持PASV模式，因为客户端无法穿过防火墙打开FTP服务器的高端端口；而许多内网的客户端不能用PORT模式登陆FTP服务器，因为从服务器的TCP 20无法和内部网络的客户端建立一个新的连接，造成无法工作。本次实验中我使用FTP的被动方式。

**2.3 实验方案**

2.3.1 搭建支持断点续传的FTP服务器

本次实验我使用WFTPD搭建支持断点续传的FTP服务器，并且新建了一个名称为wcy，密码为123456的用户，根目录为我的电脑桌面。





2.3.2 使用C#编写程序界面

我的FTP客户端包括一个菜单栏，其中包含文件与帮助2个菜单项。菜单栏下方是连接服务器部分，包含填写连接服务器所需的各种信息的输入框。下方是本地和服务器的目录区，使用TreeView实现，本地目录要选择本FTP客户端拥有访问权限的目录。上传下载暂停和继续按钮可以分别控制FTP客户端的行为。下方有一个显示进度的文本标签。

界面效果如下：



2.3.3 实现FTP客户端的基本功能

2.3.3.1 实现FTP客户端的菜单

实现帮助菜单，文件菜单里的上传下载暂停和继续菜单项通过调用相应功能函数实现。

2.3.3.2 实现选择本地目录的功能

点击按钮后弹出文件选择框，选择目录后将该目录存储在变量localPath中，并且从该路径开始遍历目录的目录结构，以该路径为treeView的根节点若遍历到目录则添加为子节点并递归遍历，若遍历到文件则添加为叶节点。节点的Tag为其路径，显示出来的Text为其名字。最终在treeView中形成本地目录为根的目录树。

中文乱码问题通过创建StreamReader对象时指定Encoding.Default解决。

2.3.3.3 实现连接FTP服务器的功能

点击按钮后将服务器ip、用户名和密码存储在变量中。实现GetDirctory方法获取ftp服务器上当前目录中的子目录，实现GetFile方法获取ftp服务器上当前目录中的文件，从"ftp://" + ip + "/"服务器根路径开始遍历目录中的目录，以根路径为treeView的根节点，使用GetDirctory方法遍历ftp服务器上当前目录中的子目录并添加为子节点并递归遍历，使用GetFile方法遍历ftp服务器上当前目录中的文件则添加为叶节点。节点的Tag为其路径，显示出来的Text为其名字。最终在treeView中形成ftp服务器根路径为根的目录树。

获取ftp服务器上当前目录中的目录及文件使用了一个FtpWebRequest对象，对其初始化时给定了ftp服务器的路径，之后设置了连接使用的用户名和密码，并设置其方法为获取目录中文件详情WebRequestMethods.Ftp.ListDirectoryDetails。然后从FtpWebRequest对象中获取了ftp服务器的回复WebResponse对象后使用WebResponse对象创建StreamReader对象读取返回信息，之后通过对Windows和Linux下的两种ftp服务器返回的不同格式的目录详情进行解析分别获取目录和文件信息。

连接后可以使用上传和下载功能。

2.3.3.4 实现上传功能

上传时先判断是否选择了要上传的文件，默认上传到服务器的根目录下，直接使用"ftp://" + ip + "/"会报错550，路径加上用户名则可解决该问题，实际使用的服务器路径为"ftp://" + user + "@" + ip + "/" + file。上传时同样是使用FtpWebRequest对象发起ftp请求，此时请求方法为上传文件WebRequestMethods.Ftp.AppendFile。使用一个FileStream对象对本地文件进行读取，同时通过ftpWebRequest.GetRequestStream()方法创建一个Stream对象对服务器根目录进行写入，从而实现上传功能。

上传时会更新进度，直接修改label4.Text = percent + "%";时无法实时显示可以通过添加Application.DoEvents();解决，同时控制相关按钮的状态。

2.3.3.5 实现下载功能

下载时先判断是否选择了要下载的文件，和要下载到的本地目录。下载时同样是使用FtpWebRequest对象发起ftp请求，此时请求方法为下载文件WebRequestMethods.Ftp.DownloadFile。通过ftpWebResponse.GetResponseStream()方法创建一个Stream对象对服务器根目录进行读取，同时使用一个FileStream对象对本地文件进行写入，从而实现下载功能。

下载时会更新进度，直接修改label4.Text = percent + "%";时无法实时显示可以通过添加Application.DoEvents();解决，同时控制相关按钮的状态。

2.3.3.6 实现暂停功能

暂停功能由关闭正在使用的流及WebResponse对象实现，同时控制相关按钮的状态。

2.3.3.7 实现继续功能

继续功能分两种情况实现，当前正在上传时通过FtpWebRequest对象获取已上传的文件的长度并以此为起点继续上传，当前正在下载时通过FileStream对象获取已下载的文件的长度并以此为起点继续下载。获取服务器上的文件长度使用FtpWebRequest对象的方法为WebRequestMethods.Ftp.GetFileSize。继续功能比正常的上传下载多了获取当前已操作的文件的长度的步骤，也会更新进度，同时控制相关按钮的状态。

**结论**

**1. 程序主要界面及结果**

连接前：



连接后显示服务器目录树：



选择本地目录后显示本地目录树：



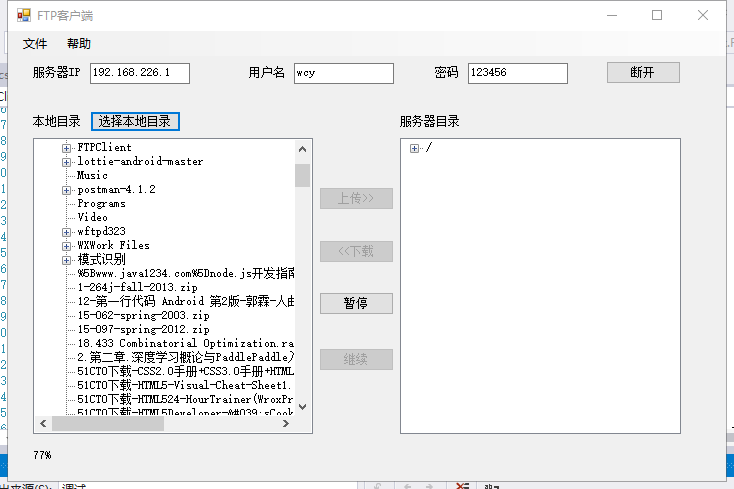
下载文件显示进度（截取暂停后的状态，别的太快截不到）：



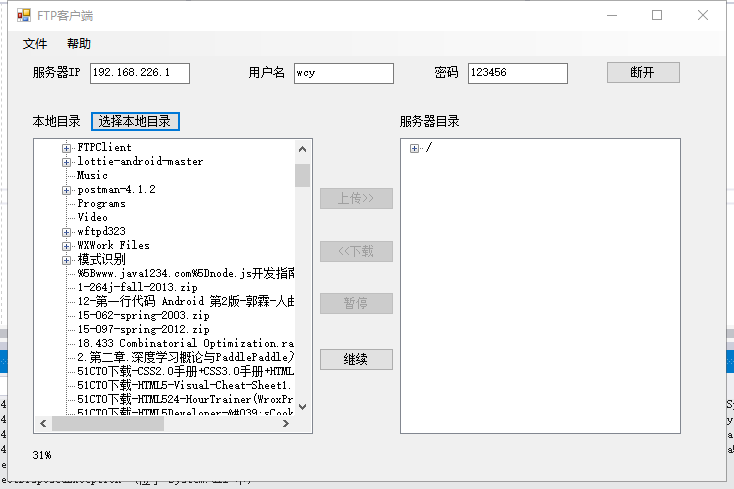
继续后下载完毕：



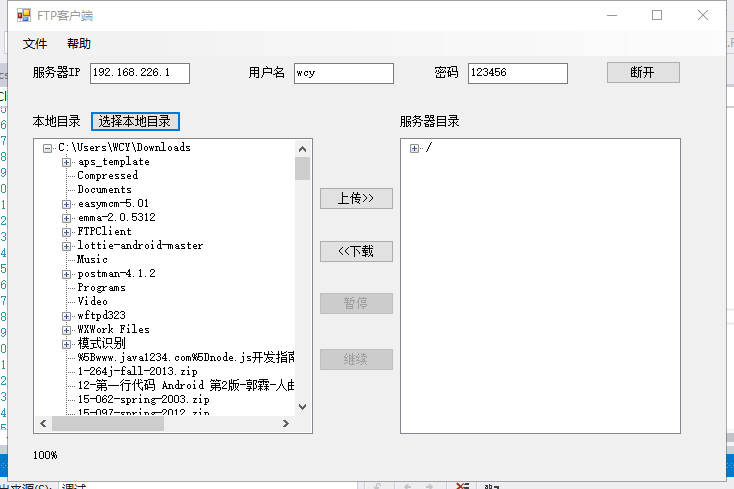
上传文件显示进度：



上传文件暂停后：



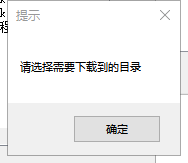
继续后上传完毕：



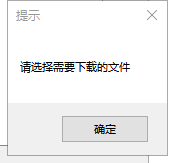
断开连接清空服务器相关信息：



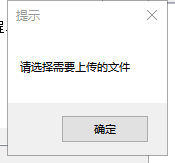
下载未选择本地目录时提示：



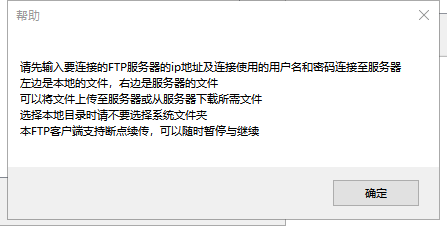
下载未选择远程文件时提示:



上传未选择本地文件时提示：



帮助提示：



**2. 程序源程序**

已上传至GitHub，地址：

<https://github.com/wcy00000000000000/NetworkClassFinalProject>

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Net;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace FTPClient

{

public partial class Form1 : Form

{

private FtpWebRequest ftpWebRequest;

private FtpWebResponse ftpWebResponse;

private String localPath = null;

private String ip;

private String user;

private String pwd;

private String ftpPath;

private String filePath;

private Stream readStream = null;

private FileStream writeStream = null;

private long totalByte = 0;

private bool isUpload = false;

public Form1()

{

InitializeComponent();

}

private void 帮助ToolStripMenuItem\_Click(object sender, EventArgs e)

{

MessageBox.Show("请先输入要连接的FTP服务器的ip地址及连接使用的用户名和密码连接至服务器\n" +

"左边是本地的文件，右边是服务器的文件\n可以将文件上传至服务器或从服务器下载所需文件\n" +

"选择本地目录时请不要选择系统文件夹\n本FTP客户端支持断点续传，可以随时暂停与继续", "帮助");

}

private void 上传ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Upload();

}

private void 下载ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Download();

}

private void 暂停ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Stop();

}

private void 继续ToolStripMenuItem\_Click(object sender, EventArgs e)

{

Resume();

}

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

{

if (button1.Text.Equals("连接"))

{

ip = textBox1.Text;

user = textBox2.Text;

pwd = textBox3.Text;

ftpPath = "ftp://" + ip;

上传ToolStripMenuItem.Enabled = true;

下载ToolStripMenuItem.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

button1.Text = "断开";

TreeNode root = new TreeNode("/");

root.Tag = ftpPath + "/";

root.Text = "/";

treeView2.Nodes.Add(root);

FillFTPTree(root, ftpPath);

}

else if (button1.Text.Equals("断开"))

{

上传ToolStripMenuItem.Enabled = false;

下载ToolStripMenuItem.Enabled = false;

暂停ToolStripMenuItem.Enabled = false;

继续ToolStripMenuItem.Enabled = false;

button2.Enabled = false;

button3.Enabled = false;

button4.Enabled = false;

button5.Enabled = false;

treeView2.Nodes.Clear();

button1.Text = "连接";

if (ftpWebRequest != null)

{

ftpWebRequest.Abort();

ftpWebRequest = null;

}

if (ftpWebResponse != null)

{

ftpWebResponse.Close();

ftpWebResponse = null;

}

}

}

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

{

Upload();

}

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

{

Download();

}

private void button4\_Click(object sender, EventArgs e)

{

Stop();

}

private void button5\_Click(object sender, EventArgs e)

{

Resume();

}

private void button6\_Click(object sender, EventArgs e)

{

FolderBrowserDialog folderBrowserDialog = new FolderBrowserDialog();

if (folderBrowserDialog.ShowDialog() == DialogResult.OK)

{

localPath = folderBrowserDialog.SelectedPath;

FillTreeView(treeView1, folderBrowserDialog.SelectedPath);

}

}

private void FillTreeView(TreeView tree, String path)

{

try

{

tree.Nodes.Clear();

TreeNode root = new TreeNode(path);

root.Tag = path;

root.Text = path;

tree.Nodes.Add(root);

FillTreeViewNode(root, path);

}

catch (Exception e)

{

Console.WriteLine(e.StackTrace);

}

}

private void FillTreeViewNode(TreeNode treeNode, String path)

{

if (Directory.Exists(path) == false)

{

return;

}

DirectoryInfo directory = new DirectoryInfo(path);

FileInfo[] fileInfos = directory.GetFiles();

DirectoryInfo[] dirs = directory.GetDirectories();

int i = 0;

foreach (DirectoryInfo d in dirs)

{

TreeNode node = new TreeNode();

node.Text = d.Name;

node.Tag = path + "\\" + d.Name;

treeNode.Nodes.Add(node);

string subPath = path + "\\" + d.Name;

FillTreeViewNode(treeNode.Nodes[i], subPath);

i++;

}

foreach (FileInfo f in fileInfos)

{

TreeNode node = new TreeNode();

node.Text = f.Name;

node.Tag = path + "\\" + f.Name;

treeNode.Nodes.Add(node);

}

}

private void Upload()

{

if (treeView1.SelectedNode == null || treeView1.SelectedNode.Nodes.Count > 0)

{

MessageBox.Show("请选择需要上传的文件", "提示");

return;

}

isUpload = true;

filePath = treeView1.SelectedNode.Tag.ToString();

暂停ToolStripMenuItem.Enabled = true;

上传ToolStripMenuItem.Enabled = false;

下载ToolStripMenuItem.Enabled = false;

button2.Enabled = false;

button3.Enabled = false;

button4.Enabled = true;

FileInfo fileInfo = new FileInfo(filePath);

totalByte = (long)fileInfo.Length;

string file = fileInfo.Name;

long startPosition = 0;

long startByte = startPosition;

int percent = (int)(startByte \* 100 / totalByte);

label4.Text = percent + "%";

Application.DoEvents();

ftpWebRequest = (FtpWebRequest)FtpWebRequest.Create(new Uri("ftp://" + user + "@" + ip + "/" + file));

ftpWebRequest.Credentials = new NetworkCredential(user, pwd);

ftpWebRequest.KeepAlive = false;

ftpWebRequest.Method = WebRequestMethods.Ftp.AppendFile;

ftpWebRequest.UseBinary = true;

ftpWebRequest.ContentLength = fileInfo.Length;

int buffLength = 2048;

byte[] buff = new byte[buffLength];

writeStream = fileInfo.OpenRead();

try

{

readStream = ftpWebRequest.GetRequestStream();

writeStream.Seek(startPosition, 0);

int contentLen = writeStream.Read(buff, 0, buffLength);

while (contentLen != 0)

{

readStream.Write(buff, 0, contentLen);

contentLen = writeStream.Read(buff, 0, buffLength);

startByte += contentLen;

percent = (int)(startByte \* 100 / totalByte);

label4.Text = percent + "%";

Application.DoEvents();

}

readStream.Close();

writeStream.Close();

暂停ToolStripMenuItem.Enabled = false;

上传ToolStripMenuItem.Enabled = true;

下载ToolStripMenuItem.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

button4.Enabled = false;

isUpload = false;

label4.Text = "100%";

Application.DoEvents();

}

catch

{

}

finally

{

if (writeStream != null)

{

writeStream.Close();

}

if (readStream != null)

{

readStream.Close();

}

}

}

private void Download()

{

if (treeView2.SelectedNode == null || treeView2.SelectedNode.Nodes.Count > 0)

{

MessageBox.Show("请选择需要下载的文件", "提示");

return;

}

if (localPath == null || localPath.Length == 0)

{

MessageBox.Show("请选择需要下载到的目录", "提示");

return;

}

filePath = treeView2.SelectedNode.Tag.ToString();

暂停ToolStripMenuItem.Enabled = true;

上传ToolStripMenuItem.Enabled = false;

下载ToolStripMenuItem.Enabled = false;

button2.Enabled = false;

button3.Enabled = false;

button4.Enabled = true;

button5.Enabled = false;

FtpWebRequest remoteFileLenReq;

try

{

writeStream = new FileStream(localPath + "\\" + treeView2.SelectedNode.Text, FileMode.Append);

remoteFileLenReq = (FtpWebRequest)FtpWebRequest.Create(filePath);

remoteFileLenReq.Credentials = new NetworkCredential(user, pwd);

remoteFileLenReq.UseBinary = true;

remoteFileLenReq.ContentOffset = 0;

remoteFileLenReq.Method = WebRequestMethods.Ftp.GetFileSize;

FtpWebResponse response = (FtpWebResponse)remoteFileLenReq.GetResponse();

totalByte = response.ContentLength;

response.Close();

ftpWebRequest = (FtpWebRequest)FtpWebRequest.Create(filePath);

ftpWebRequest.Credentials = new NetworkCredential(user, pwd);

ftpWebRequest.UseBinary = true;

ftpWebRequest.KeepAlive = false;

ftpWebRequest.ContentOffset = 0;

ftpWebRequest.Method = WebRequestMethods.Ftp.DownloadFile;

ftpWebResponse = (FtpWebResponse)ftpWebRequest.GetResponse();

readStream = ftpWebResponse.GetResponseStream();

long downloadedByte = 0;

int bufferSize = 512;

byte[] btArray = new byte[bufferSize];

int contentSize = readStream.Read(btArray, 0, btArray.Length);

while (contentSize > 0)

{

downloadedByte += contentSize;

int percent = (int)(downloadedByte \* 100 / totalByte);

label4.Text = percent + "%";

Application.DoEvents();

writeStream.Write(btArray, 0, contentSize);

contentSize = readStream.Read(btArray, 0, btArray.Length);

}

暂停ToolStripMenuItem.Enabled = false;

上传ToolStripMenuItem.Enabled = true;

下载ToolStripMenuItem.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

button4.Enabled = false;

button5.Enabled = false;

readStream.Close();

writeStream.Close();

ftpWebResponse.Close();

return;

}

catch (Exception e)

{

Console.WriteLine(e.StackTrace);

}

finally

{

if (readStream != null)

{

readStream.Close();

}

if (writeStream != null)

{

writeStream.Close();

}

}

}

private void Stop()

{

button4.Enabled = false;

button5.Enabled = true;

暂停ToolStripMenuItem.Enabled = false;

继续ToolStripMenuItem.Enabled = true;

button2.Enabled = false;

button3.Enabled = false;

button4.Enabled = false;

button5.Enabled = true;

if (readStream != null)

{

readStream.Close();

}

if (writeStream != null)

{

writeStream.Close();

}

if (readStream != null)

{

readStream.Close();

}

if (ftpWebResponse != null)

{

ftpWebResponse.Close();

}

}

private void Resume()

{

button5.Enabled = false;

button4.Enabled = true;

继续ToolStripMenuItem.Enabled = false;

暂停ToolStripMenuItem.Enabled = true;

button4.Enabled = true;

button5.Enabled = false;

if (isUpload)

{

if (treeView1.SelectedNode == null || treeView1.SelectedNode.Nodes.Count > 0)

{

MessageBox.Show("请选择需要上传的文件", "提示");

return;

}

isUpload = true;

filePath = treeView1.SelectedNode.Tag.ToString();

暂停ToolStripMenuItem.Enabled = true;

上传ToolStripMenuItem.Enabled = false;

下载ToolStripMenuItem.Enabled = false;

button2.Enabled = false;

button3.Enabled = false;

button4.Enabled = true;

FileInfo fileInfo = new FileInfo(filePath);

string file = fileInfo.Name;

FtpWebRequest reqFTP;

reqFTP = (FtpWebRequest)FtpWebRequest.Create(ftpPath + "/" + file);

reqFTP.KeepAlive = false;

reqFTP.UseBinary = true;

reqFTP.Credentials = new NetworkCredential(user, pwd);

reqFTP.Method = WebRequestMethods.Ftp.GetFileSize;

FtpWebResponse response = (FtpWebResponse)reqFTP.GetResponse();

long startPosition = response.ContentLength;

long startByte = startPosition;

int percent = (int)(startByte \* 100 / totalByte);

label4.Text = percent + "%";

Application.DoEvents();

ftpWebRequest = (FtpWebRequest)FtpWebRequest.Create(new Uri(ftpPath + "/" + file));

ftpWebRequest.Credentials = new NetworkCredential(user, pwd);

ftpWebRequest.KeepAlive = false;

ftpWebRequest.Method = WebRequestMethods.Ftp.AppendFile;

ftpWebRequest.UseBinary = true;

ftpWebRequest.ContentLength = fileInfo.Length;

int buffLength = 2048;

byte[] buff = new byte[buffLength];

writeStream = fileInfo.OpenRead();

try

{

readStream = ftpWebRequest.GetRequestStream();

writeStream.Seek(startPosition, 0);

int contentLen = writeStream.Read(buff, 0, buffLength);

while (contentLen != 0)

{

readStream.Write(buff, 0, contentLen);

contentLen = writeStream.Read(buff, 0, buffLength);

startByte += contentLen;

percent = (int)(startByte \* 100 / totalByte);

label4.Text = percent + "%";

Application.DoEvents();

}

readStream.Close();

writeStream.Close();

暂停ToolStripMenuItem.Enabled = false;

上传ToolStripMenuItem.Enabled = true;

下载ToolStripMenuItem.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

button4.Enabled = false;

isUpload = false;

}

catch

{

throw;

}

finally

{

if (writeStream != null)

{

writeStream.Close();

}

if (readStream != null)

{

readStream.Close();

}

}

}

else

{

try

{

writeStream = new FileStream(localPath + "\\" + treeView2.SelectedNode.Text, FileMode.Append);

long startPosition = writeStream.Length;

if (startPosition >= totalByte)

{

writeStream.Close();

return;

}

ftpWebRequest = (FtpWebRequest)FtpWebRequest.Create(filePath);

ftpWebRequest.Credentials = new NetworkCredential(user, pwd);

ftpWebRequest.UseBinary = true;

ftpWebRequest.KeepAlive = false;

ftpWebRequest.ContentOffset = startPosition;

ftpWebRequest.Method = WebRequestMethods.Ftp.DownloadFile;

ftpWebResponse = (FtpWebResponse)ftpWebRequest.GetResponse();

readStream = ftpWebResponse.GetResponseStream();

long downloadedByte = startPosition;

int bufferSize = 512;

byte[] btArray = new byte[bufferSize];

int contentSize = readStream.Read(btArray, 0, btArray.Length);

while (contentSize > 0)

{

downloadedByte += contentSize;

int percent = (int)(downloadedByte \* 100 / totalByte);

label4.Text = percent + "%";

Application.DoEvents();

writeStream.Write(btArray, 0, contentSize);

contentSize = readStream.Read(btArray, 0, btArray.Length);

}

暂停ToolStripMenuItem.Enabled = false;

上传ToolStripMenuItem.Enabled = true;

下载ToolStripMenuItem.Enabled = true;

button2.Enabled = true;

button3.Enabled = true;

button4.Enabled = false;

button5.Enabled = false;

readStream.Close();

writeStream.Close();

ftpWebResponse.Close();

return;

}

catch (Exception e)

{

Console.WriteLine(e.StackTrace);

}

finally

{

if (readStream != null)

{

readStream.Close();

}

if (writeStream != null)

{

writeStream.Close();

}

}

}

}

private void FillFTPTree(TreeNode treeNode, String path)

{

int i = 0;

foreach (String d in GetDirctory(path))

{

TreeNode node = new TreeNode();

node.Text = d;

node.Tag = path + "/" + d;

treeNode.Nodes.Add(node);

string subPath = path + "/" + d;

FillFTPTree(treeNode.Nodes[i], subPath);

i++;

}

foreach (string f in GetFile(path))

{

TreeNode node = new TreeNode();

node.Text = f;

node.Tag = path + "/" + f;

treeNode.Nodes.Add(node);

}

}

private List<string> GetDirctory(string path)

{

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

try

{

ftpWebRequest = (FtpWebRequest)FtpWebRequest.Create(path + "/");

ftpWebRequest.Credentials = new NetworkCredential(user, pwd);

ftpWebRequest.Method = WebRequestMethods.Ftp.ListDirectoryDetails;

WebResponse response = ftpWebRequest.GetResponse();

StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);

string line = reader.ReadLine();

while (line != null)

{

if (line[0] >= '0' && line[0] <= '9' && line.Contains("<DIR>"))

{

//Windows

string msg = line.Substring(line.LastIndexOf("<DIR>") + 5).Trim();

strs.Add(msg);

}

else if (line[0] == 'd')

{

//Linux

string msg = line.Substring(53).Trim();

strs.Add(msg);

}

line = reader.ReadLine();

}

reader.Close();

response.Close();

return strs;

}

catch (Exception e)

{

Console.WriteLine(e.StackTrace);

}

return strs;

}

private List<string> GetFile(String path)

{

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

try

{

ftpWebRequest = (FtpWebRequest)FtpWebRequest.Create(path + "/");

ftpWebRequest.Credentials = new NetworkCredential(user, pwd);

ftpWebRequest.Method = WebRequestMethods.Ftp.ListDirectoryDetails;

WebResponse response = ftpWebRequest.GetResponse();

StreamReader reader = new StreamReader(response.GetResponseStream(), Encoding.Default);

string line = reader.ReadLine();

while (line != null)

{

if (line[0] >= '0' && line[0] <= '9' && !line.Contains("<DIR>") && line.Length > 39)

{

string msg = line.Substring(39).Trim();

strs.Add(msg);

}

else if (line[0] != 'd' && line.Length > 53)

{

//Linux

string msg = line.Substring(53).Trim();

strs.Add(msg);

}

line = reader.ReadLine();

}

reader.Close();

response.Close();

return strs;

}

catch (Exception e)

{

Console.WriteLine(e.StackTrace);

}

return strs;

}

}

}

**参考文献**

[1]

[2]

[3]

|  |
| --- |
| 【结论】：本次实验中我使用C#完成了一个FTP客户端的编写，该客户端拥有图形用户化界面，可以连接FTP服务器，上传下载文件，并且支持断点续传功能。 |
| 【小结】：通过本次实验，我对FTP网络协议和网络编程方面有了更深刻的认识。本次实验中我学会了如何搭建FTP服务器，学习了FTP协议与其工作模式和使用的命令，加强了对网络协议的认识和使用C#进行网络编程的能力。 |
| 指导老师评语及成绩 |
| 【评语】：  成 绩： 指导老师签名：  批阅日期：2019.5.10 |