网络编程设计开发文档

学生：代领

学号：919113100209

专业： 软件工程

教师： 吕建勇

目录

[系统需求分析 3](#_Toc91164257)

[系统设计 3](#_Toc91164258)

[系统架构 3](#_Toc91164259)

[系统流程 4](#_Toc91164260)

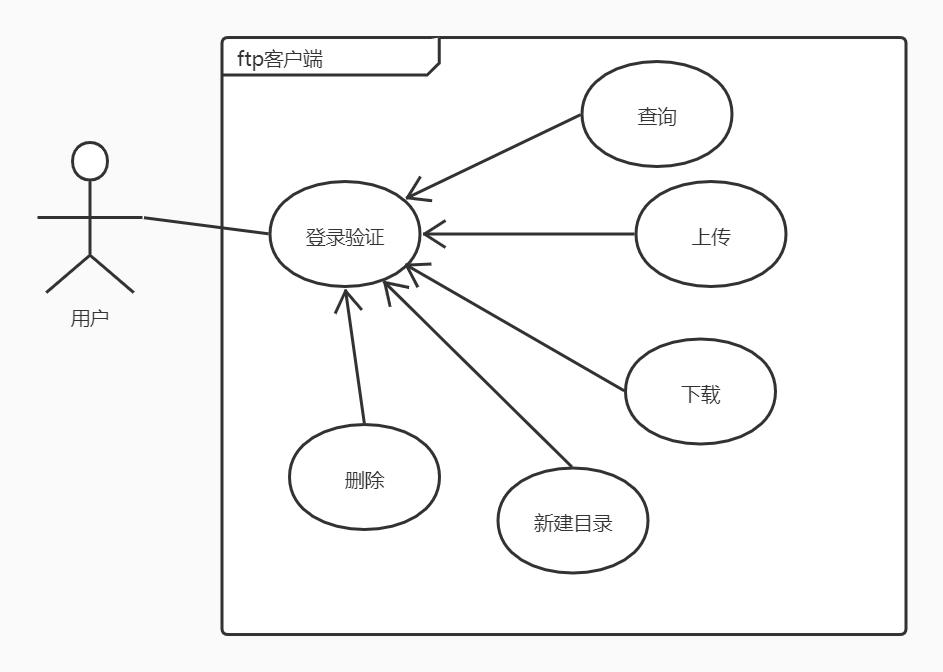
[详细设计及实现 4](#_Toc91164261)

[个人心得 10](#_Toc91164262)

# 系统需求分析

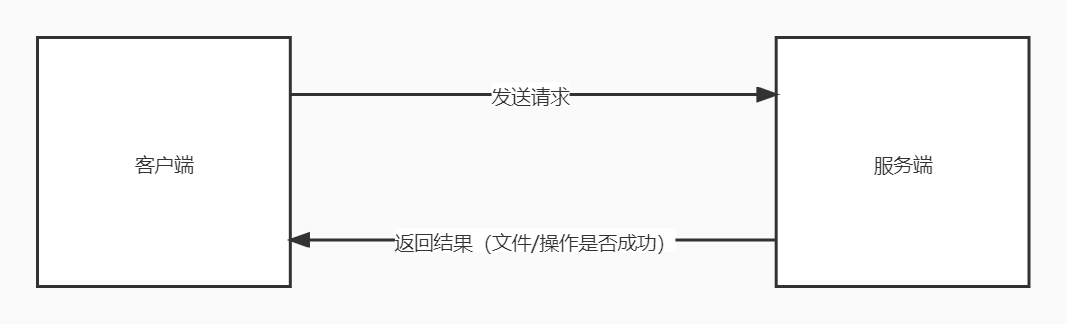
本项目作为网络编程课程设计的作业，使用visual studio 2019的MFC平台，制作一个WIN32程序，该程序实现了ftp客户端。该客户端需实现以下功能：

文件上传、下载、新建目录等操作。

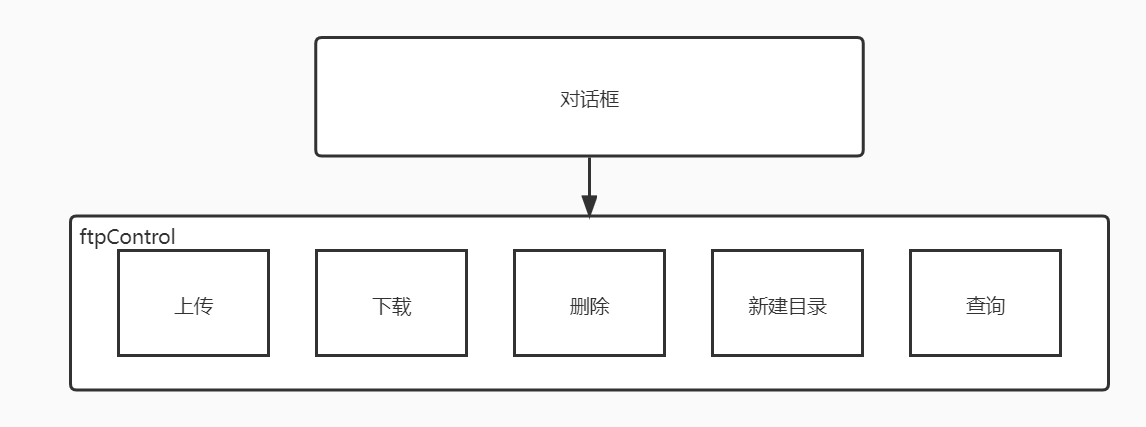


# 系统设计

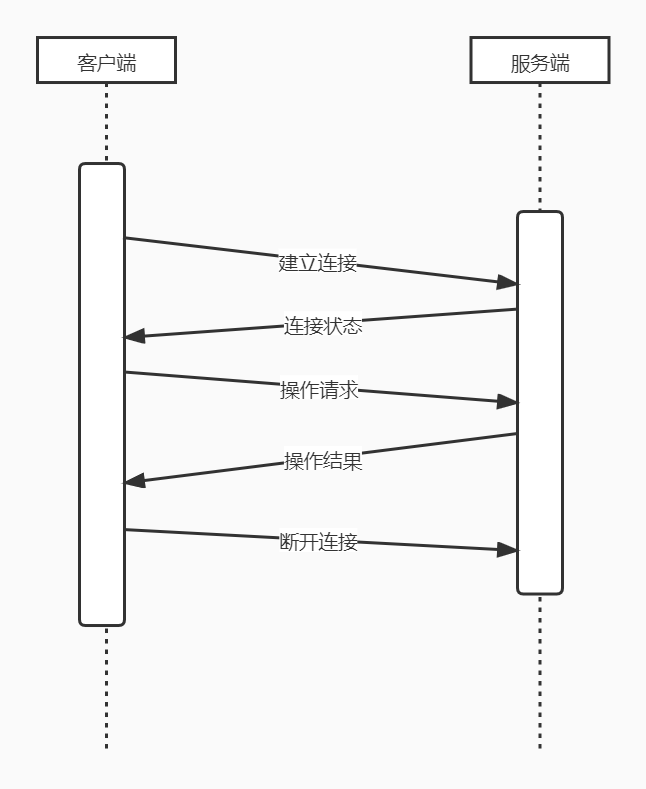
### 系统架构



### 功能模块



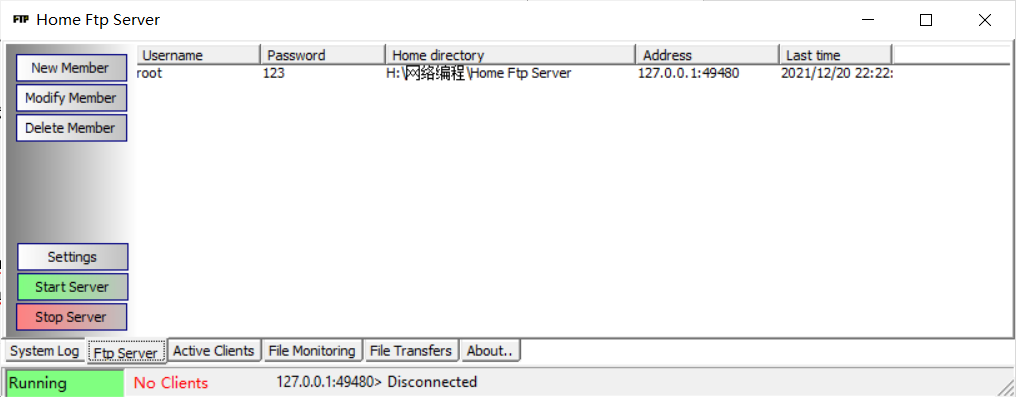
### 系统流程



### 详细设计及实现

本项目使用Home Ftp Server作为ftp服务端。

点击Start Server运行服务端，服务端开始间听请求。



#### 开始界面

BOOL CMyFtpClientApp::InitInstance()：初始化界面。

BOOL CMyFtpClientDlg::OnInitDialog()：初始化，默认ip为127.0.0.1，用户名为root，密码为123。将除连接以外的3个按钮非使能。

效果如下：



#### 点击连接

虽然名为连接，但是考虑到用户操作频率并不高，所以可以每次进行操作的时候再连接，以此节约双方资源。

void CMyFtpClientDlg::FtpQuery()：进行一次文件夹查询，第一次查询。流程：取出当前连接相关的变量。开始一个线程，执行回调函数：Query

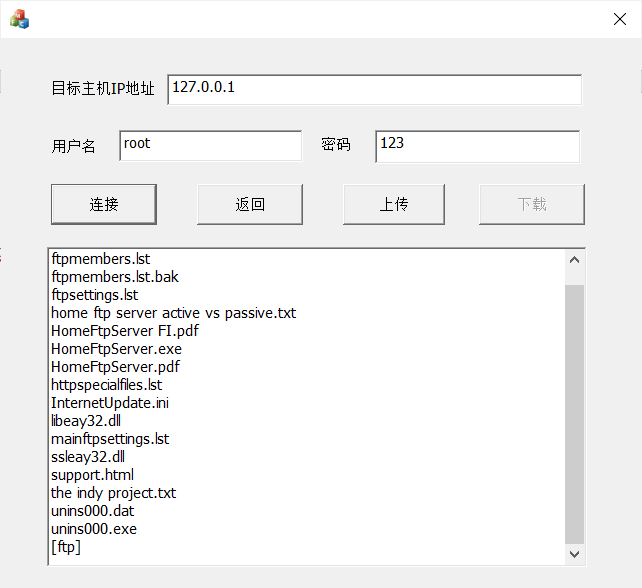
UINT Query(LPVOID pParam)：使用CInternetSession建立会话，创建CFtpConnection将取出的变量作为参数传入会话用于创建连接的函数GetFtpConnection()，得到连接对象。

将创建好的连接对象传入CFtpFileFind的构造函数得到一个CFtpFileFind的对象，然后用这个对象FindFile函数获取服务端根目录下的所有文件。将查找到的所有文件加入CListBox，最后更新界面，使能返回后上传按钮。

连接建立关键代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | CInternetSession\* pSession = NULL;//会话  CFtpConnection\* pConnection = NULL;//连接  CFtpFileFind\* pFileFind = NULL;//查询  CString fileName;  **BOOL** isContinue=0;    **while** (pList->GetCount()) {      pList->DeleteString(0);  }  //Internet会话  pSession = **new** CInternetSession(AfxGetAppName(), 1, PRE\_CONFIG\_INTERNET\_ACCESS);  **try**{      // 连接服务器      pConnection = pSession->GetFtpConnection(ip, user, pwd);  }  **catch** (CInternetException\* e) {      // 连接失败      AfxMessageBox("连接失败!\n", MB\_OK | MB\_ICONSTOP);      e->Delete();      pConnection = NULL;  } |

效果如下：



#### 双击文件或文件夹

void CMyFtpClientDlg::OnDblclkList1()：

如果是文件，则调用下载操作；如果是文件夹，则将该文件夹名加入路径。使用一个vector来储存路径，每次进入一个文件夹，就把这个文件夹名push进去；退出则将最顶层的文件夹名pop出来。

例如，双击上图ftp效果如下：



#### 返回操作

将vector的最顶层文件夹名pop出来，然后当前这个vector中存的路径。

在上图的基础上操作，效果如下：



#### 右键操作

如果没有选中任何文件或者文档，则灰化删除下载重命名按钮。

点击菜单选项即可执行相应操作。

右键效果如下图所示：



#### 上传操作

先获取本地需要上传的文件路径，得到路径之后建立连接进行上传。

#### 新建目录

new一个获取名称的对话框对象，用以获取名称。

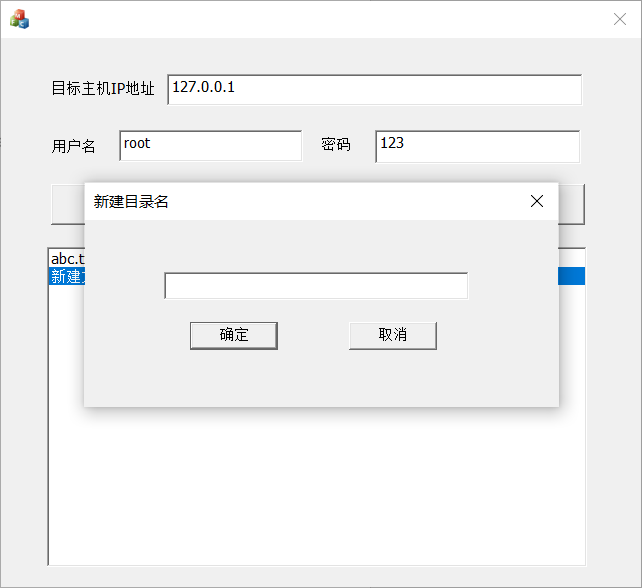
然后调用CreateDirectory函数进行创建目录



#### 重命名

new一个获取名称的对话框对象，用以获取名称。

然后调用rename函数进行重命名。



#### 下载操作

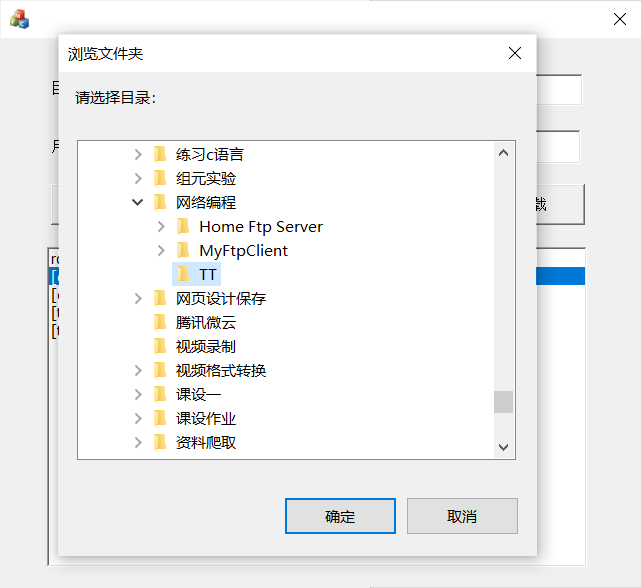
选中文件之后，即可下载。

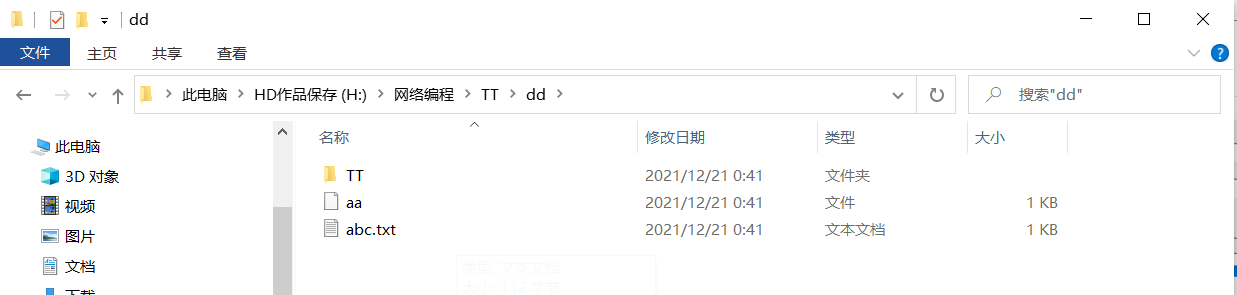
但是为了下载更加方便，我增加了递归下载文件的功能，虽然这个功能会给网络增加很多负担，但是也提供了很大的便捷。

核心代码如下：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | **void** MyDownloadFolder(CFtpConnection\* pConnection,CString path,CString destPath) {      pConnection->SetCurrentDirectoryA(path);      CFtpFileFind finder(pConnection);  **BOOL** isContinue = finder.FindFile(path);      queue<CString> q;  **while** (isContinue) {          isContinue = finder.FindNextFileA();          CString fileName = finder.GetFileName();  **if** (finder.IsDirectory()) {              q.push(fileName);              CreateDirectory(destPath + "/" + fileName, NULL);          }  **else** {              pConnection->GetFile(path + "/" + fileName, destPath + "/" + fileName);          }      }      finder.Close();  **while** (!q.empty()) {          CString curDir = q.front();          q.pop();          MyDownloadFolder(pConnection, path + "/" + curDir, destPath + "/" + curDir);      }  } |

例如下载dd文件夹





可以看见按照原来的目录结构保存到了本地。

# 个人心得

通过此次课程设计的编写，我学到了网络编程的要领。相较于其他的编程任务，网络编程更加侧重于处理线程管理、网络请求收发等操作。

此外，还学会了MFC这个工具，可以通过MFC这个工具开发一些简单的Win32小程序。

在这个项目中，最棘手的问题是递归下载文件夹的操作，最开始Debug的时候很困惑，因为自己认为逻辑上是对的，实际上没有考虑到两个CFtpFind是不可以使用同一个connection的，每一次需要重新修改connection的根目录。并且，在查询完一个目录之后，还需要把CFtpFind关掉。这些问题，还是由于没有深入理解源码所导致。可见，静下心读源码，完全搞懂原理，有时候比上网搜索问题更节约时间。

最后，感谢老师的教诲，带我们入门网络编程，学会一门工具，可以更加深入编程世界的海洋。