# 上机作业三：使用UDP实现可靠的文件传输实验报告

## 1 实验要求

1. 下层使用UDP协议（即使用数据报套接字完成本次程序）；
2. 完成客户端和服务器端程序；
3. 实现可靠的文件传输：能可靠下载文件，能同时下载文件。

## 2 实验环境

Windows 10、Visual Studio 2019 preview、Debug x64、MFC

## 3 实验原理

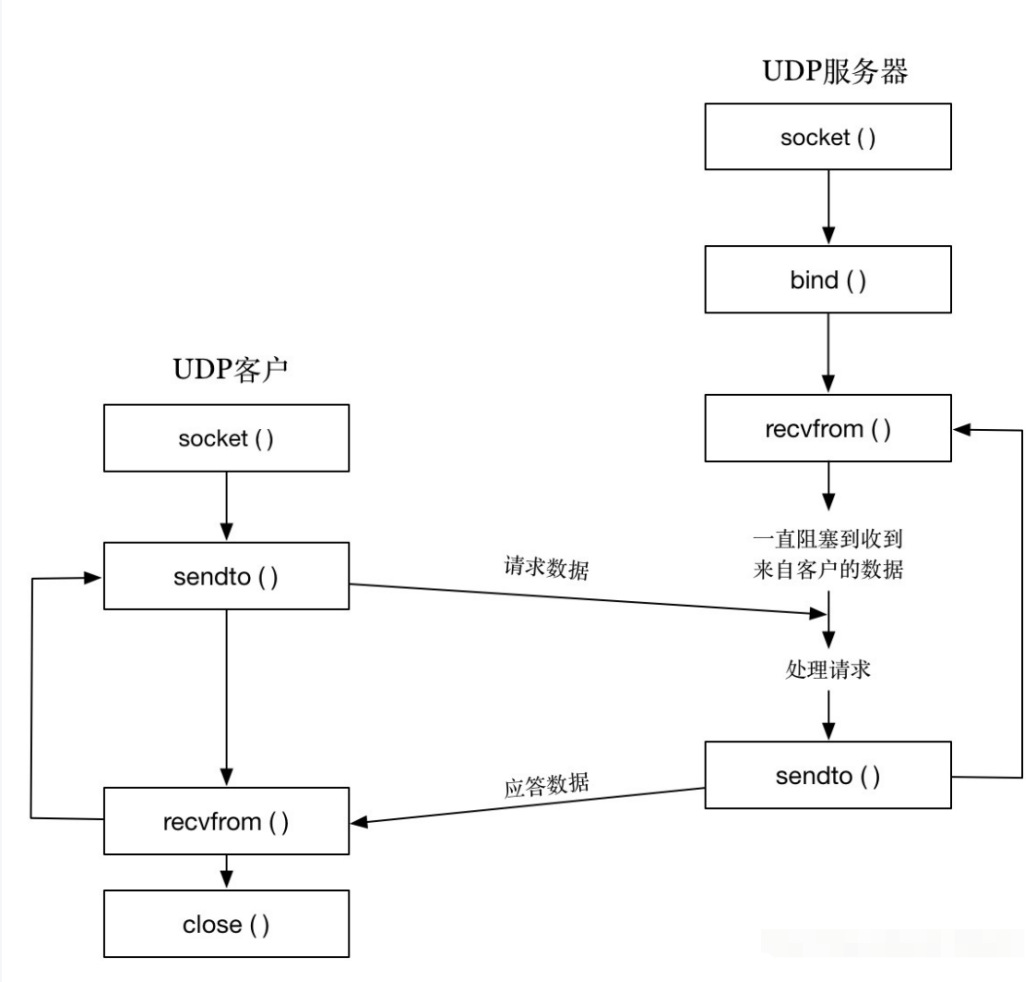
### 3.1 UDP简介

1. UDP(User Datagram Protocol)传输与IP传输非常类似。你可以将UDP协议看作IP协议暴露在传输层的一个接口。UDP协议同样以数据包(datagram)的方式传输，它的传输方式也是"Best Effort"的，所以UDP协议也是不可靠的(unreliable)。那么，我们为什么不直接使用IP协议而要额外增加一个UDP协议呢？ 一个重要的原因是IP协议中并没有端口(port)的概念。IP协议进行的是IP地址到IP地址的传输，这意味者两台计算机之间的对话。但每台计算机中需要有多个通信通道，并将多个通信通道分配给不同的进程使用。一个端口就代表了这样的一个通信通道。正如我们在邮局和邮差中提到的收信人的概念一样。UDP协议实现了端口，从而让数据包可以在送到IP地址的基础上，进一步可以送到某个端口。
2. UDP是一个无连接协议，传输数据之前源端和终端不建立连接，当UDP它想传送时就简单地去抓取来自应用程序的数据，并尽可能快地把它扔到网络上。在发送端，UDP传送数据的速度仅仅是受应用程序生成数据的速度、计算机的能力和传输带宽的限制;在接收端，UDP把每个消息段放在队列中，应用程序每次从队列中读一个消息段。
3. 由于传输数据不建立连接，因此也就不需要维护连接状态，包括收发状态等，因此一台服务机可同时向多个客户机传输相同的消息。
4. UDP信息包的标题很短，只有8个字节，相对于TCP的20个字节信息包的额外开销很小。
5. 吞吐量不受拥挤控制算法的调节，只受应用软件生成数据的速率、传输带宽、源端和终端主机性能的限制。
6. UDP使用尽最大努力交付，即不保证可靠交付，因此主机不需要维持复杂的链接状态表(这里面有许多参数)。
7. UDP是面向报文的。发送方的UDP对应用程序交下来的报文，在添加首部后就向下交付给IP层。既不拆分，也不合并，而是保留这些报文的边界，因此，应用程序需要选择合适的报文大小。
8. UDP属于传输层,下面我们由下至上一步一步来看: 以太网(Ethernet)数据帧的长度必须在46-1500字节之间,这是由以太网的物理特性决定的。 这个1500字节被称为链路层的MTU(最大传输单元)。 但这并不是指链路层的长度被限制在1500字节，其实这个MTU指的是链路层的数据区并不包括链路层的首部和尾部的18个字节。所以事实上这个1500字节就是网络层IP数据报的长度限制。因为IP数据报的首部为20字节，所以IP数据报的数据区长度最大为1480字节。而这个1480字节就是用来放TCP传来的TCP报文段

或UDP传来的UDP数据报的。又因为UDP数据报的首部8字节，所以UDP数据报的数据区最大长度为1472字节。这个1472字节就是我们可以使用的字节数。

9. 在UDP套接字程序中，客户不需要与服务器建立连接，而只管直接使用`sendto`函数给服务器发送数据报。同样的，服务器不需要接受来自客户的连接，而只管调用`recvfrom`函数，等待来自某个客户的数据到达。

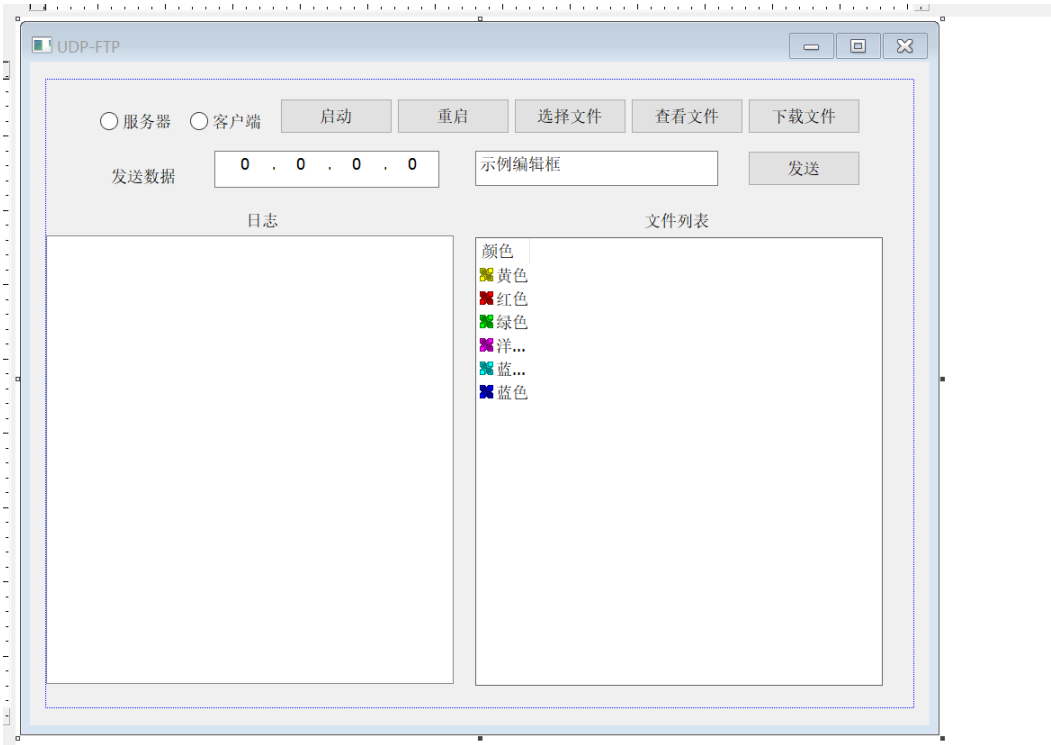
## 3.2 实验原理图




# 4 实现步骤

## 4.1 创建MFC应用程序

1. 在资源视图中构建以下界面




2. 修改控件的属性，如report，组合服务器与客户端，取消排序，滚动栏等。

## 4.2 宏定义

在头文件里定义文件名的长度以及数据的长度

```
#define Max 1025
#define MTU 1024
#define T 2
#define FILENAMELEN 99
#define FILENUM 10
```

因为UDP的MTU最好小于1472，所以取1024。

## 4.3 结构声明

在对话框类的头文件里声明数据包的结构

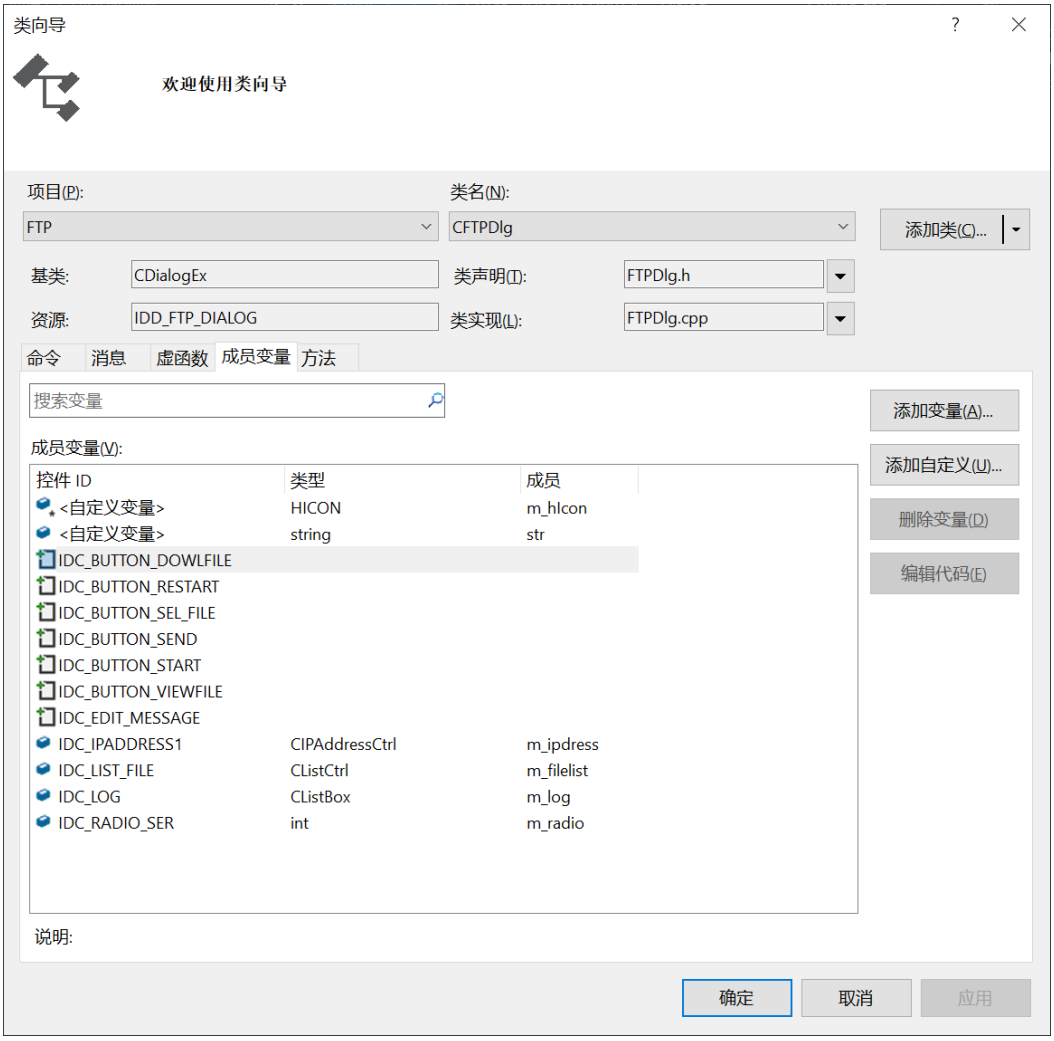
```
typedef struct _FileName
{
    char filename[FILENAMELEN];//文件名

}FileName;

typedef struct _Packet
{
    int flag;//传输的标记
    long Number;//数据包的序号
    long Total;//总大小
    char filename[100];//文件名
    BYTE Data[MTU];//数据
    int length;//数据的长度
```

```
    int Totallength;//文件的总长度
    FileName mfilelist[FILENUM];//文件列表

    _Packet()//初始化函数
    {
        flag = 0;
        Number = 0;
        Total = 0;
        memset(filename, 0, 0);
        for (int mindex = 0; mindex < MTU; mindex++)
        {
            Data[mindex] = 0;
        }
        length = 0;
        Totallength = 0;
        for (int mindex = 0; mindex < FILENUM; mindex++)
        {
            memset(mfilelist[mindex].filename, 0, FILENAMELEN);
        }
    }

    void init()
    {
        flag = 0;
        Number = 0;
        Total = 0;
        memset(filename, 0, 0);
        for (int mindex = 0; mindex < MTU; mindex++)
        {
            Data[mindex] = 0;
        }
        length = 0;
        Totallength = 0;
        for (int mindex = 0; mindex < FILENUM; mindex++)
        {
            memset(mfilelist[mindex].filename, 0, FILENAMELEN);
        }
    }

    void initdata()
    {
        for (int mindex = 0; mindex < MTU; mindex++)
        {
            Data[mindex] = 0;
        }
    }

    void initfilelist()
    {
        for (int mindex = 0; mindex < FILENUM; mindex++)
        {
            memset(mfilelist[mindex].filename, 0, FILENAMELEN);
```

```
        }
    }

}Packet;
```

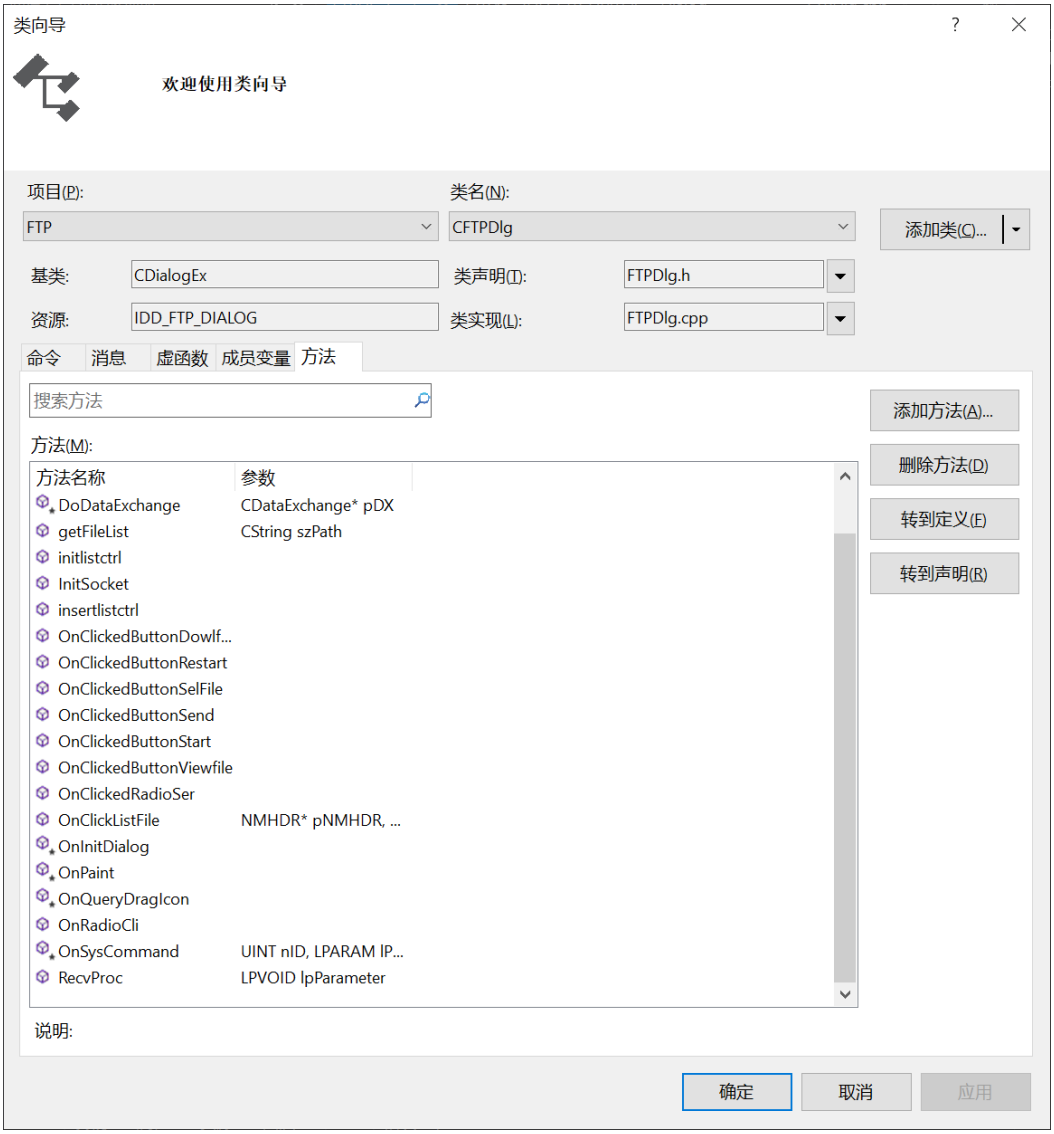
## 4.4 变量声明

使用类向导对对话框类里的控件进行绑定，结果如下




## 4.5 方法声明

MFC的方法同样可以使用类向导声明，比如：`getFileList`,所有的方法如下：




## 4.6 创建哈希表

哈希表一个用来保存写文件，一个读文件，还有一个是服务器保存文件列表

```
map<int, CString> map_filelist;

//服务器写入文件的句柄,用来提供多客户端同时上传文件
map<SOCKADDR_IN*, CFile*> map_readfile;
map<SOCKADDR_IN*, CFile*> map_writfile;
```

## 4.7 初始化

主要是初始化控件与套接字

```
// TODO: 在此添加额外的初始化代码
    m_log.SetHorizontalExtent(500);
    initlistctrl();
    InitSocket();

    IndexList = -1;

    GetDlgItem(IDC_BUTTON_SEL_FILE)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON_VIEWFILE)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON_DOWLFILE)->EnableWindow(FALSE);
    GetDlgItem(IDC_BUTTON_SEND)->EnableWindow(FALSE);
```

### 4.7.1 初始化文件列表

```
// 初始化文件列表
void CFTPDlg::initlistctrl()
{
    // TODO: 在此处添加实现代码.
    LONG lStyle;
    lStyle = GetWindowLong(m_filelist.m_hWnd, GWL_STYLE);//获取当前窗口style
    lStyle &= ~LVS_TYPEMASK; //清除显示方式位
    lStyle |= LVS_REPORT; //设置style
    SetWindowLong(m_filelist.m_hWnd, GWL_STYLE, lStyle);//设置style

    DWORD dwStyle = m_filelist.GetExtendedStyle();
    dwStyle |= LVS_EX_FULLROWSELECT;//选中某行使整行高亮（只适用与report风格的listctrl)
    dwStyle |= LVS_EX_GRIDLINES;//网格线（只适用与report风格的listctrl)
    m_filelist.SetExtendedStyle(dwStyle); //设置扩展风格
    m_filelist.InsertColumn(0, _T("文件名"), LVCFMT_LEFT, 315);//插入列
}
```

### 4.7.2 创建套接字

```
// 创建套接字
bool CFTPDlg::InitSocket()
{
    // TODO: 在此处添加实现代码.
    m_socket=socket(AF_INET,SOCK_DGRAM,0);
    if(INVALID_SOCKET==m_socket)
    {
        MessageBox(_T("套接字创建失败! "));
        return FALSE;
    }
    return TRUE;
}
```

## 4.8 选择客户端或者服务器

```
void CFTPDlg::OnClickedRadioSer()
{
    // TODO: 在此添加控件通知处理程序代码
    m_radio = 0;

    UpdateData(FALSE);
}


void CFTPDlg::OnRadioCli()
{
    // TODO: 在此添加命令处理程序代码
    m_radio = 1;
    m_ipdress.SetAddress(127, 0, 0, 1);//设置控件默认值
    UpdateData(FALSE);
}
```

## 4.9 点击启动

### 4.9.1 服务器

如果是服务器启动则绑定端口，获取当前文件夹下的文件并显示，设置按键状态，创建接收消息的线程。

```
void CFTPDlg::OnClickedButtonStart()
{
    // TODO: 在此添加控件通知处理程序代码
    if (m_radio == 0)//如果是服务器就绑定端口号
    {
        BindSocket();
        getFileList(GetExePath());

        m_log.AddString(_T("服务器已启动! "));
        //移动指针到末尾
```

```
        int nCount = m_log.GetCount();
        if (nCount > 0)
            m_log.SetCurSel(nCount - 1);
        //设置按钮状态
        GetDlgItem(IDC_RADIO_SER)->EnableWindow(FALSE);
        GetDlgItem(IDC_RADIO_CLI)->EnableWindow(FALSE);
        GetDlgItem(IDC_BUTTON_START)->EnableWindow(FALSE);

        GetDlgItem(IDC_BUTTON_SEL_FILE)->EnableWindow(FALSE);
        GetDlgItem(IDC_BUTTON_VIEWFILE)->EnableWindow(FALSE);
        GetDlgItem(IDC_BUTTON_DOWLFILE)->EnableWindow(FALSE);
        GetDlgItem(IDC_BUTTON_SEND)->EnableWindow(TRUE);
        GetDlgItem(IDC_IPADDRESS1)->EnableWindow(FALSE);
        /*GetDlgItem(IDC_STATIC)->EnableWindow(FALSE);

        GetDlgItem(IDC_EDIT_MESSAGE)->EnableWindow(FALSE);*/

    }
    HANDLE hThread = CreateThread(NULL, 0, RecvProc, this, 0,
NULL);//创建接收消息线程
    CloseHandle(hThread);

}
```

服务器绑定套接字端口

```
// 绑定套接字端口
void CFTPDlg::BindSocket()
{
    // TODO: 在此处添加实现代码.
    if (m_radio == 0)
    {
        SOCKADDR_IN addrSock;//创建SOCKADDR_IN结构体变量
        addrSock.sin_family = AF_INET;//使用IPV4地址
        addrSock.sin_port = htons(6000);//端口号
        addrSock.sin_addr.S_un.S_addr = htonl(INADDR_ANY);//IP地
址

        int retval;
        retval = bind(m_socket, (SOCKADDR*)&addrSock,
sizeof(SOCKADDR));//绑定
        if (SOCKET_ERROR == retval)
        {
            closesocket(m_socket);
            MessageBox(_T("绑定失败!"));
            return;
        }
    }

    return;
}
```

服务器获取文件夹下的文件

```
// 获取文件夹下的文件
void CFTPDlg::getFileList(CString szPath)
{
    // TODO: 在此处添加实现代码.
    m_filelist.DeleteAllItems();
    CFileFind finder;
    //建立CString对象，设置检索匹配字符串
    CString strWildcard(szPath);
    strWildcard += _T("\\*.*");

    //文件内部检索
    BOOL bWorking = finder.FindFile(strWildcard);
    int i; //用于定位字符，作为下标
    int index = 0;    //用于列表索引

    CString fontName, lastType, filepath, filename, filesize;//存储切割结果
    while (bWorking)
    {
        bWorking = finder.FindNextFile();
        filename = finder.GetFileName();
        long imageSize = finder.GetLength();
        if (filename != _T(".") && filename != _T(".."))
        {
            filepath = finder.GetFilePath();
            for (i = 0; i < filename.GetLength(); i++)
            {
                if (filename[i] == '.')
                {
                    fontName = filename.Mid(0, i);
                    lastType = filename.Mid(i + 1, filename.GetLength());

                    CString filen;
                    filen.Format(_T("%s%s%s"),fontName, _T("."), lastType);
                    m_filelist.InsertItem(index, filen);
                    map_filelist[mapindex++] = filen;
                    //添加到服务器列表
                }
            }
            index++;
        }

    }
    finder.Close();
}
```

服务器插入到控件

```
void CFTPDlg::insertlistctrl()
{
    // TODO: 在此处添加实现代码.
    m_filelist.DeleteAllItems();
    int index = 0;

    for (; index < MTU; index++)
    {
        if (list_filelist[index] == "")
        {
            break;
        }
        m_filelist.InsertItem(index, _T(""));//插入行
        m_filelist.SetItemText(index, 0, list_filelist[index]);//
设置数据
    }
}
```

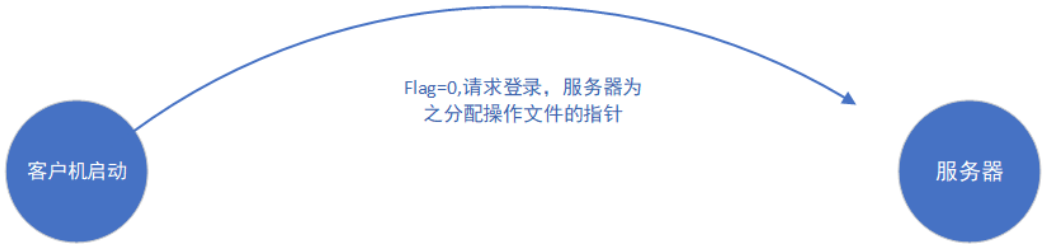
### 4.9.2 客户机

如果是客户机就需要和服务器建立第一次连接，让服务器分配指针。

```
    else
    {
        if (m_ipdress.IsBlank())
        {
            MessageBox(_T("请先填入服务器IP地址"));
            return;
        }
        DWORD dwIP;
        ((CIPAddressCtrl*)GetDlgItem(IDC_IPADDRESS1))-
>GetAddress(dwIP);//获取填入的IP地址

        SOCKADDR_IN addrTo;
        addrTo.sin_family = AF_INET;
        addrTo.sin_port = htons(6000);
        addrTo.sin_addr.S_un.S_addr = htonl(dwIP);

        Packet FileData;
        FileData.flag = 0;
        strcpy((char*)FileData.Data, "请求登录服务器");

        sendto(m_socket, (char*)&FileData, sizeof(FileData), 0,
(SOCKADDR*)&addrTo, sizeof(SOCKADDR));//发送给服务器
        char tempBuf[MTU * 2];
```

```
        sprintf(tempBuf, "向IP为: %s发送消息:\r\n%s",
inet_ntoa(addrTo.sin_addr), FileData.Data);
        CString strt(tempBuf);


        m_log.AddString(strt);
        int nCount = m_log.GetCount();
        if (nCount > 0)
            m_log.SetCurSel(nCount - 1);
        //设置按键
        GetDlgItem(IDC_RADIO_SER)->EnableWindow(FALSE);
        GetDlgItem(IDC_RADIO_CLI)->EnableWindow(FALSE);
        GetDlgItem(IDC_BUTTON_START)->EnableWindow(FALSE);

        GetDlgItem(IDC_BUTTON_SEL_FILE)->EnableWindow(TRUE);
        GetDlgItem(IDC_BUTTON_VIEWFILE)->EnableWindow(TRUE);
        GetDlgItem(IDC_BUTTON_DOWLFILE)->EnableWindow(TRUE);
        GetDlgItem(IDC_BUTTON_SEND)->EnableWindow(TRUE);
        UpdateData(FALSE);

    }
```

然后服务器在接收消息线程接收消息分配指针

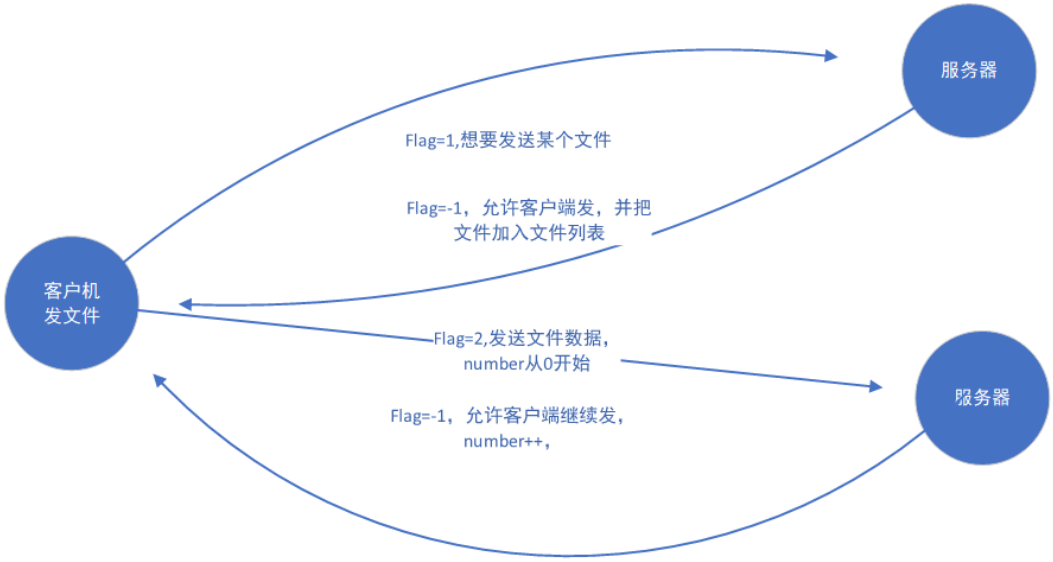



### 4.9.3 服务器接收消息

```
//接收消息线程
DWORD WINAPI CFTPDlg::RecvProc(LPVOID lpParameter)
{
    CFTPDlg* p_dlg = (CFTPDlg*)lpParameter;

    SOCKET sock = m_socket;

    SOCKADDR_IN addrFrom;
    int len = sizeof(SOCKADDR);

    Packet FileData;
    int retval;
    char tempBuf[MTU * 2];

    while (TRUE)//一直循环对话
    {
        retval = recvfrom(sock, (char*)&FileData,
sizeof(FileData), 0, (SOCKADDR*)&addrFrom, &len);
        addrFromC = addrFrom;
```

```
        if (SOCKET_ERROR == retval)
            break;
        if ( FileData.flag == 2||FileData.flag==-5)
        {
            wsprintfA(tempBuf, "收到IP为：%s的消息。\r\n",
inet_ntoa(addrFrom.sin_addr));
        }
        else
        {
            sprintf(tempBuf, "收到IP为：%s的消息:\r\n%s",
inet_ntoa(addrFrom.sin_addr), FileData.Data);
        }
        CString strt(tempBuf);
        p_dlg->m_log.AddString(strt);

        //服务器收到：0代表从客户端第一次发来消息
        if (FileData.flag == 0)
        {
            //看该客户端的文件指针存不存在
            bool isexist = false;
            for (auto it = map_writfile.begin(); it !=
map_writfile.end(); ++it)
            {
                if (it->first->sin_family == addrFrom.sin_family
&& it->first->sin_port == addrFrom.sin_port && it->first-
>sin_addr.S_un.S_addr == addrFrom.sin_addr.S_un.S_addr)
                {
                    isexist = true;
                }
            }
            if (isexist == false)//如果不存在就创建
            {
                SOCKADDR_IN* addrclient = new SOCKADDR_IN;
                *addrclient = addrFrom;
                map_writfile[addrclient] = new CFile;
                map_readfile[addrclient] = new CFile;
            }
        }
```

## 4.10 客户端上传文件




### 4.10.1 客户端发起请求

首先客户端点击选择文件按钮

```
void CFTPDlg::OnClickedButtonSelFile()
{
    // TODO: 在此添加控件通知处理程序代码
    char szchar[Max];
    memset(szchar, 0, Max);

    Packet FileData;
    CFileDialog fileDlg(TRUE);//打开文件窗口
    fileDlg.m_ofn.lpstrFilter = _T("所有文件\0*.*\0World文档
(*.doc;*.docx)\0*doc;*.docx\0Excel表格(*.xls)\0*.xls\0文本文档
(*.txt;*.pdf)\0*.txt;*.pdf\0图片(*.jpg)\0*.jpg\0压缩包
(*.zip;*.rar)\0*.zip;*.rar\0\0");
    fileDlg.m_ofn.lpstrTitle = _T("打开文件");//定义打开对话框的标题
    if (fileDlg.DoModal() == IDOK)
    {
        file.Open(fileDlg.m_ofn.lpstrFile, CFile::modeRead);
        CString filename = fileDlg.m_ofn.lpstrFileTitle;//获取文件
名
        wsprintfA(FileData.filename, "%ls", filename);
        long g_FileLength = file.GetLength();//获取文件的长度
        //计算包的总数
        FileData.Total = g_FileLength / MTU - 1;
        if (g_FileLength % MTU != 0)
        {
            FileData.Total = g_FileLength / MTU;
        }
        FileData.Number = 0;
        FileData.flag = 1;
        FileData.length = g_FileLength;
        FileData.Totallength = g_FileLength;

        DWORD dwIP;
```

```
        ((CIPAddressCtrl*)GetDlgItem(IDC_IPADDRESS1))-
>GetAddress(dwIP);

        SOCKADDR_IN addrTo;
        addrTo.sin_family = AF_INET;
        addrTo.sin_port = htons(6000);
        addrTo.sin_addr.S_un.S_addr = htonl(dwIP);

        char tempBuf[MTU * 2];
        strcpy((char*)FileData.Data, "请求向服务器上传文件");
        sprintf(tempBuf, "向IP为：%s发送消息:\r\n%s",
inet_ntoa(addrTo.sin_addr), FileData.Data);

        sendto(m_socket, (char*)&FileData, sizeof(FileData), 0,
(SOCKADDR*)&addrTo, sizeof(SOCKADDR));
        CString strt(tempBuf);

        m_log.AddString(strt);
        int nCount = m_log.GetCount();
        if (nCount > 0)
            m_log.SetCurSel(nCount - 1);
    }
}
```

### 4.10.2 服务器相应并返回消息

然后再接收线程里对数据进行接收和显示，显示部分在flag=0那一步已经做了，所以不需要重复

```
//服务器收到：客户端选择了“选择按钮”即将要给服务器发送消息，并把flag置为-1
        if (FileData.flag == 1)
        {
            //服务端接收到了文件名
            CFile* filew = NULL;
            for (auto it = map_writfile.begin(); it !=
map_writfile.end(); ++it)
            {
                if (it->first->sin_family == addrFrom.sin_family
&& it->first->sin_port == addrFrom.sin_port && it->first-
>sin_addr.S_un.S_addr == addrFrom.sin_addr.S_un.S_addr)
                {
                    filew = it->second;
                }
            }

            //获取程序当前运行的路径，并把接收的文件存放在当前路径下
            FileData.initdata();
            CString mfilepath = GetExePath();
            mfilepath += FileData.filename;

            //把接收的文件名存入服务器列表中
```

```
            map_filelist[mapindex++] = FileData.filename;

            //服务器创建新的文件
            filew->Open(mfilepath, CFile::modeCreate |
CFile::modeWrite);
            Packet FileDatatemp = FileData;

            //标志设置为-1允许接收数据
            FileDatatemp.flag = -1;

            CString temptex(FileData.filename) ;
            CString temptex1(FileData.filename);
            temptex.Format(_T("%s %s"), _T("服务器允许接收文件:"),
temptex1);

            //strcpy((char*)FileDatatemp.Data,
(char*)temptex.GetBuffer());
            char tempbuf[MTU * 2] = {0};
            wsprintfA(tempbuf, "%ls", temptex);
            strcpy((char*)FileData.Data, tempbuf);
            sendto(sock, (char*)&FileDatatemp,
sizeof(FileDatatemp), 0, (SOCKADDR*)&addrFrom, sizeof(SOCKADDR));
            p_dlg->m_filelist.InsertItem(p_dlg-
>m_filelist.GetItemCount() , _T(""));//插入行
            p_dlg->m_filelist.SetItemText(p_dlg-
>m_filelist.GetItemCount()-1 , 0, temptex1);//设置数据
            sprintf(tempBuf, "向IP为: %s 发送消息:\r\n %s",
inet_ntoa(addrFrom.sin_addr), tempbuf);
            CString strt(tempBuf);
            p_dlg->m_log.AddString(strt);
            int nCount = p_dlg->m_log.GetCount();
            if (nCount > 0)
                p_dlg->m_log.SetCurSel(nCount - 1);
            continue;
        }
```

### 4.10.3 客户端向服务器传输数据

```
//客户端收到: 服务端允许上传文件数据断并把flag置为2让服务器写入数据
        if (FileData.flag == -1)
        {
            //最后一个包, 如果数据不满MTU的处理
            FileData.initdata();
            if (FileData.Number == FileData.Total)
            {
                file.Read(FileData.Data, FileData.Totallength -
FileData.Number * MTU);
                FileData.length = FileData.Totallength -
FileData.Number * MTU;
            }
            else
```

```
            {
                file.Read(FileData.Data, MTU);
                FileData.length = MTU;
            }
            //flag=2，让对方写入
            Packet FileDatatemp = FileData;
            FileDatatemp.flag = 2;
            sendto(sock, (char*)&FileDatatemp,
sizeof(FileDatatemp), 0, (SOCKADDR*)&addrFrom, sizeof(SOCKADDR));

            wsprintfA(tempBuf, "向IP为：%s发送消息.\r\n",
inet_ntoa(addrFrom.sin_addr));
            CString strt(tempBuf);
            p_dlg->m_log.AddString(strt);

            //如果包的数量等于总数量则传送完毕
            if (FileData.Number == FileData.Total)
            {
                file.Close();
                FileData.Number = 0;//初始化

                p_dlg->m_log.AddString(_T("传送完成!"));
                int nCount = p_dlg->m_log.GetCount();
                if (nCount > 0)
                    p_dlg->m_log.SetCurSel(nCount - 1);
            }
            continue;
        }
```

### 4.10.4 服务器回应收到数据

```
//服务器收到：客户端发来的消息即服务器需要写入文件，并把flag置为-1让客户端继
续发送
        if (FileData.flag == 2)
        {
            CFile* filew = NULL;
            for (auto it = map_writfile.begin(); it !=
map_writfile.end(); ++it)
            {
                if (it->first->sin_family == addrFrom.sin_family
&& it->first->sin_port == addrFrom.sin_port && it->first-
>sin_addr.S_un.S_addr == addrFrom.sin_addr.S_un.S_addr)
                {
                    filew = it->second;
                }
            }

            //服务端收到客户端上传的文件数据段
            filew->Write(FileData.Data, FileData.length);

            if (FileData.Number == FileData.Total)
```

```
            {
                filew->Close();
                FileData.Number = 0;//初始化

                p_dlg->m_log.AddString(_T("接收完成!"));
                int nCount = p_dlg->m_log.GetCount();
                if (nCount > 0)
                    p_dlg->m_log.SetCurSel(nCount - 1);

                continue;
            }
            FileData.initdata();
            FileData.Number++;//把文件的数量加一为了看有没有接收完毕
            Packet FileDatatemp = FileData;
            FileDatatemp.flag = -1;


            strcpy((char*)FileDatatemp.Data, "服务器允许继续上传文件
段\n");

            sendto(sock, (char*)&FileDatatemp,
sizeof(FileDatatemp), 0, (SOCKADDR*)&addrFrom, sizeof(SOCKADDR));

            sprintf(tempBuf, "向IP为: %s发送消息:\r\n%s",
inet_ntoa(addrFrom.sin_addr), FileDatatemp.Data);
            CString strt(tempBuf);
            p_dlg->m_log.AddString(strt);
            continue;
        }
```

## 4.11 客户端获取文件列表


### 4.11.1 客户端发起请求

```
void CFTPDlg::OnClickedButtonViewfile()
{
    // TODO: 在此添加控件通知处理程序代码
    Packet FileData;
    FileData.flag = 3;
    strcpy((char*)FileData.Data, "请求文件列表");

    DWORD dwIP;
```

```
    ((CIPAddressCtrl*)GetDlgItem(IDC_IPADDRESS1))-
>GetAddress(dwIP);

    SOCKADDR_IN addrTo;
    addrTo.sin_family = AF_INET;
    addrTo.sin_port = htons(6000);
    addrTo.sin_addr.S_un.S_addr = htonl(dwIP);

    sendto(m_socket, (char*)&FileData, sizeof(FileData), 0,
(SOCKADDR*)&addrTo, sizeof(SOCKADDR));
    char tempBuf[MTU * 2];
    sprintf(tempBuf, "向IP为: %s发送消息:\r\n%s",
inet_ntoa(addrTo.sin_addr), FileData.Data);
    CString strt(tempBuf);

    m_log.AddString(strt);
    int nCount = m_log.GetCount();
    if (nCount > 0)
        m_log.SetCurSel(nCount - 1);
}
```

### 4.11.2 服务器回送列表

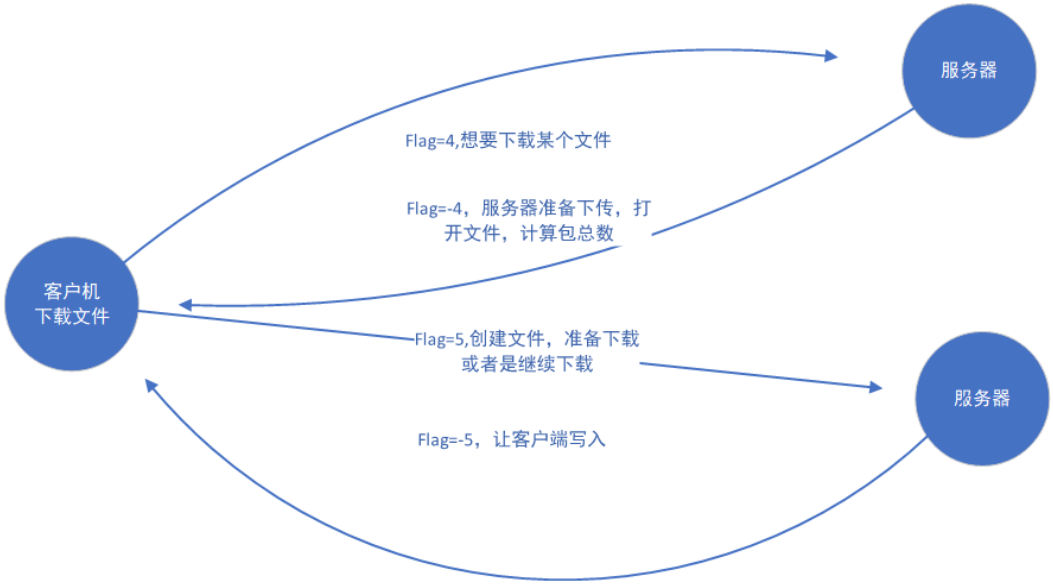
```
//服务端收到文件列表请求并把flag置为-3
        if (FileData.flag ==3)
        {
            FileData.initdata();
            Packet FileDatatemp = FileData;
            FileDatatemp.initfilelist();
            int mindex = 0;
            //把文件名称写入包里的文件列表
            for (auto it = map_filelist.begin(); it !=
map_filelist.end(); ++it)
            {

wsprintfA(FileDatatemp.mfilelist[mindex++].filename, "%ls", it-
>second);

                if (mindex == FILENUM)
                {

                    p_dlg->m_log.AddString(_T("文件超出个数!"));
                    int nCount = p_dlg->m_log.GetCount();
                    if (nCount > 0)
                        p_dlg->m_log.SetCurSel(nCount - 1);
                    break;
                }
            }
            FileDatatemp.flag = -3;
            strcpy((char*)FileDatatemp.Data, "服务器下传文件列表");
```

```
            sendto(sock, (char*)&FileDatatemp,
sizeof(FileDatatemp), 0, (SOCKADDR*)&addrFrom, sizeof(SOCKADDR));

            sprintf(tempBuf, "向IP为：%s发送消息:\r\n%s",
inet_ntoa(addrFrom.sin_addr), FileDatatemp.Data);
            CString strt(tempBuf);

            p_dlg->m_log.AddString(strt);
            int nCount = p_dlg->m_log.GetCount();
            if (nCount > 0)
                p_dlg->m_log.SetCurSel(nCount - 1);
            continue;
        }
```

### 4.11.3 客户端自己处理不发给服务器

```
//客户端收到：文件列表回复
        if (FileData.flag == -3)
        {
            FileData.initdata();
            //先初始化本地列表
            for (int mindex = 0; mindex < MTU; mindex++)
            {
                list_filelist[mindex] = "";
            }

            //把服务器的文件列表复制到本地
            for (int mindex = 0; mindex < FILENUM; mindex++)
            {
                list_filelist[mindex] =
FileData.mfilelist[mindex].filename;
            }
            p_dlg->insertlistctrl();//插入到控件

            continue;
        }
```

## 4.12 客户端请求下载文件




### 4.12.1 客户端发起请求

首先点击列表控件，选择要下载的文件，获取索引

```
void CFTPDlg::OnClickListFile(NMHDR* pNMHDR, LRESULT* pResult)
{
    LPNMITEMACTIVATE pNMItemActivate =
reinterpret_cast<LPNMITEMACTIVATE>(pNMHDR);
    // TODO: 在此添加控件通知处理程序代码
    NM_LISTVIEW* pNMListView = (NM_LISTVIEW*)pNMHDR;
    IndexList = pNMListView->iItem;

    *pResult = 0;
}
```

然后点击下载按钮，把文件名传到服务器

```
void CFTPDlg::OnClickedButtonDowlfile()
{
    // TODO: 在此添加控件通知处理程序代码
    // TODO: 在此添加控件通知处理程序代码
    if (IndexList == -1)
    {
        MessageBox(_T("请选择一个文件"));
        return;
    }

    CString needfilename = m_filelist.GetItemText(IndexList, 0);

    Packet FileData;
    FileData.flag = 4;
    memset(FileData.filename, 0, 100);
    wsprintfA(FileData.filename, "%ls", needfilename);
    needfilename = _T("请求下载文件: ") + needfilename;
```

```
    char tempbuf[MTU * 2] = { 0 };
    wsprintfA(tempbuf, "%ls", needfilename);
    strcpy((char*)FileData.Data, tempbuf);

    DWORD dwIP;
    ((CIPAddressCtrl*)GetDlgItem(IDC_IPADDRESS1))-
>GetAddress(dwIP);

    SOCKADDR_IN addrTo;
    addrTo.sin_family = AF_INET;
    addrTo.sin_port = htons(6000);
    addrTo.sin_addr.S_un.S_addr = htonl(dwIP);

    sendto(m_socket, (char*)&FileData, sizeof(FileData), 0,
(SOCKADDR*)&addrTo, sizeof(SOCKADDR));
    char tempBuf[MTU * 2];
    sprintf(tempBuf, "向IP为: %s发送消息:\r\n%s",
inet_ntoa(addrTo.sin_addr), FileData.Data);
    CString strt(tempBuf);

    m_log.AddString(strt);
    int nCount = m_log.GetCount();
    if (nCount > 0)
        m_log.SetCurSel(nCount - 1);
}
```

### 4.12.2 服务器收到请求

```
if (FileData.flag == 4)
        {
            CFile* filedownload = NULL;
            for (auto it = map_readfile.begin(); it !=
map_readfile.end(); ++it)
            {
                if (it->first->sin_family == addrFrom.sin_family
&& it->first->sin_port == addrFrom.sin_port && it->first-
>sin_addr.S_un.S_addr == addrFrom.sin_addr.S_un.S_addr)
                {
                    filedownload = it->second;
                }
            }

            //服务器收到下载请求,打开待下载文件
            CString mfilepath = GetExePath();
            mfilepath += FileData.filename;
            CString tempfilename(FileData.filename);
            //服务器打开待下载的文件
            filedownload->Open(mfilepath, CFile::modeRead);

            long downloadFileLength = filedownload-
>GetLength();//获取文件的长度
```

```
            FileData.init();
            wsprintfA(FileData.filename, "%ls", tempfilename);
            //计算包的总数
            FileData.Total = downloadFileLength / MTU - 1;
            if (downloadFileLength % MTU != 0)
            {
                FileData.Total = downloadFileLength / MTU;
            }
            FileData.Number = 0;
            FileData.flag = -4;
            FileData.length = downloadFileLength;
            FileData.Totallength = downloadFileLength;

            strcpy((char*)FileData.Data, "服务器准备下传文件");
            sprintf(tempBuf, "向IP为: %s发送消息:\r\n%s",
inet_ntoa(addrFrom.sin_addr), FileData.Data);

            sendto(m_socket, (char*)&FileData, sizeof(FileData),
0, (SOCKADDR*)&addrFrom, sizeof(SOCKADDR));
            CString strt(tempBuf);

            p_dlg->m_log.AddString(strt);
            int nCount = p_dlg->m_log.GetCount();
            if (nCount > 0)
                p_dlg->m_log.SetCurSel(nCount - 1);
            continue;
        }
```

### 4.12.3 客户端收到下载请求的回复

```
if (FileData.flag == -4)
        {
            //客户端收到下载请求的回复
            //获取程序当前运行的路径，并把接收的文件存放在当前路径下
            FileData.initdata();
            CString mfilepath = GetExePath();
            mfilepath += FileData.filename;

            //客户端创建新的文件
            filedownload.Open(mfilepath, CFile::modeCreate |
CFile::modeWrite);
            Packet FileDatatemp = FileData;

            //标志设置为5准备开始接收数据
            FileDatatemp.flag = 5;

            CString temptex (FileDatatemp.filename);
            temptex.Format(_T("%s%s"), _T("客户端准备接收文件完毕:
"), temptex);
```

```
            char tempbuf[MTU * 2] = { 0 };
            wsprintfA(tempbuf, "%ls", temptex);
            strcpy((char*)FileData.Data, tempbuf);
            sendto(sock, (char*)&FileDatatemp,
sizeof(FileDatatemp), 0, (SOCKADDR*)&addrFrom, sizeof(SOCKADDR));

            sprintf(tempBuf, "向IP为：%s发送消息:\r\n%s",
inet_ntoa(addrFrom.sin_addr), tempbuf);
            CString strt(tempBuf);

            p_dlg->m_log.AddString(strt);
            int nCount = p_dlg->m_log.GetCount();
            if (nCount > 0)
                p_dlg->m_log.SetCurSel(nCount - 1);

            continue;
        }
```

4.12.4

## 服务器收到客户端回应开始下传数据

```
if (FileData.flag == 5)
        {
            //服务器开始发送文件段到客户端
            CFile* filedownload = NULL;
            for (auto it = map_readfile.begin(); it !=
map_readfile.end(); ++it)
            {
                if (it->first->sin_family == addrFrom.sin_family
&& it->first->sin_port == addrFrom.sin_port && it->first-
>sin_addr.S_un.S_addr == addrFrom.sin_addr.S_un.S_addr)
                {
                    filedownload = it->second;
                }
            }

            FileData.initdata();
            if (FileData.Number == FileData.Total)
            {
                filedownload->Read(FileData.Data,
FileData.Totallength - FileData.Number * MTU);
                FileData.length = FileData.Totallength -
FileData.Number * MTU;
            }
            else
            {
                filedownload->Read(FileData.Data, MTU);
                FileData.length = MTU;
            }
            //flag=-5，让客户端写入
            Packet FileDatatemp = FileData;
```

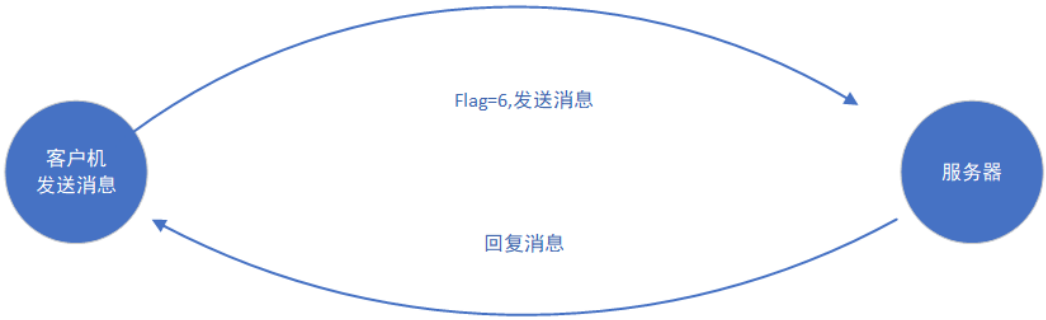
```
            FileDatatemp.flag = -5;
            sendto(sock, (char*)&FileDatatemp,
sizeof(FileDatatemp), 0, (SOCKADDR*)&addrFrom, sizeof(SOCKADDR));

            wsprintfA(tempBuf, "向IP为：%s发送消息。\r\n",
inet_ntoa(addrFrom.sin_addr));
            CString strt(tempBuf);
            p_dlg->m_log.AddString(strt);

            if (FileData.Number == FileData.Total)
            {
                filedownload->Close();
                FileData.Number = 0;//初始化

                p_dlg->m_log.AddString(_T("传送完毕！"));
                int nCount = p_dlg->m_log.GetCount();
                if (nCount > 0)
                    p_dlg->m_log.SetCurSel(nCount - 1);
            }
            continue;
        }
```

### 4.12.5 客户端收到服务器的数据并回应

```
f (FileData.flag == -5)
        {
            //客户端接收服务器发送的文件段
            filedownload.Write(FileData.Data, FileData.length);
            if (FileData.Number == FileData.Total)
            {
                filedownload.Close();
                FileData.Number = 0;//初始化

                p_dlg->m_log.AddString(_T("接收完毕！"));
                int nCount = p_dlg->m_log.GetCount();
                if (nCount > 0)
                    p_dlg->m_log.SetCurSel(nCount - 1);
                continue;
            }
            FileData.initdata();
            FileData.Number++;
            Packet FileDatatemp = FileData;
            FileDatatemp.flag = 5;


            strcpy((char*)FileDatatemp.Data, "客户端请求继续下载文件
段");

            sendto(sock, (char*)&FileDatatemp,
sizeof(FileDatatemp), 0, (SOCKADDR*)&addrFrom, sizeof(SOCKADDR));
```

```
            sprintf(tempBuf, "向IP为：%s发送消息:\r\n%s",
inet_ntoa(addrFrom.sin_addr), FileDatatemp.Data);
            CString strt(tempBuf);
            p_dlg->m_log.AddString(strt);
            continue;
        }
```

## 4.13 发送一般消息




就在点击发送按钮时候控制就好,服务器可以立即回应。

```
void CFTPDlg::OnClickedButtonSend()
{
    // TODO: 在此添加控件通知处理程序代码
    DWORD dwIP;
    ((CIPAddressCtrl*)GetDlgItem(IDC_IPADDRESS1))-
>GetAddress(dwIP);

    SOCKADDR_IN addrTo;
    addrTo.sin_family = AF_INET;
    addrTo.sin_port = htons(6000);
    addrTo.sin_addr.S_un.S_addr = htonl(dwIP);

    CString strSend;
    GetDlgItemText(IDC_EDIT_MESSAGE, strSend);

    Packet FileData;
    FileData.flag = 6;


    char tempbuf[MTU] = { 0 };
    wsprintfA(tempbuf, "%ls", strSend);
    strcpy((char*)FileData.Data, tempbuf);
    if (m_radio == 1)
    {
        sendto(m_socket, (char*)&FileData, sizeof(FileData), 0,
(SOCKADDR*)&addrTo, sizeof(SOCKADDR));
    }
    else
    {
```

```
        sendto(m_socket, (char*)&FileData, sizeof(FileData), 0,
(SOCKADDR*)&addrFromC, sizeof(SOCKADDR));
    }
    char tempBuf2[MTU * 2];
    sprintf(tempBuf2, "向IP为: %s发送消息:\r\n%s",
inet_ntoa(addrTo.sin_addr), tempbuf);
    CString mm(tempBuf2);

    m_log.AddString(mm);
    int nCount = m_log.GetCount();
    if (nCount > 0)
        m_log.SetCurSel(nCount - 1);
    SetDlgItemText(IDC_EDIT_MESSAGE, _T(""));


}
```

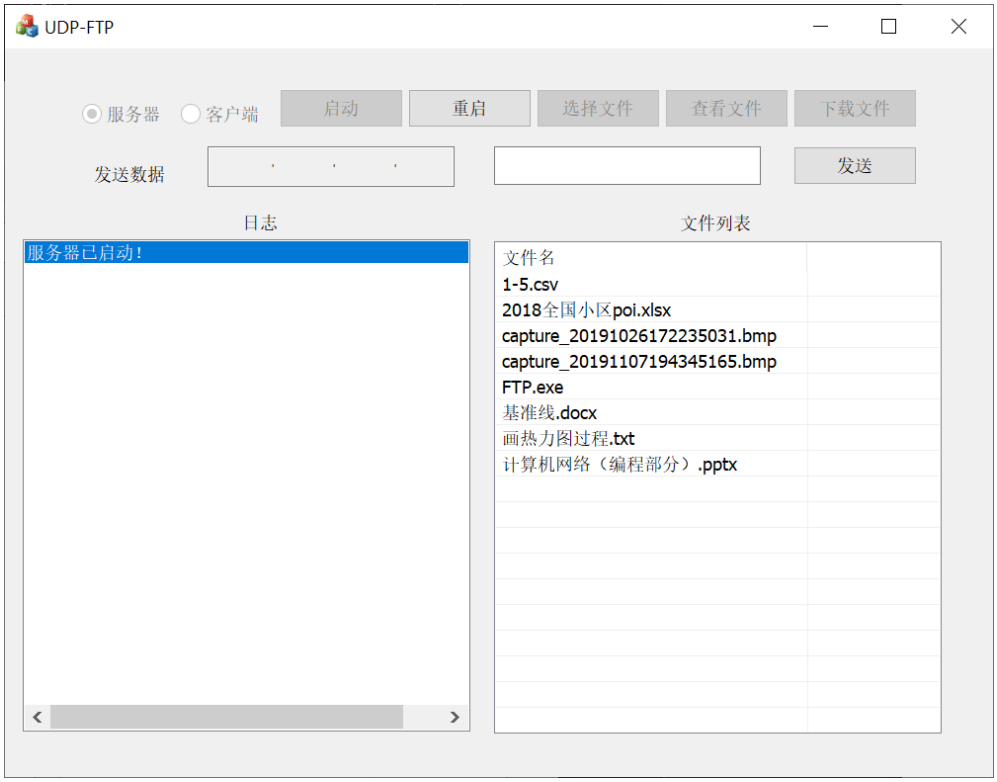
## 4.14 重启

也是关掉目前运行的APP然后重新打开

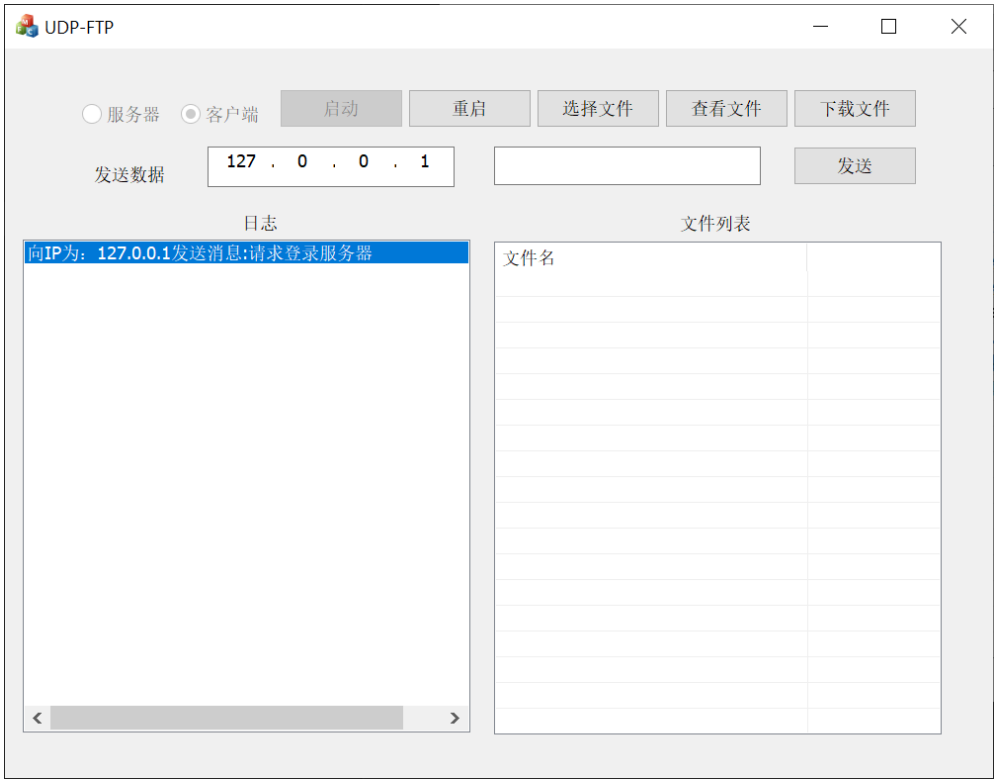
```
void CFTPDlg::OnClickedButtonRestart()
{
    // TODO: 在此添加控件通知处理程序代码
    ::PostMessage(AfxGetMainWnd()->m_hWnd, WM_SYSCOMMAND,
SC_CLOSE, NULL);
    //获取exe程序当前路径
    extern CFTPApp theApp;
    TCHAR szAppName[MAX_PATH];
    ::GetModuleFileName(theApp.m_hInstance, szAppName, MAX_PATH);
    CString strAppFullName;
    strAppFullName.Format(_T("%s"), szAppName);
    //重启程序
    STARTUPINFO StartInfo;
    PROCESS_INFORMATION procStruct;
    memset(&StartInfo, 0, sizeof(STARTUPINFO));
    StartInfo.cb = sizeof(STARTUPINFO);
    ::CreateProcess(
        (LPCTSTR)strAppFullName,
        NULL,
        NULL,
        NULL,
        FALSE,
        NORMAL_PRIORITY_CLASS,
        NULL,
        NULL,
        &StartInfo,
        &procStruct);
}
```

# 5 效果演示

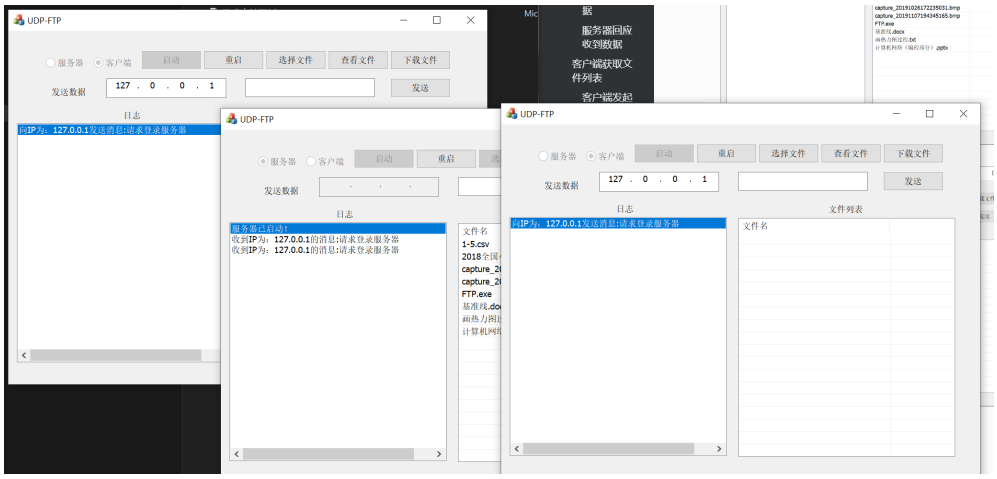
1. 服务器启动



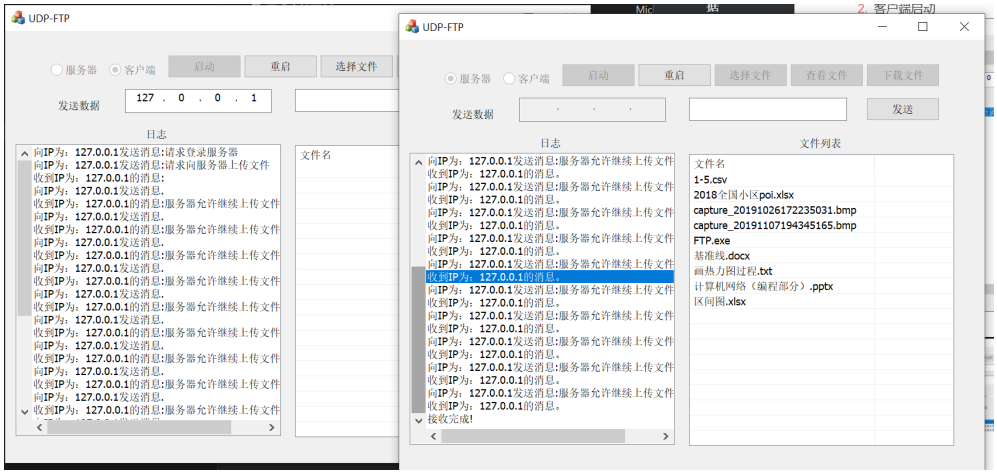

2. 客户端启动



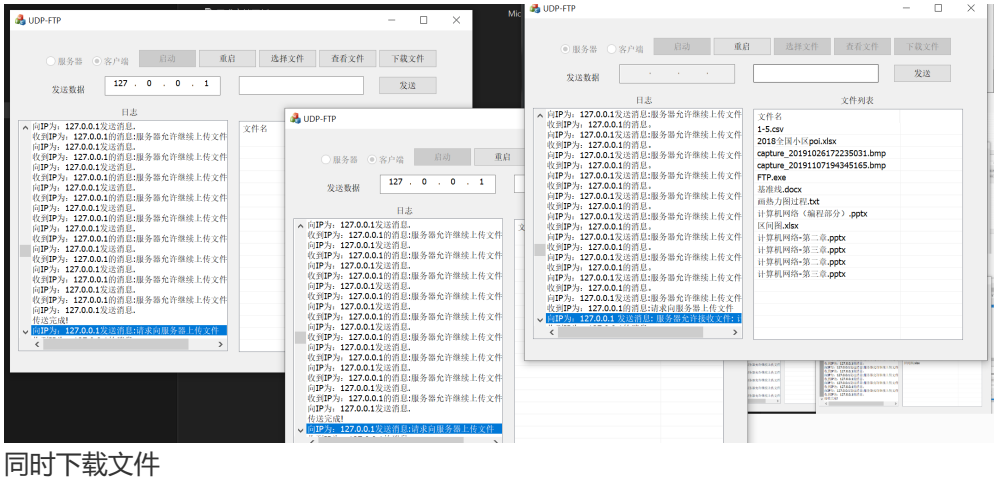
3. 多客户端



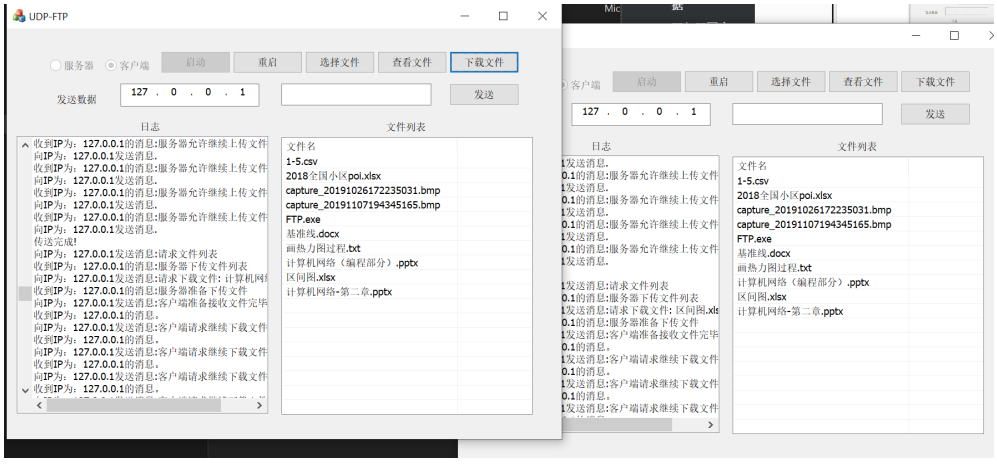
4. 上传文件，任意格式均支持



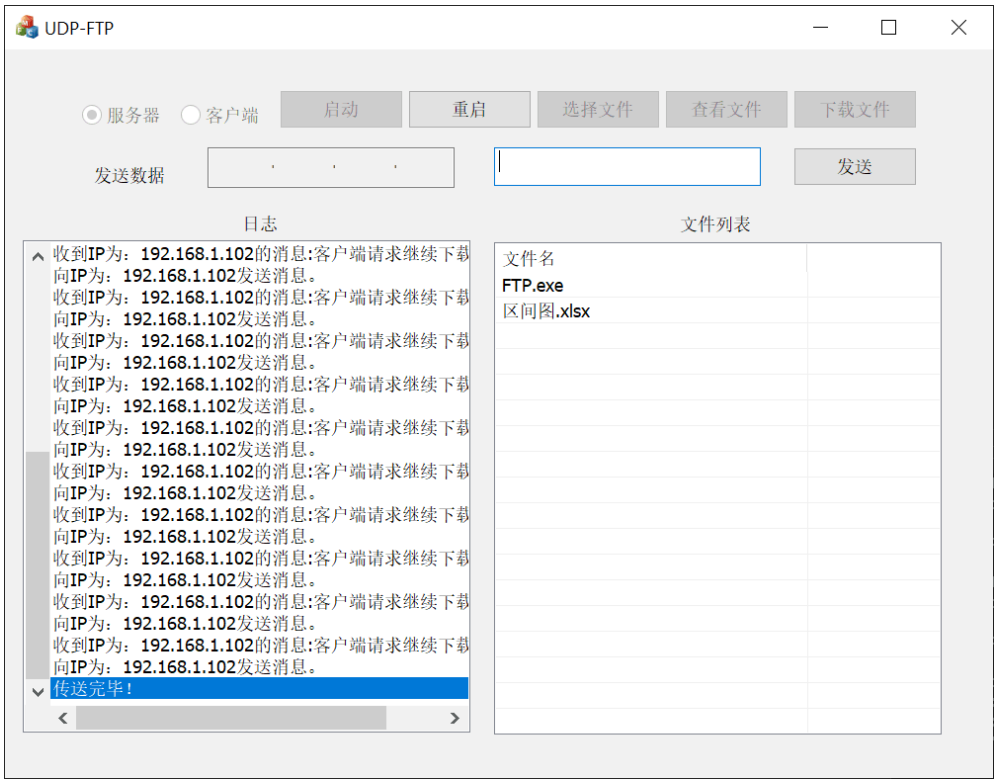
5. 支持同时上传



6. 同时下载文件



7. 支持局域网内传输




# 6 总结与反思

1. 实验中最大的难题是解决乱码问题，在这个地方花了很大的功夫，一步一步调试。
2. 对数据的处理需要分片处理。
3. 不能同时对同一个文件进行读和写，多客户端不能同时上传或者下载同一个文件
4. 服务器与客户端相互收到一个确认一个保证了可靠性