原 基于Socket的文件传输（使用CSocket类）

2013年09月25日 18:38:33　阅读数：15098

本软件使用 MFC 采用面向对象的方法实现了基于 Socket 的文件传输。这是原来研究生课程的结课作业，实现了Socket的发送和接收，以及读取ini配置文件等操作。使用了CSocket类

以下是当时 结课 作业 的正文：

一． 软件特点如下：

1. 采用了多线程的方法，文件传输时使用 AfxBeginThread() 开启新线程

```
void CClientsockDlg::OnBnClickedSend()

{

pThreadSend = AfxBeginThread(Thread_Send, this ); /

}
```

文件的发送和接收都开起了新线程

```
UINTThread_Send(LPVOID lpParam)

{

代码略 …

}
```

2. 支持从配置文件 configuration.ini 中获取服务器参数

采用 GetPrivateProfileString() 和 GetPrivateProfileInt() 分别获取位于 ServerConfiguration.ini 文件中的 String 类型的 IP 和 int 类型的 port

```
CString IP;

int port;

GetPrivateProfileString

(L "ServerConfiguration" ,L "IP" ,L " 没有读取到数据 !" ,IP.GetBuffer(10),10,L ".\\configuration.ini" );

port=GetPrivateProfileInt(L "ServerConfiguration" ,L "port" ,0,L ".\\configuration.ini" );
```

3. 采用了面向对象的设计方式，功能之间按模块划分

MFC 本身具有良好的面向对象的特性，本程序严格按照 MFC 框架结构编写代码，每个按钮对应一个功能函数，降低了代码之间的耦合性，有利于程序的扩展和复用。

```
void CServersockDlg::OnBnClickedChoose()

void CServersockDlg::OnBnClickedSend()

void CServersockDlg::OnBnClickedRecvdata()

void CServersockDlg::OnBnClickedAbout()

void CServersockDlg::OnBnClickedWriteini()
```

4. 采用了 CSocket 类，代码相对更简单

CSocket 类是 MFC 框架对 socket 编程中的 winsockAPI 的封装，因此通过这个类管理收发数据更加便利。代码也跟那个既简单易懂。

```
// 创建
if(!Clientsock.Socket())
{
CString str;
str.Format(_T("Socket 创建失败 :%d"),GetLastError());
AfxMessageBox(str);
}
// 连接
if(!Clientsock.Connect(IP,port))
{
CString str;
str.Format(_T("Socket 连接失败 :%d"),GetLastError());
AfxMessageBox(str);
```

```
}
else
{
AfxMessageBox(_T("Socket 连接成功 "));
代码略 …
// 发送
while(nSize<FindFileData.nFileSizeLow)
{
szBuff = new char[1024];
memset(szBuff,0x00,1024);
nSend =file.Read(szBuff,1024);
Clientsock.Send(szBuff,nSend);// 发送数据
nSize += nSend;
}
file.Close();
delete szBuff;
Clientsock.Close();
(dlg->GetDlgItem(IDC_SEND))->EnableWindow(TRUE);
AfxMessageBox(_T(" 文件发送成功 "));
dlg->SetDlgItemTextW(IDC_FILEPATHNAME,_T(""));
}
return 0;
```

5. 支持数据在服务器与客户端之间双向传输

本程序不但可以从客户端往服务器端传文件，而且可以从服务器端往客户端传文件。

但是互传文件的方式并不是完全相同的。

服务器端不管是接收文件还是发送文件始终是对绑定的端口进行监听。

```
// 绑定

if (!Serversock.Bind(port))

{

CString str;

str.Format(_T( "Socket 绑定失败 : %d" ),GetLastError());

AfxMessageBox(str);

}

// 监听

if (!Serversock.Listen(10))

{

CString str;

str.Format(_T( "Socket 监听失败 :%d" ),GetLastError());

AfxMessageBox(str);

}
```

客户端不管是接收文件还是发送文件始终是进行连接。

```
if (!Clientsock.Connect(IP,port))

{

CString str;

str.Format(_T( "Socket 连接失 ì 败 :%d" ),GetLastError());

AfxMessageBox(str);

}

else

{
```

```
略 ...
```

6. 完全图形化操作界面

二． 软件使用说明

客户端主界面如图所示：

u 单击“选择文件”弹出文件对话框，选择一个要发送的文件，同时保存文件的路径。

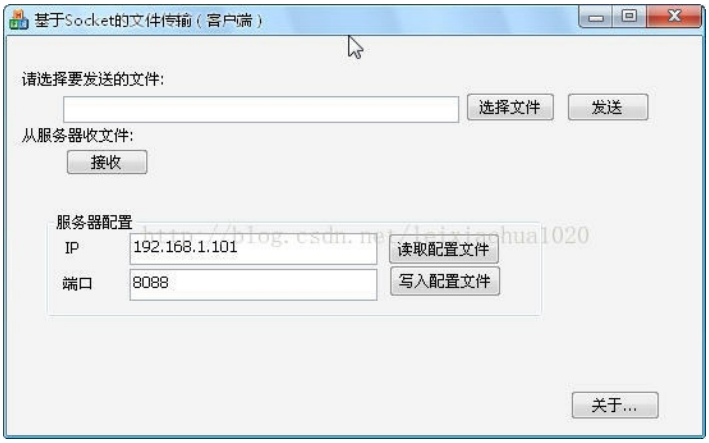
u 单击“发送”则会读取 ServerConfiguration.ini 文件中的配置信息（ IP 和 port），并根据此信息建立 Socket 连接，发送文件。注意：服务器端应该先单击了“接受客户端数据”，否则发送失败。

u 单击“接收”也会读取 ServerConfiguration.ini 文件中的配置信息（ IP 和 port），并根据此信息建立 Socket 连接，接收文件。注意：服务器端应该先选择了向客户端发送的文件，并单击了“发送”，否则接受失败。

u 单击“读取配置文件”，会从 ServerConfiguration.ini 文件中读取配置信息，并以可编辑的文本形式显示出来，修改完后，单击“写入配置文件”，会将修改后的信息保存到配置文件中。

u 单击“关于”可以了解到软件相关信息。

u 代码注释里有更详细的说明



服务器端主界面如图所示

u 单击“接受客户端数据”，开始监听客户端的链接。

u 单击“选择文件”弹出文件对话框，选择一个要发送的文件，同时保存文件的路径。

u 单击“发送”则会读取 ServerConfiguration.ini 文件中的配置信息（ port ），并监听对应端口，准备发送文件。注意：客户端选择“接收”以后才能发送成功。

u 单击“读取配置文件”，会从 ServerConfiguration.ini 文件中读取配置信息，并以可编辑的文本形式显示出来，修改完后，单击“写入配置文件”，会将修改后的信息保存到配置文件中。但是服务器的 IP 是不可以修改的，它是在程序开始运行时从服务器所在机器的网卡上获取的。

u 单击“关于”可以了解到软件相关信息。

u 代码注释里有更详细的说明

代码下载地址： http://download.csdn.net/detail/leixiaohua1020/6320417

在此附上客户端使用CSocket发起连接的代码

```cpp
//----------------------------发送文件的线程----------------------------
UINT Thread_Send(LPVOID lpParam)
{
    CClientsockDlg *dlg=(CClientsockDlg *)lpParam;
    (dlg->GetDlgItem(IDC_SEND))->EnableWindow(FALSE);

    CSocket Clientsock; //definition socket.
    if(!AfxSocketInit())
    {
        AfxMessageBox(IDP_SOCKETS_INIT_FAILED);
    }

    CString IP;
    int port;
    GetPrivateProfileString(L"ServerConfiguration",L"IP",L"没有读取到数据！",IP.GetBuffer(100),100,L".\\configuration.ini");
    port=GetPrivateProfileInt(L"ServerConfiguration",L"port",0,L".\\configuration.ini");
    //创建
    if(!Clientsock.Socket())
    {
        CString str;
        str.Format(_T("Socket创建失败: %d"),GetLastError());
        AfxMessageBox(str);
    }
    //连接
//  if(!Clientsock.Connect(_T("127.0.0.1"),8088))
    if(!Clientsock.Connect(IP,port))
    {
        CString str;
        str.Format(_T("Socket连接失败: %d"),GetLastError());
        AfxMessageBox(str);
    }
    else
    {
        AfxMessageBox(_T("Socket连接成功"));
        WIN32_FIND_DATA FindFileData;
        CString strPathName; //定义用来保存发送文件路径的CString对象
        dlg->GetDlgItemTextW(IDC_FILEPATHNAME,strPathName);
        FindClose(FindFirstFile(strPathName,&FindFileData));
        Clientsock.Send(&FindFileData,sizeof(WIN32_FIND_DATA));

        CFile file;
        if(!file.Open(strPathName,CFile::modeRead|CFile::typeBinary))
        {
            AfxMessageBox(_T("文件不存在"));
            return 1;
        }

        UINT nSize = 0;
        UINT nSend = 0;

        char *szBuff=NULL;
    //发送
        while(nSize<FindFileData.nFileSizeLow)
        {
            szBuff = new char[1024];
            memset(szBuff,0x00,1024);
            nSend = file.Read(szBuff,1024);
            Clientsock.Send(szBuff,nSend);//发送数据
            nSize += nSend;
        }
        file.Close();
        delete szBuff;
        Clientsock.Close();
        (dlg->GetDlgItem(IDC_SEND))->EnableWindow(TRUE);
        AfxMessageBox(_T("文件发送成功"));
        dlg->SetDlgItemTextW(IDC_FILEPATHNAME,_T(""));
    }
    return 0;
}
```

以及服务器端使用CSocket监听的代码：

```cpp
//----------------------------监听文件的线程----------------------------
UINT Thread_Func(LPVOID lpParam)  //接收文件的线程函数
{
    CServersockDlg *dlg = (CServersockDlg *)lpParam; //获取对话框指针
    (dlg->GetDlgItem(IDC_RECVDATA))->EnableWindow(FALSE);

    if(!AfxSocketInit())
    {
        AfxMessageBox(IDP_SOCKETS_INIT_FAILED);
    }

    CString IP;
    int port;
    GetPrivateProfileString(L"ServerConfiguration",L"IP",L"没有读取到数据！",IP.GetBuffer(100),100,L".\\configuration.ini");
```

```
    port=GetPrivateProfileInt(L"ServerConfiguration",L"port",0,L".\\configuration.ini");

    char errBuf[100]={0};// 临时缓存

    SYSTEMTIME t; //系统时间结构

    CFile logErrorfile;
    if(!logErrorfile.Open(_T("logErrorfile.txt"),CFile::modeCreate|CFile::modeReadWrite))
    {
        return 1;
    }

    CSocket Serversock;
    CSocket Clientsock;
    //创建
    if(!Serversock.Socket())
    {
        CString str;
        str.Format(_T("Socket创建失败: %d"),GetLastError());
        AfxMessageBox(str);
    }

    BOOL bOptVal = TRUE;
    int bOptLen = sizeof(BOOL);
    Serversock.SetSockOpt(SO_REUSEADDR,(void *)&bOptVal,bOptLen,SOL_SOCKET);

    //绑定
    if(!Serversock.Bind(port))
    {
        CString str;
        str.Format(_T("Socket绑定失败: %d"),GetLastError());
        AfxMessageBox(str);
    }
    //监听
    if(!Serversock.Listen(10))
    {
        CString str;
        str.Format(_T("Socket监听失败: %d"),GetLastError());
        AfxMessageBox(str);
    }

    GetLocalTime(&t);
    sprintf_s(errBuf,"服务器已经启动...正在等待接收文件...\r\n时间：%d年%d月%d日 %2d:%2d:%2d \r\n",t.wYear,t.wMonth,t.wDay,
        t.wHour,t.wMinute,t.wSecond);
    int len = strlen(errBuf);
    logErrorfile.Write(errBuf,len);
    AfxMessageBox(_T("启动成功等待接收文件"));
    while(1)
    {
        //AfxMessageBox(_T("服务器启动成功..."));
        if(!Serversock.Accept(Clientsock)) //等待接收
        {
            continue;
        }
        else
        {
            WIN32_FIND_DATA FileInfo;
            Clientsock.Receive(&FileInfo,sizeof(WIN32_FIND_DATA));

            CFile file;
            file.Open(FileInfo.cFileName,CFile::modeCreate|CFile::modeWrite);
            //AfxMessageBox(FileInfo.cFileName);
            int length = sizeof(FileInfo.cFileName);
            logErrorfile.Write(FileInfo.cFileName,length);
            //Receive文件的数据

            UINT nSize = 0;
            UINT nData = 0;

            char *szBuff=NULL;

            while(nSize<FileInfo.nFileSizeLow)
            {
                szBuff = new char[1024];
                memset(szBuff,0x00,1024);
                nData=Clientsock.Receive(szBuff,1024);
                file.Write(szBuff,nData);
                nSize+=nData;
            }

            delete szBuff;
            Serversock.Close();
            Clientsock.Close();
            file.Close();
            (dlg->GetDlgItem(IDC_RECVDATA))->EnableWindow(TRUE);
            sprintf_s(errBuf,"文件接收成功...\r\n时间：%d年%d月%d日 %2d:%2d:%2d \r\n",t.wYear,t.wMonth,t.wDay,
        t.wHour,t.wMinute,t.wSecond);
            int len = strlen(errBuf);
            logErrorfile.Write(errBuf,len);
            //AfxMessageBox(_T("文件接收成功..."));
            break;
```

```
106.            }
107.        }
108.        return 0;
109.    }
```


文章标签： socket 面向对象 文件传输 对话框 CSocket

个人分类： 纯编程 计算机网络