**飞鸽局域网程序设计与实现**

2017年3月4日

目录

[一、概述 3](#_Toc476513339)

[二、开发目标 3](#_Toc476513340)

[三、参考文档 3](#_Toc476513341)

[四、开发工具和平台 3](#_Toc476513342)

[五、测试工具 4](#_Toc476513343)

[六、程序结构 4](#_Toc476513344)

[1.类定义 4](#_Toc476513345)

[2.主要类接口 4](#_Toc476513346)

[2.1.IMComm类 4](#_Toc476513347)

[2.2.Im\_send\_file类 5](#_Toc476513348)

[2.3.InstantMessage类 5](#_Toc476513349)

[3. 线程定义 6](#_Toc476513350)

[七、功能模块详细设计 7](#_Toc476513351)

[1.显示局域网内主机信息(IP和主机名称) 7](#_Toc476513352)

[2.发送与接收文本信息 8](#_Toc476513353)

[3.发送与接收文件 8](#_Toc476513354)

[八、体会 9](#_Toc476513355)

[附：实现结果 10](#_Toc476513356)

1. 概述

在本次飞鸽互联网程序实现上，全程使用win32API，这要比使用MFC难度增大了很多，特别是在界面处理上的listview的checkbox和标题栏的事件绑定监听，花费了大量的时间。这次的程序监听了3个端口。主要是对socket和多线程的应用。对于局域网内其他运行本程序的主机信息的查找是本应用程序的重点，对于后面的文本数据和文件的传输都大同小异了。我们对于IP查找和文本数据传输、文件数据传输分别定义了通讯协议。除了处理文件传输时，文件名的获取和缓存区的设定有点麻烦外，其他的还是比较简单的。

对于我们这个飞鸽局域网应用程序，我们完成了局域网主机信息查找显示、文本信息单发/群发、文本单发/群发等功能的设计和实现。同时在界面实现部分，我们都listview进行了深入的开发，并且对标题栏绑定了全选和取消功能，方便用户选择使用。同时使用多线程，避免了界面的出现“死掉”的情况。

总之，我们的界面使用了多线程控制，socket使用TCP发送文本数据，UDP寻找主机信息和文本传输，全程使用Microsoft Visual Studio开发。收获巨大。

1. 开发目标

设计并实现了win32 API开发标准的，拥有显示局域网内所有运行本程序的主机信息、单发/群发文本信息、单发/群发文件等功能的win32窗体应用程序软件。

1. 参考文档

《Windows程序设计》第二版，王艳平，10.1~10.3

《VC++深入详解》，孙鑫，第14章

《Visual C++6.0程序设计从入门到精通》11.1~11.4

[http://www.lellansin.com/tutorials/windows-%E7%BC%96%E7%A8%8B%E4%B9%8B%E8%B7%AF#windows-gdi](http://www.lellansin.com/tutorials/windows-编程之路" \l "windows-gdi)

http://blog.csdn.net/column/details/win32-dev.html?&page=2

1. 开发工具和平台

开发工具：Microsoft Visual Studio 2010

平台：windows 10

1. 测试工具

Microsoft Visual Studio 2010

1. 程序结构
2. 类定义

|  |  |
| --- | --- |
| Class | Specification |
| IMComm | 工具类，主要用于查找到局域网内所有运行本程序的主机信息，以及文本信息的接受和发送。 |
| Im\_send\_file | 工具类，主要用于文件的接收和发送。 |
| InstantMessage | 主函数入口，只要用于界面的绘制、消息命令的接收和响应处理，以及界面内容的填充。 |
| Resource, InstantMessage.rc | 资源文件类，主要用于定义资源文件。 |

1. 主要类接口
   1. IMComm类

|  |  |
| --- | --- |
| Classmember | Function |
| IMComm(); | 构造函数，初始化参数 |
| ~IMComm(); | 析构函数，释放参数 |
| void getOtherAddr(vector<nameIP> &otherAddrs); | 用于通过接收端口为9999的UPD广播消息，来查找局域网内其他的主机信息，并存储在 |
| void sendOtherAddr(vector<nameIP> otherAddrs); | 用于给局域网内其他主机发送端口为9999的UPD广播消息，以便其他主机能发现本机的信息 |
| void sendMessageTo(nameIP, std::string); | 用于给IP为nameIP的主机发送string消息，端口为10000 |
| void m\_send(std::string); | 用于控制需要给哪些发送主机发送string消息 |
| void setChecked(int index, bool state) | 用于设置otherAddrs中的第index +1 个ip选中状态 |
| DWORD CALLBACK MainListner(LPVOID lParam) | 用于监听其他主机发送过来的文本消息，端口为10000 |

* 1. Im\_send\_file类

|  |  |
| --- | --- |
| Classmember | Function |
| void start\_listen\_file(); | 用于创建监听其他主机发来的文件的线程 |
| void send\_file\_to(char\* szFileName, SOCKADDR\_IN dest); | 用于创建给IP为dest的主机发送名称为szFileName的文本的线程 |
| DWORD CALLBACK FileListenerThread(LPVOID lpVoid); | 用于监听其他主机发来的文件，端口12345 |
| DWORD CALLBACK FileRcvThread(LPVOID lpVoid); | 用于接收lpvoid发来的文件，端口12345 |
| DWORD CALLBACK SendFileThread(LPVOID lpVoid) | 用于给lpvoid发送文件，端口12345 |

* 1. InstantMessage类

|  |  |
| --- | --- |
| Classmember | Function |
| int APIENTRY wWinMain(\_In\_ HINSTANCE hInstance,  \_In\_opt\_ HINSTANCE hPrevInstance,  \_In\_ LPWSTR lpCmdLine,  \_In\_ int nCmdShow) | 主函数入口 |
| ATOM MyRegisterClass(HINSTANCE hInstance); | 注册所创建的窗口类 |
| BOOL InitInstance(HINSTANCE, int); | 初始化界面以及参数 |
| LRESULT CALLBACK WndProc(HWND, UINT, WPARAM, LPARAM); | 主函数回执函数，用于对消息的处理 |
| INT\_PTR CALLBACK About(HWND, UINT, WPARAM, LPARAM); | 副窗口，介绍版本信息 |
| void listView(); | 用于显示list view窗口内容，即主机信息列表 |
| void checkBox(); | 用于记录选择主机情况 |
| void ThreadSend(); | 用于发送UDP广播消息的线程 |
| void ThreadRec(); | 用于接收UDP广播消息的线程 |
| void charTowchar(const char \*chr, wchar\_t \*wchar, int size); | 用于将char\*转化为wchar\* |
| inline char\* UnicodeToAnsi(const wchar\_t\* szStr) | 用于将wchar\*转化为char\* |

1. 线程定义

|  |  |
| --- | --- |
| Thread | Function |
| DWORD CALLBACK MainListner(LPVOID lParam) | 用于监听其他主机发送过来的文本消息，端口为10000 |
| DWORD CALLBACK FileListenerThread(LPVOID lpVoid); | 用于监听其他主机发来的文件，端口12345 |
| DWORD CALLBACK FileRcvThread(LPVOID lpVoid); | 用于接收lpvoid发来的文件，端口12345 |
| DWORD CALLBACK SendFileThread(LPVOID lpVoid) | 用于给lpvoid发送文件，端口12345 |
| void ThreadSend(); | 用于发送UDP广播消息的线程 |
| void ThreadRec(); | 用于接收UDP广播消息的线程 |

1. 功能模块详细设计
2. 显示局域网内主机信息(IP和主机名称)

void ThreadSend(); 用于发送UDP广播消息的线程

void ThreadRec(); 用于接收UDP广播消息的线程

void getOtherAddr(vector<nameIP> &otherAddrs); 用于通过接收端口为9999的UPD广播消息，来查找局域网内其他的主机信息，并存储在

void sendOtherAddr(vector<nameIP> otherAddrs); 用于给局域网内其他主机发送端口为9999的UPD广播消息，以便其他主机能发现本机的信息

void listView(); 用于显示list view窗口内容，即主机信息列表

void checkBox(); 用于记录选择主机情况

1. 发送与接收文本信息

void sendMessageTo(nameIP, std::string); 用于给IP为nameIP的主机发送string消息，端口为10000

void m\_send(std::string); 用于控制需要给哪些发送主机发送string消息

void setChecked(int index, bool state) 用于设置otherAddrs中的第index +1 个ip选中状态

DWORD CALLBACK MainListner(LPVOID lParam) 用于监听其他主机发送过来的文本消息，端口为10000

1. 发送与接收文件

void send\_file\_to(char\* szFileName, SOCKADDR\_IN dest); 用于创建给IP为dest的主机发送名称为szFileName的文本的线程

DWORD CALLBACK SendFileThread(LPVOID lpVoid) 用于给lpvoid发送文件，端口12345

void start\_listen\_file(); 用于创建监听其他主机发来的文件的线程

DWORD CALLBACK FileListenerThread(LPVOID lpVoid); 用于监听其他主机发来的文件，端口12345

DWORD CALLBACK FileRcvThread(LPVOID lpVoid); 用于接收lpvoid发来的文件，端口12345

1. 体会

在本次大作业中，使用了win32窗体开发API以及C++ Socket网络编程的相关知识，同时对C++的基础知识和windows窗体开发有了很好的巩固和了解，并且在本次试验中还使用了大量的C++线程，包括无参线程和有参线程。在老师检查实验过程中，老师还提出了很多很好的建议，比如将一些常用的基本数据处理方法封装，积累自己的开发库。最后还通过文档的编写，对软件开发流程进行了回顾，对软件开发的过程的理解进一步加深。虽然我们因为中途有个比赛需要提交耽误了三天，导致没有按时检查作业，但是我们人加班加点的完成了，并且收获巨大。

附：实现结果

