**局域网内即时通讯软件**

**BS00580_**

**班级：计科13-2班**

**学号：1306010201、11、26**

**姓名：陈国文、贾彦勇、张弛**

**成绩：**

**电子与信息工程学院**

**计算机科学系**

[1 设计内容 1](#_Toc16252)

[1.1 课题背景 1](#_Toc4293)

[1.2 本课题主要研究背景 1](#_Toc10905)

[1.3 本课题主要研究思路和方法 1](#_Toc12290)

[1.4 研究项目的目的以及意义 2](#_Toc29511)

[1.5 Win Socket 的寻址方式 2](#_Toc2666)

[2设计目标 3](#_Toc20450)

[3需求分析 3](#_Toc5574)

[3.1 系统设计目标和可行性分析 3](#_Toc7266)

[3.1.1系统设计目标 3](#_Toc28196)

[3.1.2可行性分析 3](#_Toc15565)

[3.2 系统总体需求分析 4](#_Toc26356)

[3.3 组织结构的分析 4](#_Toc16460)

[3.4开发运行环境 4](#_Toc7754)

[3.5 系统开发语言 5](#_Toc12780)

[3.5.1 理论知识 5](#_Toc27429)

[3.6 C++ 中的MFC和Socket 6](#_Toc11337)

[4设计思想 8](#_Toc1762)

[4.1聊天模块的整体构架 9](#_Toc12217)

[4.2 文件传输模块的主要流程图 9](#_Toc4929)

[图4 业务流程图 10](#_Toc30387)

[5具体实现 10](#_Toc27052)

[5.1聊天的主登录界面 10](#_Toc8836)

[5.1.1 单个用户之间的文字聊天 10](#_Toc17824)

[5.3文件传输的实现 12](#_Toc10897)

[6主要数据附录 14](#_Toc11107)

[7参考文献 16](#_Toc2736)

# 1 设计内容

1.1 课题背景

当时间走向 21 世纪，当今时代也由工业经济渐渐成为了知识经济、信息经济的时代。信息的需求量的不停增加，促使了计算机通信的日新月异！虽然网络的使用频率与使用的范围都越来越广泛，但是人们使用的网络的用途主要的还是只有两个：

(1)．相互之间的沟通，

(2)．资源共享。 局域网作为网络的一个重要组成部分，它不但用户与外界连接的枢纽，同时也是近距离用户之间沟通与资源共享的一个重要途径。在这种情况下，如何使局 域网内的用户便捷的勾通，以及快速的资源共享，也就成为了人们对网络研究的 重点之一。

本系统仿飞鸽传输，是一个局域网内的通迅工具，主要通过 Winsockt 完成。 它据有的功能有：

(1)局域网内用户间简单的文字聊天功能；

(2)局域网内用户间相互的文件传输。

## 1.2 本课题主要研究背景

随着计算机和通讯技术的发展，计算机的网络化已经渗透到各个应用领域， 尤其是以 TCP/IP 协议之间的网络互连发展很快，计算机应用系统的开发由传统的单机方式转向到了网络环境的开发。大部分编程语言都提供了基于 Windows Sockets 网络编程接口的 Win socket 控件，它使开发 TCP/IP 通讯变得简单方便。 网络之间进行数据通讯时，使用地址来标识网络中的计算机，有了地址以后， 数据才会准确地发送到地址所指向的计算机。TCP/IP 协议使用 IP 地址为网络中计算机的标识，而且每台计算机的 IP 地址是唯一的，一般情况下，每台计算机 运行的程序不止一个，因此要在应用程序中建立连接的话，还需要一个地址标识。 在 TCP/IP 协议书中使用端口(PORT)作为计算机上运行的应用程序的标识， 因此 TCP/IP 协议中一个有效的网络地址包括 IP 地址和端口地址两个部分。TCP/IP 协议提供了两种计算机之间的通讯方式：TCP 方式和 UDP 方式。

TCP 是一种面向连接的服务，在文件传输时会先行建立连接，通过链接提供双向、有序且无重复的数据流服务、以及流量控制、差错检测和纠错等服务，从而确保文件传输的正确性；UDP 则是一种无连接的服务，它在传输文件时不会确保传输端的文件能够正确无误的传送到接收端，所以当使用 UDP 时，通常要自己作接收确认的工作。

## 1.3 本课题主要研究思路和方法

本系统的开发采用基于 win socket 的广播发送模式，进行文字聊天和文件传输的设计与开发。因此，本设计首先分析该系统的相关功能，结合本设计的相关要求写出需求分析；其次，综合运用以前所学的相关知识，选择所熟悉的开发工具进行本设计的开发；在设计中以需求分析为基础，写出系统开发计划、实现流程及相关问题的实现方法；最后，在系统开发完毕后，进行调试和运行， 做好调试和运行的相关记录。

## 1.4 研究项目的目的以及意义

随着互联网技术的迅猛发展，网络给人们带来了很多便利，比如人们借助于网络进行相互交流变得更加方便。聊天工具作为互联网中运用最为广泛的通信工

具之一，如今都已被广大网民接受。很多人都使用过飞鸽传书软件，飞鸽传书发展到现在已经具有了很多功能：有文字聊天、文件、文件夹传输的基本功能，也有密封、加锁等较为复杂的功能。本毕业设计的目的主要是为了检查使学生综合运用以前所学知识（包括以前所学的一些关于网络、编程、信息安全等知识）的能力，模拟局域网飞鸽传书软件，开发一个天系统软件，本毕业设计主要实现局域网内文字聊天及文件传输等功能。

## 1.5 Win Socket 的寻址方式

　　因为 Winsock 要兼容多个协议，所以必须使用通用的寻址方式。TCP/IP 使用地址和端口号来指定一个地址，但是其它协议也许采用不同的形式。如果 Winsock 强迫使用特定的寻址方式，添加其他协议就不大可能了。在 Winsock 中应用程序员通过 SOCKADDR\_IN 结构来指定 IP 地址和端口号。定义如下：

Struct sockaddr\_in{

　　　　　　　　short sin\_family;

　　　　　　　　u\_short sin\_port;

　　　　　　　　struct in\_addr sin\_addr;

　　　　　　　　har sin\_zero[8];

　　　　};

应用程序可以使用 inet\_addr 函数将一个小数点分隔的十进制 IP 地址字符 串转化成由 32 位二进制数表示的 IP 地址。inet\_ntoa 函数将一个网络字节顺序的 32 位 IP 地址转化字符串。注意：inet\_addr 返回的 32 位二进制数是用网络 顺序存储的

# 2设计目标

根据对飞鸽传输系统的调研，我认为本设计需要满足以下几个系统设计目

标：

(1)实用性原则：真正为局域网的实际工作服务，按照需求的轻重缓急，合理设计本系统。

(2)可靠性原则：必须为用户提供安全的服务，尤其是要保证传输的稳定性。

(3)友好性原则：使用本系统的用户相当一部分对计算机知识了解很少，所以系统操作上要求简单方便，便于用户掌握。

(4)可扩展性原则：采用开放的标准和接口，便于系统向更大的规模和功能扩展。

# 3需求分析

## 3.1 系统设计目标和可行性分析

### 3.1.1系统设计目标

根据对飞鸽传输系统的调研，我认为本设计需要满足以下几个系统设计目标：

(1)实用性原则：真正为局域网的实际工作服务，按照需求的轻重缓急，合理设计本系统。

(2)可靠性原则：必须为用户提供安全的服务，尤其是要保证传输的稳定性。

(3)友好性原则：使用本系统的用户相当一部分对计算机知识了解很少，所以系统操作上要求简单方便，便于用户掌握。

(4)可扩展性原则：采用开放的标准和接口，便于系统向更大的规模和功能扩展。

### 3.1.2可行性分析

技术可行性：本系统以Visual Studio 2015为主要编程环境，再结合网络通信基 本 原理和Socket编程及语音处理API技术，技术成熟，可行。

经济可行性：这个系统开发开发成较低，经济效益会远大于开发成本，可行。

操作可行性：系统硬件为PC机，系统win 10，而且编程环境Visual Studio 2015为专业程序员使用工具，比较易于使用，可行。

## 3.2 系统总体需求分析

根据用户需求，本系统分为两个子系统:

1. ：局域网内简单文字聊天子系统， 它是在局域网内选择相应的用户，以便对之发送文字信息;
2. 文件传输系统，它是在先把需要传输的文件转换成相就的数据流传到了目的主机再由之还原。

本系统的文字聊天系统实现的功能类似于ＱＱ用户之间的聊天。在设计系统时，应根据简洁、流畅的特点安排布局，在编制程序时应充分考虑这一点传输的稳定、快捷，真正做到"简洁、高效、流畅、安全"的使用环境。用户使用的过程为，先是用户登开启本软件，软件自动检索局域网的其它用户，并以之成为用户列表，让用户选择。当用户发现所要连接的用户之后，双击该用户，则出现发送窗口。用户用之来发送消息或者文件。接收方可以选择接收或者拒绝相应的传输文件。

## 3.3 组织结构的分析

以下为组织结构图：

文件传输

在线用户

到达用户主机

数据传输

选择交互

服务器主机

文字聊天

图1 组织结构

## 3.4开发运行环境

开发此系统所需的基本软、硬件环境为：

(1).win10或win7;

(2).Pentium100 及以上档次的 PC 及其兼容机;

(3).128M 以上内存措施;

(4).1G 以上可用硬盘空间;

(5).VGA 显示器;

以上的配置均要求为在同一局域网内的两台或两台以上。

## 3.5 系统开发语言

本系统采用 VC/VC++语言开发。VC/VC++是迅速发展和推广的一种计算机语言，近年来，已经相当的完善。它主要特点有两个方面：一是全面兼容 C，二是支持面向对向的方法。.

### 3.5.1 理论知识

3.5.1.1 C/S 软件架构技术

C/S（Client/Server）结构，即大家熟知的客户机/服务器结构。它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到Client 端和 Server 端来实现，降低了系统的通讯开销。目前大多数应用软件系统都是 Client/Server 形式的两层结构，由于现在的软件应用系统正在向分布式的 Web 应用发展，Web 和 Client/Server 应用都可以进行同样的业务处理，应用不同的模块共享逻辑组件；因此，内部的和外部的用户都可以访问新的和现有的应用系统，通过现有应用系统中的逻辑可以扩展出新的应用系统。这也就是目前应用系统的发展方向。

传统的 C／S 体系结构虽然采用的是开放模式，但这只是系统开发一级的开放性，在特定的应用中无论是 Client 端还是 Server 端都还需要特定的软件支持。由于没能提供用户真正期望的开放环境，C/S 结构的软件需要针对不同的操作系 统系统开发不同版本的软件，加之产品的更新换代十分快，已经很难适应百台电脑以上局域网用户同时使用。

3.5.1.2 用户数据报协议(UDP)

UDP（User Datagram Protocol）：用户数据包协议，它和 TCP 一样位于传输层，和 IP 协议配合使用，在传输数据时省去包头，但它不能提供数据包的重传，所以适合传输较短的文件。

由于大多数网络应用程序都在同一台机器上运行，计算机上必须能够确保目的地机器上的软件程序能从源地址机器处获得数据包，以及源计算机能收到正确的回复。这是通过使用 UDP 的"端口号"完成的。源端口号标识了请求域名服务的本地机的应用程序，同时需要将所有由目的站生成的响应包都指定到源主机的这个端口上。与 TCP 不同，UDP 并不提供对 IP 协议的可靠机制、流控制以及错误恢复功能等。由于 UDP 比较简单，UDP 头包含很少的字节，比 TCP 负载消耗少，所以 UDP 适用于不需要 TCP 可靠机制的情形，比如，当高层协议或应用程序提供错误和流控制功能的时候。UDP 是传输层协议，服务于很多知名应用层协议，包括网络文件系统（NFS）、简单网络管理协议（SNMP）、域名系统（DNS）以及简单文件传输系统（TFTP）。

3.5.1.3 广播

广播（broadcasting）是多点投递的最普遍的形式，它向每一个目的站投递一个分组的拷贝。它可以通过多个单次分组的投递完成，也可以通过单独的连接传递分组的拷贝，直到每个接收方均收到一个拷贝为止。在多数网络中，用户是通过把分组分送给一个特殊保留的地址即广播地址（broadcast address）来进行广播投递，它的主要缺点是会耗费大量的主机资源和网络资源。

## 3.6 C++ 中的MFC和Socket

（1） Socket及Window Socket API简介

套接字（Socket）是一种双向的通信接口，可以通过这个端口与任何一个具有Socket端口的计算机通信，套接字是网络通信的基础。Socket在Windows以句柄的形式被创建。使用Socket进行网络通信必须包含下面的几种信息：双方认可的协议，本地主机的IP地址，本地进程的协议端口，对方主机的IP地址，对方进程的协议端口。

Socket可分为： 1] 数据报套接字（Datagram Sockets）—— 对于在TCP/IP上实现的WinSock，数据报套接字使用用户数据报协议（UDP）。数据报套接字提供了一种不可靠的、非连接的数据包通信方式。 2] 流式套接字（Stream Sockets）—— 流式套接字使用传输控制协议（TCP）。流式套接字可以将数据按顺序无重复地发送到目的地，它提供的是一种可靠的、面向连接的数据传输方式。不管是对单个的数据报，还是对数据包，流式套接字都提供了一种流式数据传输。

（1.1）Socket

套接字（Socket）是一种双向的通信接口，可以通过这个端口与任何一个具有Socket端口的计算机通信，套接字是网络通信的基础。Socket在Windows以句柄的形式被创建。使用Socket进行网络通信必须包含下面的几种信息：双方认可的协议，本地主机的IP地址，本地进程的协议端口，对方主机的IP地址，对方进程的协议端口。

Socket可分为： 1] 数据报套接字（Datagram Sockets）—— 对于在TCP/IP上实现的WinSock，数据报套接字使用用户数据报协议（UDP）。数据报套接字提供了一种不可靠的、非连接的数据包通信方式。 2] 流式套接字（Stream Sockets）—— 流式套接字使用传输控制协议（TCP）。流式套接字可以将数据按顺序无重复地发送到目的地，它提供的是一种可靠的、面向连接的数据传输方式。不管是对单个的数据报，还是对数据包，流式套接字都提供了一种流式数据传输。

MFC Socket程序编程的步骤：  
1：加载套接字库，创建套接字(AsfSocket()/socket())；  
2：绑定套接字到一个IP地址和一个端口上(bind())；  
3：将套接字设置为监听模式等待连接请求(listen())；  
4：请求到来后，接受连接请求，返回一个新的对应于此次连接的套接字(accept())；  
5：用返回的套接字和客户端进行通信(send()/recv())；  
6：返回，等待另一连接请求；  
7：关闭套接字，关闭加载的套接字库(closesocket())。

（1.2）socket函数  
socket的创建函数，其定义为：SOCKET PASCAL FAR socket (int af, int type, int protocol)

第一个参数为int af，代表网络地址族，目前只有一种取值是有效的，即AF\_INET，代表internet地址族；

第二个参数为int type，代表网络协议类型，SOCK\_DGRAM代表UDP协议，SOCK\_STREAM代表TCP协议；

第三个参数为int protocol，指定网络地址族的特殊协议，目前无用，赋值0即可。  
返回值为SOCKET，若返回INVALID\_SOCKET则失败。

（1.3）sockaddr\_in、in\_addr类型，inet\_addr、inet\_ntoa函数

sockaddr\_in定义了socket发送和接收数据包的地址

sockaddr\_in的含义比in\_addr的含义要广泛，其各个字段的含义和取值如下

第一个字段short sin\_family，代表网络地址族，如前所述，只能取值AF\_INET；  
第二个字段u\_short sin\_port，代表IP地址端口，由程序员指定；  
第三个字段struct in\_addr sin\_addr，代表IP地址；  
第四个字段char    sin\_zero[8]，很搞笑，是为了保证sockaddr\_in与SOCKADDR类型的长度相等而填充进来的字段。

（1.4）常用的WinSock函数

VC++对网络编程的支持有socket支持，Winlnet支持，MAPI和ISAPI支持等。其中Windows Sockets API是TCP/IP网络环境里，也是Internet上进行开发最为通用的API。最早美国加州大学Berkeley分校在UNIX下为TCP/IP协议开发了一个API，这个API就是著名的Berkeley Socket接口（套接字）。在桌面操作系统进入Windows时代后，仍然继承了Socket方法。从表一可以看出，主要的WinSock API函数有哪些。

[表1] WinSock API函数

|  |  |
| --- | --- |
| 函数 | 功能 |
| WSAStartup() | 连结应用程序与 Windows Sockets DLL 的第一个函数 |
| WSACleanup() | 结束 Windows Sockets DLL 的使用 |
| socket() | 建立Socket |
| closesocket() | 关闭某一Socket |
| bind() | 将一本地地址与一个SOCKET描述字连接在一起 |
| listen() | 设定 Socket 为监听状态，准备被连接 |
| accept() | 接受某一Socket的连接要求，以完成面向连接的客户端 Socket 的连接请求。 |
| connect() | 要求连接某一Socket到指定的网络上服务端 |
| recv() | 从面向连接的 Socket 接收信息 |
| send() | 使用面向连接的 Socket 发送信息 |
| WSAAsyncSelect() | 要求某一 Socket 有事件 (event) 发生时通知使用者 |

（2） MFC Socket编程技术

MFC是对API的封装，MFC针对WinSock API封装了两个类：CAsyncSocket和Csocket。因此，使用这两个类进行Socket网络程序的编写会比使用API方便一些。

MFC中Socket相关类的继承关系如下图：

CObject

CAsyncSock

CSocket

图2 MFC中Socket相关类的继承关系

CAsyncSocket对WinSock API进行了较低程度的封装，缺省情况下建立的Socket是异步操作方式，即非阻塞的，因此一些阻塞操作，需要自己动手完成。另外，由于它对API的封装程度不高，以前在WinSock API中需要自己完成的工作（比如：字节顺序的转换），仍然需要自己完成。当然有许多程序员喜欢使用该类甚至WinSock API，原因是这些方法具有很大的灵活性，给程序员以较大的发挥空间。为了给程序员提供更方便的接口以自动处理网络通信中的一些任务，MFC在CAsycnSocket类的基础上派生了CSocket类，它提供了比CAsyncSocket更高层的WinSock API接口。CSocket的网络操作是同步方式，即阻塞式的。CSocket建立的Socket不是阻塞的，而是对非阻塞的Socket进行处理后实现的阻塞操作。在阻塞任务执行过程中，Windows线程的消息循环仍然有效，也就是说CSocket类的阻塞操作不影响Windows的消息循环。

# 4设计思想

　 本系统的开发采用基于 win socket 的广播发送模式，进行文字聊天和文件传输的设计与开发。因此，本次设计的思想有以下几个方面：

第一，分析该系统的相关功能，结合本设计的相关要求写出需求分析；需求分析的基本任务是准确的回答“系统必须做什么？”这个问题。这一阶段是对系统认识最为全面、准确、清晰、得体的一个比较关键的时期。而且需求分析的结果是系统开发的基础，关系到工程的成败和软件产品的质量。因此，必须用行之有效的方法对软件需求进行严格的审查验证，为下一步的总体设计打好基础。本次设计着重于文字聊天和文件传输功能的实现。在局域网中在线用户之间的信息交流。对于聊天功能的实现可以使基于点到点的，也就是说，在两个用户之间的文字传输，可以把每一在线用户设成一个类，实际是对这个类的操作，在具体实现中可以在一个对话框中（类似qq聊天的窗口）具有文字信息的接收与传输；也可以是基于多播的形式（类似qq群的功能），在一个群组里的成员都可与接收到某一用户发在群组里的信息。对于文件传输功能的实现，是在与目标用户建立连接后把数据转换成socket流进行传输。

第二，综合运用以前所学的相关知识，选择所熟悉的开发工具进行本设计的开发；在设计中以需求分析为基础，写出系统开发计划、实现流程及相关问题的实现方法；在具体实现中有以下几个重要问题,需要解决：

首先，发现机制的实现。如何使局域网的在线用户刷新在显示本地在线用户列表上。

其次，文本消息的发送与接收。如何利用socket实现文本信息的发送与接收。

然后，文件的发送与接收。

第三，在系统开发完毕后，进行调试和运行，做好调试和运行的相关记录。

## 4.1聊天模块的整体构架

　　本系统文字聊天是采用 C/S 结构，所以每个客户端也是作为服务器端的，

当系统发送信息给局域网的用户，主要由单播模块和广播模块来实现，而其中的

单播模块和用户信息模块是具有依赖性的，在接收信息的时候又有两个模块来完

成，一是信息接收的模块，另一个是信息查看的模块。

由此可以看出本模块主要分为两模块：发送模块和接收模块。发送模块：单 播模块，广播模块，用户信息模块，信息记录模块接收模块：接收信息模块，查看信息模块

综上所述，系统的架构图为：

系

统

主

页

面

信息接收

信息写入

显示用户信息

查看信息

多播发送

单播发送

图3 系统架构

## 4.2 文件传输模块的主要流程图

由于本模块采用 C/S 架构故将系统运行载体分为两个部分：客户端与服务器端。文件传输需要建立有效的连接后，通过 TCP/IP 协议实现点对点的文件传输。通信双方是通过三次握手建立连接，实现文字、文件传输，最后关闭套节字连接。

其传输的整个流程图如下：

断开连接

断开连接请求

接收文件流（Socket）

写入信息流

套字节连接请求

连接相应（3次握手）

服务器端主机

客户端主机

# 图4 业务流程图

# 5具体实现

## 5.1聊天的主登录界面

由于系统的用户设计是针对的用户日常活动，而非专门从事计算机操作的人员，所以本系统力求界面友好、简单，以便给用户提供一个舒适的使用环境。在文字聊天模块的主要目标是简捷，方便。为了方便用户选取单个用户进行信息发送和查看在线用户的信息，所以设计 了一个 在线用户列表 来进行显示如下图，在 在线用户列表 中包括了用户名组、在线用户群组等。

### 5.1.1 单个用户之间的文字聊天

当有用户打开本系统的时候将会向局域网中的所有用户发送一个握手信息来获取网络中的在线用户信息并添加到列表中显示出来，同时也把自己的网络资源信息发送给对方，使自己被添加到其他用户的列表中。

登录界面,如图 所示：

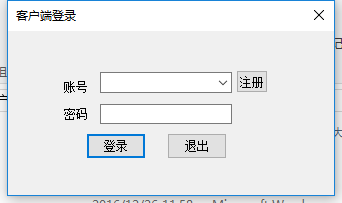


图5 登陆界面

注册界面：

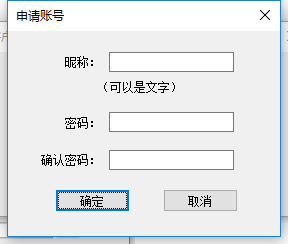


图6 注册界面

在线列表界面：

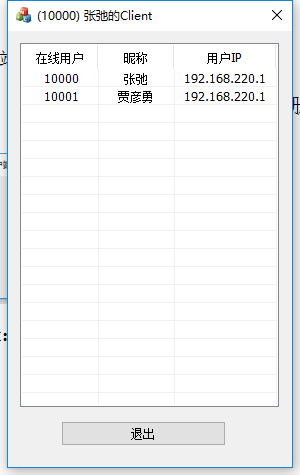


图7 在线列表

与单个用户进行通信：

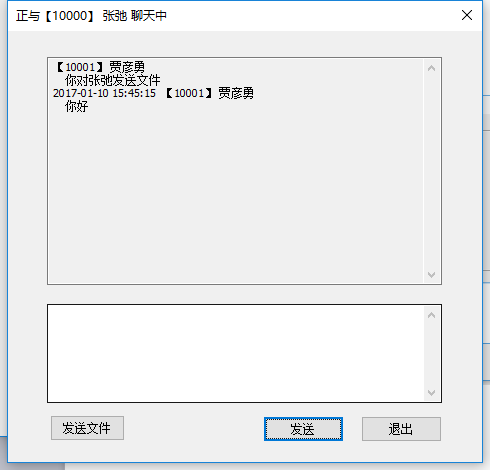


图8 聊天对话框

## 5.3文件传输的实现

与文字聊天一样，系统的文件、文件夹传输模块也尽量地要求简单易行，便于操作。当然，作为文件的传输，速度和安全也就成为了重要的问题，要能达到局域网较为单一的环境要求。

文件传输的过程中流程为首先客端的主机通过 TCP/IP 协议建立链接，然后客户机发送套节字请求，通过 3 次握手连接响应，再转换数据为 socket 流传送最近通过 3 次握手断开连接响应。

选择用户并单击文件按钮把相应所要传送的文件选中，完成选择即可传输。

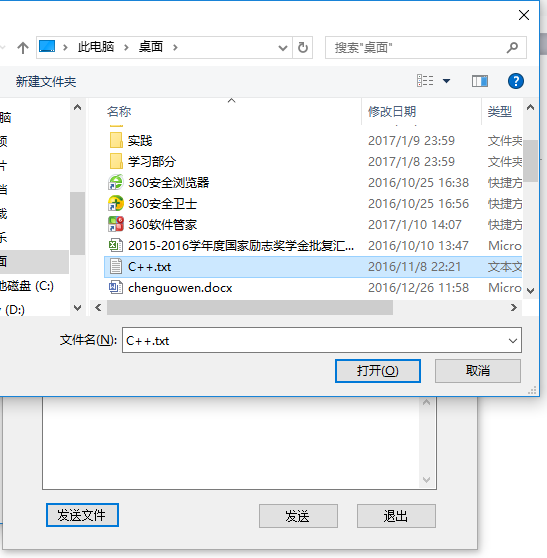
文件发送： 

图9 文件发送

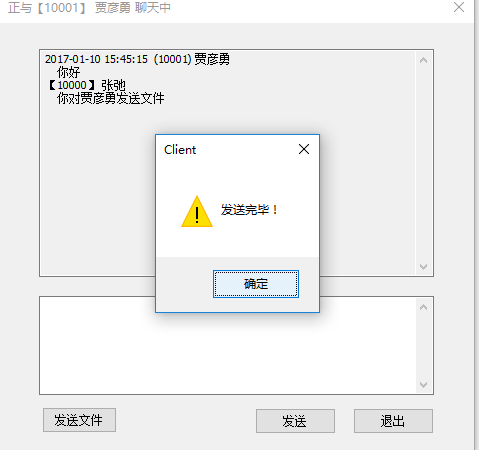


图10 完成提示

接收方弹出一个提醒对话框提示接收完成，并保存在当前运行目录下。

# 6主要数据附录

class UserMsg

{

public:

char ID[6];

char NAME[21];

char PASSWORD[21];

};

class CHead

{

public:

CHead(void);

~CHead(void);

public:

int type;//信息类型

int tag;

char fromID[6];//信息源客户端用户名

char toID[6];//信息目的客户端用户名

class UserMsg usermsg;

char ip[32];

char msg[1024];//信息

};

typedef struct \_SOCKET\_STREAM\_FILE\_INFO {

TCHAR szFileTitle[128]; //文件的标题名

DWORD dwFileAttributes; //文件的属性

FILETIME ftCreationTime; //文件的创建时间

FILETIME ftLastAccessTime; //文件的最后访问时间

FILETIME ftLastWriteTime; //文件的最后修改时间

DWORD nFileSizeHigh; //文件大小的高位双字

DWORD nFileSizeLow; //文件大小的低位双字

DWORD dwReserved0; //保留，为0

DWORD dwReserved1; //保留，为0

} SOCKET\_STREAM\_FILE\_INFO, \*PSOCKET\_STREAM\_FILE\_INFO;

class CClientApp : public CWinApp

{

public:

CClientApp();

CHead head;

CClientDlg \*active;

CLogin \*log;

CUdpSocket \*m\_udp;

CTcpSocket \*m\_tcp;

int search;//服务器搜寻标识

int tag;//账号密码匹配标识

CLogin login;//登录菜单

CString serverip;//记录服务器IP

CString m\_localip;//本地IP

CString m\_userID;//账号

CString m\_userNAME;//用户名

map<CString,CChat \*> m\_userMsg;//记录聊天对话框

map<CString,CString> m\_MsgRecode;//记录聊天信息

map<CString,CString> m\_clientMsg;//记录在线客户端ID == ip

map<CString,CString> m\_client;//ID == NAME

};

extern CClientApp theApp; //全局变量存放信息

# 7参考文献

[1] Andrew S.Tanenbaum．计算机网络[M]．北京：清华大学出版社，1998。

[2] Anonymous．网络最高安全技术指南[M]．北京：机械工业出版社，1998。

[3] David A.Solomon． Windows NT 技术内幕[M]．北京：清华大学出版社，1999。

[4] Douglas E.Comer． TCP/IP 网络互连技术[M]．北京：清华大学出版社，1998。

[5] David J.Kruglinski． Visual C++ 技术内幕[M]．北京：清华大学出版社，1999。

[6]梁普选 .Visual C++ 序设计与实践.北京：清华大学出版社，2005。

[7]百度文库。www.baidu.com

[8]孙鑫MFC教学视频

[9]csdn在线文库 www.csdn.com.cn