

# 开发过程文档

建筑与环境学院-力学-软件工程

张艺馨-2019141470413

李洁- 2019141470450

## 相关知识

### 即时通讯 (Instant Messaging) :

是目前 Internet 上最为流行的通讯方式, 各种各样的即时通讯软件也层出不穷; 服务提供商也提供了越来越丰富的通讯服务功能。 不容置疑, Internet 已经成为真正的信息高速公路。从实际工程应用角度出发, 以计算机网络原理为指导, 结合当前网络中的一些常用技术, 编程实现基于 C/S 架构的网络聊天工具是切实可行的。

目前, 中国市场上的企业级即时通信工具主要包括: 信鸽、视高科技的视高可视协同办公平台、263EM、群英 CC2010、通软联合的 GoCom、腾讯公司的 RTX、IBM 的 Lotus Sametime、点击科技的 GKE、中国互联网办公室的 imo、中国移动的企业飞信、华夏易联的 e-Link、擎旗的 UcStar 等。相对于个人即时通信工具而言, 企业级即时通信工具更加强调安全性、实用性、稳定性和扩展性。

实时通信 (Instant Messaging, 简称 IM) 是一个实时通信系统, 允许两人或多人使用网络实时的传递文字消息、文件、语音与视频交流。

### 套接字 (socket) :

应用层通过传输层进行数据通信时, TCP 和 UDP 会遇到同时为多个应用程序进程提供并发服务的问题。多个 TCP 连接或多个应用程序进程可能需要通过同一个 TCP 协议端口传输数据。为了区别不同的应用程序进程和连接, 许多计算机操作系统为应用程序与 TCP / IP 协议交互提供了称为套接字 (Socket) 的接口。

区分不同应用程序进程间的网络通信和连接, 主要有 3 个参数: 通信的目的 IP 地址、使用的传输层协议 (TCP 或 UDP) 和使用的端口号。Socket 原意是 “插座”。通过将这 3 个参数结合起来, 与一个 “插座” Socket 绑定, 应用层就可以和传输层通过套接字接口, 区分来自不同应用程序进程或网络连接的通信, 实现数据传输的并发服务。

输入/输出多路复用——select()

`select()` 调用用来检测一个或多个套接字的状态。对每一个套接字来说，这个调用可以请求读、写或错误状态方面的信息。请求给定状态的套接字集合由一个 `fd_set` 结构指示。在返回时，此结构被更新，以反映那些满足特定条件的套接字的子集，同时，`select()` 调用返回满足条件的套接字的数目，其调用格式如下：

```
int PASCAL FAR select(int nfds, fd_set FAR * readfds, fd_set FAR *
writefds, fd_set FAR * exceptfds, const struct timeval FAR * timeout);
```

参数 `nfds` 指明被检查的套接字描述符的值域，此变量一般被忽略。

参数 `readfds` 指向要做读检测的套接字描述符集合的指针，调用者希望从中读取数据。参数 `writefds` 指向要做写检测的套接字描述符集合的指针。`exceptfds` 指向要检测是否出错的套接字描述符集合的指针。`timeout` 指向 `select()` 函数等待的最大时间，如果设为 `NULL` 则为阻塞操作。`select()` 返回包含在 `fd_set` 结构中已准备好的套接字描述符的总数目，或者是发生错误则返回 `SOCKET_ERROR`。

## socket 编程原理 - 网络字节序

UNIX 系统的 I/O 命令集，是从 Maltics 和早期系统中的命令演变出来的，其模式为打开—读/写—关闭（open—write—read—close）。在一个用户进程进行 I/O 操作时，它首先调用“打开”获得对指定文件或设备的使用权，并返回称为文件描述符的整型数，以描述用户在打开的文件或设备上进行的 I/O 操作的进程。然后这个用户进程多次调用“读/写”以传输数据。当所有的传输操作完成后，用户进程关闭调用，通知操作系统已经完成了对某对象的使用。

TCP/IP 协议被集成到 UNIX 内核中时，相当于在 UNIX 系统引入了一种新型的 I/O 操作。UNIX 用户进程与网络协议的交互作用比用户进程与传统的 I/O 设备相互作用复杂得多。

在 UNIX 系统中，网络应用编程界面有两类：UNIX BSD 的套接字（socket）和 UNIX System V 的 TLI。由于 Sun 公司采用了支持 TCP/IP 的 UNIX BSD 操作系统，使 TCP/IP 的应用有更大的发展，其网络应用编程界面——套接字（socket）在网络软件中被广泛应用，至今已引进微机操作系统 DOS 和 Windows 系统中，成为开发网络应用软件的强有力工具，

### 使用 MFC CSocket 异步非阻塞转同步阻塞方式传输文件：

阻塞套接字是指执行此套接字的网络调用时，直到成功才返回，否则一直阻塞在此网络调用上，比如调用 `recv()` 函数读取网络缓冲区中的数据，如果没有数据到达，将一直挂在 `recv()` 这个函数调用上，直到读到一些数据，此函数调用才返回；

非阻塞套接字是指执行此套接字的网络调用时，不管是否执行成功，都立即返回。比如调用 `recv()` 函数读取网络缓冲区中数据，不管是否读到数据都立即返回，而不会一

直挂在此函数调用上。在实际 Windows 网络通信软件开发中，异步非阻塞套接字是用的最多的。平常所说的 C/S（客户端/服务器）结构的软件就是异步非阻塞模式的。

MFC 提供了一个异步类 CAsyncSocket，它封装了异步、非阻塞 Socket 的基本功能，用它做常用的网络通信软件很方便。但它屏蔽了 Socket 的异步、非阻塞等概念，开发人员无需了解异步、非阻塞 Socket 的原理和工作机制。

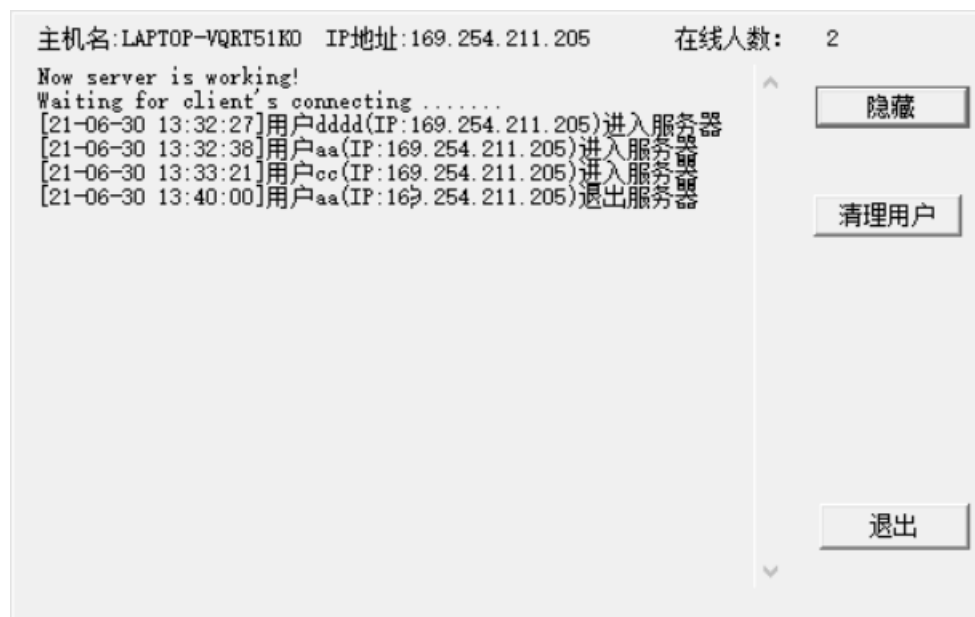
```
DWORD dwTemp = 0;
BOOL bTest = sockSend.AsyncSelect(0); //由于 CSocket 实际是异步，将它
变为同步（阻塞）方式。
sockSend.IOctl(FIONBIO, &dwTemp); //用 IOctl 要将 AsyncSelect 的第一个
参数为 0，参看 MSDN
//读取文件内容
if((int)(uiLength - uiTotal) < iBufSize)
    iSize = uiLength - uiTotal; //当小于缓冲区 iTEST 时的处理
iSize=file.Read(pBuf, iSize); //得到读取的字节数
int iCount=0;
//发送定长文件数据
while(iCount<iSize)
{
    iNumByte = sockSend.Send(pBuf, iSize-iCount); //注意 iNumByte
为实际的发送字节数，不要以 iSize 为准
    if(iNumByte == SOCKET_ERROR)
    {
        AfxMessageBox("发送错误！");
        goto ExitLabel1;
    }
    iCount+=iNumByte;
    if(iCount<iSize)
    {
        file.Seek(iSize-iCount, CFile::current);
    }
}
```

## 基本功能点：

（包含局域网聊天和文件传输功能）

- (1) 可以对指定的人发消息或者对全体发消息
- (2) 聊天可以实现屏蔽和私聊功能，屏蔽可以只接受发给自己的信息，私聊时发送信息只让放送方知道。
- (3) 可以单线程发送消息，接受可以选择接受文件或者拒绝传输。
- (4) 用户列表显示在线的用户
- (5) 界面可以最小化到托盘。在界面最小化时发送信息时，接收方状况类似与 QQ，可以发送屏幕抖动。
- (6) 支持一台机器多个客户端登陆。以用户名识别，用户名不可相同。
- (7) 服务器可以显示在线人数，显示登陆退出信息。服务器可以最小化，可以清理用户

运行图片如下：



ddd

(06-30 13:33:04)aa对大家说:你好

[21-06-30 13:33:21]系统消息:  
cc进入聊天室

(06-30 13:33:35)aa对ddd说:不

(06-30 13:33:59)aa对你说:不

cc对大家发送了窗口抖动

[21-06-30 13:40:00]系统消息:  
用户aa退出聊天室

发送到: 大家

☐ 私聊

☐ 过滤

退出

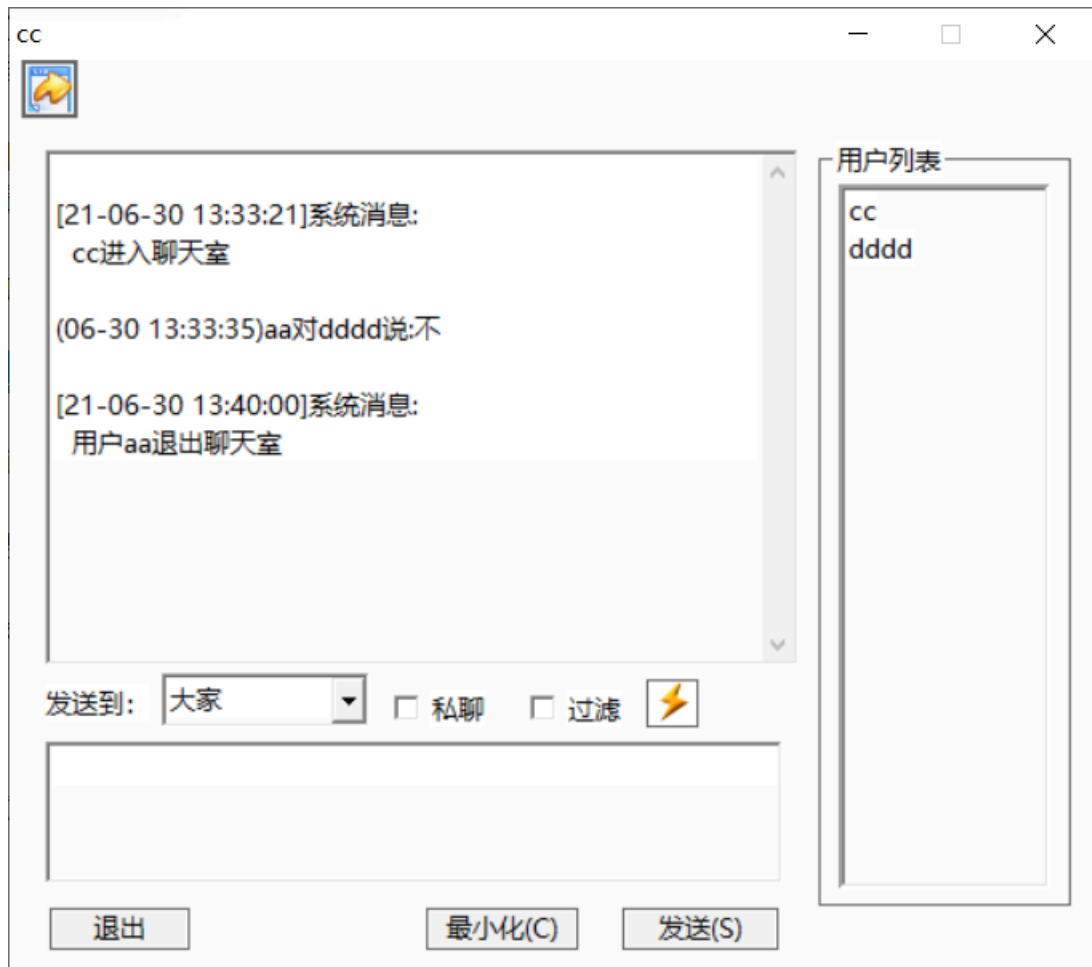
最小化(C)

发送(S)

用户列表

cc

ddd



## 问题:

- 1) 本系统在文件传输时只能单线程传输。因为传输界面不是动态创建，所以多文件传输也难以实现，多线程多文件的传输等并没有实现。

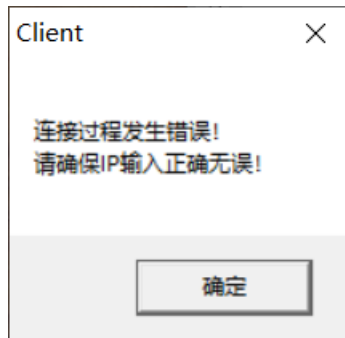
单线程传输:



多线程传输:



- 2) 类似于 QQ 的群，但是仅能支持一台机器多个客户端登陆。否则会出现下列错误提示：



- 3) 编写过程遇到了许多困难，通过组员之间的讨论交流，以及网上的帮助，大部分得到解决。