

实验题目： 实现 Socket 编程

姓名 王雨朦

学号 3012216083

专业 计算机科学与技术

班级 2012 级 3 班

天津大学计算机科学与技术学院

2015 年 6 月 28 日

一、 题目和实现的功能

选题：SOCKET 实现基本的微信功能

功能：相当于网络聊天室的网络通信

二、 环境

环境：win7+vc6.0

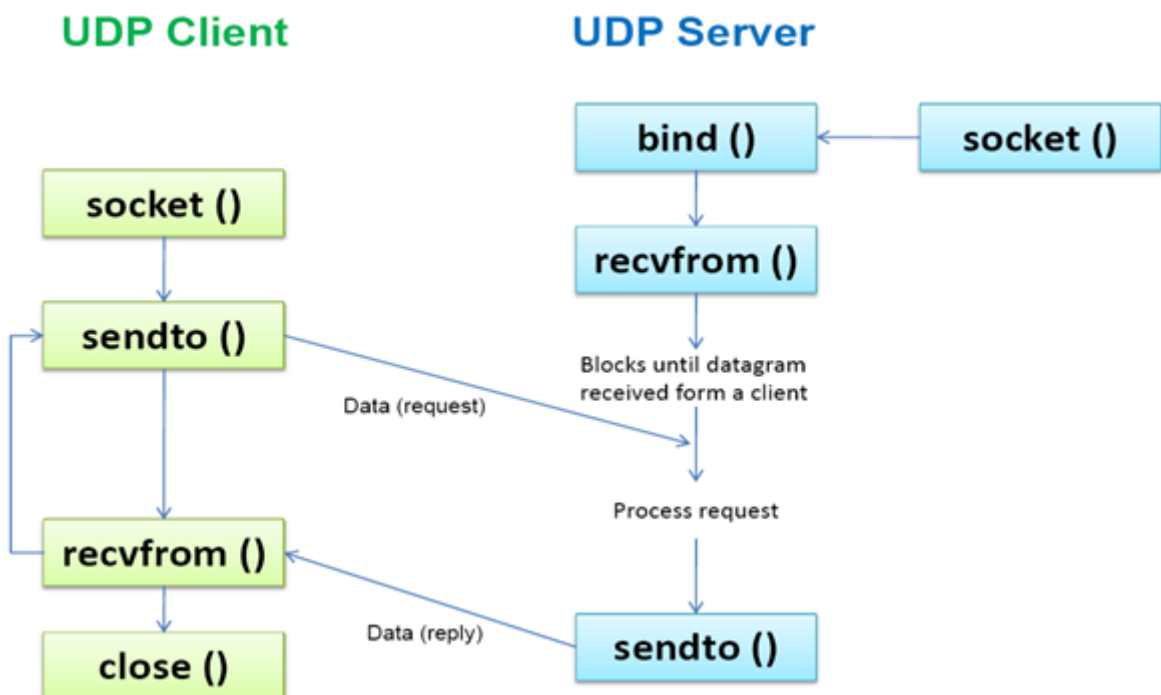
语言：c++

编程模型：winsock+UDP

选择套接字类型：UDP

由于自己笔记本的 VS2013 一系列的软件长期报错无法使用，打算近期重装一次系统，所以只能借用同学 ACM 实验室的一台电脑来实现 SOCKRT，配置有点低。

对于这个 wechat 软件会采用 UDP 的方式实现，是因为 UDP 不需要建立连接而且没有数据确认和数据重传的机制，实时性较高；而在 TCP 协议中是以三步握手方式进行连接，当聊天的时候要先连接再聊天，如果不聊天时就不知道是连接还是断开好。因此在 TCP 中这种连接数据确认和重传机制会影响聊天的效率。



图表 1 基于 UDP 的编程模型

三、 实现原理

基于 UDP 的 socket 编程服务器端为接收端，客户端为发送端。发送数据为 `sendto`, 接收数据为 `recvfrom`.

```
int WSStartup (  
    WORD wVersionRequested,
```

```
LPWSADATA lpWSADATA
);
```

wVersionRequested 用来指定准备加载 Winsock 库的版本，高位字节指定副版本，低位字节指定主版本，可用 MAKEWORD(x, y) (其中 x 是高位字节，y 是低位字节) 方便的获得 wVersionRequested 的正确值。

lpWSADATA 是指向 WSADATA 结构的指针，WSAStartup 把其加载的库版本的有关信息添在这个结构中。

对于每一个 WSAStartup 的成功调用，最后都应该调用 WSACleanup 以便释放资源。

如果调用成功，将返回套接字描述符。如果调用失败，将返回 INVALID_SOCKET，错误信息可通过 WSAGetLastError 返回。

```
SOCKET socket (
    int af,                //指定地址族, 对于 TCP/IP 只能是 AF_INET (PF_INET)
    int type,              //SOCK_STREAM, SOCK_DGRAM
    int protocol           //推荐为零, 可自动选择协议
);
```

在为我们的网络程序指定端口号时，我们要用 1024 以上的端口

htonl 把一个 u_long 类型从主机字节序转换为网络字节序

htons 把一个 u_short 类型从主机字节序转换为网络字节序

✧ 注意：

要加载库函数 ws2_32.lib

启动顺序应遵循先服务器后客户机，否则容易出错。

发送字符时应该多加一个空字符作为结束字符。

```
SOCKET accept (
    SOCKET s,
    struct sockaddr FAR* addr,
    int FAR* addrlen           //必须在传入一个 addrlen 之前为它
                                赋初始值, 否则调用失败
); //int len=sizeof(SOCKADDR);
```

inet_addr: The Windows Sockets inet_addr function converts a string containing an (Ipv4) Internet Protocol dotted address into a proper address for the IN_ADDR structure.

```
unsigned long inet_addr (
    const char FAR * cp
); //用来把 IP 地址转化为 ULONG 类型
```

inet_ntoa: The Windows Sockets inet_ntoa function converts an (Ipv4) Internet network address into a string in Internet standard dotted format.

```
char FAR * inet_ntoa (
    struct in_addr in
);
```

返回一个点分十进制地址值

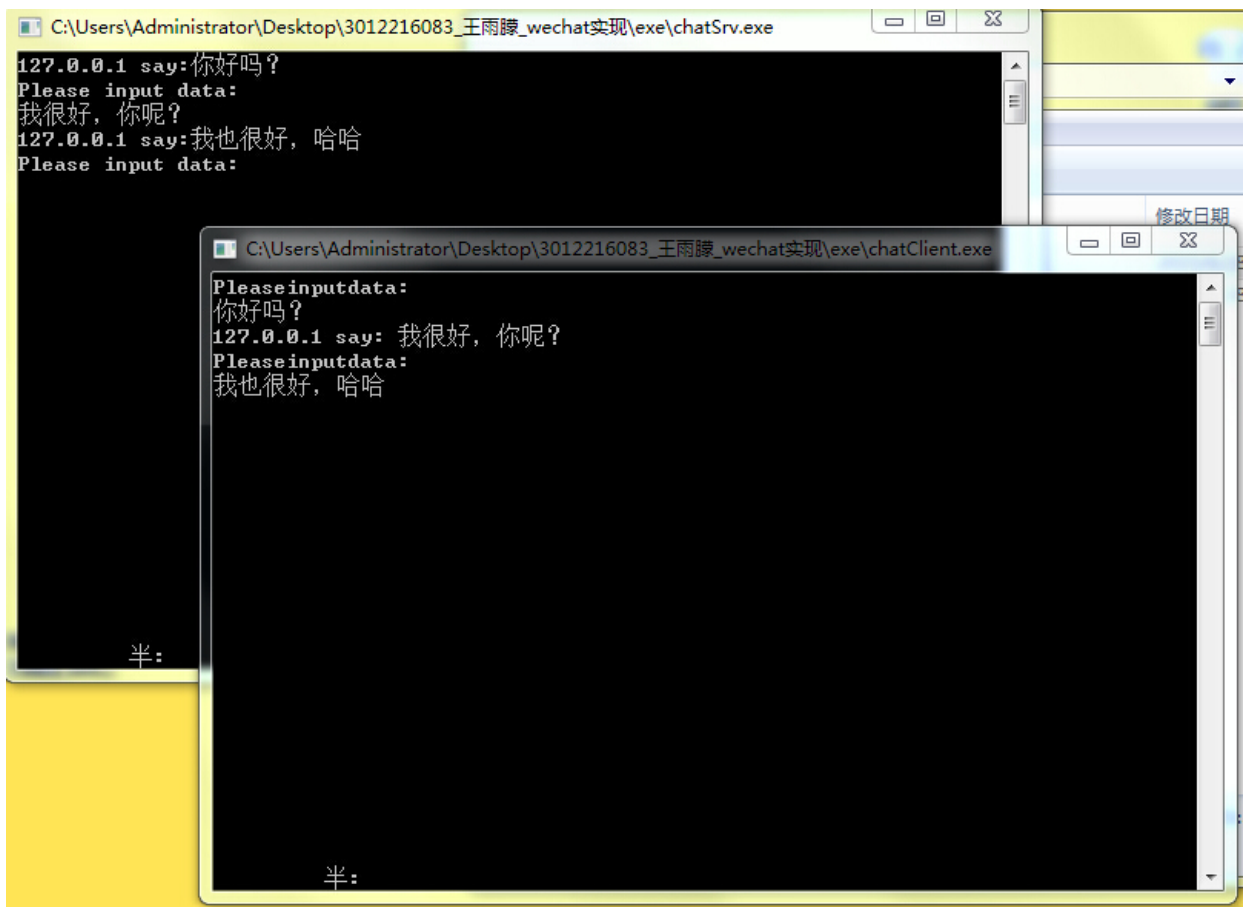
```
int bind (
    SOCKET s,
```

```

    const struct sockaddr FAR*   name,
    int namelen
);
SOCKADDR: The SOCKADDR structure varies depending on the protocol selected.
Except for the sa_family field, SOCKADDR contents are expressed in network
byte order.
struct sockaddr {
                u_short          sa_family;
                char              sa_data[14];
};

```

四、 实现结果



五、 心得体会

这次实验通过建立一个基本的网络聊天, 实现了 socket 的基本应用。在实验中, 可以有一些小小的改进。每次客户端和服务端只能给对方发送一次信息, 要等待对方的接受。基于此, 可以有一些改进。比如: 代码里有个关闭套接字的函数, close 套接字样, 不要关闭, 一直使用就可以了, 但是 accept 那里你也要判断一下不要在产生一个新的套接字, 就一个使用最早那个套接字就可以。