

# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## 实验报告

LAB REPORT



## 数据通信

Winsock ex5

姓 名: 袁炜程

学 号: 516030910287

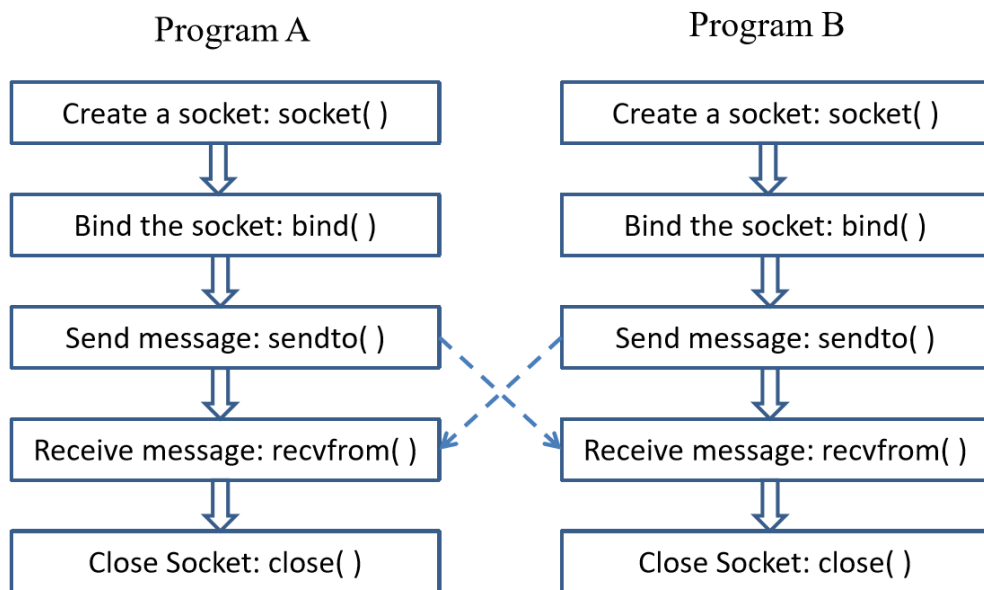
班 级: F1603602

## 一、实验要求

Write a chatting program using datagram socket and UDP protocol. People using your chatting program can send message to each other, as long as the IP address and Port number are known to each other, and the program is running.

## 二、实验原理

### 1、UDP 通信基本步骤



### peer-to-peer model

即是一个对称的模型，因此 `UDPServer` 和 `UPDClient` 代码一样。

### 2、重要函数 `sendto` 和 `recvfrom`

`sendto()`: 向某一地址发送数据。若无错误发生，`send()` 返回所发送数据的总数（这个数字可能小于 `len` 中所规定的大小）。否则的话，返回 `SOCKET_ERROR` 错误，应用程序可通过 `WSAGetLastError()` 获取相应错误代码。

简述:

向一指定目的地发送数据。

```
#include <winsock.h>
```

```
int PASCAL FAR sendto( SOCKET s, const char FAR* buf, int len, int flags,  
const struct sockaddr FAR* to, int tolen);
```

s: 一个标识套接口的描述字。

buf: 包含待发送数据的缓冲区。

len: buf缓冲区中数据的长度。

flags: 调用方式标志位。

to: (可选) 指针, 指向目的套接口的地址。

tolen: to所指地址的长度。

recvfrom(): 接收一个数据报并保存源地址。若无错误发生, recvfrom() 返回读入的字节数。如果连接已中止, 返回 0。否则的话, 返回 SOCKET\_ERROR 错误, 应用程序可通过 WSAGetLastError() 获取相应错误代码。

简述:

接收一个数据报并保存源地址。

```
#include <winsock.h>
```

```
int PASCAL FAR recvfrom( SOCKET s, char FAR* buf, int len, int flags,  
struct sockaddr FAR* from, int FAR* fromlen);
```

s: 标识一个已连接套接口的描述字。

buf: 接收数据缓冲区。

len: 缓冲区长度。

flags: 调用操作方式。

from: (可选) 指针, 指向装有源地址的缓冲区。

fromlen: (可选) 指针, 指向from缓冲区长度值。

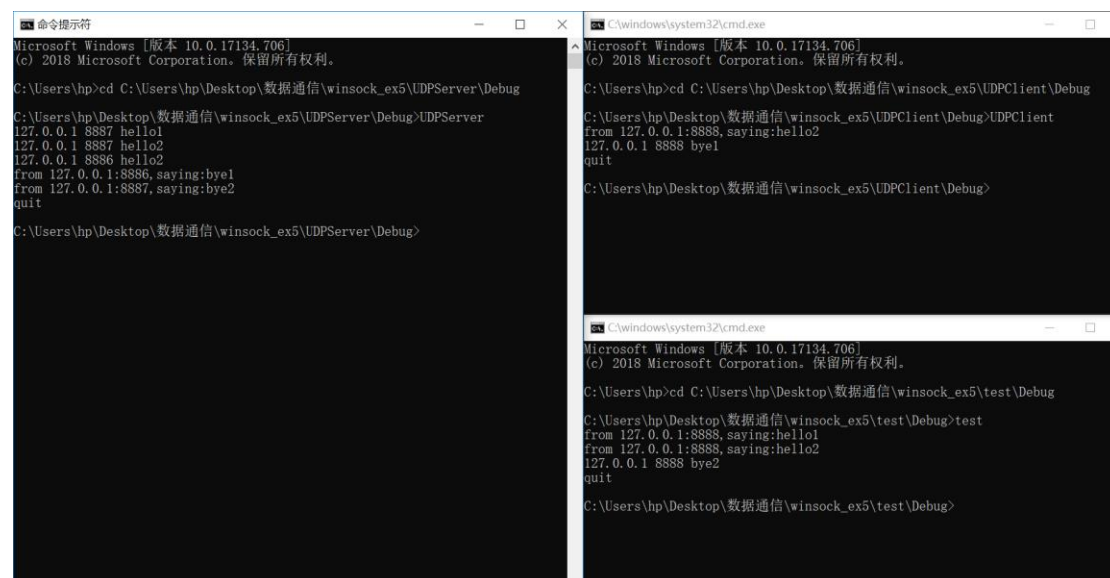
### 3、UDP Server 具体实现

首先新建 socket, 并选择 UDP 协议, 绑定 socket, 然后开始循环, 处理发送和接受消息。由于 sendto 和 recvfrom 都是阻塞的函数, 考虑用到多线程处理。

我的逻辑是首先处理接受消息，这应该是一个多线程，因为能同时收到多方的消息；而对于 `sendto`，阻塞也没有关系，不会出现同时对多方发送，同时发给错误的地址也没有关系，因此直接写在循环中即可，并且经过尝试，应该写在 `recvfrom` 多线程的后面。为了实现能 `sendto` 多个客户端，需要在发送消息时指明消息发送对象的 ip 地址和端口号，这需要对发送数据的格式作一定要求。我的格式为：“ip port message” 比如 “192.168.1.102 8888 hello”，然后就需要在发送前处理这段信息。我选择的是 `string.h` 中的 `strtok` 函数来以空格分割串，返回指向各字串的指针，取 3 个指针分别得到参数 ip 地址、端口号、数据传给 `sendto` 函数。

### 三、实验结果

#### 1、本机三个聊天程序通信（UDPClient 8886，UDPServer 8888，test 8887）



```
Microsoft Windows [版本 10.0.17134.706]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\hp>cd C:\Users\hp\Desktop\数据通信\winsock_ex5\UDPServer\Debug

C:\Users\hp\Desktop\数据通信\winsock_ex5\UDPServer\Debug>UDPServer
127.0.0.1 8887 hello1
127.0.0.1 8887 hello2
127.0.0.1 8886 hello2
from 127.0.0.1:8886, saying:bye1
from 127.0.0.1:8887, saying:bye2
quit
C:\Users\hp\Desktop\数据通信\winsock_ex5\UDPServer\Debug>
```

```
Microsoft Windows [版本 10.0.17134.706]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\hp>cd C:\Users\hp\Desktop\数据通信\winsock_ex5\UDPClient\Debug

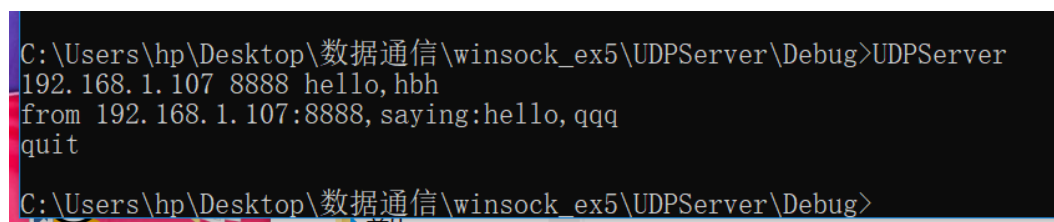
C:\Users\hp\Desktop\数据通信\winsock_ex5\UDPClient\Debug>UDPClient
from 127.0.0.1:8888, saying:hello2
127.0.0.1 8888 bye1
quit
C:\Users\hp\Desktop\数据通信\winsock_ex5\UDPClient\Debug>
```

```
Microsoft Windows [版本 10.0.17134.706]
(c) 2018 Microsoft Corporation. 保留所有权利。

C:\Users\hp>cd C:\Users\hp\Desktop\数据通信\winsock_ex5\test\Debug

C:\Users\hp\Desktop\数据通信\winsock_ex5\test\Debug>test
from 127.0.0.1:8888, saying:hello1
from 127.0.0.1:8888, saying:hello2
127.0.0.1 8888 bye2
quit
C:\Users\hp\Desktop\数据通信\winsock_ex5\test\Debug>
```

#### 2、与局域网中其他聊天程序通信（与胡秉晖同学的 UDPChatting 聊天）



```
C:\Users\hp\Desktop\数据通信\winsock_ex5\UDPServer\Debug>UDPServer
192.168.1.107 8888 hello, hbh
from 192.168.1.107:8888, saying:hello, qq
quit
C:\Users\hp\Desktop\数据通信\winsock_ex5\UDPServer\Debug>
```

### 四、实验思考

现在的实现结果基本完成了功能，但是跟实际的聊天工具仍有很大差距。主要是消息格式太繁琐，而且也没有异常处理，对用户不太友好。最理想的情况是服务器在与一个客户端通信的过程中，开启一个新的线程，并同时打开一个新的命令行，在那个命令行中就不需要每次发送消息都要输入 ip 地址和端口号。具体实现上来说，首先在打开服务器的时候，主窗口会提示用户输入 ip 地址和端口号，然后弹出新命令行与客户端通信；服务器作为客户端时接受消息，如果有通信请求，也打开一个命令行与另一个客户端交流，这样更贴合实际。希望能在之后有时间的情况下尝试非对称的聊天工具，更好地了解 UDP 的 Winsock 编程和 UDP 机制。