

**SHANGHAI JIAO TONG UNIVERSITY**

**实验报告**

LAB REPORT



**数据通信**

Winsock ex5

姓 名: 裘炜程

学 号: 516030910287

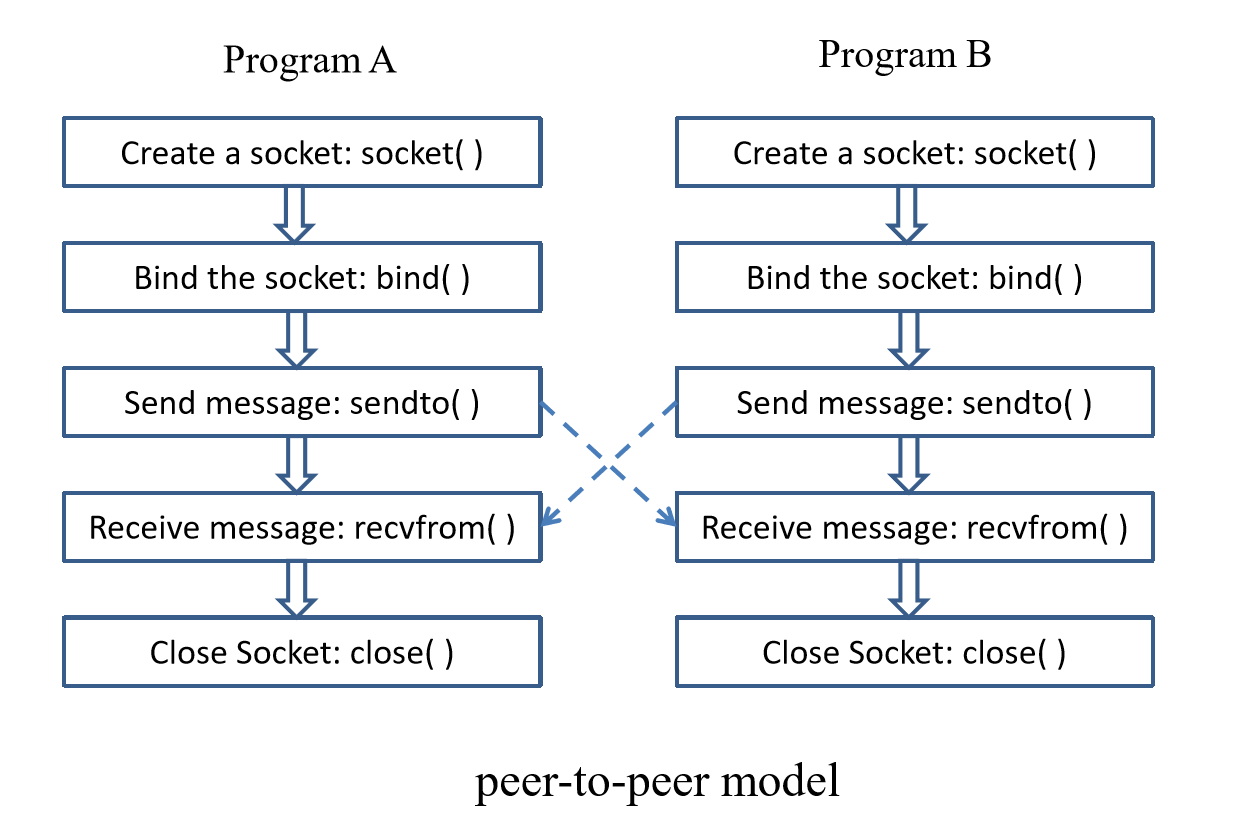
班 级: F1603602

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



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。



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。

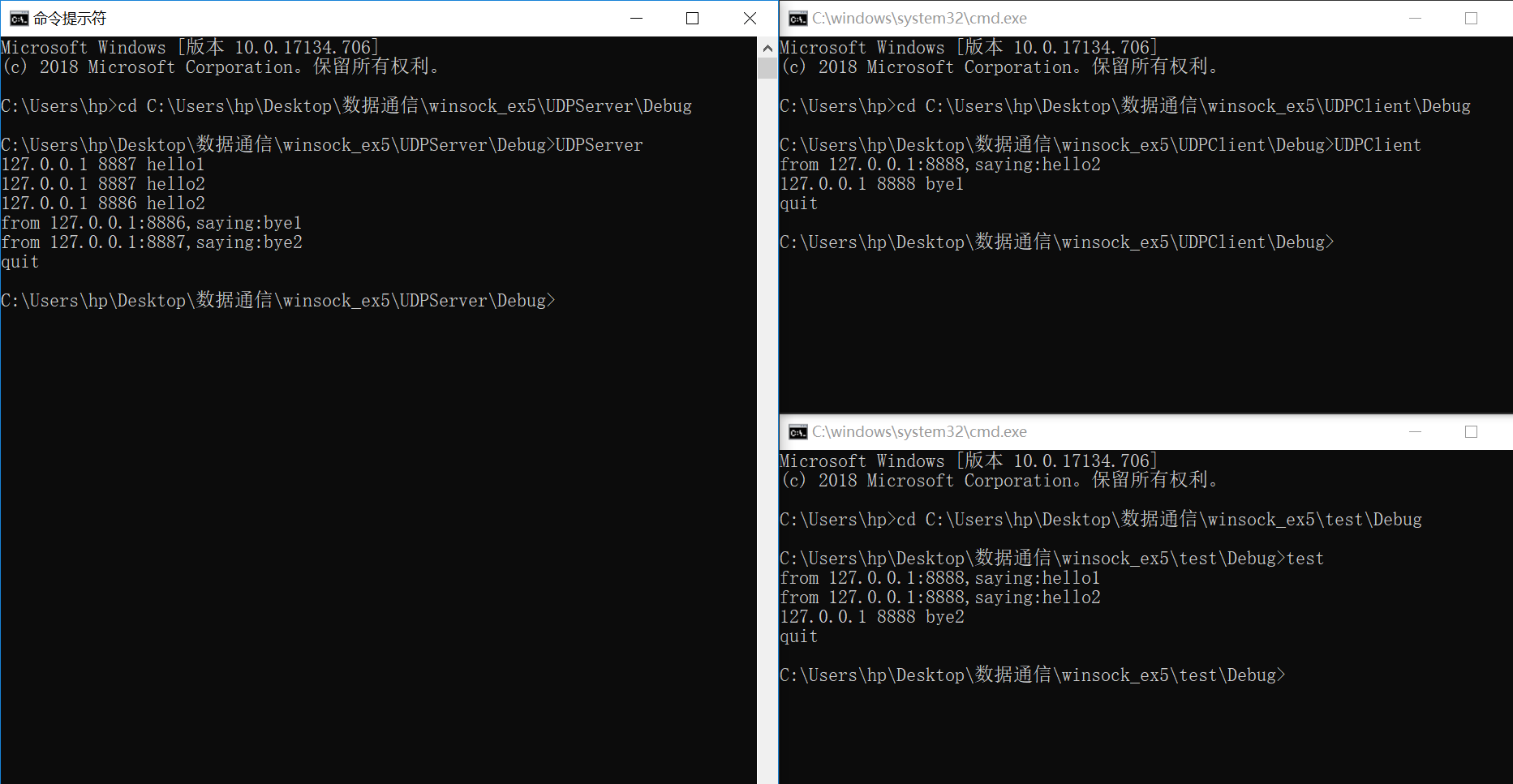


3、UDPServer具体实现

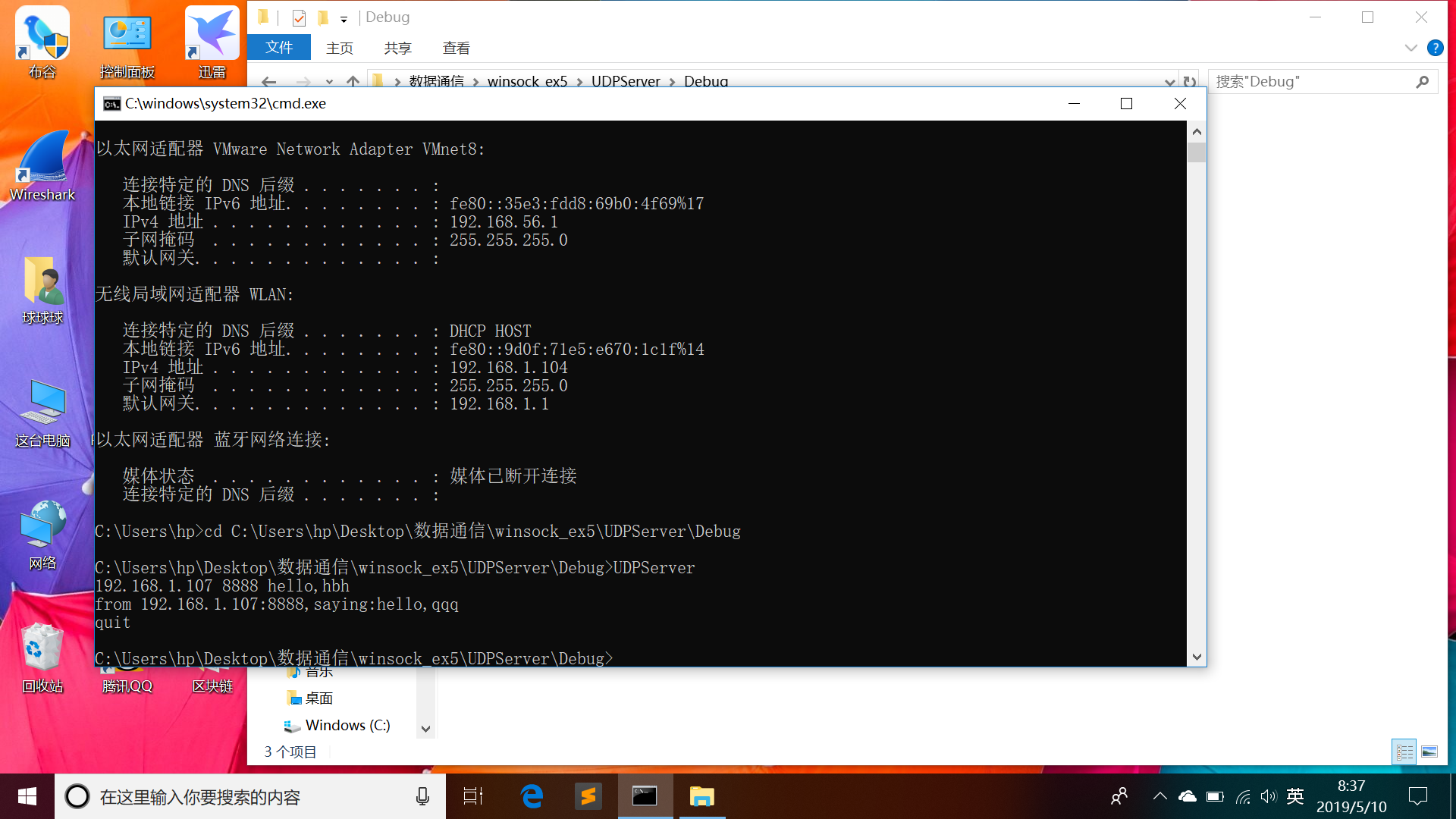
首先新建socket，并选择UDP协议，绑定socket，然后开始循环，处理发送和接受消息。由于sendto和recvfrom都是阻塞的函数，考虑用到多线程处理。我的逻辑是首先处理接受消息，这应该是一个多线程，因为能同时收到多方的消息；而对于sendto，阻塞也没有关系，不会出现同时对多方发送，同时发给错误的地址也没有关系，因此直接写在循环中即可，并且经过尝试，应该写在recvfrom多线程的后面。为了实现能sendto多个客户端，需要在发送消息时指明消息发送对象的ip地址和端口号，这需要对发送数据的格式作一定要求。我的格式为：“ip port message”比如“192.168.1.102 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）



2、与局域网中其他聊天程序通信（与胡秉晖同学的UDPChatting聊天）



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。