

数据通信作业-4

姓名： 费扬 学号： 519021910917 日期： 2022.04.25

一、 实验名称及内容

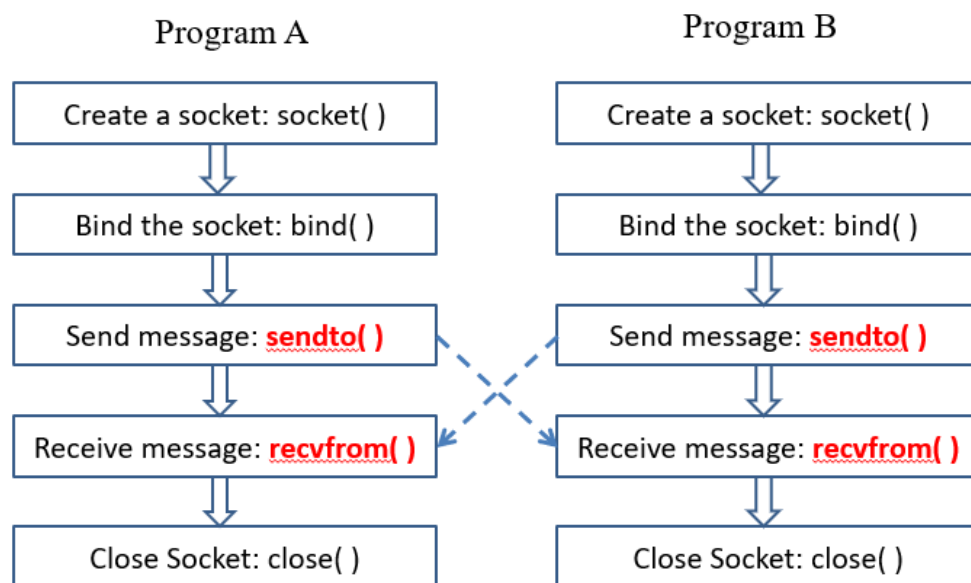
Assignment4: udpchatting

使用 winsock 编程，主要目标任务为：

Task: 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.

即使用数据报套接字和UDP协议编写聊天程序。使用聊天程序的人可以互相发送消息，只要该程序正在运行并且对方知道IP地址和端口号即可。

二、 实验过程和结果



peer-to-peer model

使用统一结构

还有 client 加 server 结构的通信方式，分开写两个程序即可，基本方法类似，这里没有使用。

Task:

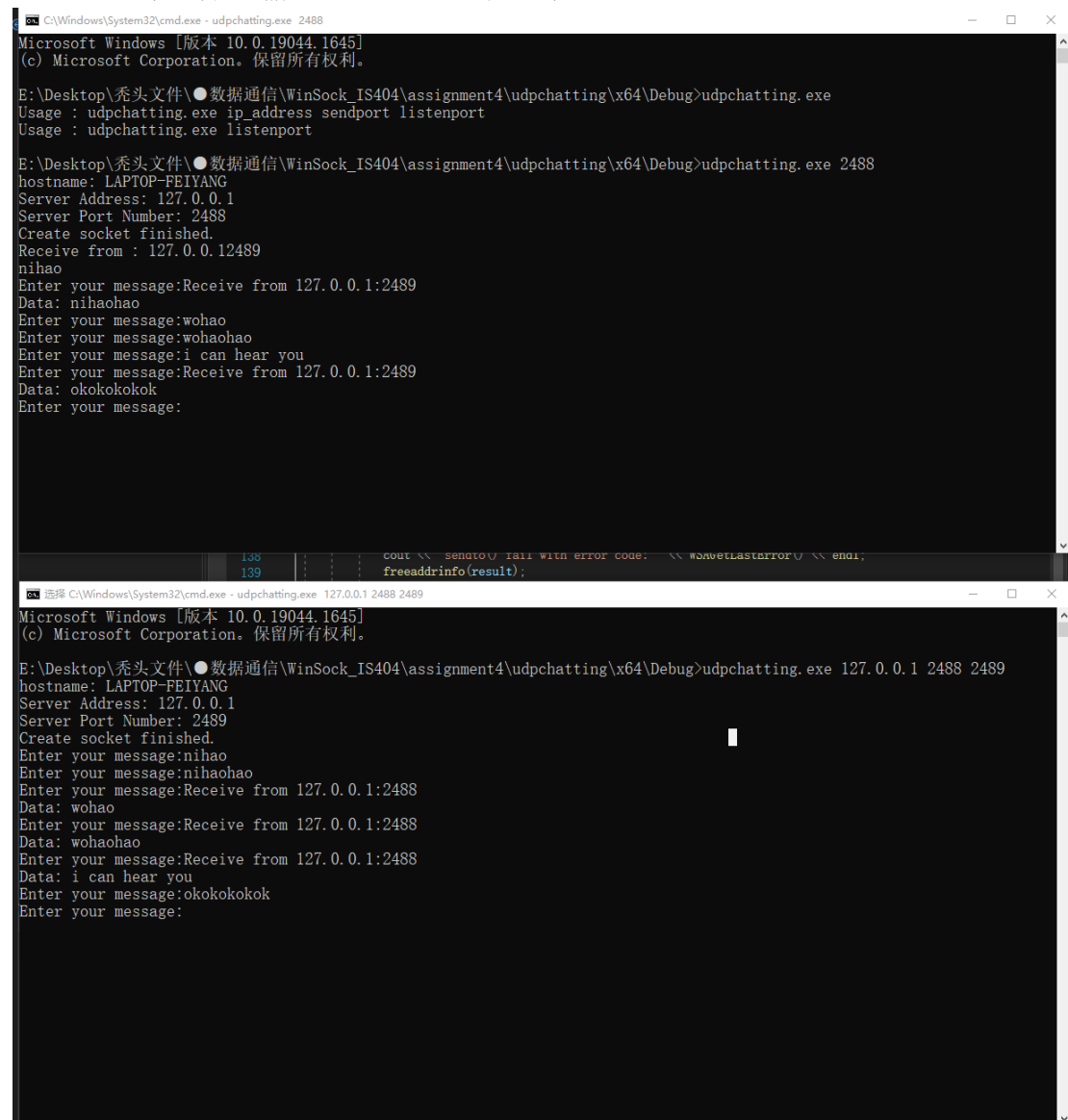
基本操作步骤：类似 TCP 传输通信，在其基础上修改用于 udp 协议：

udpchatting:

1. Set isHolder to differ two modes.
2. Initialize Winsock
3. Creates a socket.
4. Bind the socket.
5. Sends and receives data.
6. Disconnects.

代码见打包文件中 udpchatting.cpp，不予展示：

之后在命令行操作输出，需打开两个命令行进行对话：



```
C:\Windows\System32\cmd.exe - udpchatting.exe 2488
Microsoft Windows [版本 10.0.19044.1645]
(c) Microsoft Corporation. 保留所有权利。

E:\Desktop\秃头文件\●数据通信\WinSock_IS404\assignment4\udpchatting\x64\Debug>udpchatting.exe
Usage : udpchatting.exe ip_address sendport listenport
Usage : udpchatting.exe listenport

E:\Desktop\秃头文件\●数据通信\WinSock_IS404\assignment4\udpchatting\x64\Debug>udpchatting.exe 2488
hostname: LAPTOP-FEIJANG
Server Address: 127.0.0.1
Server Port Number: 2488
Create socket finished.
Receive from : 127.0.0.1:2489
nihao
Enter your message:Receive from 127.0.0.1:2489
Data: nihao
Enter your message:wohao
Enter your message:wohaohao
Enter your message:i can hear you
Enter your message:Receive from 127.0.0.1:2489
Data: okokokokok
Enter your message:

138 cout << "sendto() fail with error code: " << WSAGetLastError() << endl;
139 freeaddrinfo(result);

选择 C:\Windows\System32\cmd.exe - udpchatting.exe 127.0.0.1 2488 2489
Microsoft Windows [版本 10.0.19044.1645]
(c) Microsoft Corporation. 保留所有权利。

E:\Desktop\秃头文件\●数据通信\WinSock_IS404\assignment4\udpchatting\x64\Debug>udpchatting.exe 127.0.0.1 2488 2489
hostname: LAPTOP-FEIJANG
Server Address: 127.0.0.1
Server Port Number: 2489
Create socket finished.
Enter your message:nihao
Enter your message:nihao
Enter your message:Receive from 127.0.0.1:2488
Data: wohao
Enter your message:Receive from 127.0.0.1:2488
Data: wohao
Enter your message:Receive from 127.0.0.1:2488
Data: i can hear you
Enter your message:okokokok
Enter your message:
```

在 cmd 中实现 udp 传输的对话

三、问题与思考

1. 对比 TCP 和 UDP 传输：

TCP:

TCP 编程的服务器端一般步骤是：

- 1、创建一个 socket，用函数 `socket()`;
- 2、设置 socket 属性，用函数 `setsockopt()`;
- 3、绑定 IP 地址、端口等信息到 socket 上，用函数 `bind()`;
- 4、开启监听，用函数 `listen()`;
- 5、接收客户端上来的连接，用函数 `accept()`;
- 6、收发数据，用函数 `send()`和 `recv()`，或者 `read()`和 `write()`;
- 7、关闭网络连接;
- 8、关闭监听;

TCP 编程的客户端一般步骤是：

- 1、创建一个 socket，用函数 `socket()`;
- 2、设置 socket 属性，用函数 `setsockopt()`;
- 3、绑定 IP 地址、端口等信息到 socket 上，用函数 `bind()`;
- 4、设置要连接的对方的 IP 地址和端口等属性;
- 5、连接服务器，用函数 `connect()`;
- 6、收发数据，用函数 `send()`和 `recv()`，或者 `read()`和 `write()`;
- 7、关闭网络连接;

UDP:

与之对应的 UDP 编程步骤要简单许多，分别如下：

UDP 编程的服务器端一般步骤是：

- 1、创建一个 socket，用函数 `socket()`;
- 2、设置 socket 属性，用函数 `setsockopt()`;
- 3、绑定 IP 地址、端口等信息到 socket 上，用函数 `bind()`;
- 4、循环接收数据，用函数 `recvfrom()`;
- 5、关闭网络连接;

UDP 编程的客户端一般步骤是：

- 1、创建一个 socket，用函数 `socket()`;
- 2、设置 socket 属性，用函数 `setsockopt()`;
- 3、绑定 IP 地址、端口等信息到 socket 上，用函数 `bind()`;
- 4、设置对方的 IP 地址和端口等属性;
- 5、发送数据，用函数 `sendto()`;
- 6、关闭网络连接;

TCP 与 UDP 区别总结：

- 1、TCP 面向连接（如打电话要先拨号建立连接），UDP 是无连接的，即发送数据之前不需要建立连接;
- 2、TCP 提供可靠的服务。也就是说，通过 TCP 连接传送的数据，无差错，不丢失，不重复，且按序到达;UDP 尽最大努力交付，即不保证可靠交付;
- 3、TCP 面向字节流，实际上是 TCP 把数据看成一连串无结构的字节流，UDP 是面向报

文的：UDP 没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如 IP 电话，实时视频会议等）；

4、每一条 TCP 连接只能是点到点的，UDP 支持一对一，一对多，多对一和多对多的交互通信；

5、TCP 首部开销 20 字节；UDP 的首部开销小，只有 8 个字节；

6、TCP 的逻辑通信信道是全双工的可靠信道，UDP 则是不可靠信道；

2. 10048 报错：

经查阅，发现是端口已经被占用，改用即可。清空各占用端口或是重启也可以生效。

(后来发现导致的原因是 sendport 和 listenport 输入反了...)

Ref:

Getting started with Winsock

[https://msdn.microsoft.com/en-us/library/ms738545\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ms738545(v=vs.85).aspx)

Winsock reference

[https://msdn.microsoft.com/en-us/library/ms741416\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ms741416(v=vs.85).aspx)