

廈門大學



信息学院软件工程系

《计算机网络》实验报告

题 目 实验六 利用 Socket API 实现网上点对点通信

班 级 软件工程 2018 级 1 班

姓 名 何明祥

学 号 24320182203193

实验时间 2020 年 4 月 22 日

2020 年 4 月 22 日

1 实验目的

在 Windows 或 Linux 操作系统（也可以将客户端部署在 Android、iOS 或 WinPhone 手机）下，分别基于 TCP 和 UDP 协议，利用 Socket API 实现网上点对点通信。

程序一“基于 TCP 的可靠文件传输”，功能包括：

在客户端，用户选择本地的某个文件，并发送到服务器端。

在服务器端，接收客户端传输的数据流，并按 IP 地址保存在服务器端（文件名重复的，可以覆盖）。

如果传输过程中服务器端发现客户端断开，服务器端应删除文件，并在屏幕上提示，如“IP: 1.2.3.4 发来 abcd.txt 文件过程中失去连接。”。如果客户端发现服务器端不工作，客户端应有提示“服务器 1.2.3.5:62345 失去连接”。

程序二“基于 UDP 的不可靠文件传输”，功能同上，但不能使用 TCP 协议进行传输。考虑如果传输过程中服务器端、客户端如何发现断开。

2 实验环境

Visual studio community 2017 C 语言

3 实验结果

基于 TCP 协议的点对点通信：

服务器端步骤：

- 1、加载套接字库，创建套接字(WSAStartup()/socket());
- 2、绑定套接字到一个 IP 地址和一个端口上(bind());
- 3、将套接字设置为监听模式等待连接请求(listen());

4、请求到来后，接受连接请求，返回一个新的对应于此次连接的套接字(accept());

5、用返回的套接字和客户端进行通信(send()/recv());

6、返回，等待另一连接请求；

7、关闭套接字，关闭加载的套接字库(closesocket()/WSACleanup())。

客户端步骤：

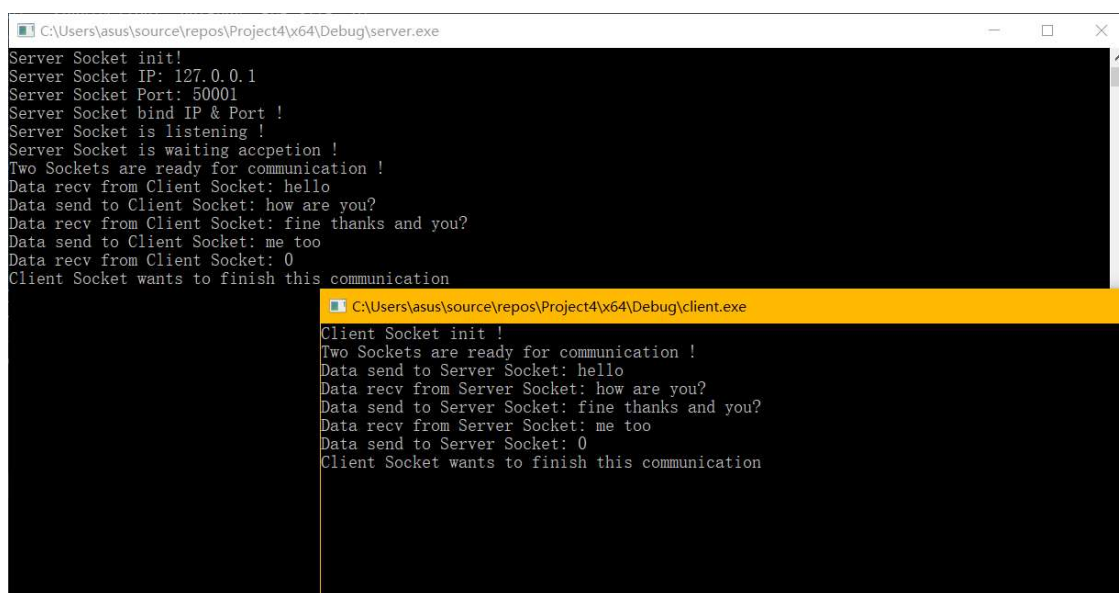
1、加载套接字库，创建套接字(WSAStartup()/socket());

2、向服务器发出连接请求(connect());

3、和服务器端进行通信(send()/recv());

4、关闭套接字，关闭加载的套接字库(closesocket()/WSACleanup())。

任意一方输入 ‘0’ 结束对话



```
C:\Users\asus\source\repos\Project4\x64\Debug\server.exe
Server Socket init!
Server Socket IP: 127.0.0.1
Server Socket Port: 50001
Server Socket bind IP & Port !
Server Socket is listening !
Server Socket is waiting accpetion !
Two Sockets are ready for communication !
Data rcv from Client Socket: hello
Data send to Client Socket: how are you?
Data rcv from Client Socket: fine thanks and you?
Data send to Client Socket: me too
Data rcv from Client Socket: 0
Client Socket wants to finish this communication

C:\Users\asus\source\repos\Project4\x64\Debug\client.exe
Client Socket init !
Two Sockets are ready for communication !
Data send to Server Socket: hello
Data rcv from Server Socket: how are you?
Data send to Server Socket: fine thanks and you?
Data rcv from Server Socket: me too
Data send to Server Socket: 0
Client Socket wants to finish this communication
```

基于 UDP 协议的点对点通信：

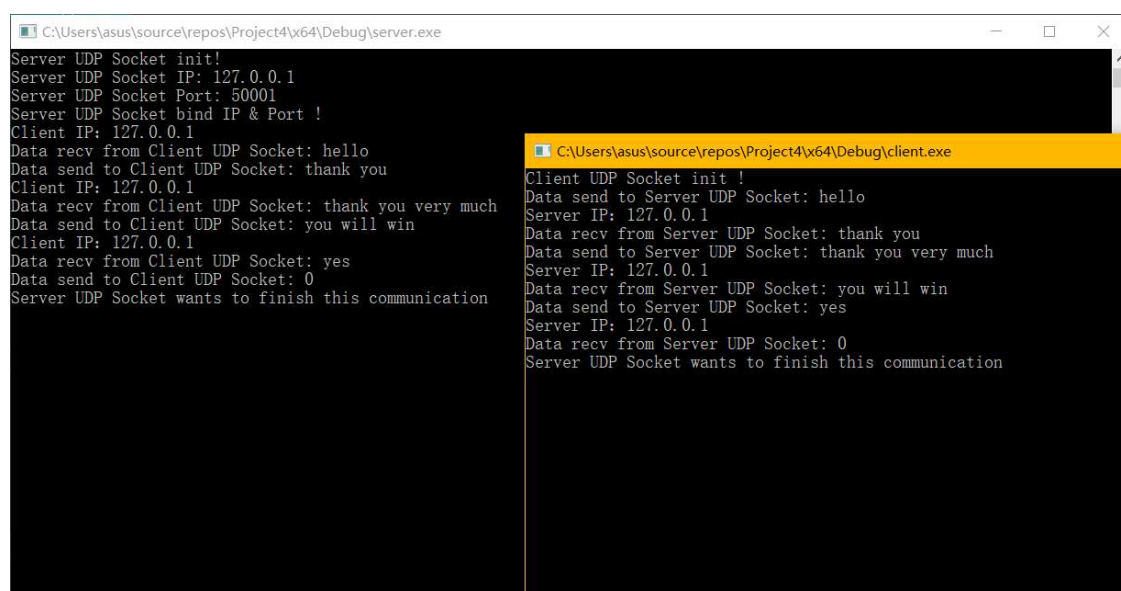
服务器端步骤：

1、加载套接字库，创建套接字(WSAStartup()/socket());

- 2、绑定套接字到一个 IP 地址和一个端口上(bind());
- 3、循环用返回的套接字和客户端进行通信(sendto()/recvfrom());
- 4、关闭套接字，关闭加载的套接字库(closesocket()/WSACleanup())。

客户端步骤:

- 1、加载套接字库，创建套接字(WSAStartup()/socket());
- 2、和服务端进行通信(sendto()/recvfrom());
- 3、关闭套接字，关闭加载的套接字库(closesocket()/WSACleanup())。



```
C:\Users\asus\source\repos\Project4\Debug\server.exe
Server UDP Socket init!
Server UDP Socket IP: 127.0.0.1
Server UDP Socket Port: 50001
Server UDP Socket bind IP & Port !
Client IP: 127.0.0.1
Data rcv from Client UDP Socket: hello
Data send to Client UDP Socket: thank you
Client IP: 127.0.0.1
Data rcv from Client UDP Socket: thank you very much
Data send to Client UDP Socket: you will win
Client IP: 127.0.0.1
Data rcv from Client UDP Socket: yes
Data send to Client UDP Socket: 0
Server UDP Socket wants to finish this communication

C:\Users\asus\source\repos\Project4\Debug\client.exe
Client UDP Socket init !
Data send to Server UDP Socket: hello
Server IP: 127.0.0.1
Data rcv from Server UDP Socket: thank you
Data send to Server UDP Socket: thank you very much
Server IP: 127.0.0.1
Data rcv from Server UDP Socket: you will win
Data send to Server UDP Socket: yes
Server IP: 127.0.0.1
Data rcv from Server UDP Socket: 0
Server UDP Socket wants to finish this communication
```

4 实验总结

对利用 Socket API 分别基于 TCP 和 UDP 协议的点对点通信过程有了更加深刻的了解。