

****

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B班**

**姓　　名 彭书浩**

**学　　号 24320182203251**

**实验时间 2020年4月22日**

**2020 年 5 月 2 日**

# 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

# 实验环境

Windows 10操作系统

-Visual Studio 2017

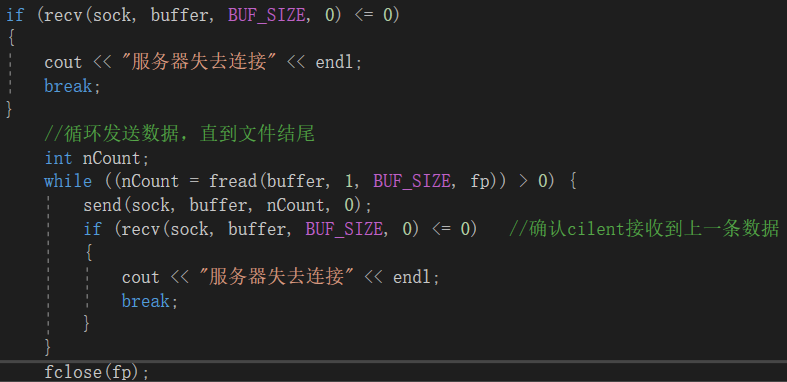
# 实验结果

1、基于TCP的可靠文件传输：

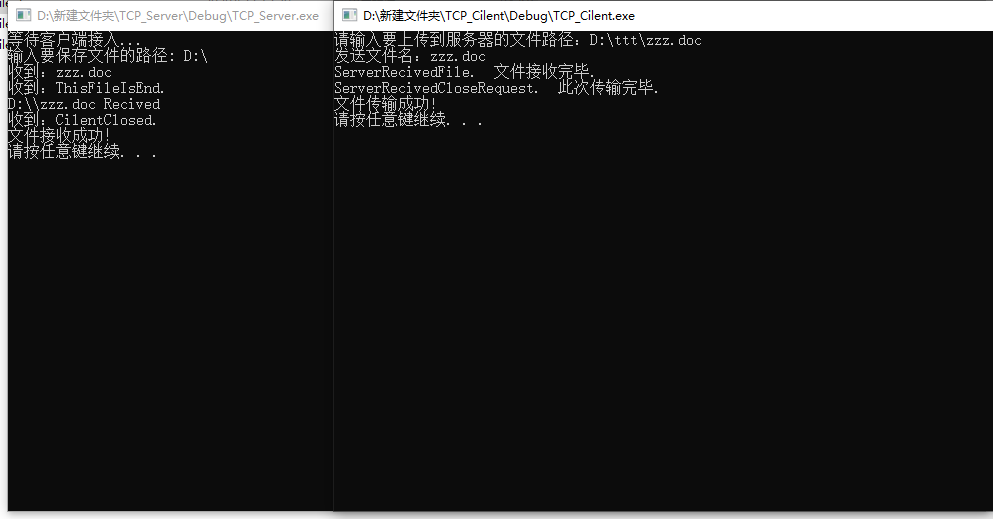
通过一次send对应一次recv来实现信息与文件数据的传输

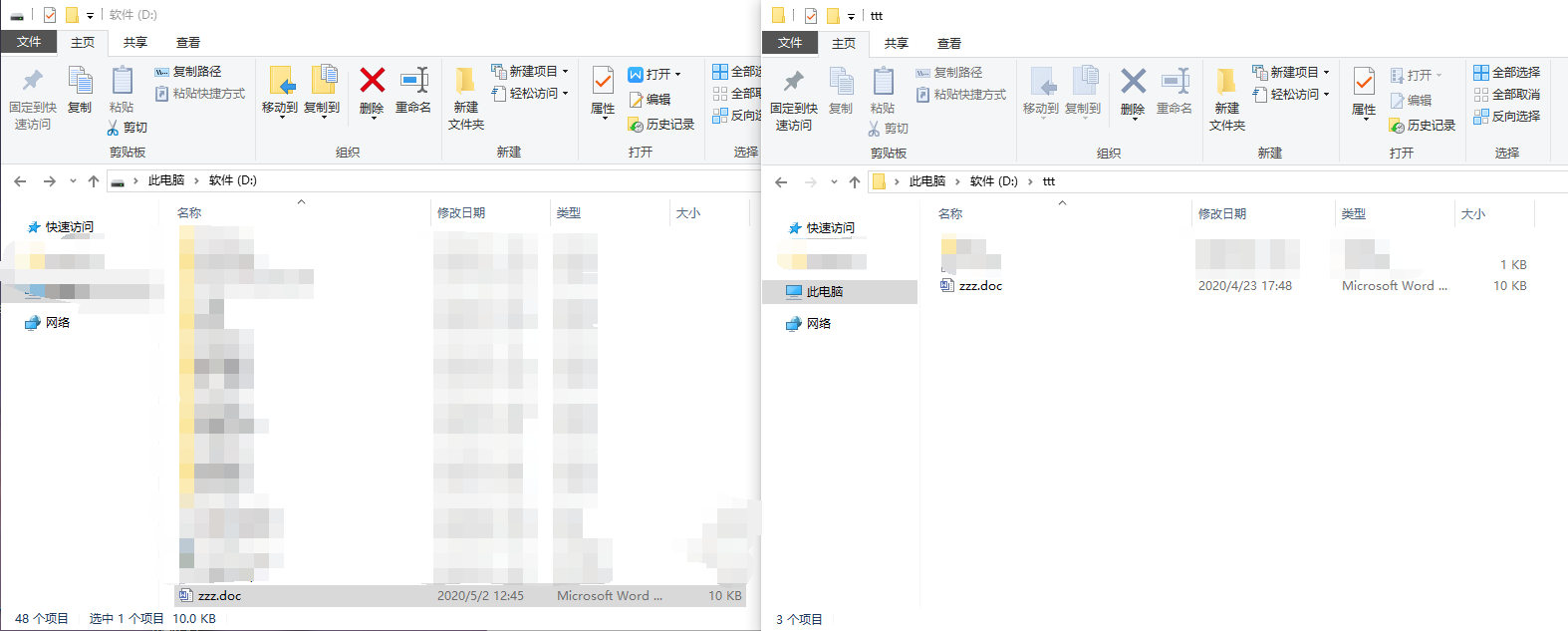
以二进制形式发送要传输的文件，设置缓存区大小循环发送至结束

如果recv的值<=0,则说明连接中断。



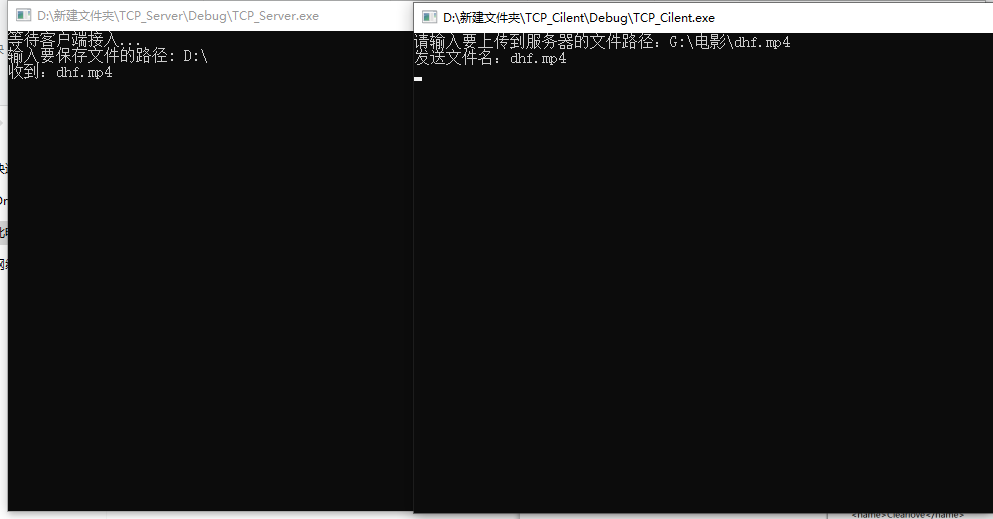
1.1、文件传输成功结果：



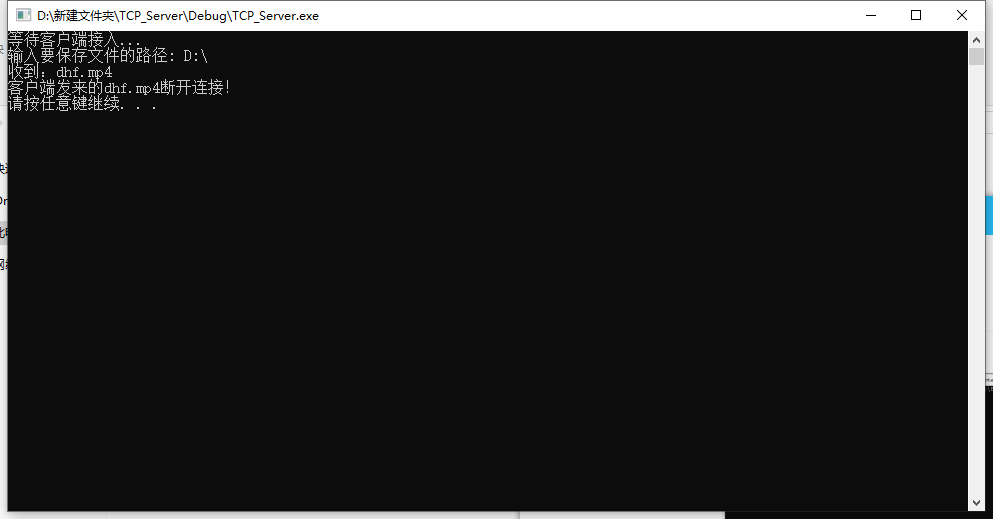


1.2、断开连接结果测试：

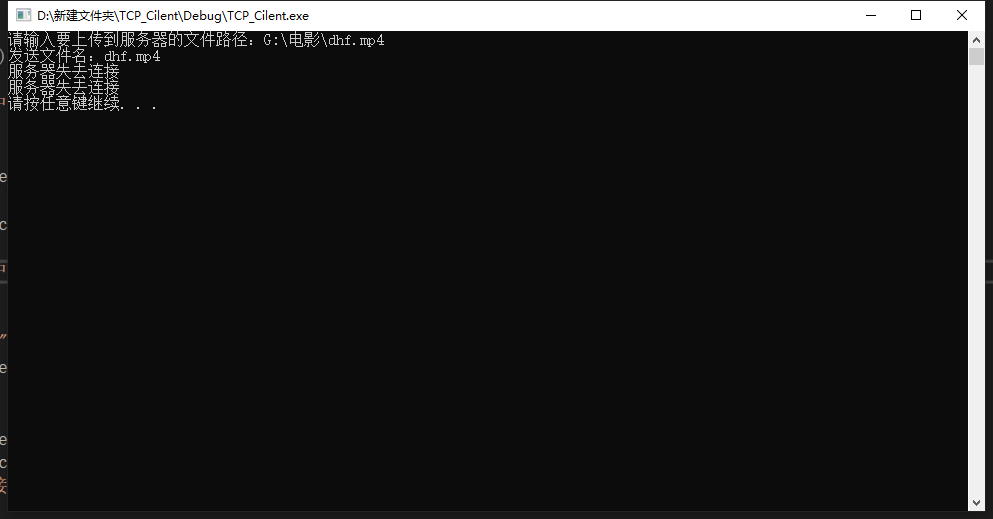
（大文件传输耗费时间，文件传输中。。。）



（关闭客户端，服务器示意图：）



（关闭服务器，客户端示意图：）

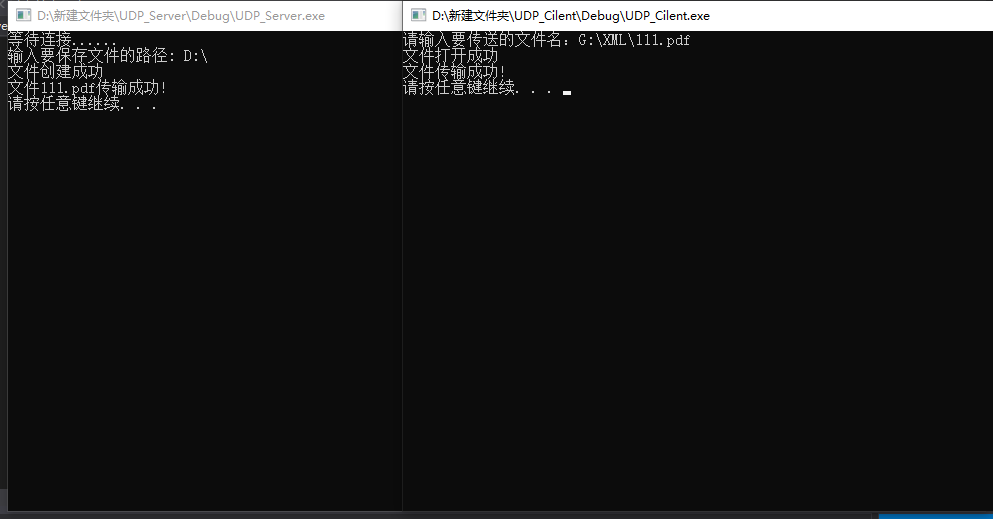


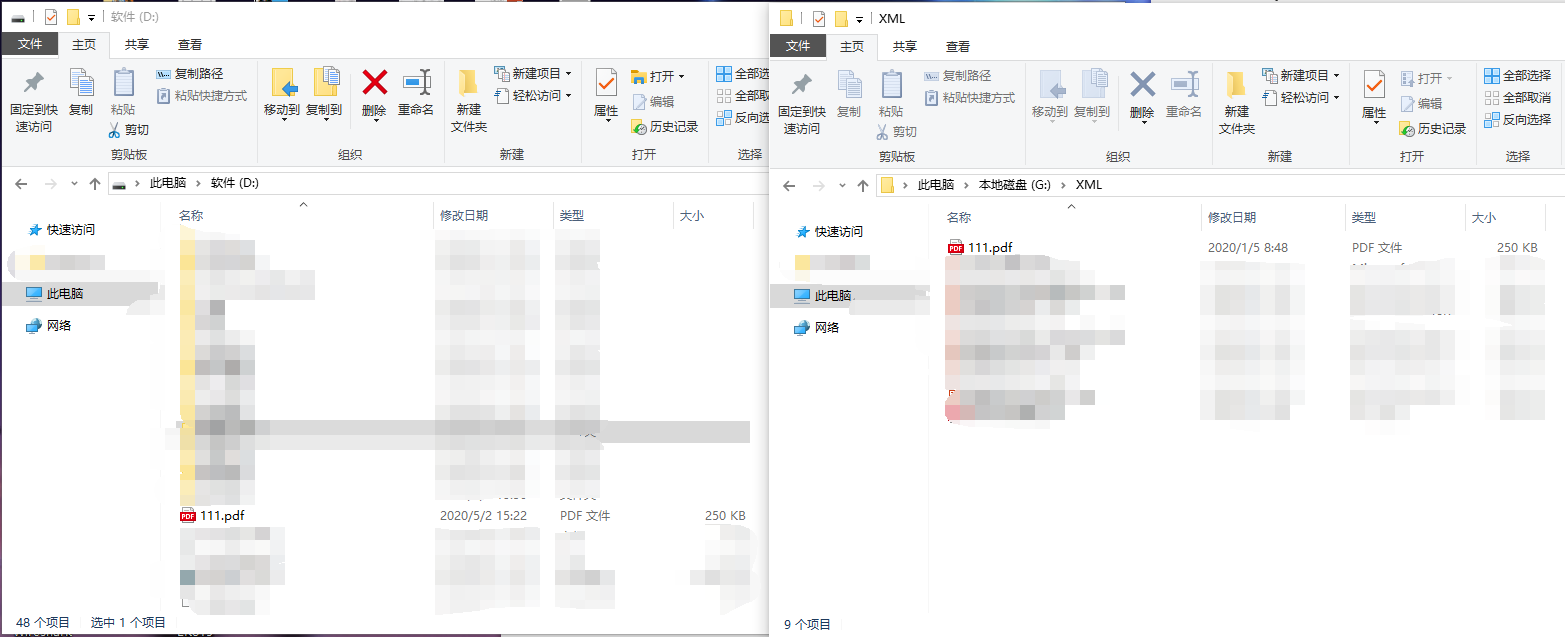
2、基于UDP的不可靠文件传输：

通过一次sendto对应一次recvfrom来实现信息与文件数据的传输

以二进制形式发送要传输的文件，设置缓存区大小循环发送至结束

连接中断用TCP的方法并不可行。





# 实验总结

1、通过对项目二的文件夹遍历传输文件的代码编写，深入学习了TCP的数据传输方式。

2、 使用TCP文件传输时，连续的send和recv时，若发送的数据未达到缓存区的大小，将不会立即把内容发送出去，而是先存入缓存区，待缓存区剩余空间不足时，才会把数据发送出去。

3、int send( SOCKET s, const char FAR \*buf, int len, int flags )：

（1）send先比较待发送数据的长度len和套接字s的发送缓冲的长度， 如果len大于s的发送缓冲区的长度，该函数返回SOCKET\_ERROR；

（2）如果len小于或者等于s的发送缓冲区的长度，那么send先检查协议是否正在发送s的发送缓冲中的数据，如果是就等待协议把数据发送完。

（3）如果len大于剩余空间大小，send就一直等待协议把s的发送缓冲中的数据发送完

（4）如果len小于剩余 空间大小，send就仅仅把buf中的数据copy到剩余空间里。

1. int recv( SOCKET s, char FAR \*buf, int len, int flags);

（1）recv先等待s的发送缓冲中的数据被协议传送完毕，如果协议在传送s的发送缓冲中的数据时出现网络错误，那么recv函数返回SOCKET\_ERROR，

（2）如果s的发送缓冲中没有数据或者数据被协议成功发送完毕后，recv先检查套接字s的接收缓冲区，如果s接收缓冲区中没有数据或者协议正在接收数据，那么recv就一直等待，直到协议把数据接收完毕。当协议把数据接收完毕，recv函数就把s的接收缓冲中的数据copy到buf中。

5、UDP没有要求保持一个连接，所以考虑可以尝试使用一个定时器来判断是否服务器与客户端断开连接。