

# 计算机网络

## 实验报告

(2022学年秋季学期)

教学班级	计科二班	专业 (方向)	计算机科学与技术
学号	20308003	姓名	曾伟超

教学班级	计科二班	专业 (方向)	计算机科学与技术
学号	20337263	姓名	俞泽斌

## 一、 实验题目

### 5.2 TCP 通信编程：

**【目的】**选择一个自己熟悉的操作系统和语言，编写采用 TCP socket 的通信程序；

**【要求】**编写服务器端和客户端程序。

客户端软件支持三个功能或者命令：远程（服务器）当前目录文件列表、上传文件、下载文件。

服务器端软件：按用户命令，支持客服端的以上功能或者命令：提供当前文件目录的文件列表、接收上传文件，按客户端用户要求提供其要求的文件（文件下载）。服务器必须提供日志记录以上操作及操作结果状态。

服务器端必须支持多个客户端的并发运行。可在屏幕显示客户端的请求及服务器端的处理日志。

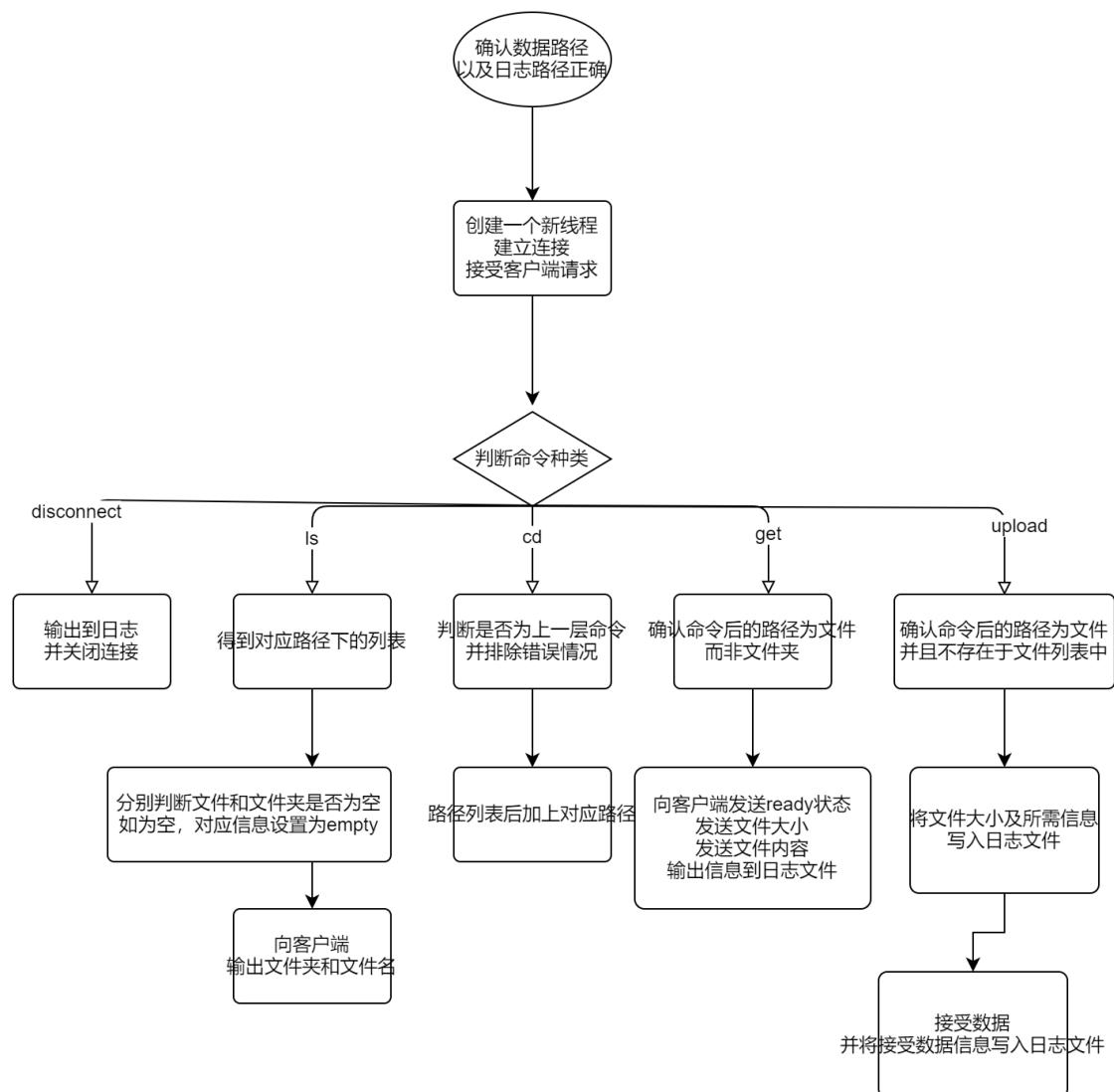
#### 【测试要求】

- 1) A 机器运行服务器软件；
- 2) 在客户端 B 机器的命令行同时运行 4 个客户端；
- 3) 在 A 机器也同时运行客户端软件 2 个客户端；
- 4) 观看服务器端运行的日志。

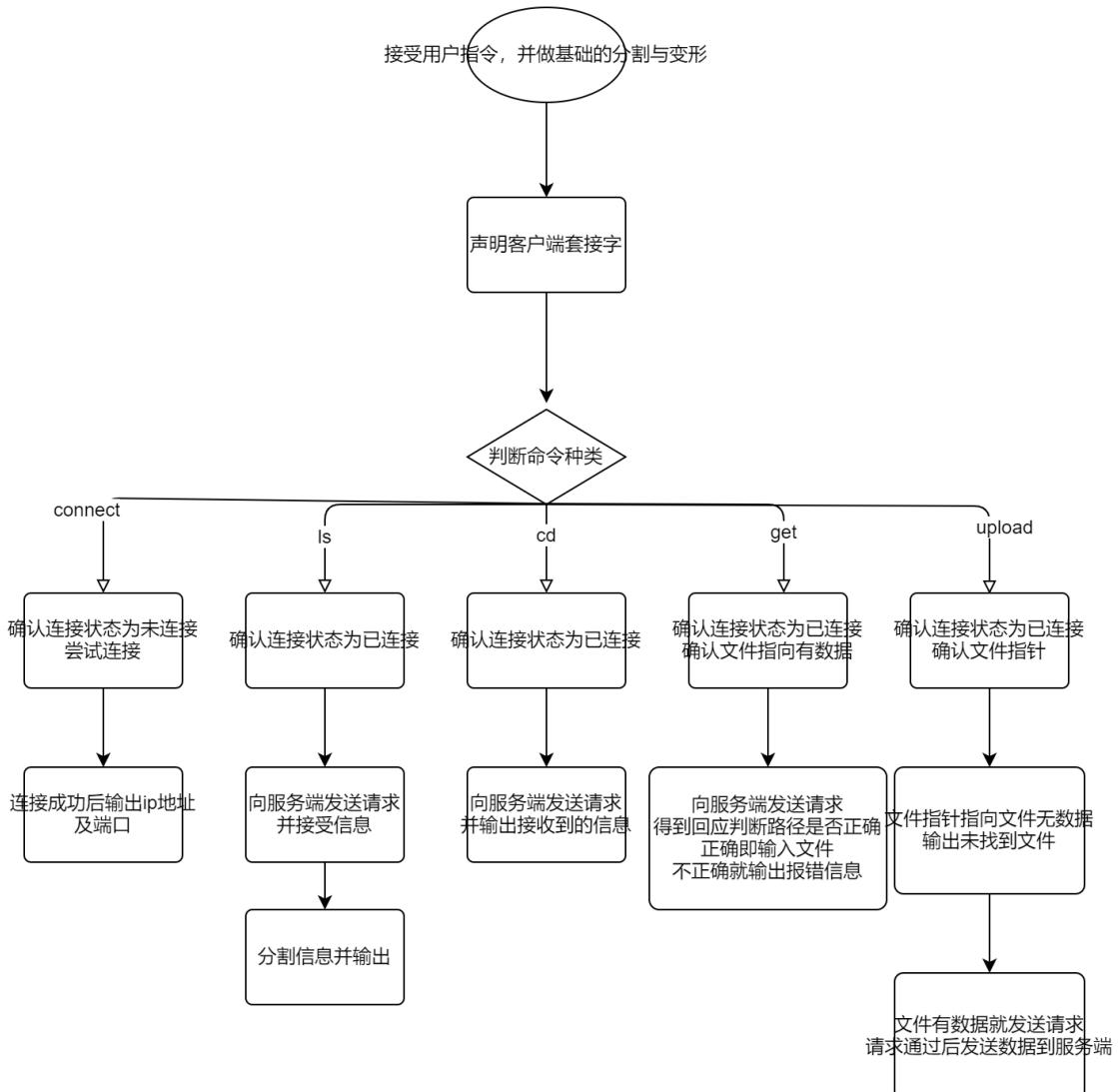
## 二、 实验内容

几个基本命令的流程图如下

服务端：



客户端：



主体部分剩下的命令的具体思想与上述几个命令差不多，由于篇幅的限制就不赘述，我们另外还实现了help,clear,mkdir,rm,pwd等命令。

首先来看客户端

```

def print_help():
    print("Support command: connect, ls, cd, get, upload, help, disconnect,
exit")
    print("connect: connect to a tcp server, usage: connect <server ip> <server
port>")
    print("ls: list files in current directory, usage: ls")
    print("cd: change current directory, usage: cd <directory>")
    print("get: download a file from server, usage: get <remote file name>")
    print("upload: upload a file to server, usage: upload <local file name>")
    print("help: print help message, usage: help")
    print("clear: clear screen, usage: clear")
    print("disconnect: disconnect from server, usage: disconnect")
    print("exit: exit the program, usage: exit")
    return
  
```

定义了一个help命令来输出所有的支持指令以及参数

然后客户端的主要部分是对于用户命令的操作，将用户命令存储到command中，如果command的长度大于1，说明是带有文件指针的命令，将指针存储到arg下并开始建立套接字

```
client = socket(AF_INET, SOCK_STREAM)
```

开始进入while循环来循环判断处理每一条命令，主要也就是对于几个边缘状态的讨论，具体可以见附录下的代码

服务端

因为要实现多线程，所以在客户端每一次建立新的连接的时候就创建一个新的线程

```
newthd = threading.Thread(target = receive_data, args = (tcpconnection,
clientaddr, LogFile, filelock, DataPath))
newthd.start()
```

服务端的操作主要是对两个文件的维护上面，一个是日志文件，另一个为数据文件，数据文件是个大类概括，里面可以包含多个文件夹，我们这里将文件夹和文件做了区分，方便命令的实现上面能够区分，如upload中需要判断所要上传的文件名与文件夹中的文件名存不存在冲突，而与文件夹的名字不冲突

```
tcpserver = socket(AF_INET, SOCK_STREAM)
tcpserver.setsockopt(SOL_SOCKET, SO_REUSEADDR, True)
tcpserver.bind(('', port))
tcpserver.listen(128)
```

服务端也通过套接字的方式，然后监听端口，得到请求后建立连接

```
tcpconnection, clientaddr = tcpserver.accept()
with open(LogFile, 'a') as f:
    filelock.acquire()
    print("Connection with", clientaddr[0] + ':' + str(clientaddr[1]), "is
established")
    f.write("Connection with " + clientaddr[0] + ':' + str(clientaddr[1]) + " is
established\n")
    filelock.release()
```

建立连接后开始准备接受客户端的请求，接受请求后也就按照请求具体的分类通过像客户端上的操作，具体也可以见附录下的代码

### 三、实验结果

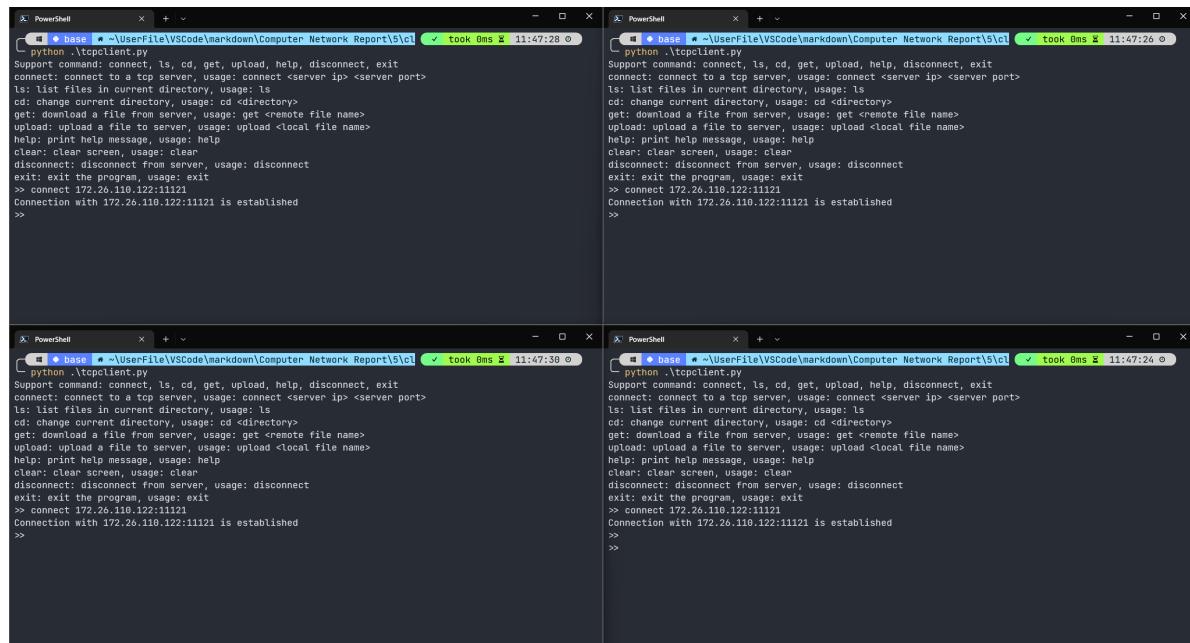
#### A 机器运行服务器软件 在客户端 B 机器的命令行同时运行 4 个客户端

首先来看两台机器的情况，A作为服务端，B作为客户端

```
C:\Windows\System32\cmd.exe - python tcpserver.py
Microsoft Windows [版本 10.0.19044.1288]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Lewis\Desktop\TCP>python tcpserver.py
Data File is Store in: Data
Server is listing on port: 11121
```

最初界面，A为服务端的界面，然后我们在B中打开四个客户端



The image shows four separate PowerShell windows, each representing a client connection to the same server. The top-left window (A) is the server, while the other three (B) are clients. Each client window displays the command being run (python .\tcpclient.py) and the server's response, which includes support commands like connect, ls, cd, get, upload, help, disconnect, and exit.

```
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>>
```

```
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>>
```

```
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>>
```

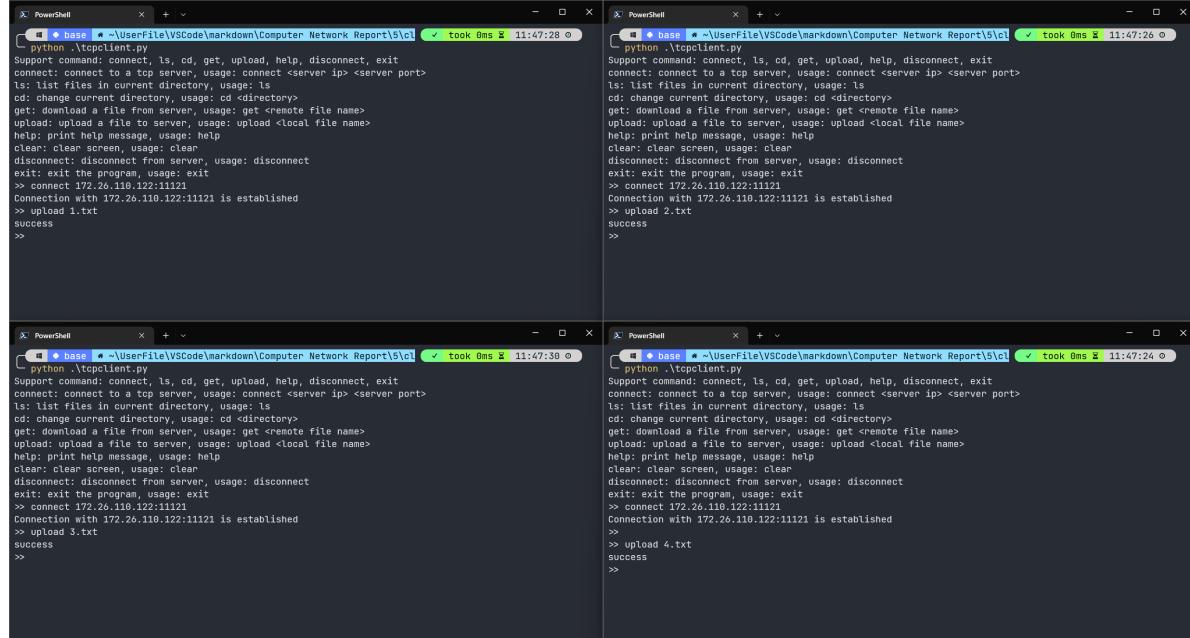
```
C:\Windows\System32\cmd.exe - python tcpserver.py
Microsoft Windows [版本 10.0.19044.1288]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Lewis\Desktop\TCP>python tcpserver.py
Data File is Store in: Data
Server is listing on port: 11121
Connection with 172.26.96.1:12559 is established
Connection with 172.26.96.1:12560 is established
Connection with 172.26.96.1:12561 is established
Connection with 172.26.96.1:12570 is established
```

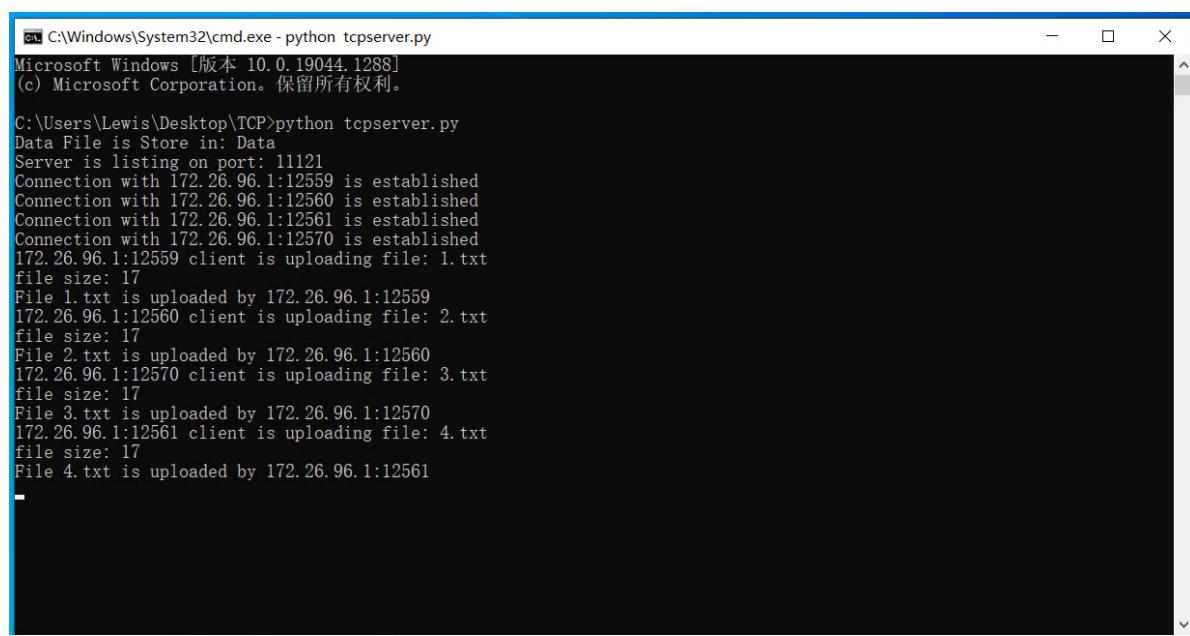
现在是打开了四个客户端你的情况，可以看到服务端界面上上述的ip地址相同，但端口不一样，说明打开并连接上了四个不同的客户端。

客户端界面上每个客户端与服务端的连接都成功。

下面看看不同客户端输入不同命令后服务端的输出



The image shows four separate PowerShell windows, each representing a different client connection to a single TCP server. The clients are running the command-line interface for the server, which includes commands like connect, ls, cd, get, upload, help, disconnect, and exit. Each window shows a sequence of these commands being entered and their corresponding responses from the server. The clients are connected to the server at 172.26.110.122:11121.

A single cmd.exe window titled 'C:\Windows\System32\cmd.exe - python tcpserver.py' displays the log of the TCP server. It shows multiple client connections being established at port 11121. The server logs the file names being uploaded by each client, such as '1.txt', '2.txt', '3.txt', and '4.txt'. The server also indicates the file size for each upload.

```
C:\Windows\System32\cmd.exe - python tcpserver.py
Microsoft Windows [版本 10.0.19044.1288]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Lewis\Desktop\TCP>python tcpserver.py
Data File is Store in: Data
Server is listing on port: 11121
Connection with 172.26.96.1:12559 is established
Connection with 172.26.96.1:12560 is established
Connection with 172.26.96.1:12561 is established
Connection with 172.26.96.1:12570 is established
172.26.96.1:12559 client is uploading file: 1.txt
file size: 17
File 1.txt is uploaded by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 2.txt
file size: 17
File 2.txt is uploaded by 172.26.96.1:12560
172.26.96.1:12570 client is uploading file: 3.txt
file size: 17
File 3.txt is uploaded by 172.26.96.1:12570
172.26.96.1:12561 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12561
```

upload：现在尝试从四个不同的客户端往服务端来upload文件，分别设置为1、2、3、4，见上图，输出了文件大小并可以从data文件夹下发现上述文件，并且客户端界面上所有的文件都上传成功传回success信息

```

PowerShell
help; print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 1.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>> get ServerMsg.txt
File ServerMsg.txt received
>> mkdir test
Directory created
>> rm 4.txt
Are you sure to remove the file 4.txt?(y/n)
Removed
>>

PowerShell
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a top server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 2.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>>

PowerShell
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a top server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 3.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>>

```

```

C:\Windows\System32\cmd.exe - python tcpserver.py
Microsoft Windows [版本 10.0.19044.128]
(c) Microsoft Corporation。保留所有权利。
C:\Users\Lewis\Desktop\TCP>python tcpserver.py
Data File is Store in: Data
Server is listing on port: 11121
Connection with 172.26.96.1:12559 is established
Connection with 172.26.96.1:12560 is established
Connection with 172.26.96.1:12561 is established
Connection with 172.26.96.1:12570 is established
172.26.96.1:12559 client is uploading file: 1.txt
file size: 17
File 1.txt is uploaded by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 2.txt
file size: 17
File 2.txt is uploaded by 172.26.96.1:12560
172.26.96.1:12570 client is uploading file: 3.txt
file size: 17
File 3.txt is uploaded by 172.26.96.1:12570
172.26.96.1:12561 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12561
File ServerMsg.txt is downloaded by 172.26.96.1:12559
Directory test is created by 172.26.96.1:12559
4.txt is removed by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12560

```

rm: 服务端: 172.26.96.1:12559将服务端的4.txt移除，客户端: rm命令后传回success信息

```

PowerShell
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 1.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>> get ServerMsg.txt
File ServerMsg.txt received
>> mkdir test
Directory created
>>

PowerShell
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a top server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 2.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>>

PowerShell
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a top server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 3.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>>

```

```
C:\Windows\System32\cmd.exe - python tcpserver.py
Microsoft Windows [版本 10.0.19044.1288]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Lewis\Desktop\TCP>python tcpserver.py
Data File is Store in: Data
Server is listing on port: 11121
Connection with 172.26.96.1:12559 is established
Connection with 172.26.96.1:12560 is established
Connection with 172.26.96.1:12561 is established
Connection with 172.26.96.1:12570 is established
172.26.96.1:12559 client is uploading file: 1.txt
file size: 17
File 1.txt is uploaded by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 2.txt
file size: 17
File 2.txt is uploaded by 172.26.96.1:12560
172.26.96.1:12570 client is uploading file: 3.txt
file size: 17
File 3.txt is uploaded by 172.26.96.1:12570
172.26.96.1:12561 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12561
File ServerMsg.txt is downloaded by 172.26.96.1:12559
Directory test is created by 172.26.96.1:12559
```

mkdir: 创建文件夹命令。客户端：输入命令，返回created信息；服务端：文件夹创建并输出创建文件夹的客户端地址

```
PowerShell ┌─┐ + ─ ─
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 1.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>> get ServerMsg.txt
File ServerMsg.txt received
>>

PowerShell ┌─┐ + ─ ─
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 2.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>>

PowerShell ┌─┐ + ─ ─
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 3.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>>

PowerShell ┌─┐ + ─ ─
python .\tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 4.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt    file
>>
```

```
C:\Windows\System32\cmd.exe - python tcpserver.py
Microsoft Windows [版本 10.0.19044.1288]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Lewis\Desktop\TCP>python tcpserver.py
Data File is Store in: Data
Server is listing on port: 11121
Connection with 172.26.96.1:12559 is established
Connection with 172.26.96.1:12560 is established
Connection with 172.26.96.1:12561 is established
Connection with 172.26.96.1:12570 is established
172.26.96.1:12559 client is uploading file: 1.txt
file size: 17
File 1.txt is uploaded by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 2.txt
file size: 17
File 2.txt is uploaded by 172.26.96.1:12560
172.26.96.1:12570 client is uploading file: 3.txt
file size: 17
File 3.txt is uploaded by 172.26.96.1:12570
172.26.96.1:12561 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12561
File ServerMsg.txt is downloaded by 172.26.96.1:12559
```

get: 客户端：输入get serverMsg.txt 的命令，返回receive信息；服务端：输出get的客户端地址

```
Client 1 (Top Left): python .\tcpclient.py
Client 2 (Top Right): python .\tcpclient.py
Client 3 (Bottom Left): python .\tcpclient.py
Client 4 (Bottom Right): python .\tcpclient.py
Server (Center): python .\tcpserver.py
```

Client 1 (Top Left) Session:

```
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 1.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>>
```

Client 2 (Top Right) Session:

```
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 2.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>>
```

Client 3 (Bottom Left) Session:

```
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 3.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>>
```

Client 4 (Bottom Right) Session:

```
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 4.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>>
```

Server (Center) Session:

```
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 2.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>>
```

Client 1 (Bottom Left) Session (After Upload):

```
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 1.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>> get ServerMsg.txt
File ServerMsg.txt received
>> mkdir test
Directory created
>> rm 4.txt
Are you sure to remove the file 4.txt?(y/n)
Removed
>>
```

Client 2 (Bottom Right) Session (After Upload):

```
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 2.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>> is
test    dir
1.txt    file
2.txt    file
3.txt    file
ServerMsg.txt   file
>>
```

Client 3 (Bottom Left) Session (After Upload):

```
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 3.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>> ls
test    dir
1.txt    file
2.txt    file
3.txt    file
ServerMsg.txt   file
>>
```

Client 4 (Bottom Right) Session (After Upload):

```
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect 172.26.110.122:11121
Connection with 172.26.110.122:11121 is established
>> upload 4.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt   file
>>
```

ls: 这里主要的是两次的ls，第一次是初始的时候四个客户端向服务端传了四个文件，通过ls命令可以看到里面有5个txt文件，第二次是通过rm操作后删除了4.txt，通过ls命令可以看到此时只有4个txt文件了，并刚好删除了4.txt

```

PowerShell session 1 (Client 1):
>> upload 1.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt  file
>> get ServerMsg.txt
File ServerMsg.txt received
>> mkdir test
Directory created
>> rm 4.txt
Are you sure to remove the file 4.txt?(y/n)
Removed
>> ls
Connection closed

PowerShell session 2 (Client 2):
>> upload 2.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt  file
>> cd test
Directory changed
>> upload 4.txt
success
>> ls
Connection closed

PowerShell session 3 (Client 3):
>> upload 3.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt  file
>> ls
test    dir
1.txt    file
2.txt    file
3.txt    file
ServerMsg.txt  file
>> cd test
Directory changed
>> rm 4.txt
>> exit
Connection closed

PowerShell session 4 (Client 4):
>> upload 4.txt
success
>> ls
1.txt    file
2.txt    file
3.txt    file
4.txt    file
ServerMsg.txt  file
>> ls
test    dir
1.txt    file
2.txt    file
3.txt    file
ServerMsg.txt  file
>> exit
Connection closed

```

```

C:\Windows\System32\cmd.exe - python tcpserver.py
C:\Users\Lewis\Desktop\TCP\python tcpserver.py
Data File is Store in: Data
Server is listing on port: 11121
Connection with 172.26.96.1:12559 is established
Connection with 172.26.96.1:12560 is established
Connection with 172.26.96.1:12561 is established
Connection with 172.26.96.1:12570 is established
172.26.96.1:12559 client is uploading file: 1.txt
file size: 17
File 1.txt is uploaded by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 2.txt
file size: 17
File 2.txt is uploaded by 172.26.96.1:12560
172.26.96.1:12570 client is uploading file: 3.txt
file size: 17
File 3.txt is uploaded by 172.26.96.1:12570
172.26.96.1:12561 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12561
File ServerMsg.txt is downloaded by 172.26.96.1:12559
Directory test is created by 172.26.96.1:12559
4.txt is removed by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12560
Connection with 172.26.96.1:12559 is closed
Connection with 172.26.96.1:12560 is closed
Connection with 172.26.96.1:12570 is closed
Connection with 172.26.96.1:12561 is closed

```

close: 四个客户端关闭连接，返回closed，服务端中显示了具体的ip地址以及关闭信息

## 在 A 机器也同时运行客户端软件 2 个客户端

```

C:\Windows\System32\cmd.exe - python tcpclient.py
Microsoft Windows [版本 10.0.19044.1288]
(c) Microsoft Corporation. 保留所有权利。
C:\Users\Lewis\Desktop\TCP\python tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect, exit
connect: connect to a tcp server, usage: connect <server ip> <server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect localhost:11121
Connection with localhost:11121 is established

```

在A机器上同时运行了两个客户端

```

C:\Windows\System32\cmd.exe - python tcpclient.py
Microsoft Windows [版本 10.0.19044.1288]
(c) Microsoft Corporation. 保留所有权利。
C:\Users\Lewis\Desktop\TCP\python tcpclient.py
Support command: connect, ls, cd, get, upload, help, disconnect
connect: connect to a tcp server, usage: connect <server ip><server port>
ls: list files in current directory, usage: ls
cd: change current directory, usage: cd <directory>
get: download a file from server, usage: get <remote file name>
upload: upload a file to server, usage: upload <local file name>
help: print help message, usage: help
clear: clear screen, usage: clear
disconnect: disconnect from server, usage: disconnect
exit: exit the program, usage: exit
>> connect localhost:11121
Connection with localhost:11121 is established
>> upload localhost_test.txt
success
>> ls
test dir
1.txt file
2.txt file
3.txt file
localhost test.txt file
ServerMsg.txt file
>> -

```

然后在A上的两个客户端一个运行upload命令，另一个通过ls命令发现第一个客户端所上传的test.txt也在第二个客户端上显示

```

C:\Windows\System32\cmd.exe - python tcpserver.py
Server is listing on port: 11121
Connection with 172.26.96.1:12559 is established
Connection with 172.26.96.1:12560 is established
Connection with 172.26.96.1:12561 is established
Connection with 172.26.96.1:12570 is established
172.26.96.1:12559 client is uploading file: 1.txt
file size: 17
File 1.txt is uploaded by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 2.txt
file size: 17
File 2.txt is uploaded by 172.26.96.1:12560
172.26.96.1:12570 client is uploading file: 3.txt
file size: 17
File 3.txt is uploaded by 172.26.96.1:12570
172.26.96.1:12561 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12561
File ServerMsg.txt is downloaded by 172.26.96.1:12559
Directory test is created by 172.26.96.1:12559
4.txt is removed by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12560
Connection with 172.26.96.1:12559 is closed
Connection with 172.26.96.1:12560 is closed
Connection with 172.26.96.1:12570 is closed
Connection with 172.26.96.1:12561 is closed
Connection with 127.0.0.1:50561 is established
Connection with 127.0.0.1:50562 is established

```

在A上同时运行服务端，可以看到在完成了第一个实验后连接关闭，然后又与自己机器上打开的客户端建立连接

```

C:\Windows\System32\cmd.exe - python tcpserver.py
Connection with 172.26.96.1:12561 is established
Connection with 172.26.96.1:12570 is established
172.26.96.1:12559 client is uploading file: 1.txt
file size: 17
File 1.txt is uploaded by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 2.txt
file size: 17
File 2.txt is uploaded by 172.26.96.1:12560
172.26.96.1:12570 client is uploading file: 3.txt
file size: 17
File 3.txt is uploaded by 172.26.96.1:12570
172.26.96.1:12561 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12561
File ServerMsg.txt is downloaded by 172.26.96.1:12559
Directory test is created by 172.26.96.1:12559
4.txt is removed by 172.26.96.1:12559
172.26.96.1:12560 client is uploading file: 4.txt
file size: 17
File 4.txt is uploaded by 172.26.96.1:12560
Connection with 172.26.96.1:12559 is closed
Connection with 172.26.96.1:12560 is closed
Connection with 172.26.96.1:12570 is closed
Connection with 172.26.96.1:12561 is closed
Connection with 127.0.0.1:50561 is established
Connection with 127.0.0.1:50562 is established
127.0.0.1:50562 client is uploading file: localhost_test.txt
file size: 13
File localhost_test.txt is uploaded by 127.0.0.1:50562

```

完成了upload以及ls命令后的A的服务端所输出的页面，可以看到一个客户端完成了upload文件localhost\_test.txt，实验结束

## **观看服务器端运行的日志**

日志信息放在server.log文件中，与实验报告一同上交。