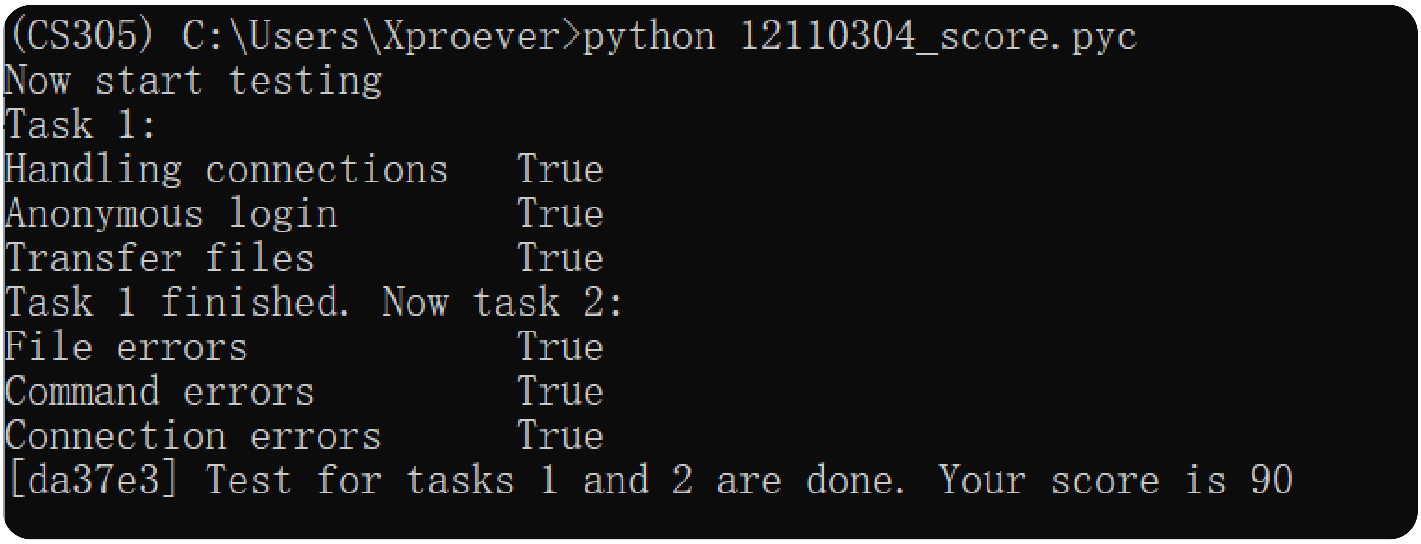
CS305 2023Spring Programing Assignment 1

12110304 徐春晖

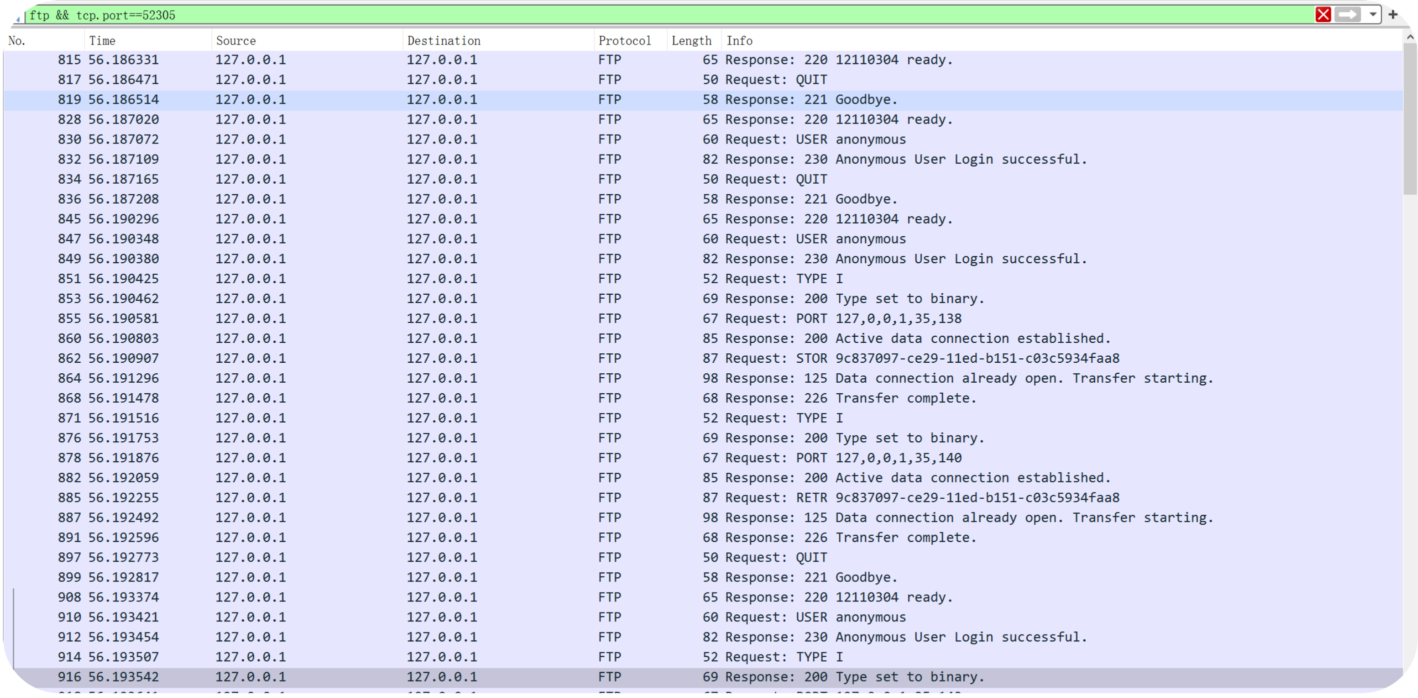
Basic Task：

脚本评分：

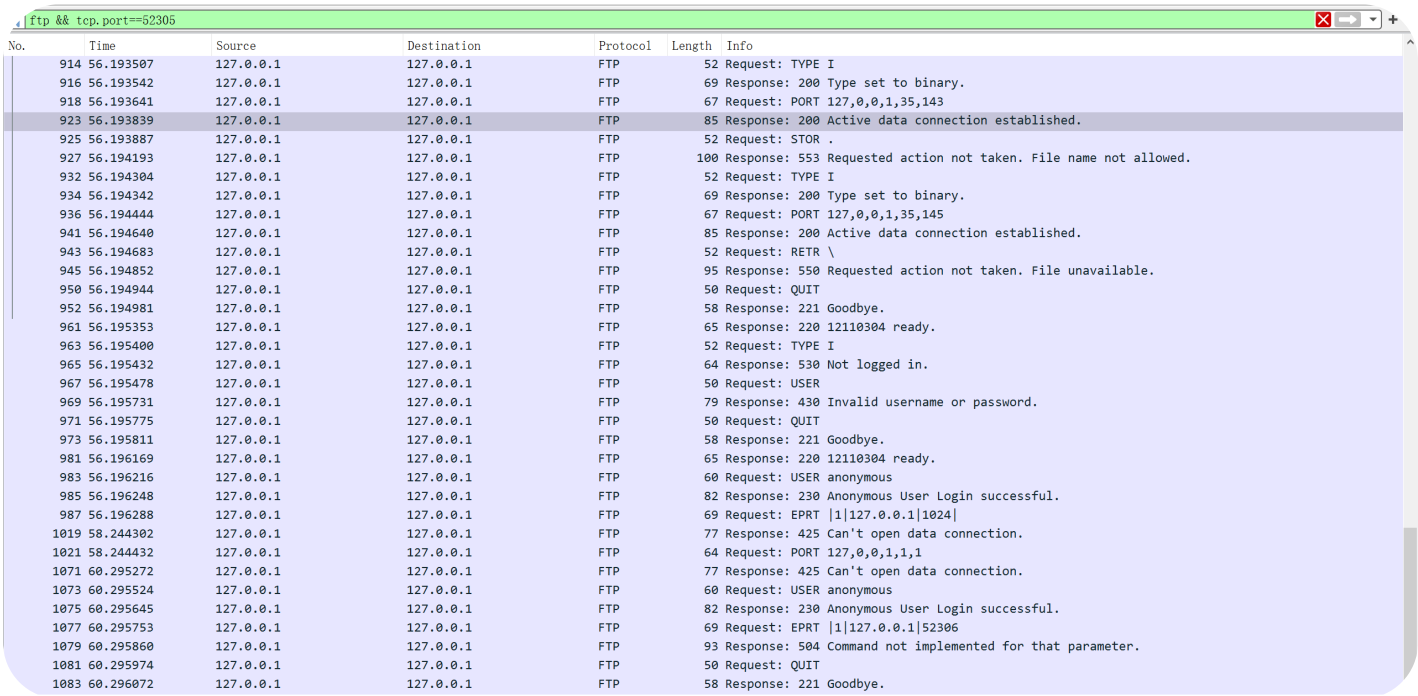


```
(CS305) C:\Users\Xproever>python 12110304_score.pyc
Now start testing
Task 1:
Handling connections    True
Anonymous login         True
Transfer files          True
Task 1 finished. Now task 2:
File errors             True
Command errors          True
Connection errors       True
[da37e3] Test for tasks 1 and 2 are done. Your score is 90
```

Wireshark 抓包：



```
ftp && tcp.port==52305
No.     Time       Source       Destination  Protocol  Length Info
    815 56.186331  127.0.0.1    127.0.0.1    FTP       65 Response: 220 12110304 ready.
    817 56.186471  127.0.0.1    127.0.0.1    FTP       50 Request: QUIT
    819 56.186514  127.0.0.1    127.0.0.1    FTP       58 Response: 221 Goodbye.
    828 56.187020  127.0.0.1    127.0.0.1    FTP       65 Response: 220 12110304 ready.
    830 56.187072  127.0.0.1    127.0.0.1    FTP       60 Request: USER anonymous
    832 56.187109  127.0.0.1    127.0.0.1    FTP       82 Response: 230 Anonymous User Login successful.
    834 56.187165  127.0.0.1    127.0.0.1    FTP       50 Request: QUIT
    836 56.187208  127.0.0.1    127.0.0.1    FTP       58 Response: 221 Goodbye.
    845 56.190296  127.0.0.1    127.0.0.1    FTP       65 Response: 220 12110304 ready.
    847 56.190348  127.0.0.1    127.0.0.1    FTP       60 Request: USER anonymous
    849 56.190380  127.0.0.1    127.0.0.1    FTP       82 Response: 230 Anonymous User Login successful.
    851 56.190425  127.0.0.1    127.0.0.1    FTP       52 Request: TYPE I
    853 56.190462  127.0.0.1    127.0.0.1    FTP       69 Response: 200 Type set to binary.
    855 56.190581  127.0.0.1    127.0.0.1    FTP       67 Request: PORT 127,0,0,1,35,138
    860 56.190803  127.0.0.1    127.0.0.1    FTP       85 Response: 200 Active data connection established.
    862 56.190907  127.0.0.1    127.0.0.1    FTP       87 Request: STOR 9c837097-ce29-11ed-b151-c03c5934faa8
    864 56.191296  127.0.0.1    127.0.0.1    FTP       98 Response: 125 Data connection already open. Transfer starting.
    868 56.191478  127.0.0.1    127.0.0.1    FTP       68 Response: 226 Transfer complete.
    871 56.191516  127.0.0.1    127.0.0.1    FTP       52 Request: TYPE I
    876 56.191753  127.0.0.1    127.0.0.1    FTP       69 Response: 200 Type set to binary.
    878 56.191876  127.0.0.1    127.0.0.1    FTP       67 Request: PORT 127,0,0,1,35,140
    882 56.192059  127.0.0.1    127.0.0.1    FTP       85 Response: 200 Active data connection established.
    885 56.192255  127.0.0.1    127.0.0.1    FTP       87 Request: RETR 9c837097-ce29-11ed-b151-c03c5934faa8
    887 56.192492  127.0.0.1    127.0.0.1    FTP       98 Response: 125 Data connection already open. Transfer starting.
    891 56.192596  127.0.0.1    127.0.0.1    FTP       68 Response: 226 Transfer complete.
    897 56.192773  127.0.0.1    127.0.0.1    FTP       50 Request: QUIT
    899 56.192817  127.0.0.1    127.0.0.1    FTP       58 Response: 221 Goodbye.
    908 56.193374  127.0.0.1    127.0.0.1    FTP       65 Response: 220 12110304 ready.
    910 56.193421  127.0.0.1    127.0.0.1    FTP       60 Request: USER anonymous
    912 56.193454  127.0.0.1    127.0.0.1    FTP       82 Response: 230 Anonymous User Login successful.
    914 56.193507  127.0.0.1    127.0.0.1    FTP       52 Request: TYPE I
    916 56.193542  127.0.0.1    127.0.0.1    FTP       69 Response: 200 Type set to binary.
```



ftp && tcp.port==52305

No.	Time	Source	Destination	Protocol	Length	Info
914	56.193507	127.0.0.1	127.0.0.1	FTP	52	Request: TYPE I
916	56.193542	127.0.0.1	127.0.0.1	FTP	69	Response: 200 Type set to binary.
918	56.193641	127.0.0.1	127.0.0.1	FTP	67	Request: PORT 127,0,0,1,35,143
923	56.193839	127.0.0.1	127.0.0.1	FTP	85	Response: 200 Active data connection established.
925	56.193887	127.0.0.1	127.0.0.1	FTP	52	Request: STOR .
927	56.194193	127.0.0.1	127.0.0.1	FTP	100	Response: 553 Requested action not taken. File name not allowed.
932	56.194304	127.0.0.1	127.0.0.1	FTP	52	Request: TYPE I
934	56.194342	127.0.0.1	127.0.0.1	FTP	69	Response: 200 Type set to binary.
936	56.194444	127.0.0.1	127.0.0.1	FTP	67	Request: PORT 127,0,0,1,35,145
941	56.194640	127.0.0.1	127.0.0.1	FTP	85	Response: 200 Active data connection established.
943	56.194683	127.0.0.1	127.0.0.1	FTP	52	Request: RETR \
945	56.194852	127.0.0.1	127.0.0.1	FTP	95	Response: 550 Requested action not taken. File unavailable.
950	56.194944	127.0.0.1	127.0.0.1	FTP	50	Request: QUIT
952	56.194981	127.0.0.1	127.0.0.1	FTP	58	Response: 221 Goodbye.
961	56.195353	127.0.0.1	127.0.0.1	FTP	65	Response: 220 12110304 ready.
963	56.195400	127.0.0.1	127.0.0.1	FTP	52	Request: TYPE I
965	56.195432	127.0.0.1	127.0.0.1	FTP	64	Response: 530 Not logged in.
967	56.195478	127.0.0.1	127.0.0.1	FTP	50	Request: USER
969	56.195731	127.0.0.1	127.0.0.1	FTP	79	Response: 430 Invalid username or password.
971	56.195775	127.0.0.1	127.0.0.1	FTP	50	Request: QUIT
973	56.195811	127.0.0.1	127.0.0.1	FTP	58	Response: 221 Goodbye.
981	56.196169	127.0.0.1	127.0.0.1	FTP	65	Response: 220 12110304 ready.
983	56.196216	127.0.0.1	127.0.0.1	FTP	60	Request: USER anonymous
985	56.196248	127.0.0.1	127.0.0.1	FTP	82	Response: 230 Anonymous User Login successful.
987	56.196288	127.0.0.1	127.0.0.1	FTP	69	Request: EPRT |1|127.0.0.1|1024|
1019	58.244302	127.0.0.1	127.0.0.1	FTP	77	Response: 425 Can't open data connection.
1021	58.244432	127.0.0.1	127.0.0.1	FTP	64	Request: PORT 127,0,0,1,1,1
1071	60.295272	127.0.0.1	127.0.0.1	FTP	77	Response: 425 Can't open data connection.
1073	60.295524	127.0.0.1	127.0.0.1	FTP	60	Request: USER anonymous
1075	60.295645	127.0.0.1	127.0.0.1	FTP	82	Response: 230 Anonymous User Login successful.
1077	60.295753	127.0.0.1	127.0.0.1	FTP	69	Request: EPRT |1|127.0.0.1|52306
1079	60.295860	127.0.0.1	127.0.0.1	FTP	93	Response: 504 Command not implemented for that parameter.
1081	60.295974	127.0.0.1	127.0.0.1	FTP	50	Request: QUIT
1083	60.296072	127.0.0.1	127.0.0.1	FTP	58	Response: 221 Goodbye.

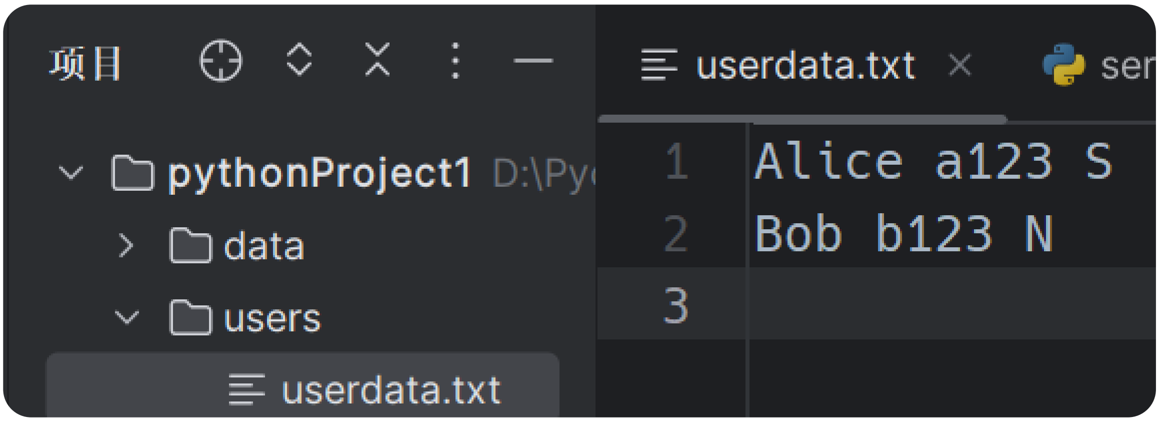
Other Task：

User login control (5 pts):

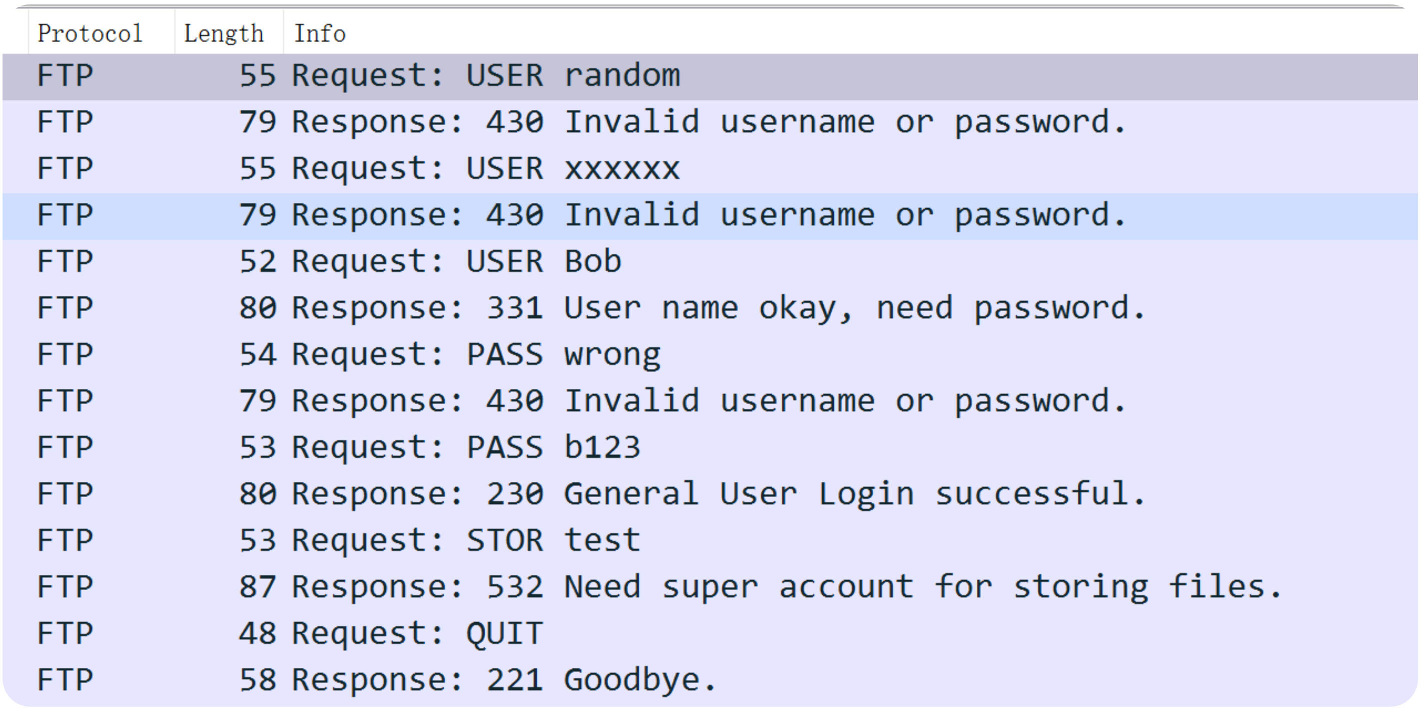
User privilege control (5 pts):

功能展示：

`Bob` 和 `Alice` 为储存在数据库中的用户，分别为普通用户和超级用户，正确密码为 `b123` ， `a123` 。



```
Alice a123 S
Bob b123 N
```



Protocol	Length	Info
FTP	55	Request: USER random
FTP	79	Response: 430 Invalid username or password.
FTP	55	Request: USER xxxxxx
FTP	79	Response: 430 Invalid username or password.
FTP	52	Request: USER Bob
FTP	80	Response: 331 User name okay, need password.
FTP	54	Request: PASS wrong
FTP	79	Response: 430 Invalid username or password.
FTP	53	Request: PASS b123
FTP	80	Response: 230 General User Login successful.
FTP	53	Request: STOR test
FTP	87	Response: 532 Need super account for storing files.
FTP	48	Request: QUIT
FTP	58	Response: 221 Goodbye.

前两次为随意输入而不在数据库中的用户名，第三次查找到用户名并返回 `331`。

然后是一次错误的密码尝试，返回 `430` ，接着是一次正确的密码传输，通过登录并识别为 General User。

接着运行指令，然而作为普通用户的 `Bob` 没有 `STOR` 功能的访问权限。（为了方便演示没有进行 `PORT` 等数据传输接口的操作，下同）

super_user

在这里，作为超级用户的 `Alice` 被识别为 Super User，并将开放所有权限。

实现思路：

下面结合代码注释阐述实现

```python
if line[:4] == "USER":
    try:
        u_name = line[5:]
        if u_name == "anonymous": # 匿名用户无需输入密码
            client.send(b"230 Anonymous User Login successful.\r\n")
            is_login = True
            is_super = True # 默认为超级用户
        else:
            for line in open("users/userdata.txt"): # 遍历 data 文件
                ud = line.split(' ') # 处理原始数据
                if ud[0] == u_name: # 如果用户名匹配成功
                    client.send(b"331 User name okay, need password.\r\n")
                    user = ud # 记录下用户信息，等待密码匹配
                    on_login = True # 准备登录状态
                    break

            if not on_login: # 遍历完成而没有匹配上
                client.send(b"430 Invalid username or password.\r\n")

    except IndexError:
        client.send(b"430 Invalid username or password.\r\n")

elif on_login and line[:4] == "PASS": # 处于准备登录状态并尝试传输密码
    if line[5:] == user[1]: # 密码匹配成功
```

```
        is_login = True  # 成功登录
        if user[2] == 'S\n': # 如过数据库中记录的为超级用户
            is_super = True # 记录超级用户权限
            client.send(b"230 Super User Login successful.\r\n")
        else: # 否则为普通用户
            is_super = False # 记录普通用户权限
            client.send(b"230 General User Login successful.\r\n")

    else: #密码错误
        client.send(b"430 Invalid username or password.\r\n")

else:
    client.send(b"530 Not logged in.\r\n")

```

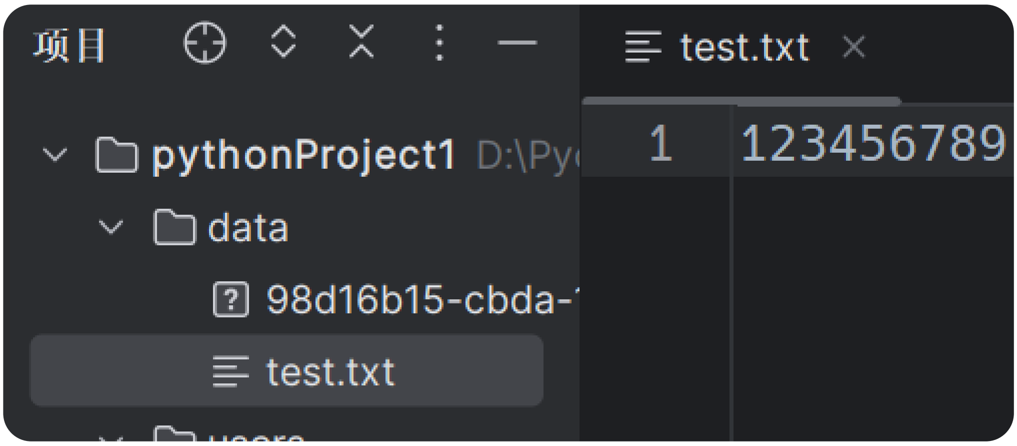
STOR 部分：

```
elif line[:4] == "STOR":
    filename = line[5:]
    if is_super:
        # 正常运行功能
    else: # 无权限
        client.send(b"532 Need super account for storing files.\r\n")
```

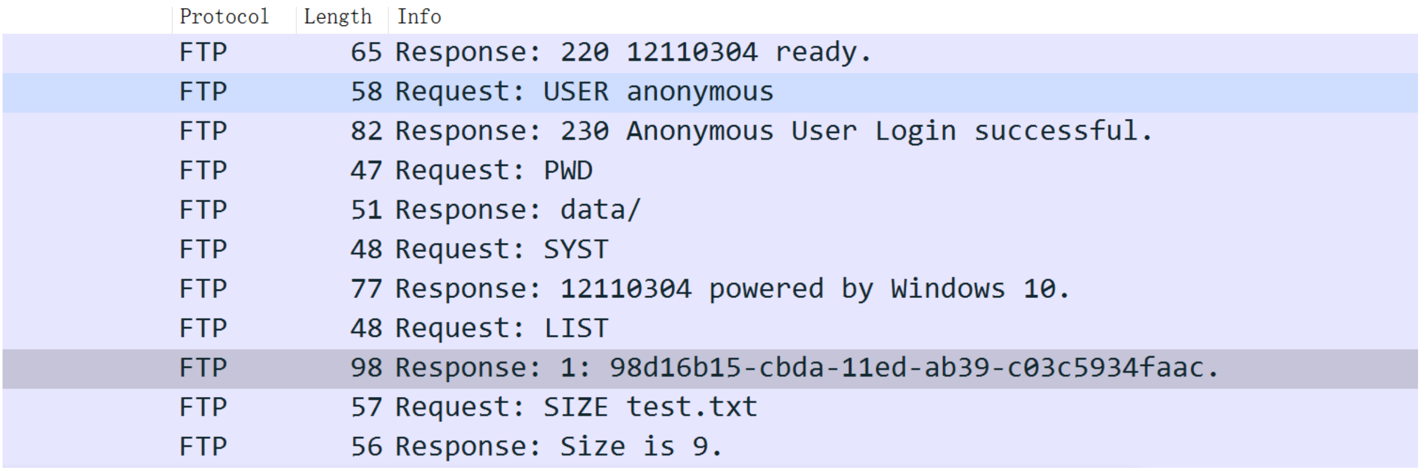
More commands (include `LIST`, `SIZE`, `SYST`, `PWD`, 2 pts each):

功能展示：

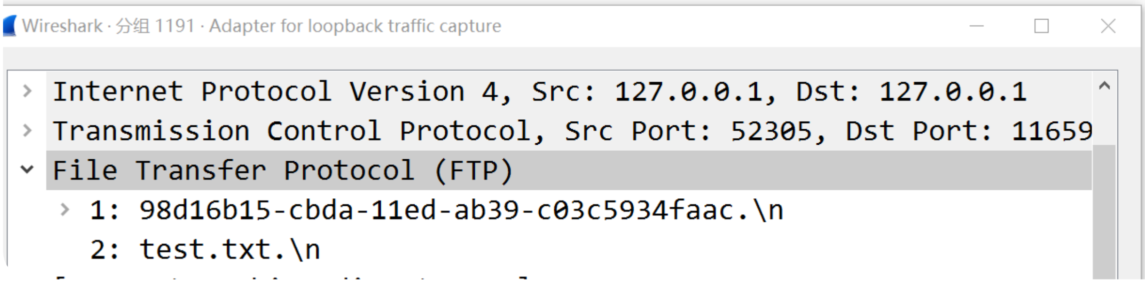
原始库里有两个文件，其中一个为手动编辑的 `text.txt` ，内容如图所示。



下图为四个命令的测试内容



```
Protocol  Length Info
FTP           65 Response: 220 12110304 ready.
FTP           58 Request: USER anonymous
FTP           82 Response: 230 Anonymous User Login successful.
FTP           47 Request: PWD
FTP           51 Response: data/
FTP           48 Request: SYST
FTP           77 Response: 12110304 powered by Windows 10.
FTP           48 Request: LIST
FTP           98 Response: 1: 98d16b15-cbda-11ed-ab39-c03c5934faac.
FTP           57 Request: SIZE test.txt
FTP           56 Response: Size is 9.
```



Wireshark · 分组 1191 · Adapter for loopback traffic capture

```
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 52305, Dst Port: 11659
v File Transfer Protocol (FTP)
  > 1: 98d16b15-cbda-11ed-ab39-c03c5934faac.\n
    2: test.txt.\n
```

下框为 `LIST` 命令的完整 Response

实现思路：

下面结合代码注释阐述实现

```python
elif line[:4] == "SIZE":
    filename = line[5:]

    try:
        with open(work_path + filename, 'rb') as f: # 打开文件
            size = f.seek(0, os.SEEK_END) # 扫描到文件底部
            client.send(b"Size is %d.\r\n" % size) # 返回文件大小
    except FileNotFoundError:
        client.send(b"550 Requested action not taken. File unavailable.\r\n")

elif line[:4] == "SYST":
    client.send(b"12110304 powered by Windows 10.\r\n") # 返回固定文本

elif line[:3] == "PWD":
    client.send((work_path + '\r\n').encode()) # 返回当前工作的目录

elif line[:4] == "LIST":
    cnt = 1
    list_data = '' # 准备传输的字符串
    for file_name in os.listdir(work_path): # 循环工作目录下的每一个文件
        list_data += "%d: %s.\n" % (cnt, file_name) # 返回的字符串内容增加
        cnt += 1 # 序号 + 1

    client.send(list_data.encode()) # 字符串编码为字节流

```