Networks Assignment 2a

Josue Soto
80631830

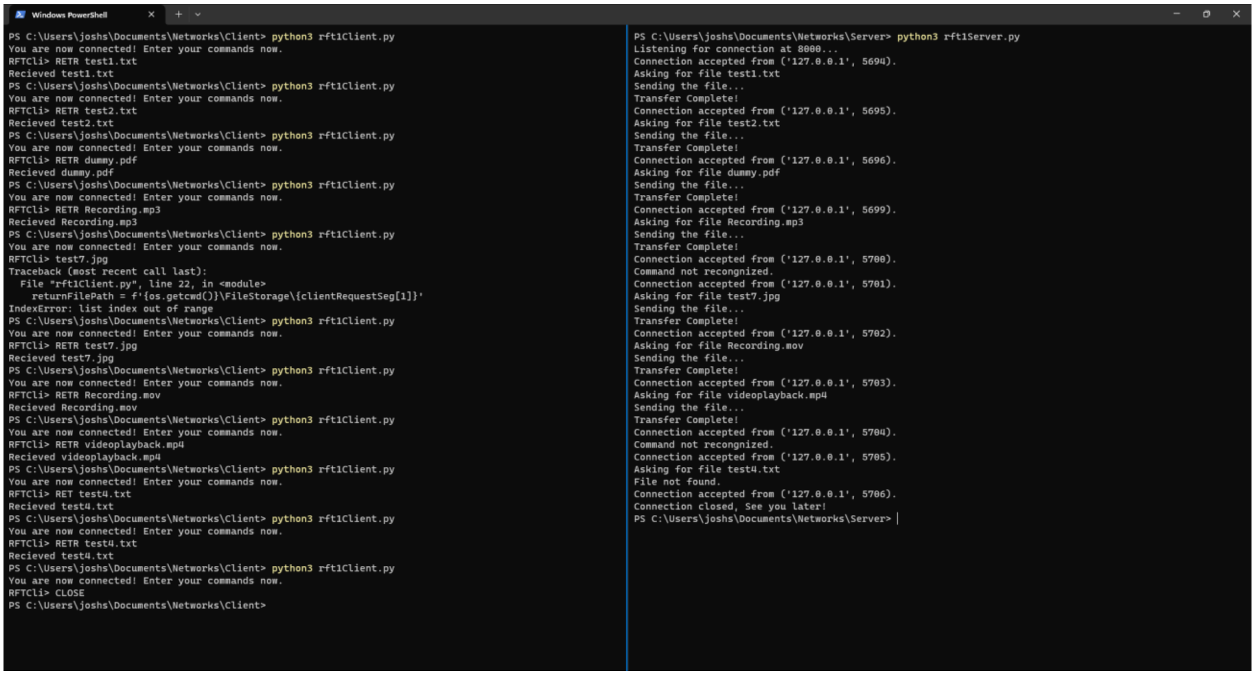
1. How I created a file transfer protocol.

Starting off I created a simple echo server and client with to act as a foundation for the system. Afterwards I set up the framework of requesting a file from the server and checking if that file was in the server's folder. Once I could determine that a file existed on the server side I began starting to implement a way for the server to send the bytes to the client. This was also modified by the fact that each packet was only allowed to be 1000 bytes in length. After some testing, I discovered that it is best to create a packets list and append in windows of 1000 bytes. I would figure out the number of packets needed by floor division of the file size, taking its result and adding one if there was any bytes left. Once I had the number of packets needed I had to just window the actual bytes into the packets list. This was done by iterating through the range of the number of packets needed and appending a slice of the file 1000 bytes at a time. Once there is < 1000 bytes left in the file then we add another packet containing the remaining data. Once the packet list is done I would continuously send every packet in the list to the client back to back. The client would then write the packets continuously into a file with the same name and stop once a packet was detected to be less than 1000 bytes.

An edge case I ran into was if the file is evenly divisible by 1000 bytes. If this was the case, I decided to additionally send a bytes message containing b"!" to signify that the file had ended. Of course, if the client request was not formatted correctly and/or the requested command or file did not exist then the server would not send any file data and only the b"!" EOF to signify that no file was sent.

2. Execution samples

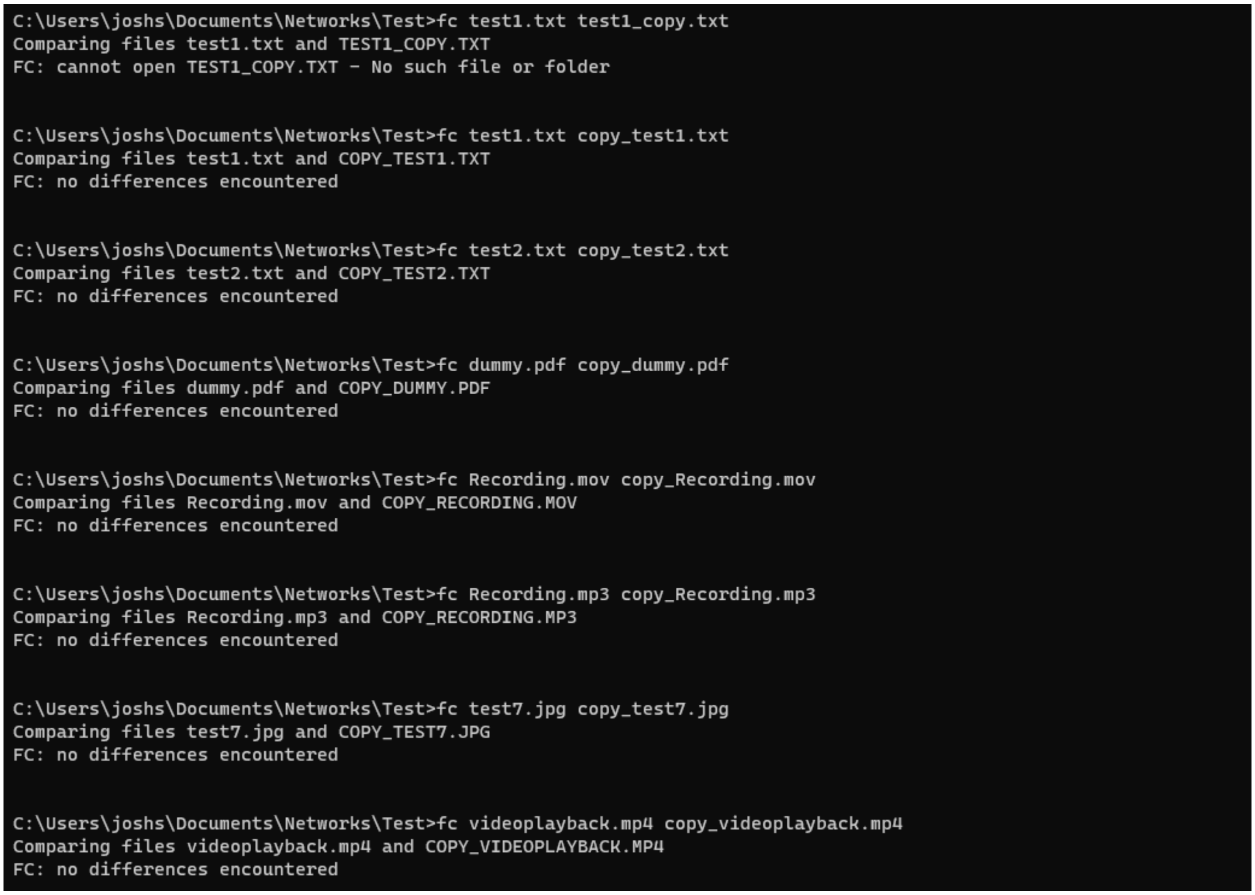
In the below example, I tested through 6 different file types established in the assignment paper. I also gave it bad instructions such as unrecognized commands, invalid commands, and files that aren't found.



```
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RETR test1.txt
Recieved test1.txt
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RETR test2.txt
Recieved test2.txt
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RETR dummy.pdf
Recieved dummy.pdf
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RETR Recording.mp3
Recieved Recording.mp3
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> test7.jpg
Traceback (most recent call last):
  File "rft1Client.py", line 22, in <module>
    returnFilePath = f'{os.getcwd()}\FileStorage\{clientRequestSeg[1]}'
IndexError: list index out of range
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RETR test7.jpg
Recieved test7.jpg
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RETR Recording.mov
Recieved Recording.mov
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RETR videoplayback.mp4
Recieved videoplayback.mp4
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RET test4.txt
Recieved test4.txt
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> RETR test4.txt
Recieved test4.txt
PS C:\Users\joshs\Documents\Networks\Client> python3 rft1Client.py
You are now connected! Enter your commands now.
RFTCli> CLOSE
PS C:\Users\joshs\Documents\Networks\Client>
```

```
PS C:\Users\joshs\Documents\Networks\Server> python3 rft1Server.py
Listening for connection at 8000...
Connection accepted from ('127.0.0.1', 5694).
Asking for file test1.txt
Sending the file...
Transfer Complete!
Connection accepted from ('127.0.0.1', 5695).
Asking for file test2.txt
Sending the file...
Transfer Complete!
Connection accepted from ('127.0.0.1', 5696).
Asking for file dummy.pdf
Sending the file...
Transfer Complete!
Connection accepted from ('127.0.0.1', 5699).
Asking for file Recording.mp3
Sending the file...
Transfer Complete!
Connection accepted from ('127.0.0.1', 5700).
Command not recongnized.
Connection accepted from ('127.0.0.1', 5701).
Asking for file test7.jpg
Sending the file...
Transfer Complete!
Connection accepted from ('127.0.0.1', 5702).
Asking for file Recording.mov
Sending the file...
Transfer Complete!
Connection accepted from ('127.0.0.1', 5703).
Asking for file videoplayback.mp4
Sending the file...
Transfer Complete!
Connection accepted from ('127.0.0.1', 5704).
Command not recongnized.
Connection accepted from ('127.0.0.1', 5705).
Asking for file test4.txt
File not found.
Connection accepted from ('127.0.0.1', 5706).
Connection closed, See you later!
PS C:\Users\joshs\Documents\Networks\Server> |
```

fc command with files.



```
C:\Users\joshs\Documents\Networks\Test>fc test1.txt test1_copy.txt
Comparing files test1.txt and TEST1_COPY.TXT
FC: cannot open TEST1_COPY.TXT - No such file or folder


C:\Users\joshs\Documents\Networks\Test>fc test1.txt copy_test1.txt
Comparing files test1.txt and COPY_TEST1.TXT
FC: no differences encountered


C:\Users\joshs\Documents\Networks\Test>fc test2.txt copy_test2.txt
Comparing files test2.txt and COPY_TEST2.TXT
FC: no differences encountered


C:\Users\joshs\Documents\Networks\Test>fc dummy.pdf copy_dummy.pdf
Comparing files dummy.pdf and COPY_DUMMY.PDF
FC: no differences encountered


C:\Users\joshs\Documents\Networks\Test>fc Recording.mov copy_Recording.mov
Comparing files Recording.mov and COPY_RECORDING.MOV
FC: no differences encountered


C:\Users\joshs\Documents\Networks\Test>fc Recording.mp3 copy_Recording.mp3
Comparing files Recording.mp3 and COPY_RECORDING.MP3
FC: no differences encountered


C:\Users\joshs\Documents\Networks\Test>fc test7.jpg copy_test7.jpg
Comparing files test7.jpg and COPY_TEST7.JPG
FC: no differences encountered


C:\Users\joshs\Documents\Networks\Test>fc videoplayback.mp4 copy_videoplayback.mp4
Comparing files videoplayback.mp4 and COPY_VIDEOPLAYBACK.MP4
FC: no differences encountered
```

3. How to use

1. Have two terminals available
2. In each terminal navigate to the respective folder, server and client.
   a. cd ./Server
   b. cd ./Client
3. For the server, type python3 rft1Server.py and type in the needed port #
4. For the client, type python3 rft1Client.py and type in the needed port # and host
   a. Recognized commands work as RETR examplename.exampleformat

4. References

Starter echo server and client functionality.
https://realpython.com/python-sockets/#echo-client-and-server