



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
MÔN: MẠNG MÁY TÍNH
LỚP: 16CNTN

BÁO CÁO ĐỒ ÁN 1

LẬP TRÌNH SOCKET

Người thực hiện:

Nguyễn Quốc Vương - 1612829

Lê Thành Công - 1612842

Nguyễn Thanh Tuấn – 1612774

Giáo viên:

Lê Hà Minh

Mục lục

I. Những hàm chính và tổ chức dữ liệu.	3
II. Đánh giá mức độ hoàn thành.	6
III. Hình ảnh các bước chạy chương trình.	7
IV. Bắt gói tin tại client bằng Wireshark của quá trình gửi nhận dữ liệu khi download một file từ server. 13	
V. Phân công nhiệm vụ.	15
VI. Tài liệu tham khảo.	15

I. Những hàm chính và tổ chức dữ liệu.

- Cấu trúc chung: Chương trình được tổ chức theo hướng đối tượng với **class Client**. Các chức năng chính được tổ chức thành từng hàm riêng lẻ và chương trình chính là hàm **void process()** xử lý dữ liệu đầu vào và gọi hàm thao tác theo đúng yêu cầu trên commandline. Tất cả các hàm chính đều không có giá trị trả về, việc xử lý lỗi được quản lý bởi thuộc tính **error** của class.
- Các hàm chức năng chính:
 - **Hàm void login():**
 - + Chức năng: Login khi đã connect được tới server.
 - + Kết quả khi gọi hàm: Thông báo người dùng nhập Username và Password để đăng nhập vào server. Nếu thành công sẽ nhận được thông báo “Logged on”, ngược lại nhận được thông báo “Login or password incorrect!”.
 - **Hàm void ls() và hàm void dir():**
 - + Chức năng: Liệt kê danh sách các tập tin, thư mục trên thư mục hiện tại của server.
 - + Kết quả: Nếu gọi lệnh “ls” sẽ liệt kê tên các tập tin, thư mục trên server, còn khi gọi lệnh “dir” sẽ liệt kê tên như lệnh “ls” nhưng chi tiết hơn thông tin từng thư mục, tập tin (phân quyền, ngày tháng cập nhật, dung lượng, tên).
 - **Hàm void put(char *filename):**
 - + Chức năng: Upload một file lên server.
 - + Tham số: filename là tên file cần upload lên server.
 - + Kết quả: Khi command lệnh “put filename” nếu filename không tồn tại sẽ báo lỗi “filename: File not found”, nếu thành công sẽ gửi thông báo “Successfully transferred "/filename". Chú ý filename có thể là đường dẫn đến file (ví dụ: put 1\filename.txt), đường dẫn sử dụng dấu ‘\’.
 - **Hàm void mput(char **filename, int count):**
 - + Chức năng: Upload nhiều file lên sever.
 - + Các tham số: filename là danh sách tên các file cần upload lên server, count là số lượng các file cần upload (tính luôn lệnh mput là 1).
 - + Kết quả: Khi command lệnh “mput filename1 filename2 ...”, chương trình sẽ hỏi lần lượt từng file có muốn Upload không (nếu nhấn phím ‘y’ hoặc Enter file sẽ được upload lên server, ngược lại bỏ qua không upload) nếu filename nào không tồn tại sẽ báo lỗi “filename: File not found”, nếu thành công sẽ gửi thông báo “Successfully transferred "/filename”.

- + Các dạng mput (có thể ghi thêm đường dẫn đến file):
 - mput filename1 filename2: Upload những file có tên trong danh sách lên server.
 - mput *: Upload tất cả các file trong thư mục hiện tại của client lên sever.
 - mput *.tail (với tail là đuôi file ví dụ *.txt, *.py, *.zip, ...): Upload tất cả các file có đuôi là tail trong thư mục hiện tại của client lên server.
- **Hàm void get(char *filename):**
 - + Chức năng: Download một file lên server.
 - + Tham số: filename là tên file cần download từ server.
 - + Kết quả: Khi command lệnh “get filename” nếu filename không tồn tại sẽ báo lỗi “filename: File not found”, nếu download thành công sẽ gửi thông báo “Successfully transferred "/filename". Chú ý filename có thể là đường dẫn đến file (ví dụ: get 1\filename.txt), đường dẫn sử dụng dấu ‘\’.
- **Hàm void mget(char **filename, int count):**
 - + Chức năng: Download nhiều file lên sever.
 - + Tham số: filename là danh sách tên các file cần download từ server, count là số lượng file cần download (tính luôn lệnh mget là 1).
 - + Kết quả: Khi command lệnh “mget filename1 filename2 ...”, chương trình sẽ hỏi lần lượt từng file có muốn Upload không (nếu nhấn phím ‘y’ hoặc Enter file sẽ được download từ server về client, ngược lại bỏ qua không download) nếu filename nào không tồn tại sẽ báo lỗi “filename: File not found”, nếu thành công sẽ gửi thông báo “Successfully transferred "/filename".
 - + Các dạng mget (có thể ghi thêm đường dẫn đến file trong thư mục con của thư mục hiện tại trên server):
 - mget filename1 filename2: Upload những file có tên trong danh sách lên server.
 - mget *: Download tất cả các file trong thư mục hiện tại của server về client.
 - mget *.tail (với tail là đuôi file ví dụ *.txt, *.py, *.zip, ...): Download tất cả các file có đuôi là tail trong thư mục hiện tại của server về client.
- **Hàm void cd(char *name):**
 - + Chức năng: Thay đổi đường dẫn trên server.
 - + Tham số: name là đường dẫn đến thư mục con cần chuyển đến.
 - + Kết quả: Khi command lệnh “cd name”, thì server sẽ chuyển sang thao tác trên thư mới chuyển đến. Nếu thành công sẽ nhận được thông

báo “CWD successful. “/name” is current directory.”, ngược lại nhận được thông báo “CWD failed. “/name”: directory not found.”.

- **Hàm void lcd(char* dir_name):**
 - + Chức năng: Thay đổi đường dẫn trên client.
 - + Tham số: dir_name đường dẫn mới đến client.
 - + Kết quả: Khi command lệnh “lcd dir_name”, nếu tồn tại thư mục và chuyển thành công sẽ nhận được thông báo “Local directory now dir_name.” (dir_name tùy vào loại lệnh lcd). Nếu thư mục không tồn tại sẽ nhận được thông báo “dir_name: File not found”.
 - + Các loại command của lcd:
 - lcd (không có đường dẫn): Sẽ chuyển đến đường dẫn mặc định là: “C:\Users\Admin”.
 - lcd C:\.... (đường dẫn cụ thể): Thư mục client sẽ chuyển sang thư mục theo đường dẫn mới.
 - lcd dir_name (dir_name là thư mục con của thư mục hiện hành của client): Chuyển sang thư mục con.
- **Hàm void delete(char* name):** (name có thể là đường dẫn tới name)
 - + Chức năng: xóa file trên thư mục hiện hành của server có tên name.
 - + Tham số: name tên file cần xóa.
 - + Kết quả: nếu command lệnh “delete name”, nếu delete thành công sẽ nhận được thông báo “File deleted successfully”, ngược lại nếu file không tồn tại sẽ nhận được “File not found.”.
- **Hàm mdelete(char* filename, int count):**
 - + Chức năng xóa nhiều file có tên trong danh sách filename.
 - + Tham số: filename là danh sách các file cần xóa, count là số lượng các file cần xóa.
 - + Kết quả: nếu command lệnh mdelete filename1 filename2 ... thì lần lượt các file sẽ được hỏi “mdelete filename...?”, nếu nhấn phím ‘y’ hoặc Enter tức đồng ý xóa và nhận được thông báo tương tự như delete, ngược lại nhấn phím khác bất kỳ sẽ bỏ qua không xóa file.
 - + Các loại command mdelete:
 - mdelete filename1 filename2: Xóa các file có tên đã liệt kê trong sau ‘mdelete’.
 - mdelete *: Xóa tất cả các file trong thư mục hiện hành của sever.
 - mdelete *.tail (với tail là loại đuôi file ví dụ *.txt, *.rar ,...): xóa tất cả các file có đuôi là tail trên thư mục hiện hành của sever.
- **Hàm void mkdir(char* name):** name có thể là đường dẫn thư mục.
 - + Công dụng: Tạo thư mục có tên name trên server.
 - + Tham số: name là tên thư mục sẽ tạo mới.

+ Kết quả: command lệnh “mkdir name” sẽ nhận được “name created successfully” nếu thành công, ngược lại nếu thư mục đã tồn tại sẽ được thông báo “Directory already exists”.

- **Hàm void rmdir(char* name):**

+ Chức năng: Xóa thư mục rỗng có tên name.

+ Tham số: name là tên thư mục cần xóa.

+ Kết quả: Nếu command “rmdir name”, nếu xóa thành công sẽ nhận được thông báo “Directory deleted successfully”, nếu thư mục không tồn tại “File not found”, nếu thư mục không phải là thư mục rỗng sẽ nhận được thông báo “Directory not empty.”.

- **Hàm void pwd():**

+ Chức năng: Hiện thị đường dẫn hiện tại trên server.

- **Hàm void pasv():**

+ Chuyển sang chế độ passive.

+ Lệnh command: “passive” hoặc “pasv”. Khi chuyển sang lệnh passive mode thì lệnh kế tiếp sau nó (gồm các lệnh get, put, ls, dir) sẽ được thực hiện dưới chế độ passive.

- **Hàm void close():**

+ Chức năng đóng kết nối đến sever.

+ Lệnh command: “exit” hoặc “quit”, khi thực hiện hiện này sẽ nhận được thông báo “Goodbye!” và thoát khỏi command line sau đó.

II. Đánh giá mức độ hoàn thành.

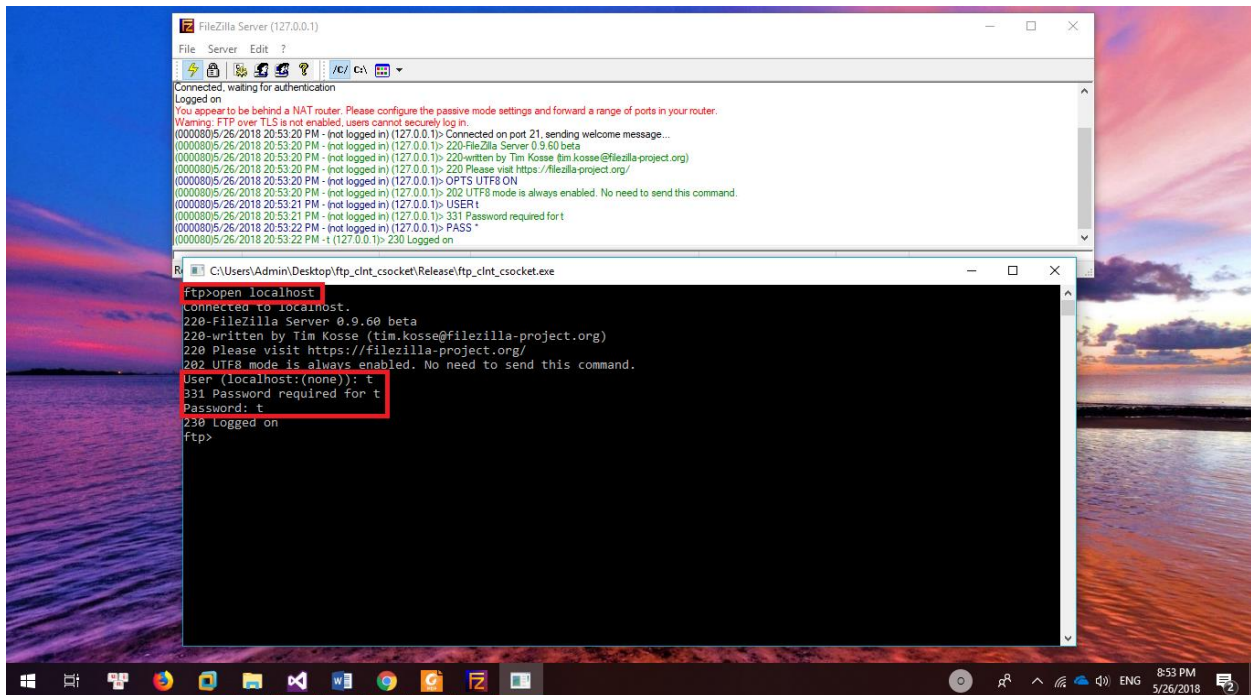
STT	Chức năng	Mức độ hoàn thành (Đơn vị: %)
1	Login vào FTP server	100
2	Liệt kê danh sách các thư mục, tập tin trên server.	100
3	Upload một file đến server	100
4	Download một file từ server	100
5	Upload nhiều file đến sever	100
6	Download nhiều file đến server	100
7	Thay đổi đường dẫn trên server	100
8	Thay đổi đường dẫn dưới client	100
9	Xóa một file trên server	100
10	Xóa nhiều file trên server	100
11	Xóa thư mục rỗng trên server	100
12	Tạo thư mục rỗng trên server	100

13	Hiện thị đường dẫn hiện tại trên server	100
14	Chuyển sang trạng thái passive	100
15	Thoát khỏi server	100
16	Hỗ trợ cả hai cơ chế active và passive	100

Bảng 1: Bảng đánh giá mức độ hoàn thành yêu cầu đề án

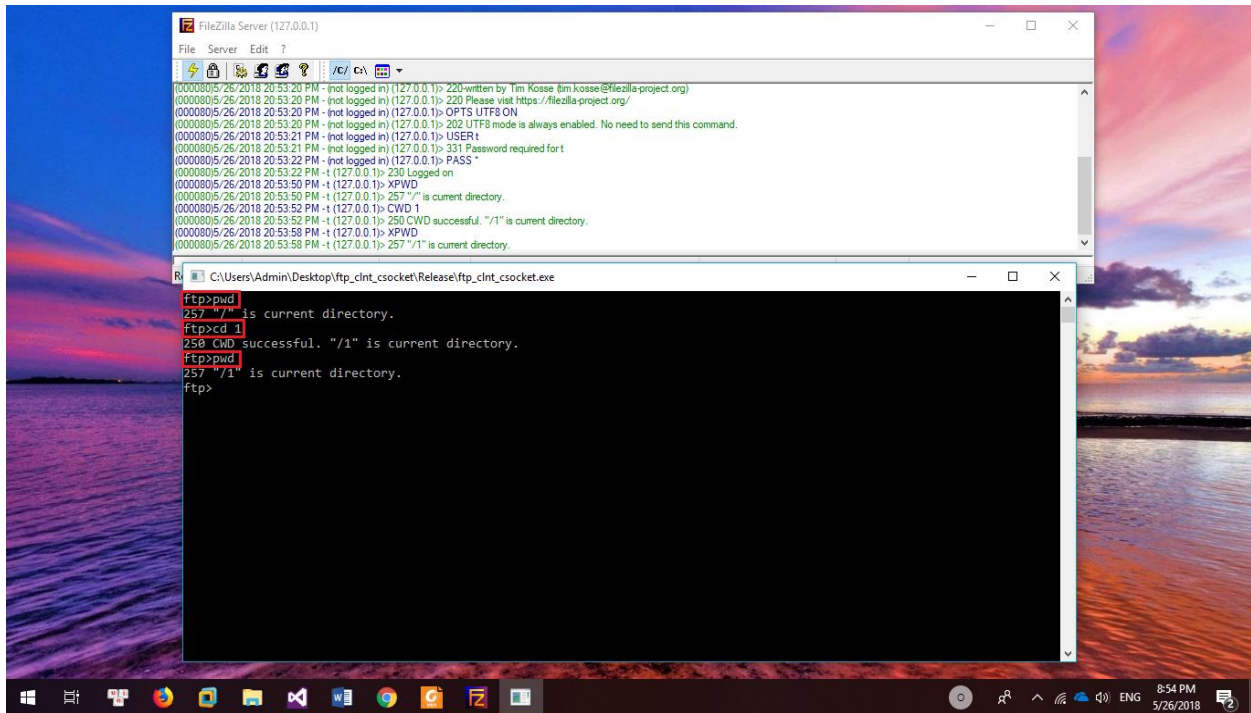
III. Hình ảnh các bước chạy chương trình.

- Kết nối và đăng nhập sever.



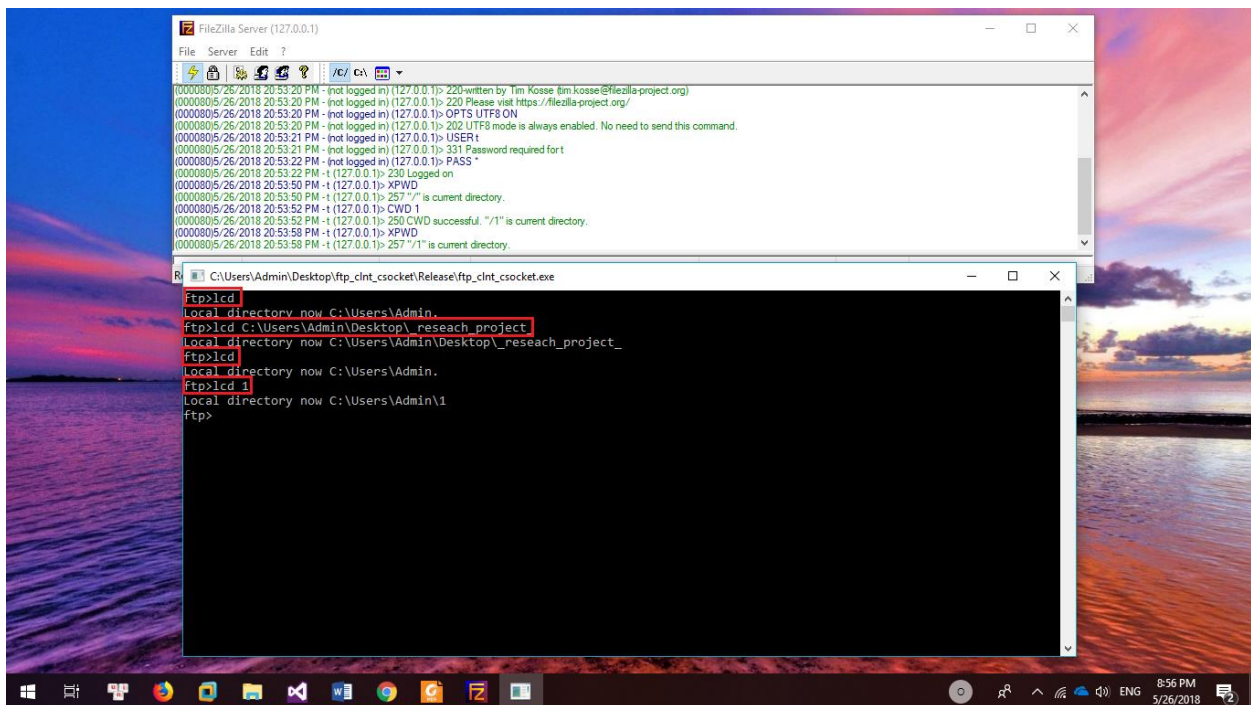
Hình 1: Đăng nhập vào server

- Lệnh pwd, cd.



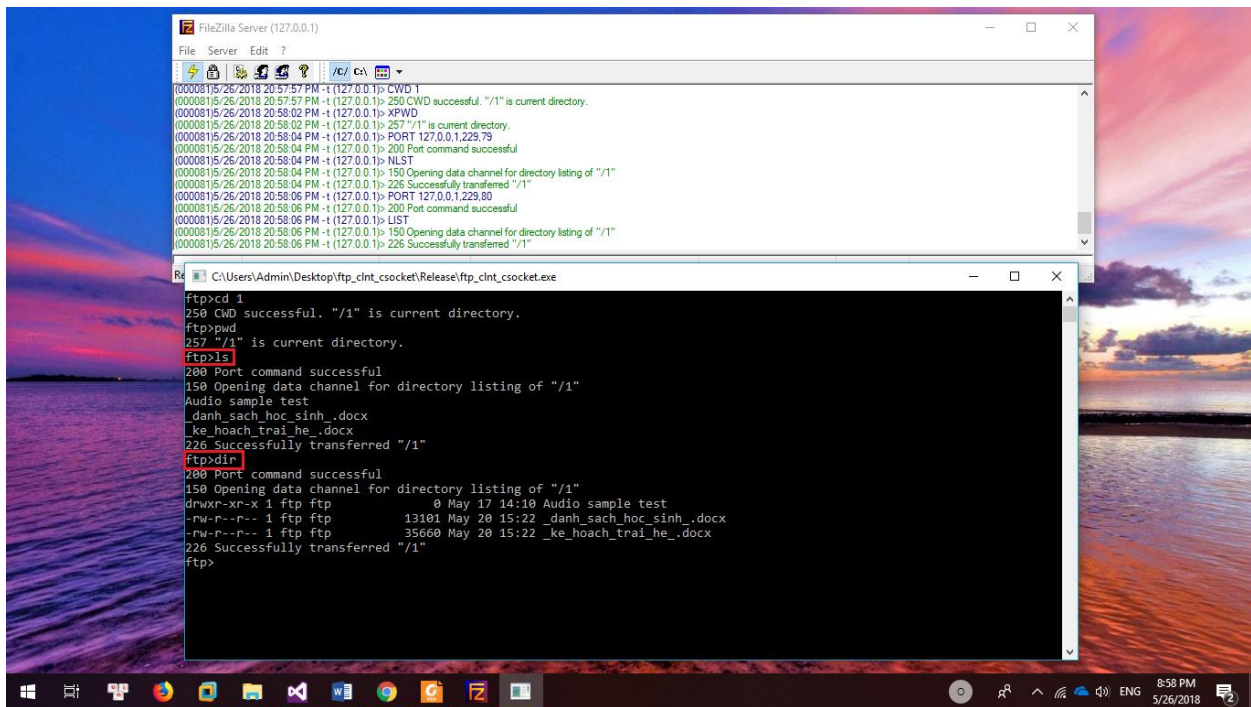
Hình 2: Lệnh thao tác với thư mục

- Lệnh lcd.



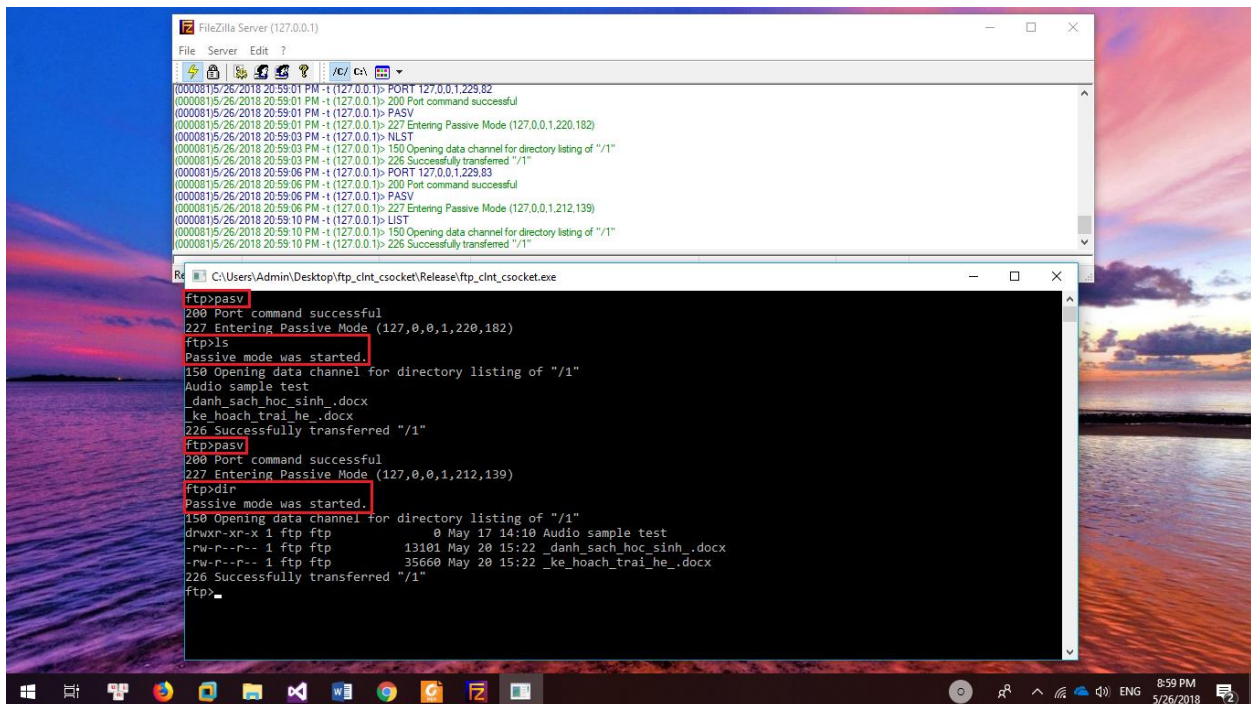
Hình 3: Thay đổi đường dẫn client

- Lệnh ls, dir.



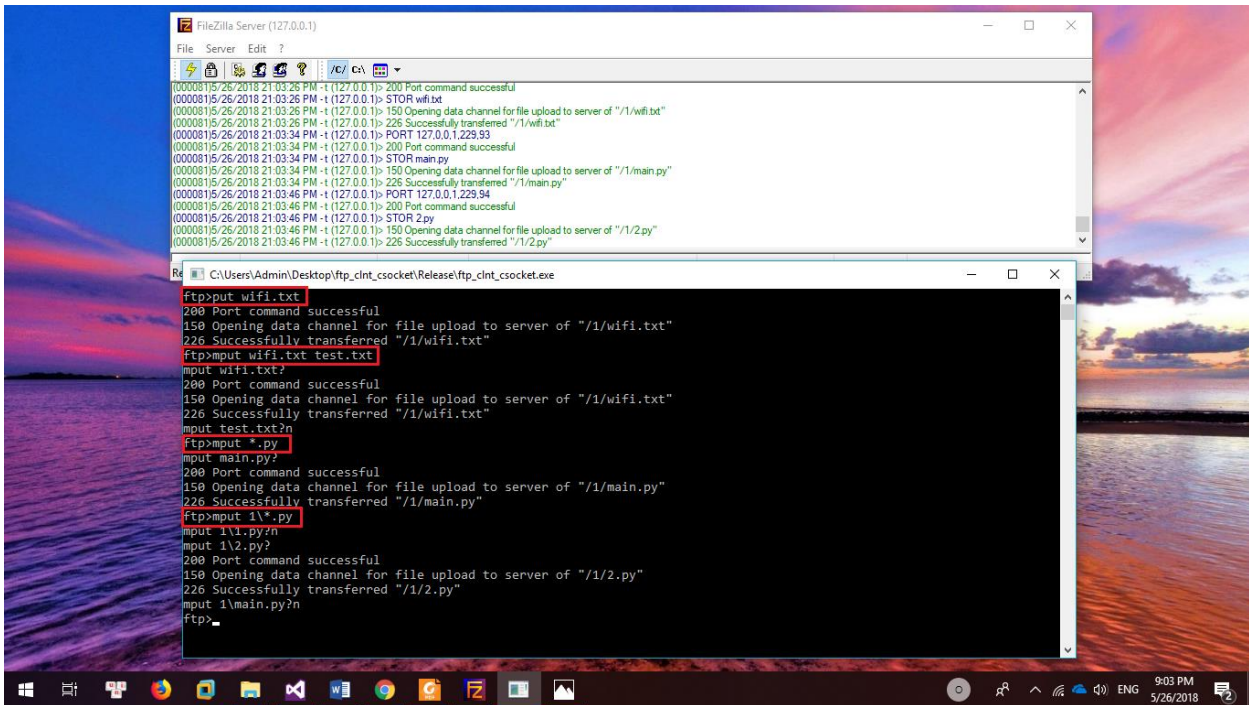
Hình 4: Liệt kê tên file và thư mục trong thư mục hiện hành trên server

- Lệnh pasv



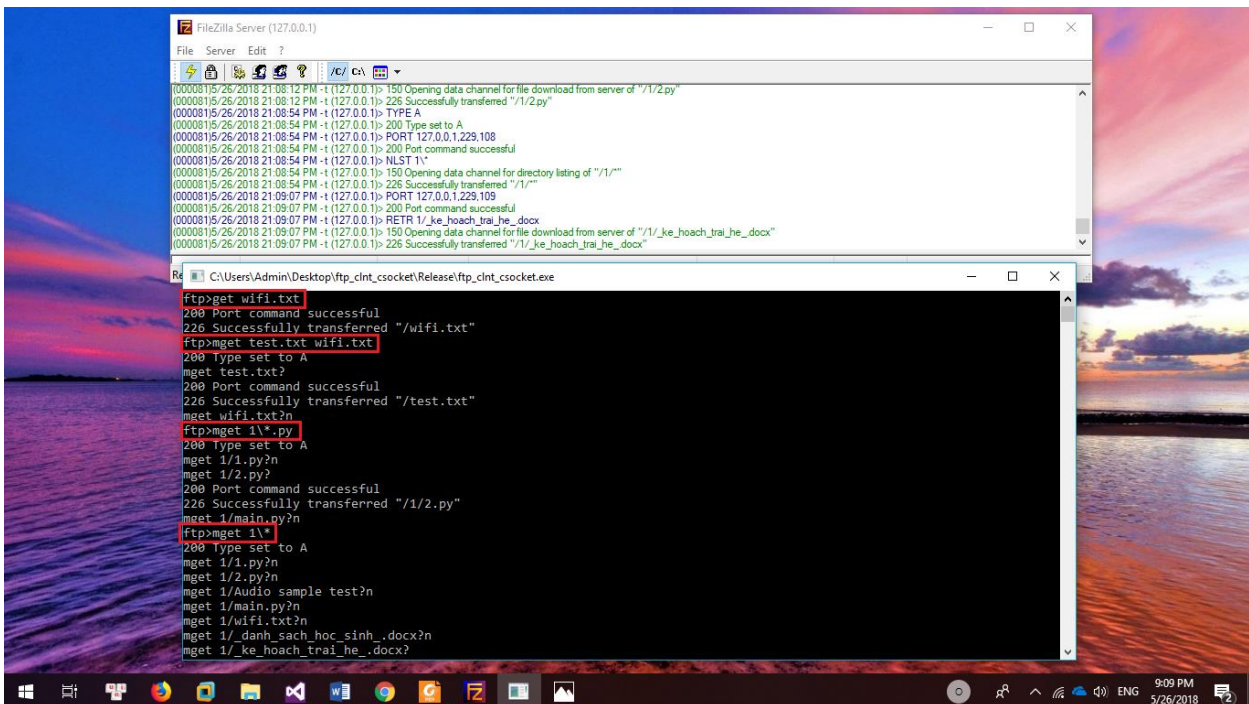
Hình 5: Chế độ passive mode

- Lệnh put, mput.



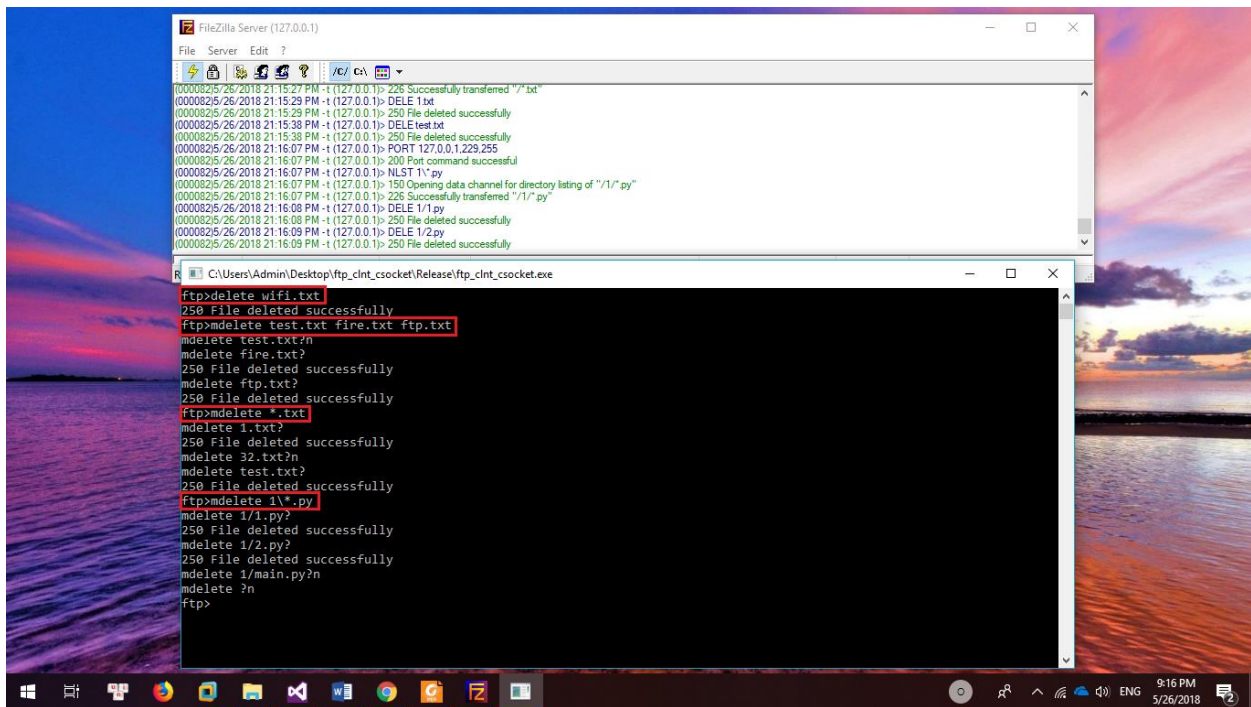
Hình 6: Upload file từ client lên server

- Lệnh get, mget.



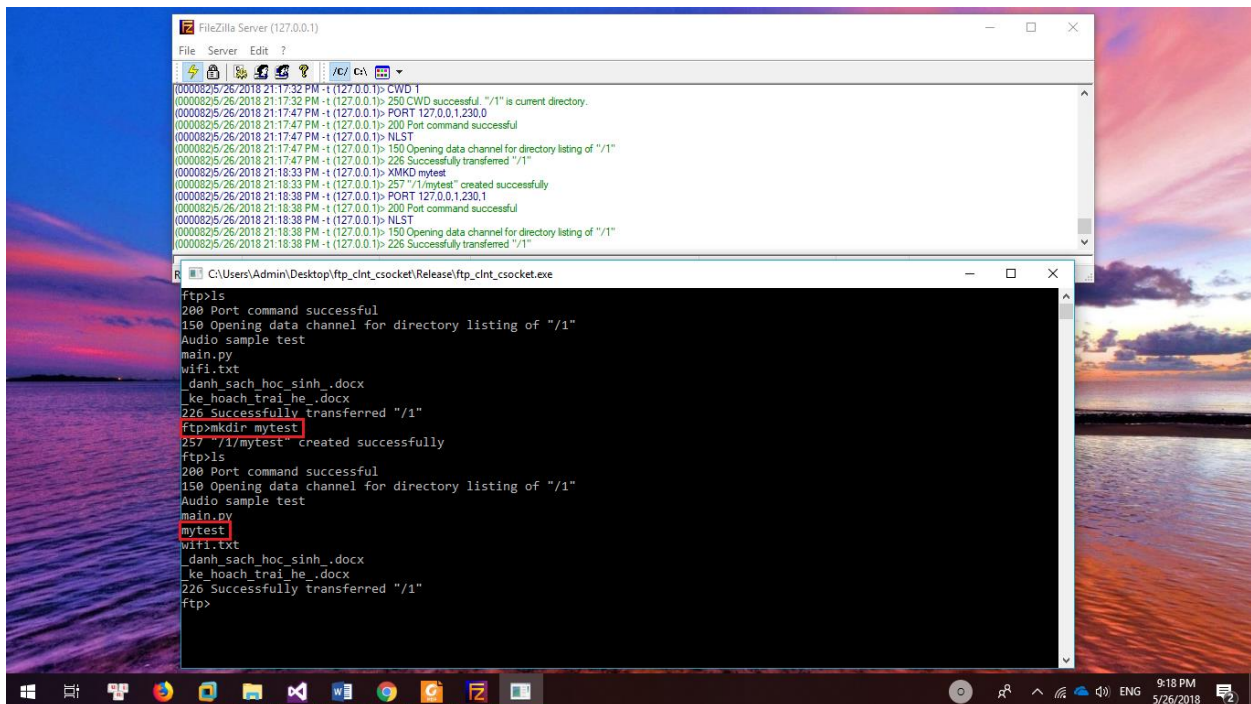
Hình 7: Download file từ server về client

- Lệnh delete, mdelete.



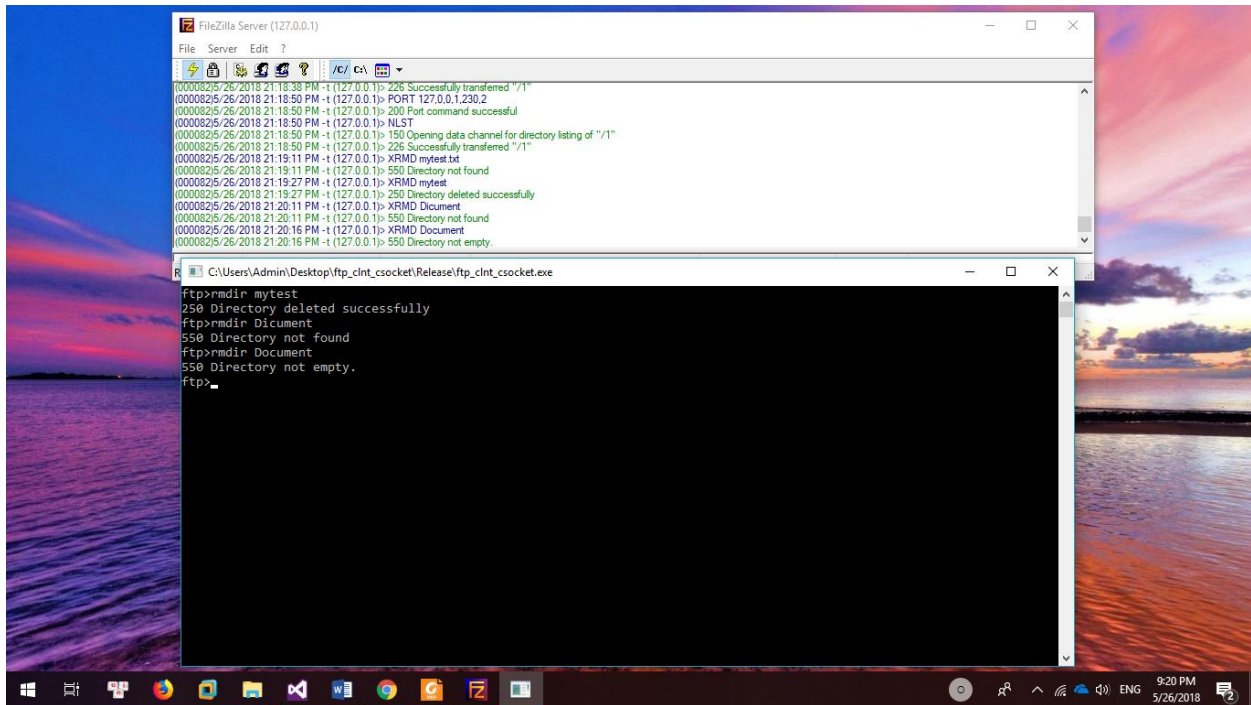
Hình 8: Xóa file trên server

- Lệnh mkdir.



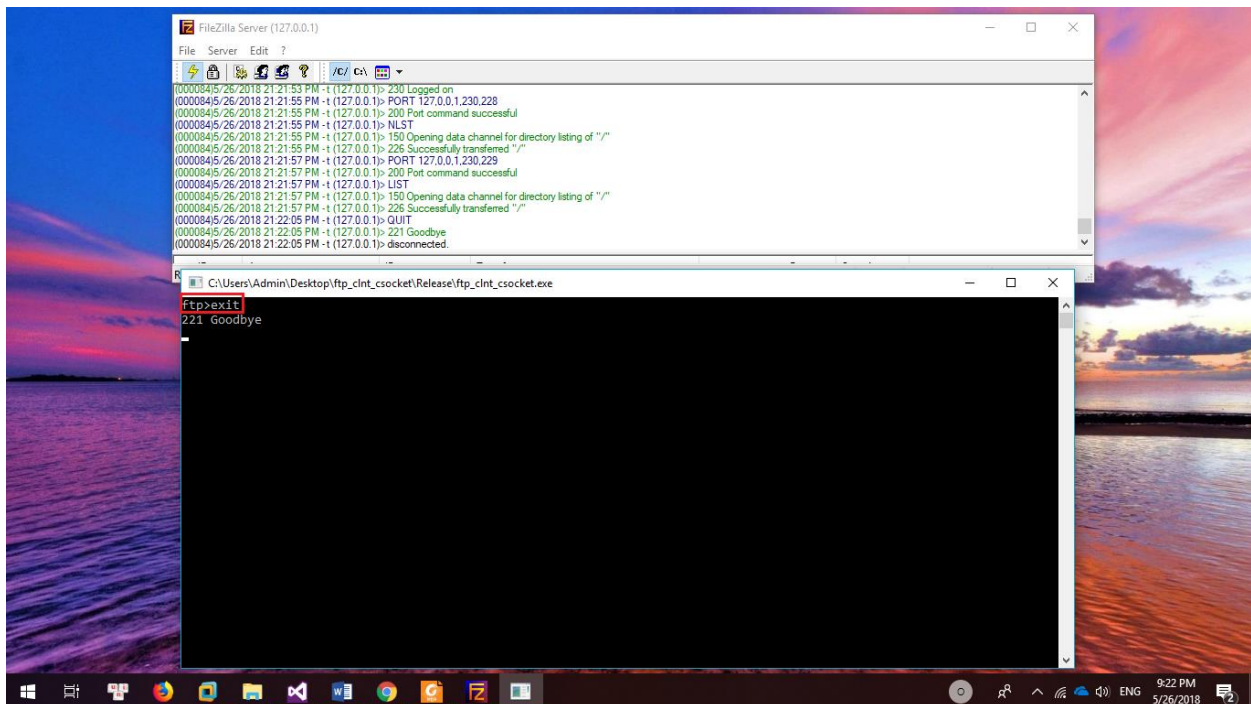
Hình 9: Tạo thư mục rỗng trên server

- **Lệnh rmdir.**



Hình 10: Xóa thư mục rỗng trên server

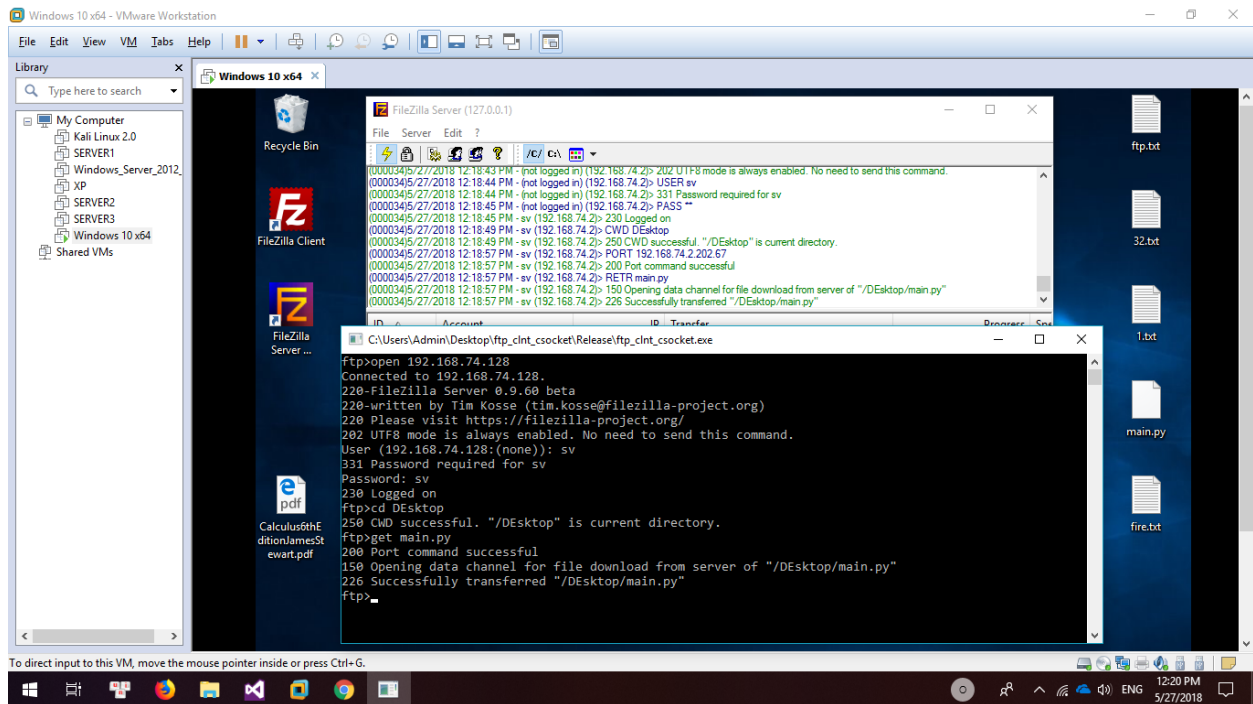
- **Lệnh exit, quit.**



Hình 11: Đóng kết nối, thoát khỏi server

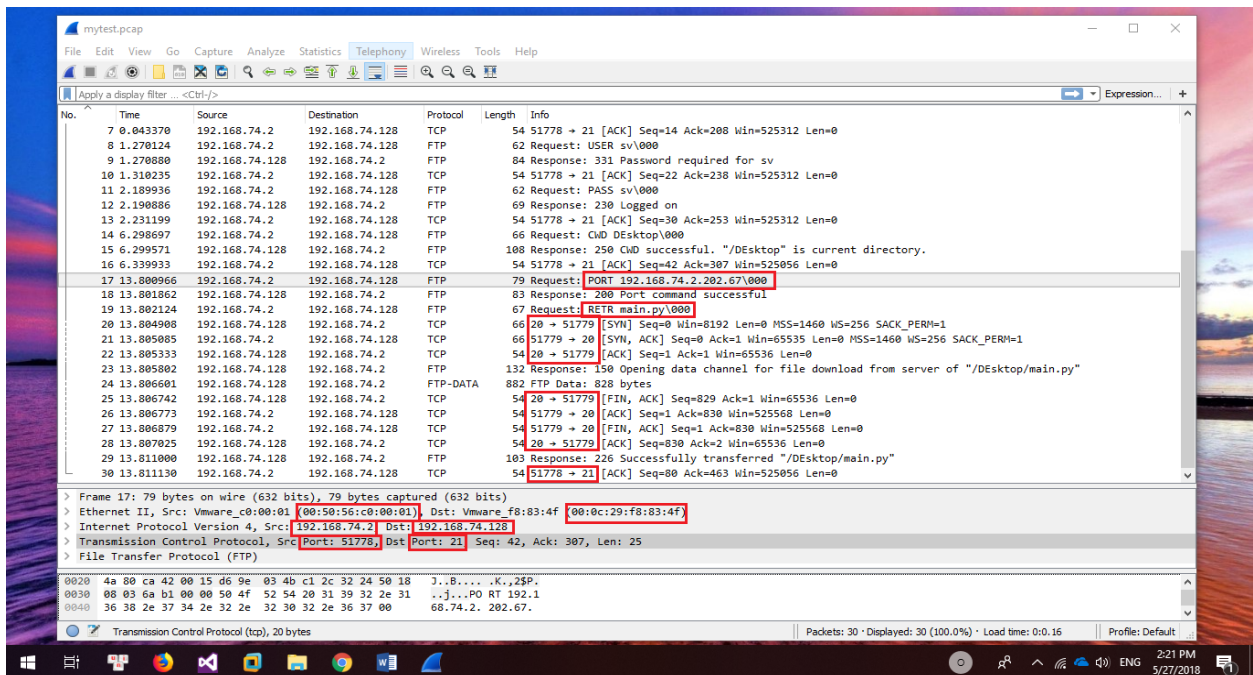
IV. Bắt gói tin tại client bằng Wireshark của quá trình gửi nhận dữ liệu khi download một file từ server.

- Download file từ server.



Hình 12: Download file từ sever (trên máy ảo) về client (trên máy thật)

- Quá trình nhận, gửi các gói tin khi tải file từ server về client.



Hình 12: Bắt gói tin quá trình download dữ liệu từ server về client bằng Wireshark

- Phân tích chi tiết:
 - + Địa chỉ MAC:
 - Client (máy thật): 00:50:56:c0:00:01
 - Server (máy ảo): 00:0c:29:f8:83:4f
 - + IP:
 - Client (máy thật): 192.168.74.2
 - Server (máy ảo): 192.168.74.128
 - + Port:
 - Server sử dụng port 20 và port 21 trong quá trình gửi nhận dữ liệu
 - Client sử dụng port 51778 và port 51779 trong quá trình gửi nhận dữ liệu.
 - + Quy trình:
 - Client gửi lệnh “PORT 192.168.74.128” lên server, sau đó server gửi về client thông báo “200 Port command successful”, tiếp đó client gửi tiếp lệnh “RETR main.py” lên server (server lúc này dùng port 21).
 - Nhận dữ liệu: Tạo kết nối và nhận dữ liệu. Server gửi về client packet với cờ “SYN”, sau đó client đáp lại với packet cờ “SYN, ACK”. Tiếp đến server gửi về client packet với cờ “ACK” chấp nhận và gửi thông điệp “150 Opening data channel for file download from server of “/Desktop/main.py””. Dữ liệu được truyền xuống client bằng kênh vừa tạo. Quá trình kết thúc trải qua 4 bước: Server gửi về client packet với cờ “FIN, ACK” tức nói đã hoàn thành và đóng kết nối truyền dữ liệu, Client liên tiếp gửi 2 packet trả lời chấp nhận với các cờ tương ứng “ACK” và “FIN, ACK”; Server nhận được gửi lại packet với cờ “ACK” và đóng kênh truyền dữ liệu. Cuối cùng server gửi thông báo “266 Successful transferred “/Desktop/main.py” về client và client gửi gói tin “ACK” để thông báo đã nhận được, quá trình hoàn thành.

V. Phân công nhiệm vụ.

STT	MSSV	Họ và tên	Công việc	Đánh giá
1	1612829	Nguyễn Quốc Vương	Hàm ls, dir, quit, exit, chế độ passive, login	Tốt
2	1612842	Lê Thành Công	Hàm delete, mdelete, cd, pwd, mkdir, rmdir, chế độ passive	Tốt
3	1612774	Nguyễn Thanh Tuấn	Viết báo cáo, hàm get, put, mput, mget, chế độ passive	Tốt

Bảng 2: Bảng phân công nhiệm vụ từng thành viên trong nhóm

VI. Tài liệu tham khảo.

- [1].WSAAsyncSelect.pdf
- [2].sockets Programming in C++.pdf, c Mani Radhakrishnan and Jon Solworth, September 24, 2004
- [3].CSocket.pdf, Khoa Công nghệ Thông tin – Trường ĐH KHTN Tp. HCM
Bộ môn Mạng Máy Tính và Viễn Thông
- [4].<https://www.cybersitesindia.com/downloads/PDF/Guide%20to%20FTP%20With%20FileZilla.pdf>
- [5].<http://www.nsftools.com/tips/MSFTP.htm>