

BÁO CÁO ĐỒ ÁN THỰC HÀNH 1 – LẬP TRÌNH SOCKET

MÔN MẠNG MÁY TÍNH

STT	Thành viên	Mã số sinh viên
1	Nguyễn Hoàng Nhân	18127017
2	Kiều Công Hậu	18127259
3	Trần Thanh Tâm	18127268

$$Mã\ đề\ tài = (7 + 9 + 8) \bmod 2 + 1 = 1$$

ĐỀ 1 – FILE SERVER

1. Đánh giá mức độ hoàn thiện của đồ án

STT	Yêu cầu	Ghi chú	Đánh giá
1	Server phải cho phép nhiều client đăng nhập và tải file cùng lúc.		100%
2	Số lượng client không giới hạn		100%
3	Client cần đăng ký username và password để đăng nhập.		100%
3	Nếu trùng username, server yêu cầu đăng kí lại username khác.		100%
4	Client sẽ sử dụng tên này trong toàn bộ quá trình trao đổi với server.		100%
5	Tại giao diện phải có mục hiển thị danh sách tất cả các client đang kết nối đến.		100%
6	Khi client đăng nhập hoặc đăng xuất (hoặc thoát chương trình), phải có dòng thông báo trên màn hình server và mọi client.		100%
7	Mọi hoạt động upload, download file của client cần ghi log hiển thị trên màn hình server.	Ngoài ra, nhóm em còn ghi xuống logfile.txt.	100%
8	Tại mỗi client cần hiển thị danh sách các file mà server chia sẻ để client có thể chọn và tải về.		100%
9	Cùng một thời điểm server cho phép nhiều client tải về cùng 1 file và kích thước có thể lên đến 200MB	Kích thước lớn hơn 200MB vẫn được.	100%
10	Cho phép client upload file lên server.		100%
11	Server cập nhật danh sách các file mà server chia sẻ đến các client khác		100%
12	Tại mỗi thời điểm chỉ cho phép 1 client upload 1 file.		100%
Tổng kết			99%

2. Kịch bản giao tiếp của chương trình

a. Giao thức Register:

Bước 1: Client → Server

- Client gửi Server 1 gói tin flag có chứa flag REGISTER
- Sau đó, gửi tiếp 4 gói tin Data lần lượt chứa các dữ liệu:
 - Chiều dài Username – là một số nguyên.
 - Username – là một chuỗi kí tự.
 - Chiều dài password – là một số nguyên.
 - Password – là một chuỗi kí tự.

Bước 2: Server → Client

- Server nhận 1 gói tin Flag và 4 gói tin Data từ Client.
- Gửi lại Client gói tin Flag chứa kết quả register:
 - REGISTER_SUCCESS – nếu thành công.
 - REGISTER_FAIL – nếu thất bại vì username đã tồn tại.

b. Giao thức Login:

Bước 1: Client → Server

- Client gửi Server 1 gói tin Flag có chứa flag LOGIN
- Sau đó, gửi tiếp 4 gói tin Data lần lượt chứa các dữ liệu:
 - Chiều dài Username – là một số nguyên.
 - Username – là một chuỗi kí tự.
 - Chiều dài password – là một số nguyên.
 - Password – là một chuỗi kí tự.

Bước 2: Server → Client

- Server nhận 1 gói tin Flag và 4 gói tin Data từ Client.
- Gửi lại Client gói tin Flag chứa kết quả login:
 - LOGIN_SUCCESS – nếu thành công.
 - LOGIN_FAIL_USERNAME – nếu thất bại vì username không tồn tại.
 - LOGIN_FAIL_PASSWORD – nếu thất bại vì password không trùng khớp.
 - LOGIN_FAIL_ONLINE – nếu thất bại vì username đó đang được Client khác sử dụng để Online.
- Nếu kết quả login là thành công, tiếp tục thực hiện bước 3. Còn không thì kết thúc quá trình login tại đây.

Bước 3: Server → Client

- Server gửi danh sách các file lưu trên Server cho Client bằng cách gửi liên tục các gói tin Flag:
 - flag NEW_FILE_LIST.
 - msgLen – chiều dài của tên file (là một số nguyên).
 - msg – tên file (là một chuỗi kí tự).
- Số gói tin cần gửi bằng số lượng file được lưu trên Server.

Bước 3: Server → All Client

- Server sẽ thông báo cho tất cả các Client còn lại biết là có 1 Client mới vừa login.

- Đầu tiên, với mỗi Client trừ Client vừa mới login ở trên, Server gửi gói tin Flag có flag NEW_USER_LOGIN.
- Sau đó, Server tiếp tục gửi cho mỗi client 2 gói tin Data với nội dung:
 - Chiều dài Username – của Client vừa login (là một số nguyên).
 - Username – của Client đó (là một chuỗi kí tự).

c. Giao thức Download

Bước 1: Client → Server

- Client gửi cho Server một thông điệp có Flag là DOWNLOAD_FILE, với nội dung Msg là chỉ số (index) của file mà Client cần download trong danh sách các file mà Server chia sẻ được hiển thị trên màn hình của Client.

Bước 2: Server → Client

- Server nhận thông điệp và xử lý. Nhận thấy Flag là DOWNLOAD_FILE, Server liền phản hồi cho Client một thông điệp với Flag là DOWNLOAD_FILE_SUCCESS.
- Client nhận thông điệp phản hồi của Server và xử lý, thấy Flag là DOWNLOAD_FILE_SUCCESS nên sẵn sàng và chờ nhận nội dung file cần tải từ Server.
- Server lấy nội dung của Msg ở thông điệp mà Client yêu cầu tải file là index của file cần tải để truy xuất database và lấy ra chính xác file tại index đó, sau đó Server bắt đầu gửi từng block byte (4096 byte) cho Client.

d. Giao thức Upload

Bước 1: Client → Server

- Client gửi cho Server một thông điệp có Flag là UPLOAD_FILE, với nội dung Msg là tên của file cần upload.

Bước 2: Server → Client

- Server nhận thông điệp và xử lý. Nhận thấy Flag là UPLOAD_FILE, Server truy xuất database của mình và kiểm tra xem tên của file mà Client muốn upload có bị trùng với các file đã tồn tại trong database hay chưa, và phản hồi lại
 - UPLOAD_FILE_FAIL – nếu tồn tại tên file trùng.
 - UPLOAD_FILE_SUCCESS – nếu Server cho phép upload, và sẵn sàng nhận.
- Client nhận thông điệp phản hồi và xử lý Flag:
 - UPLOAD_FILE_FAIL – Ghi log báo tên file bị trùng, không cho phép upload.
 - UPLOAD_FILE_SUCCESS – Client bắt đầu gửi từng block byte dữ liệu của file cho Server (4096 byte), Server nhận tuần tự từng block byte.
- Cuối cùng, Server gửi thông điệp có Flag là NEW_FILE, với nội dung Msg gồm username của Client vừa upload file lên Server + tên của file vừa upload + kích thước của file vừa upload cho tất cả các Client đang Online ngoại trừ Client upload để các Client ấy cập nhật lại danh sách các file mà Server cho phép download.

e. Giao thức Logout của Client

Bước 1: Client → Server

- Client gửi thông điệp cho Server với Flag là LOGOUT_CLIENT, nội dung Msg không quan trọng nên Msg = “a”. Client gọi hàm shutdown, tức Client không gửi dữ liệu cho Server nữa.

Bước 2: Server → Client

- Nếu Server đang gửi dữ liệu cho Client muốn logout thì Server sẽ phải gửi xong xuôi thì mới thực hiện các bước dưới đây.
- Server nhận thông điệp và xử lý, nhận thấy Flag là LOGOUT_CLIENT nên bỏ Client ra khỏi danh sách các Client online, sau đó gửi thông báo cho các Client khác là Client này đã logout với Flag là CLIENT_LOGOUT_NOTIF, nội dung của Msg là username của Client logout.
- Server gửi tiếp cho Client muốn logout một thông điệp với Flag là LOGOUT_CLIENT.
- Sau đó Server shutdown, tức không gửi dữ liệu cho Client muốn logout này nữa.
- Server đóng socket của client logout này luôn với hàm closesocket, tức không nhận dữ liệu từ Client muốn logout này nữa.
- Client nhận được thông điệp phản hồi, thấy gói tin có Flag là LOGOUT_CLIENT thì đóng socket (closesocket) luôn. Các Client đang online còn lại nhận thông điệp với Flag là CLIENT_LOGOUT_NOTIF thì ghi ra thông báo Client này đã logout.

f. Giao thức Logout của Server.

Bước 1: Server → Client

- Server gửi thông điệp với Flag là LOGOUT_SERVER cho tất cả các Client đang online, nội dung Msg không quan trọng nên Msg = “a”.
- Sau đó Server shutdown tất cả các Socket kết nối tới các Client đang online (Server không gửi dữ liệu nữa).

Bước 2: Client → Server

- Mỗi Client đang online sẽ nhận được thông điệp này và xử lý, nhận thấy Flag là LOGOUT_SERVER nên mỗi Client này sẽ gửi lại cho Server một thông điệp với Flag cũng là LOGOUT_SERVER, nội dung Msg không quan trọng nên Msg = “a”.
- Sau đó, mỗi Client này sẽ shutdown Socket của chính nó, tức không gửi dữ liệu cho Server nữa.
- Cuối cùng là mỗi Client này đóng luôn Socket đó (closesocket), tức không nhận dữ liệu từ Server nữa.
- Server nhận được thông điệp và xử lý Flag, nhận thấy Flag là LOGOUT_SERVER nên đóng Socket (closesocket) toàn bộ các socket đang kết nối tới các Client đang online, tức Server không nhận dữ liệu nữa.
- Cuối cùng, danh sách các Client đang online cập nhật lại còn 0 Client.

3. Môi trường lập trình và các framework hỗ trợ để thực thi ứng dụng

- Sử dụng ngôn ngữ lập trình C++.
- Với sự hỗ trợ của Visual Studio.
- Ngoài ra không sử dụng thêm framework hỗ trợ nào, giao diện Console tự thiết kế.

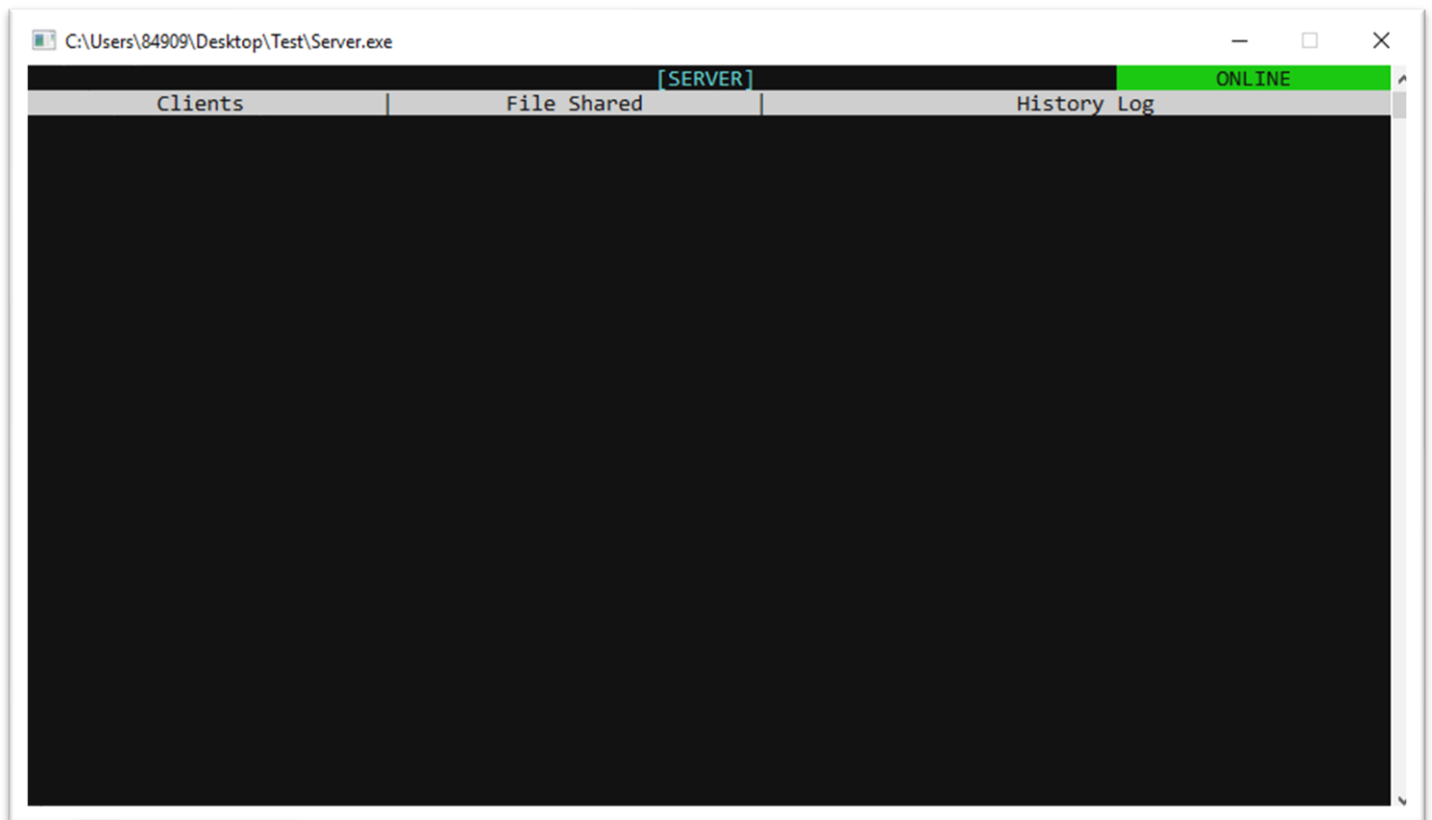
4. Bảng phân công

STT	Nhiệm vụ	Phân công
1	Thiết kế mô hình gửi nhận	Cả nhóm
2	Init file Database	Trần Thanh Tâm
3	Login	Nguyễn Hoàng Nhân
4	Register	Nguyễn Hoàng Nhân

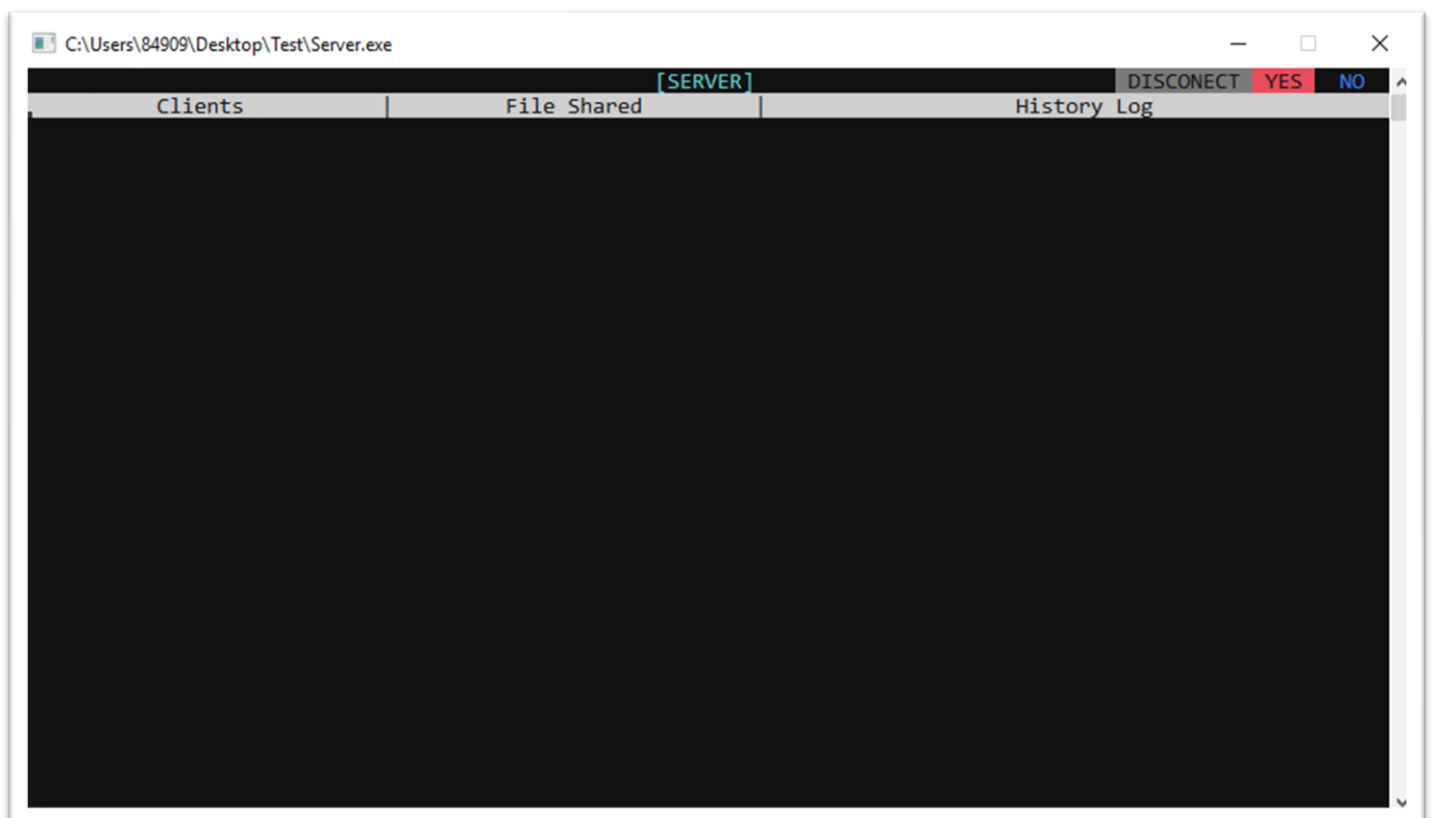
5	Upload	Kiều Công Hậu
6	Download	Kiều Công Hậu
7	Logout	Kiều Công Hậu
8	GUI	Trần Thanh Tâm
9	Báo cáo	Cả nhóm

5. Hướng dẫn sử dụng các tính năng của chương trình.

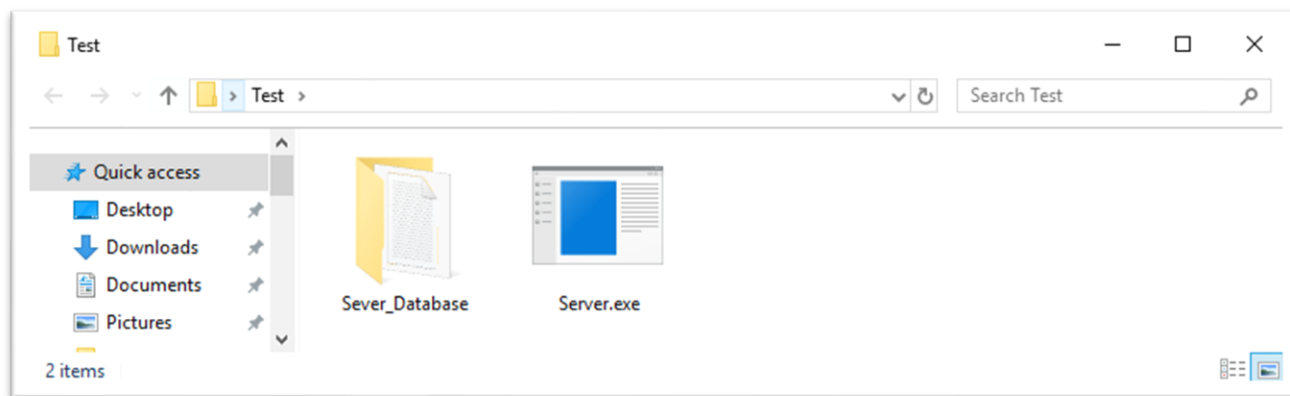
Server: Khởi động Server



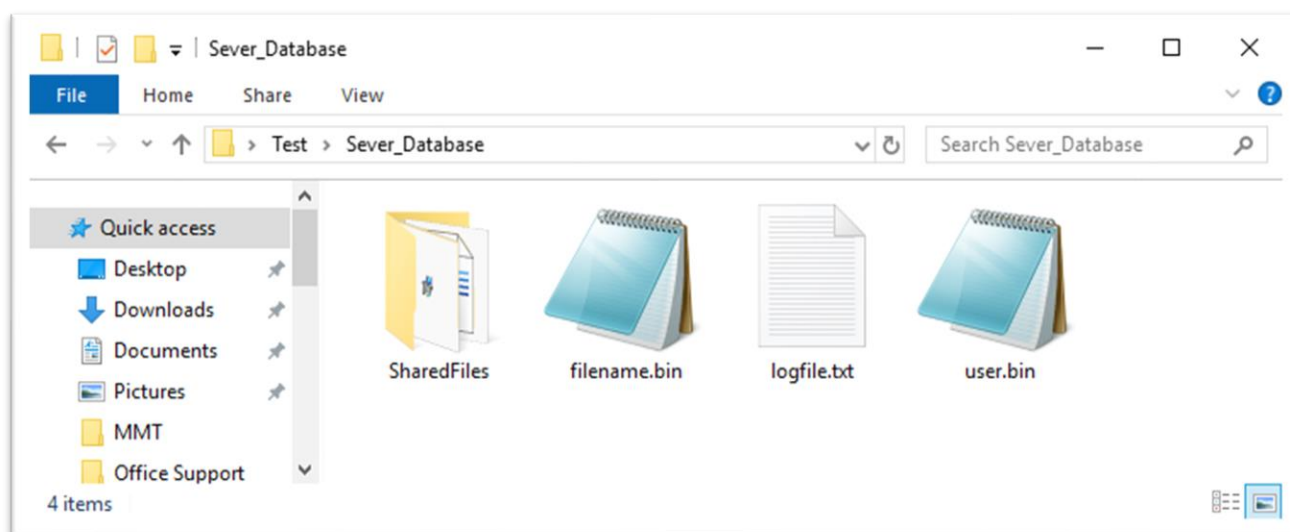
Server: Ngắt kết nối bằng thao tác các phím ESC. Sau đó, sẽ hiện lên cảnh báo DISCONNECT với hai nút YES và NO, thao tác mũi tên qua lại và ENTER để chọn. Nếu YES sẽ thoát Server và tắt chương trình. Nếu NO, thì Server vẫn hoạt động bình thường



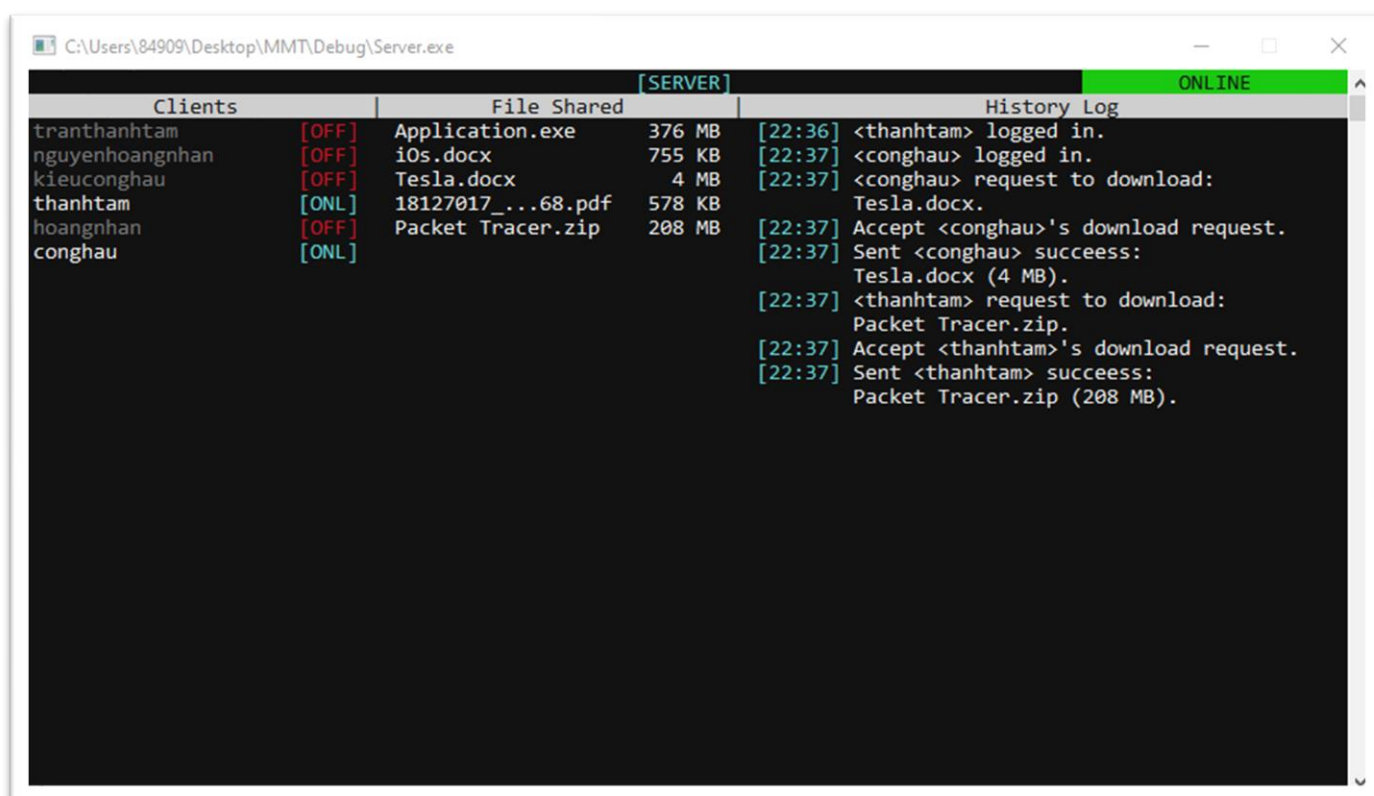
Server: Ngoài ra tại thư mục chứa Server.exe, Server sẽ tự động tạo Server_Database.



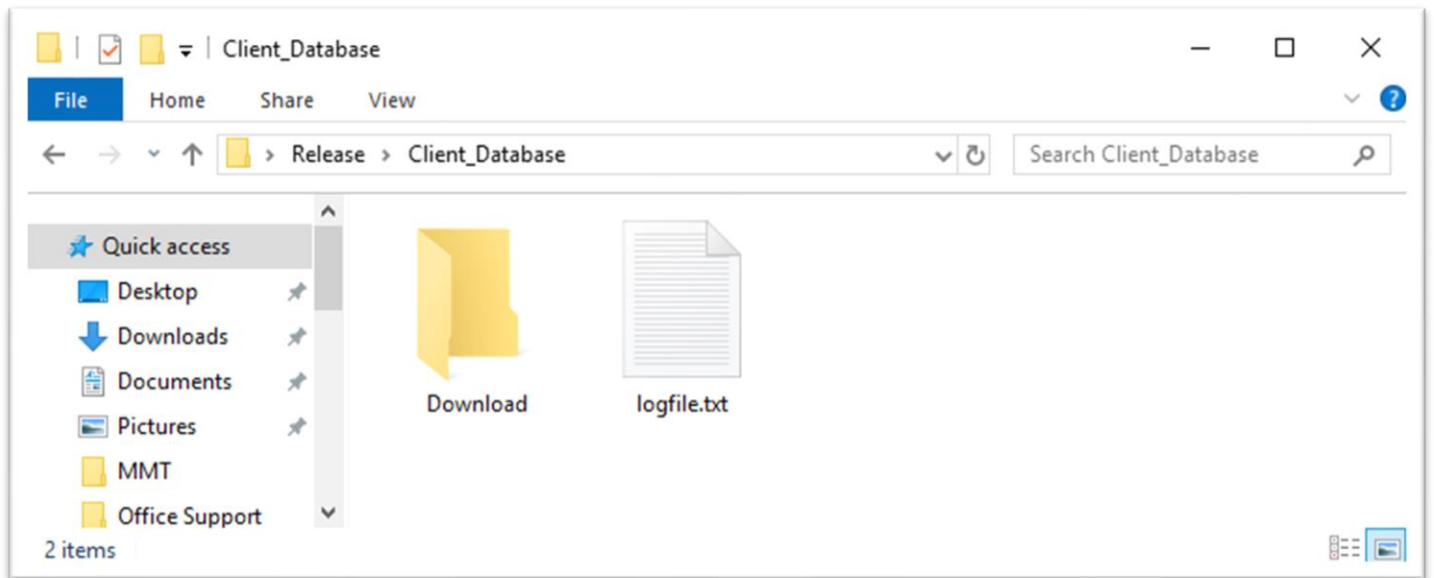
Server: Server_Database chứa các folder/file như sau.



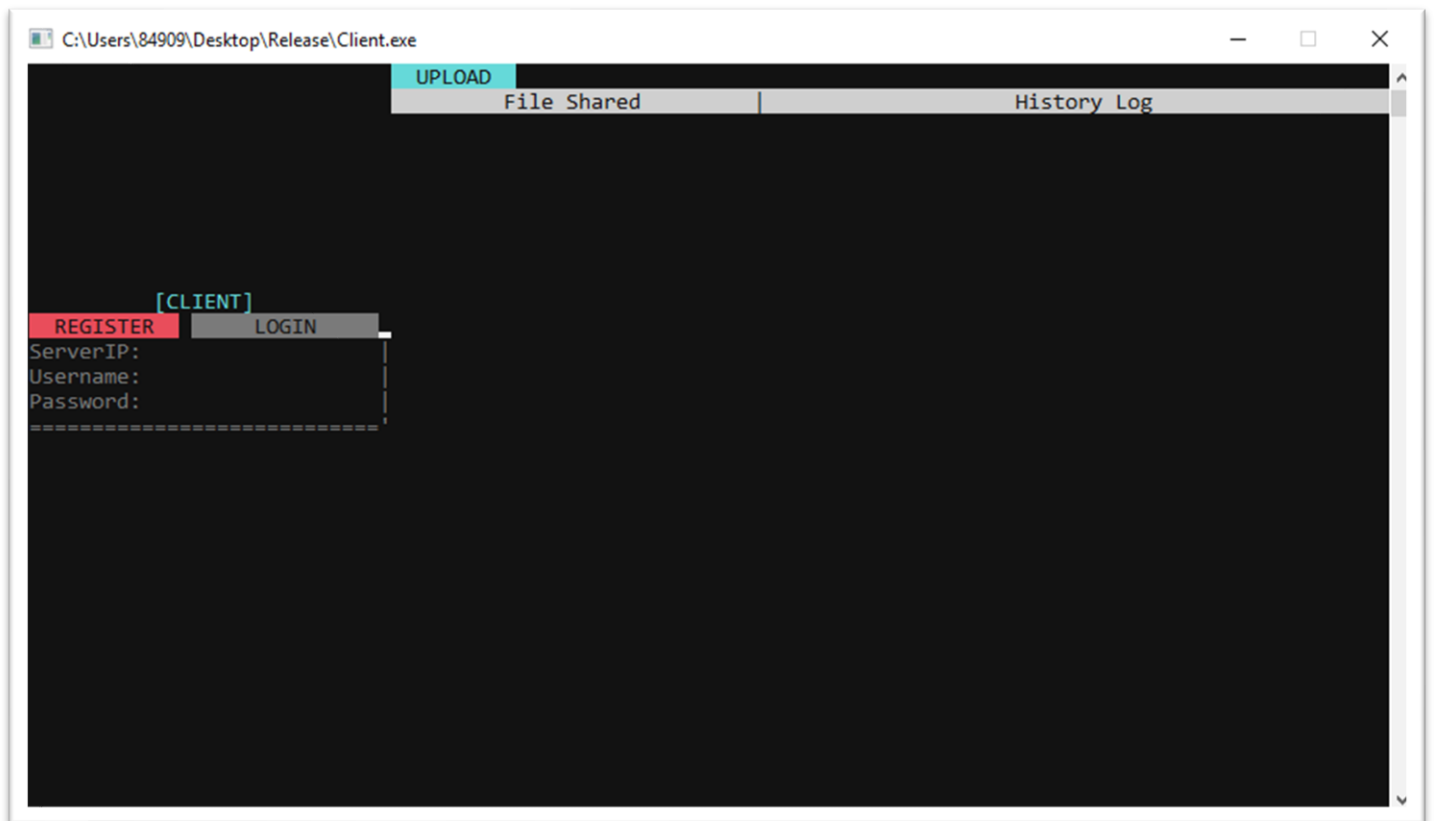
Server: Một Server hoạt động sẽ được hiển thị như sau



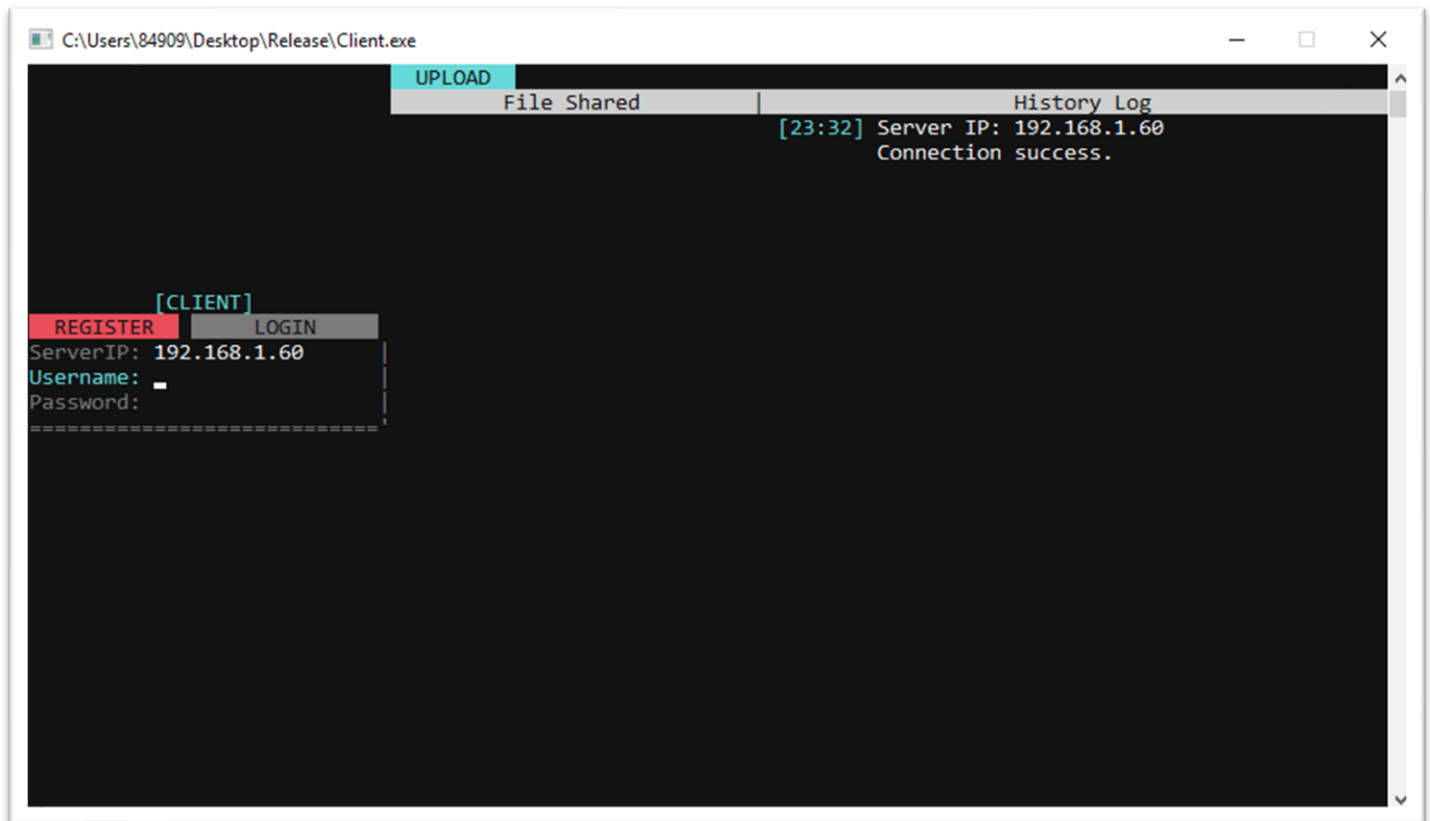
Client: Tương tự cũng có Client_Database



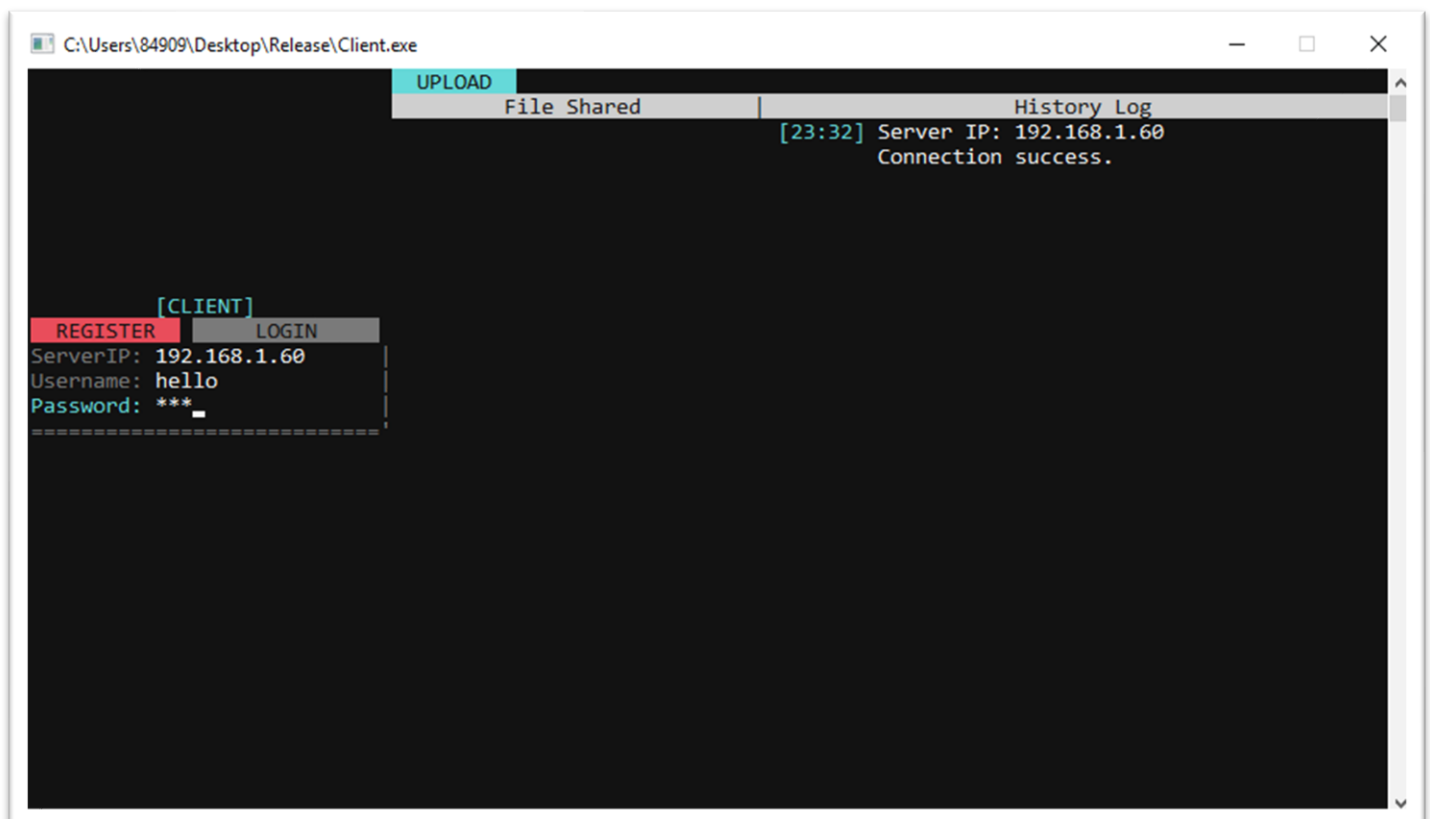
Client: Chọn 1 trong 2 chức năng REGISTER/LOGIN, bằng phím mũi tên trái phải.



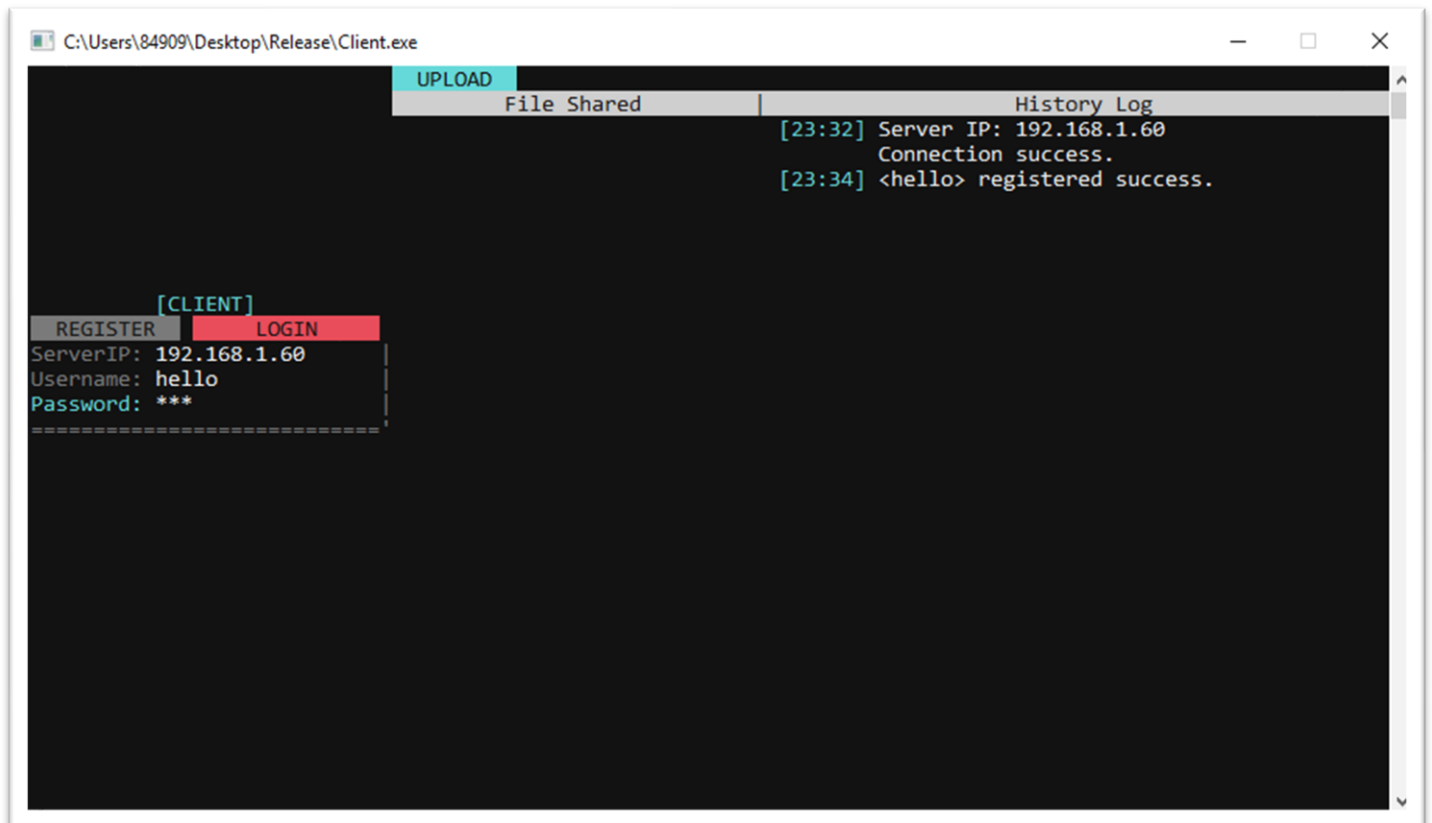
Client: Sau khi chọn sẽ nhập vào Server IP



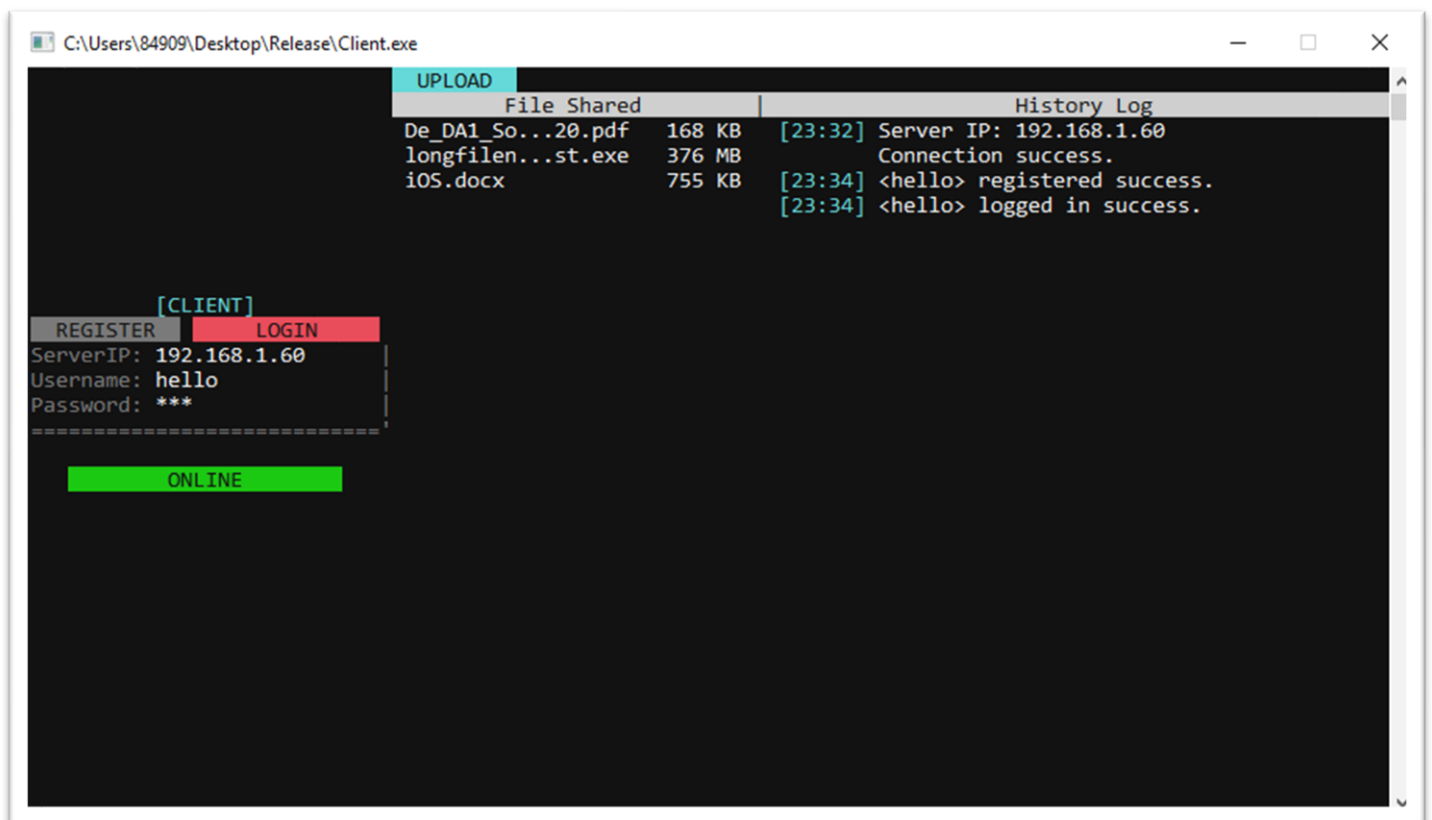
Client: Nhập thông tin để đăng kí.



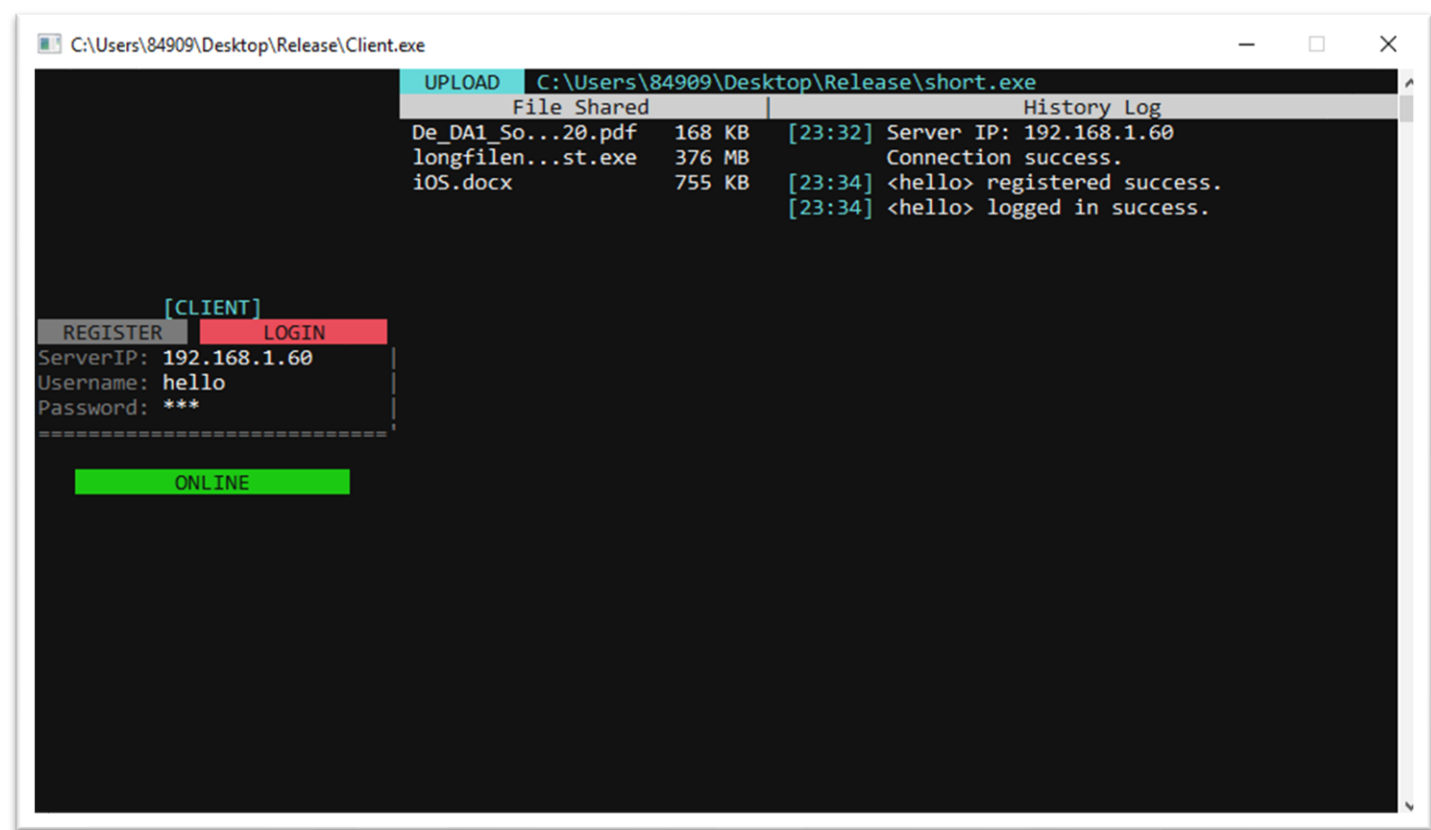
Client: Đăng nhập



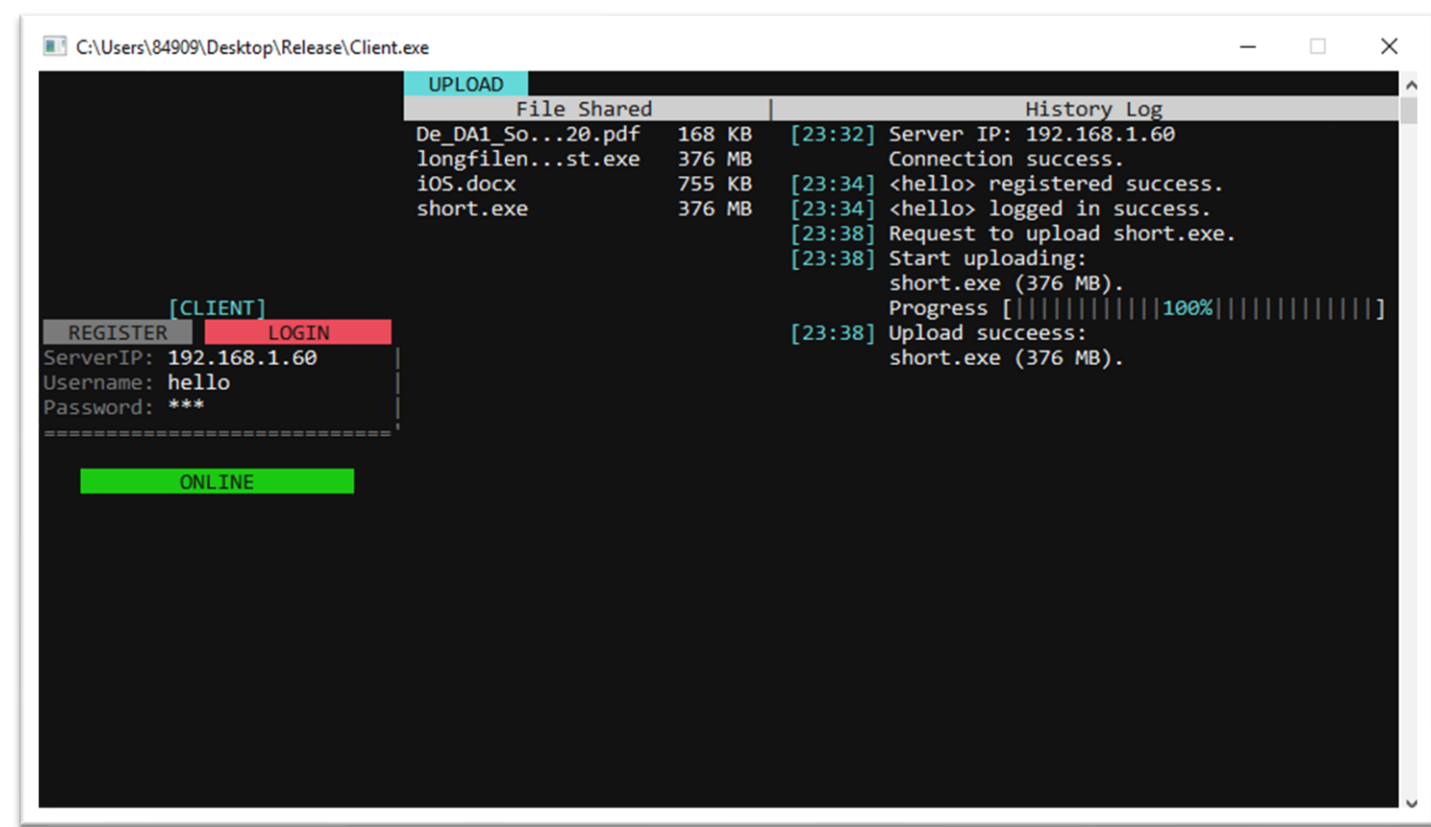
Client: Nhận danh sách file từ Server



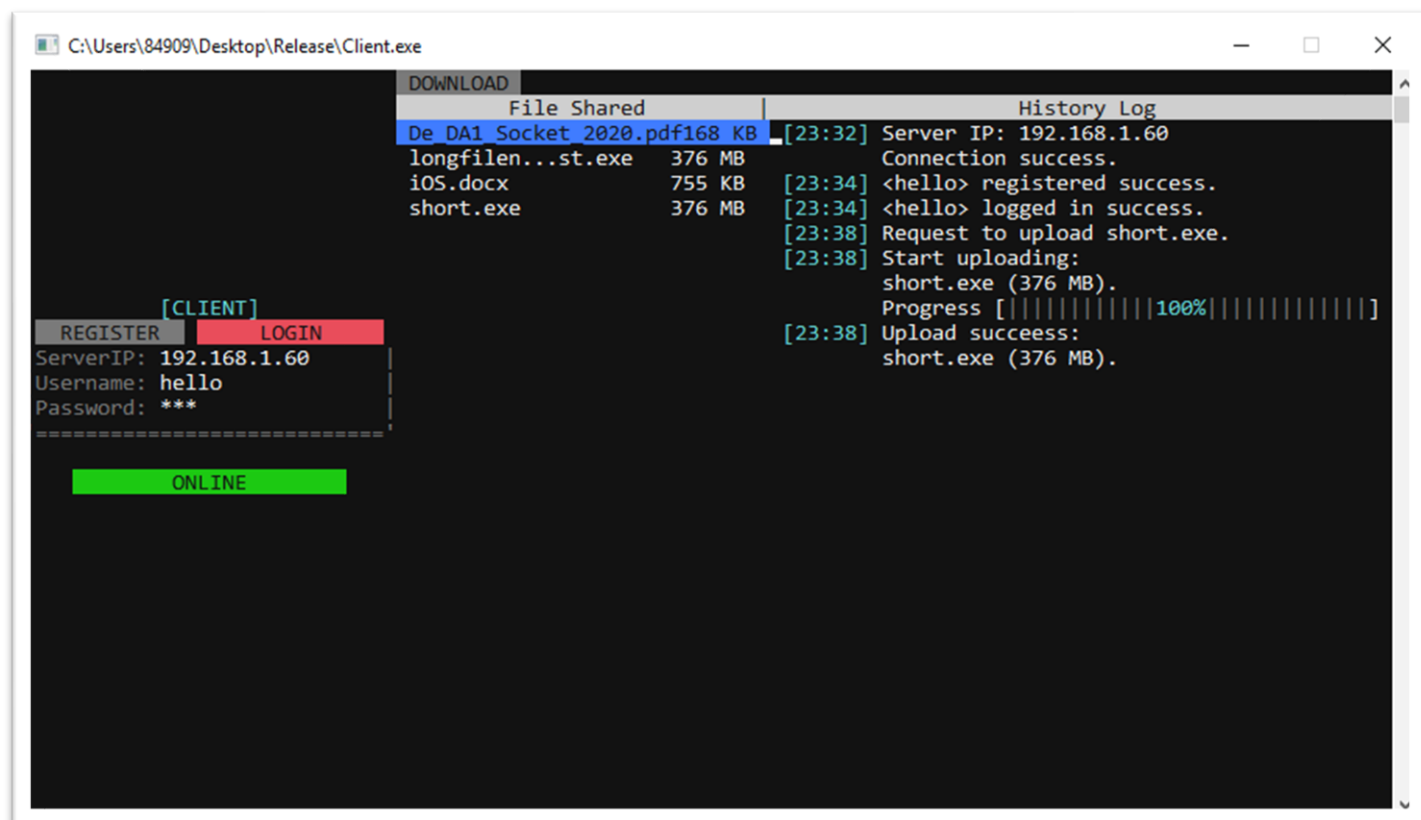
Client: Lúc này mũi tên trái phải sẽ điều khiển UPLOAD/DOWNLOAD. Ở mục UPLOAD nhấn ENTER để nhập đường dẫn file upload.



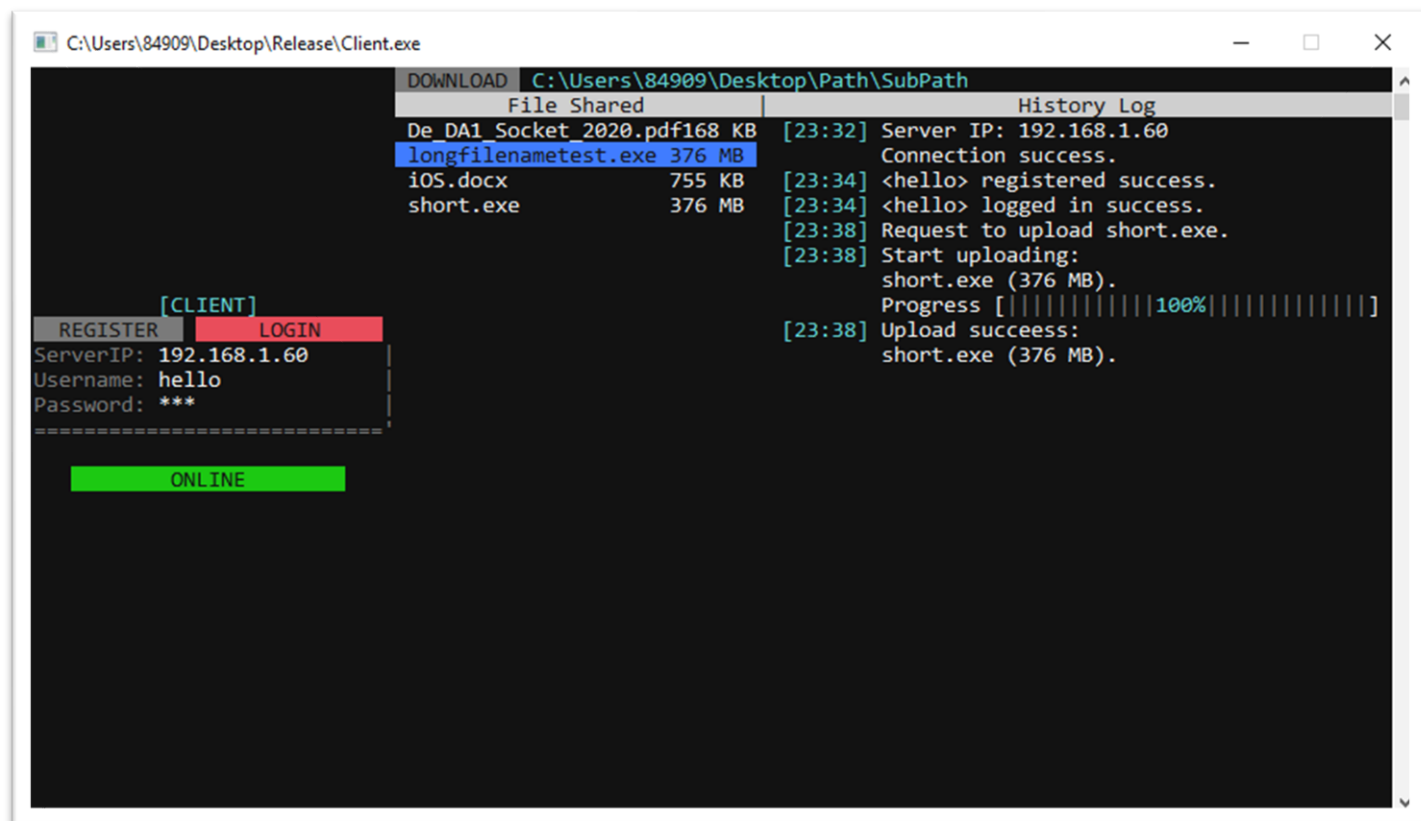
Client: Upload



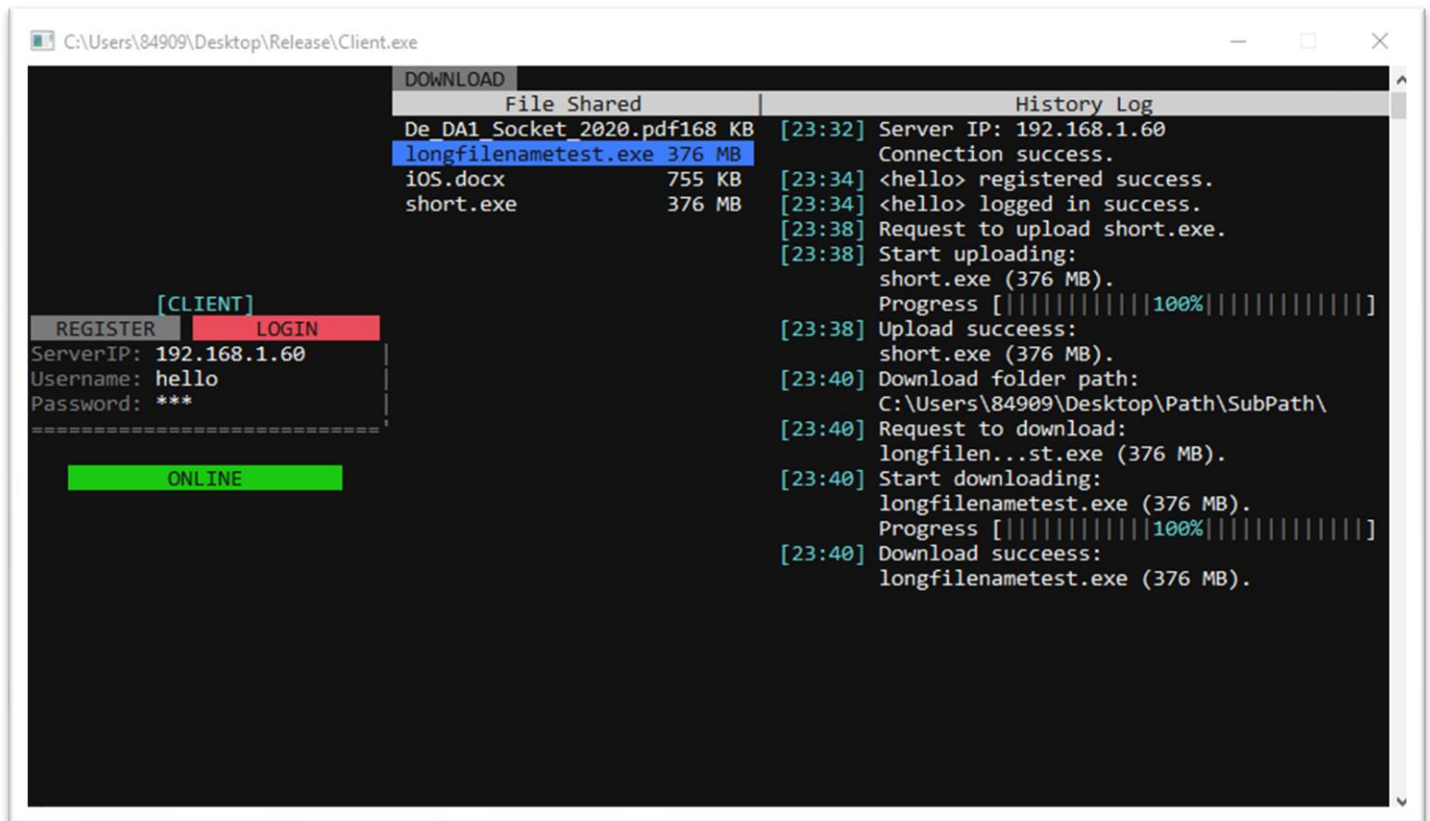
Client: Ấn trái/phải để chuyển sang mode DOWNLOAD



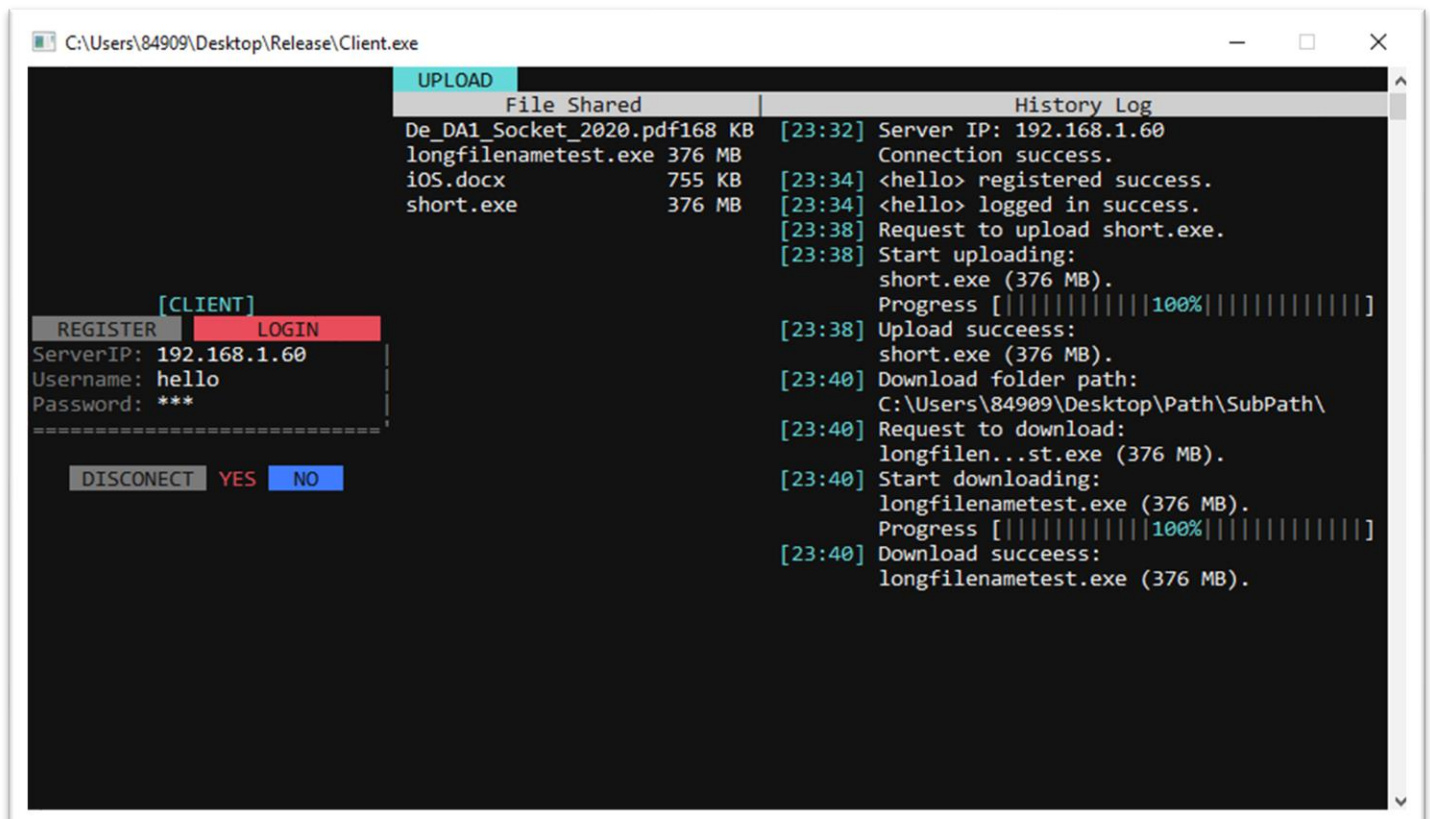
Client: Dùng mũi tên lên/xuống để chọn file muốn download. Sau đó, ENTER và nhập đường dẫn. Nếu ENTER mà không nhập đường dẫn (ENTER 2 lần), Client sẽ tải về thư mục Download mặc định của nó.



Client: Download



Client: Ấn ESC và thao tác tương tự để thoát như Server.



6. Tài liệu tham khảo

Tài liệu Winsock 2 của Microsoft: <https://docs.microsoft.com/en-us/windows/win32/winsock/getting-started-with-winsock>