**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

BÁO CÁO ĐỒ ÁN

SOCKET PROGRAMMING 1

**GIẢNG VIÊN: CHUNG THÙY LINH**

MÔN: MẠNG MÁY TÍNH

|  |  |
| --- | --- |
| **Thành Viên** | **MSVV** |
| Ôn Hạo Nguyên | 18127163 |
| Nguyễn Trịnh An | 18127058 |
| Nguyễn Lê Quốc Tài | 18127203 |

Nội dung: Chương trình mô phỏng ứng dụng File Server gồm một server và nhiều client. Server sẽ lưu trữ file để các client lên tải về. Mọi quá trình truyền file cũng như mọi thông báo (client đăng nhập, client đăng xuất) đều được thể hiện trên màn hình của server.

Yêu cầu:

- Server phải cho phép nhiều client đăng nhập và tải file cùng lúc. Trên lý thuyết, số lượng client là không giới hạn.

- Client cần đăng ký một username và password để đăng nhập. Nếu username đã được sử dụng, server sẽ yêu cầu client chọn username khác cho đến khi không bị trùng. Client sẽ sử dụng tên này trong toàn bộ quá trình trao đổi với server. Tại giao diện server phải có mục thể hiện danh sách tất cả client đang kết nối đến.

- Khi một client đăng nhập hoặc đăng xuất (hoặc thoát chương trình), phải có dòng thông báo trên màn hình server và mọi client.

Ví dụ:

- NguyenVanA log off

- LeVanC log in

- Nếu server thoát, các client đều phải nhận biết là ngắt kết nối.

- Mọi hoạt động upload, download file của client cần ghi log hiển thị trên màn hình của server.

Ví dụ:

- NguyenVanA upload file abc.txt

- LeVanC download file abc.txt

- Tại mỗi client cần hiển thị danh sách các file mà server chia sẻ để client có thể chọn tải về. Lưu ý: Cùng một thời điểm server cho phép nhiều client tải về cùng 1 file và kích thước file có thể lên đến 200 MB

- Cho phép client upload file lên server, server cập nhật danh sách các file mà server chia sẻ đến các client khác. Tại mỗi thời điểm chỉ cho phép 1 client upload 1 file.

Mức độ hoàn thành: 100% (hoàn thành tất cả các mục yêu cầu)

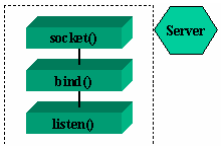
Môi trường lập trình: Visual Studio 2017, 2019

Framework hỗ trợ: MFC (Giao diện), Socket

Kịch bản giao tiếp của chương trình:

\*Mô hình Client-Server sử dụng Socket ở chế độ có kết nối (TCP)

**Giai đọan 1**: Server tạo Socket, gán số hiệu cổng và lắng nghe yêu cầu kết nối



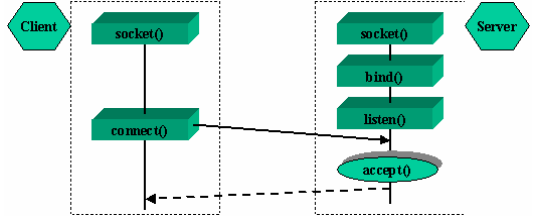
• socket(): Server yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển.

• bind(): Server yêu cầu gán số hiệu port cho socket.

• listen(): Server lắng nghe các yêu cầu kết nối từ các client trên cổng đã được gán.

🡪 Server sẵn sàng phục vụ Client.

**Giai đọan 2**: Client tạo Socket, yêu cầu thiết lập một nối kết với Server

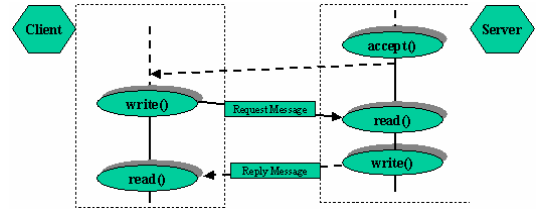


• socket(): Client yêu cầu tạo một socket để có thể sử dụng các dịch vụ của tầng vận chuyển, thông thường hệ thống tự động gán một số hiệu cổng còn trống cho socket của Client.

• connect(): Client gởi yêu cầu nối kết đến server có địa chỉ IP và Port xác định.

• accept(): Server chấp nhận nối kết của client, khi đó một kênh giao tiếp ảo được hình thành,Client và server có thể trao đổi thông tin với nhau.

**Giai đọan 3**: Trao đổi thông tin giữa Client và Server



• Sau khi chấp nhận yêu cầu nối kết, thông thường server thực hiện lệnh read() để đợi cho đến khi có thông điệp yêu cầu (Request Message) từ client gởi đến.

• Server phân tích và thực thi yêu cầu. Kết quả sẽ được gởi về client bằng lệnh write().

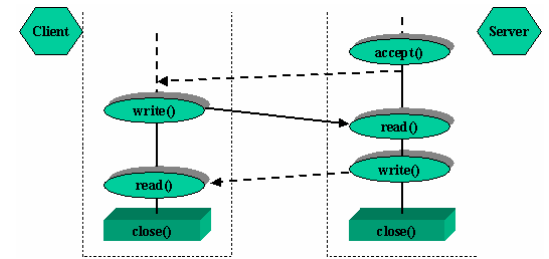
• Sau khi gởi yêu cầu bằng lệnh write(), client chờ nhận thông điệp kết quả (ReplyMessage) từ server bằng lệnh read().

- Trong giai đoạn này, việc trao đổi thông tin giữa Client và Server phải tuân thủ giao thức của ứng dụng (dạng thức và ý nghĩa của các thông điệp, qui tắc bắt tay, đồng bộ hóa, ... ). Thông thường Client sẽ là người gởi yêu cầu đến Server trước.

- Nếu chúng ta phát triển ứng dụng theo các Protocol đã định nghĩa sẳn, chúng ta phải tham khảo và tuân thủ đúng những qui định của giao thức. Bạn có thể tìm đọc mô tả chi tiết của các Protocol đã được chuẩn hóa trong các tài liệu RFC (Request For Comments).

- Ngược lại, nếu chúng ta phát triển một ứng dụng Client-Server riêng của mình, thì công việc đầu tiên chúng ta phải thực hiện là đi xây dựng Protocol cho ứng dụng

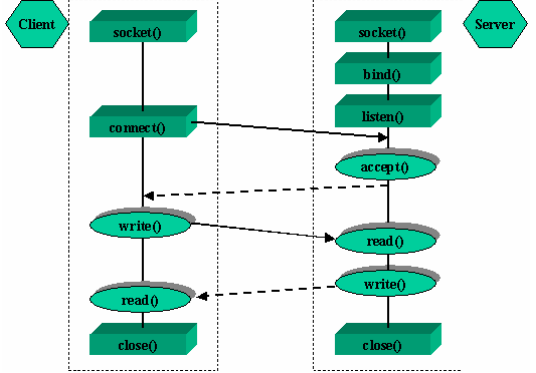
**Giai đoạn 4**: Kết thúc phiên làm việc



• Các câu lệnh read(), write() có thể được thưc hiện nhiều lần (ký hiệu bằng hình elipse).

• Kênh ảo sẽ bị xóa khi Server hoặc Client đóng socket bằng lệnh close().

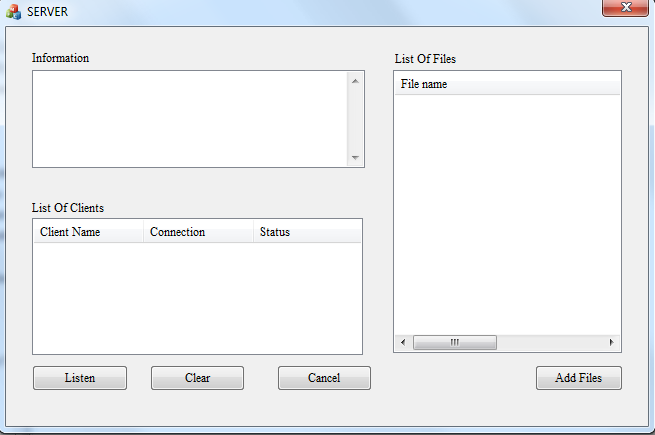
**\*\*Như vậy toàn bộ tiến trình diễn ra như sau:**



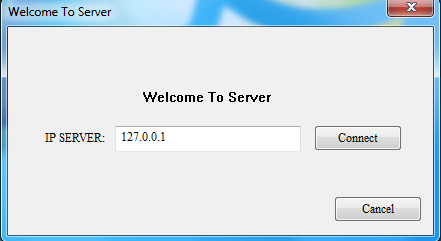
Các tính năng của chương trình

1. Login, Logout và Register:

- Đầu tiên, ta khởi động chương trình MFC\_Server.exe, ta được giao diện như sau:

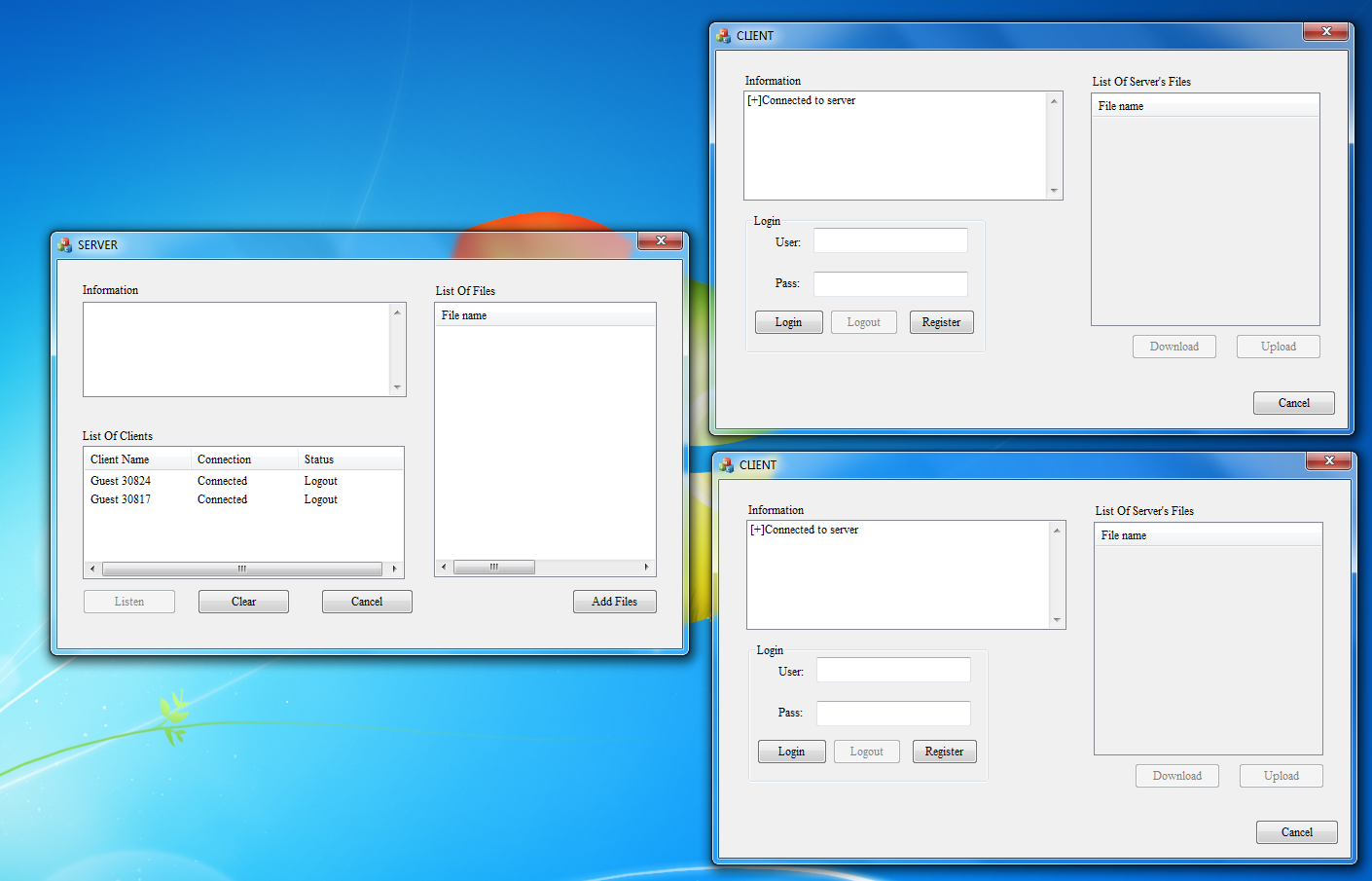


- Nhấn vào **Listen** để Server bắt đầu lắng nghe từ các Client. Sau đó, giả sử ta khởi động 2 chương trình MFC\_Client.exe, nhấn vào **Connect** để kết nào vào server.

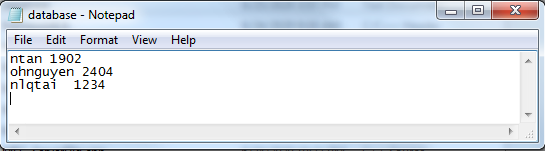


a. Login:

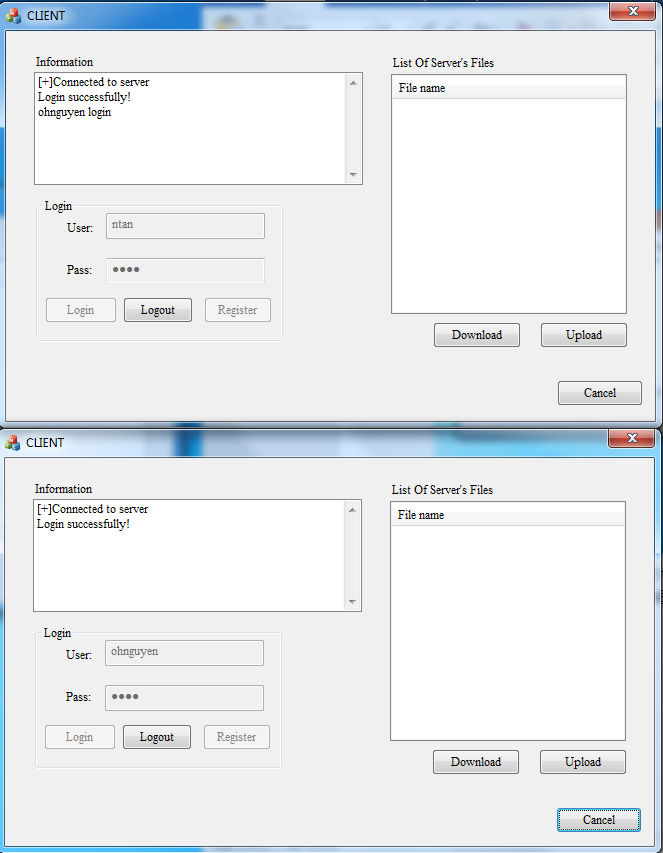
- Sau khi Connect 2 Client vào Server, ta được hiển thị như sau:



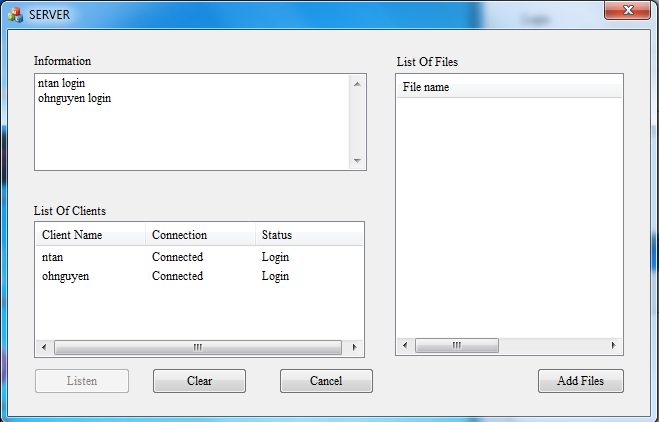
- Sau đó, ta đăng nhập vào 2 tài khoản được lưu sẵn trong *Database.txt* của bên Server.(Tài khoản ***ntan*** với mật khẩu là ***1902*** và ***ohnguyen*** với mật khẩu là ***2404***).



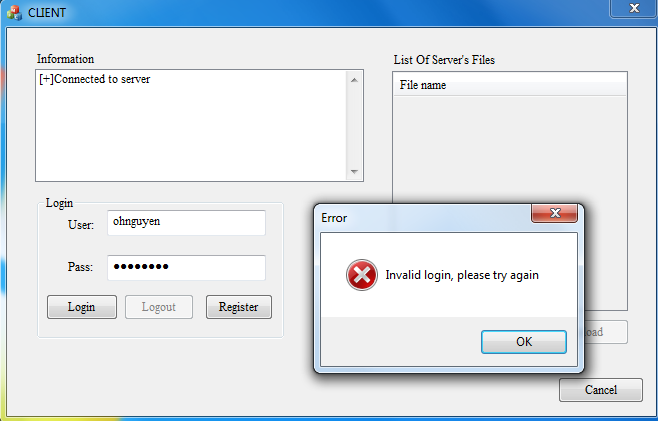
- Thông báo đăng nhập thành công vào Server từ phía Client:



- Và thông báo 2 Client đăng nhập thành công từ phía Server:

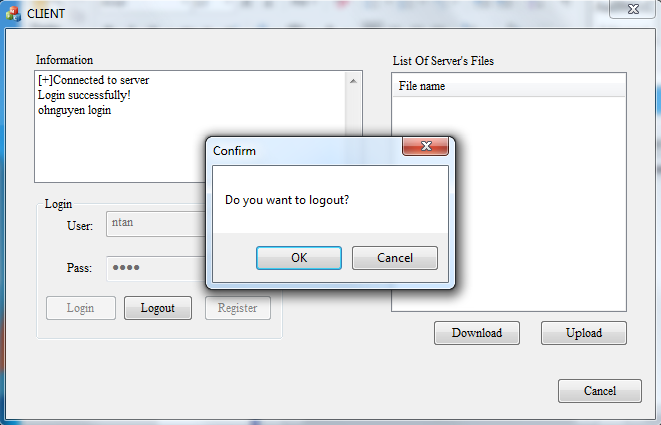


- Trường hợp nhập sai mật khẩu của tài khoản, ta sẽ nhận được thông báo sau:

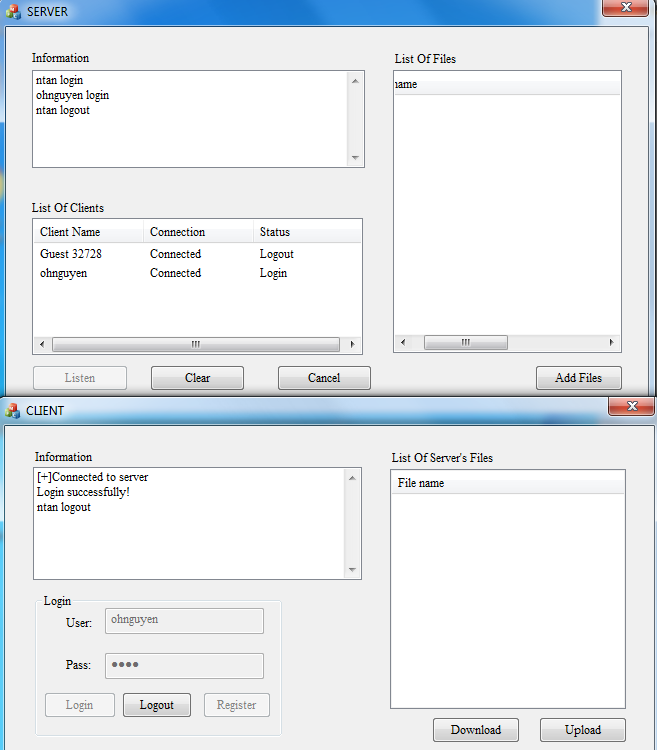


b. Logout:

- Ta thử **Logout** bên Client ntan.

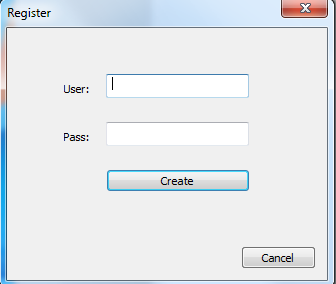


- Cả bên Server và Client đều nhận được thông báo rằng Client ntan vừa log out.

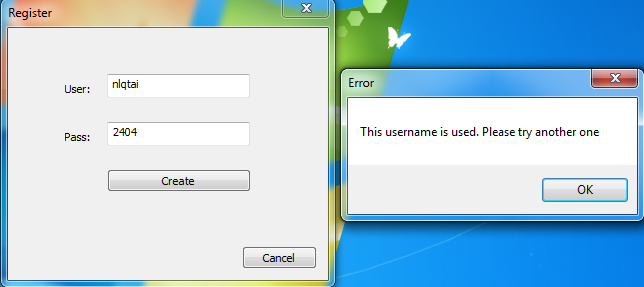


c. Register:

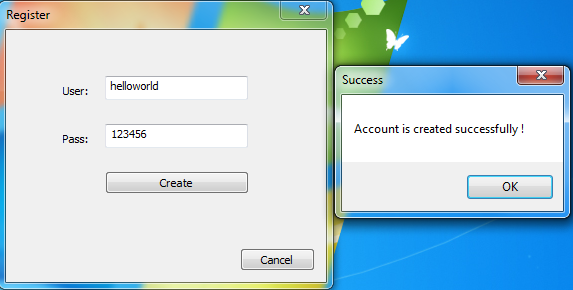
- Tiếp theo, ta sẽ sử dụng tính năng **Register** bên giao diện Client. Ta sẽ sử dụng Client ntan đã log out trước đó để đăng ký tài khoản mới.



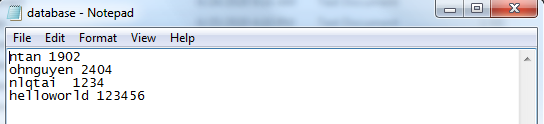
- Giả sử ta tạo User trùng với User đã có trong Database.txt.



- Ngay lập tức, một bảng thông báo hiện lên, báo hiệu rằng tài khoản này đã tồn tại. Ta cần phải tạo một tài khoản có User không trùng với dữ liệu trong Database.txt.

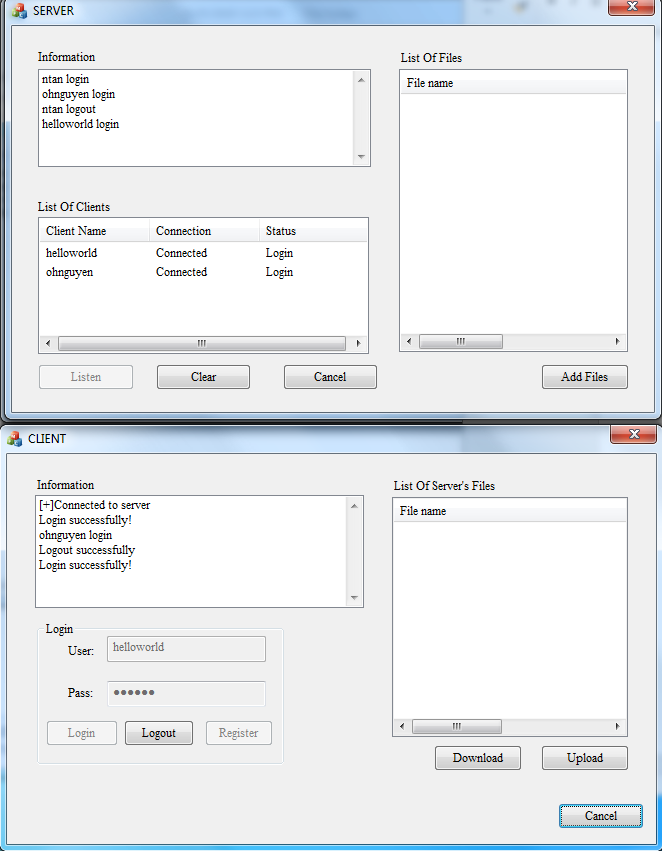


- Tiếp theo, ta thử với một User khác, và kết quả là ta đã tạo thành công một tài khoản mới.



- Trong Database.txt đã được ghi thêm một tài khoản mới.

- Và tài khoản mới đã **Login** thành công vào Server.



2. Download và Upload file:

a. Download file:

Trên dialog based của Server: **Add Files -> MFC\_Server -> Data\_Server1**

**-> Open**. Làm tương tự với **Data\_Server2** và **Data\_Server3**

- Click vào **Add Files**.

A screenshot of a social media post

Description automatically generated

- Chọn file “*Data\_Server1/2/3.txt*” rồi nhấn **Open.**

A screenshot of a social media post

Description automatically generated

- Trên dialog based client sẽ xuất hiện 3 file ta vừa add bên Server

A screenshot of a cell phone

Description automatically generated

- Click vào ô của 1 file bất kỳ để download file đó về bên Client.

A screenshot of a social media post

Description automatically generated

- Sau khi file cần download có dấu tick, nhấn **Download**

A screenshot of a social media post

Description automatically generated

- Thông báo download file “*Data\_Server1*” thành công

- Vào folder **MFC\_Client** để kiểm tra file vừa download.

A screenshot of a computer

Description automatically generated

- File “*Data\_Server1*” xuất hiện bên Client.

A screenshot of a cell phone

Description automatically generated

- Dữ liệu trong file trùng với bên Server.

A screenshot of a social media post

Description automatically generated

- Sự kiện Client download file được hiển thị trên Server.

b. Upload file:

- Trên dialog based Client chọn **Upload -> MFC\_Client -> Data\_Client1 -> Open.**

A screenshot of a cell phone

Description automatically generated

- Thông báo upload file “*Data\_Client1.txt*” thành công.

A screenshot of a cell phone

Description automatically generated

- Sự kiện Client download file và tên file client upload được hiển thị trên Server.

- Tương tự với việc download cùng 1 file ở 2 Client khác nhau:

A screenshot of a social media post

Description automatically generated

- Chọn cùng 1 file “*Data\_Server2*” ở 2 Client khác nhau để download.

A screenshot of a social media post

Description automatically generated

- Thông báo download thành công với cùng 1 file.

A screenshot of a cell phone

Description automatically generated

- Sự kiện 2 client download cùng 1 file được ghi lại bên server.

- Tương tự với file có kích thước lớn (~200MB).

A screenshot of a computer

Description automatically generated

- Thực hiện download file *test.txt* có kích thước 177MB.

A screenshot of a computer

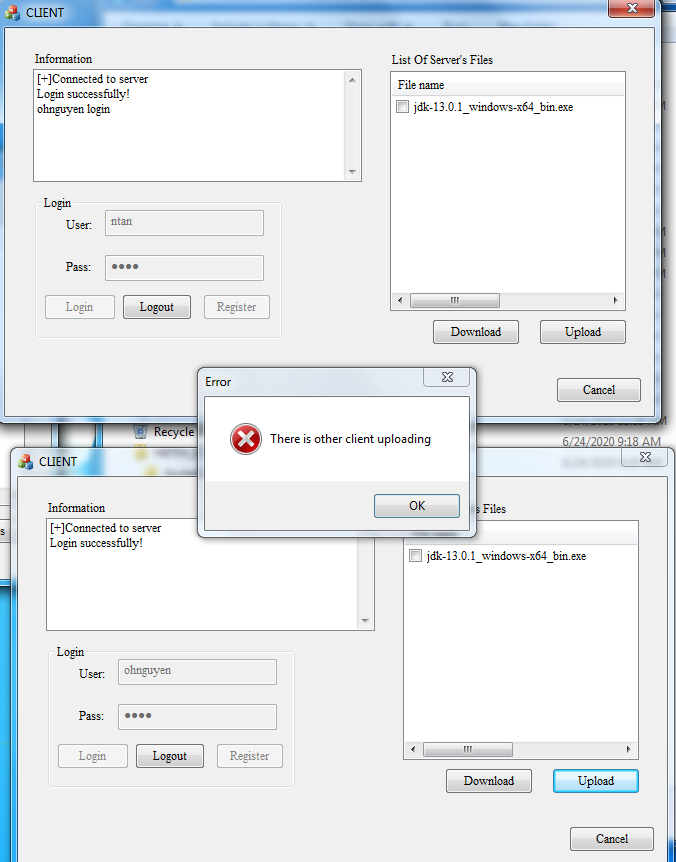
Description automatically generated

- Thông báo download thành công.

A screenshot of a computer

Description automatically generated

- File *test.txt* xuất hiện trong thư mục **MFC\_Client** với kích thước 177MB



- Thông báo lỗi khi 2 Client đồng thời Upload vào 1 thời điểm nhất định.

A screenshot of a computer

Description automatically generated

- *“Data\_Client1”* xuất hiện bên Server.

A screenshot of a social media post

Description automatically generated

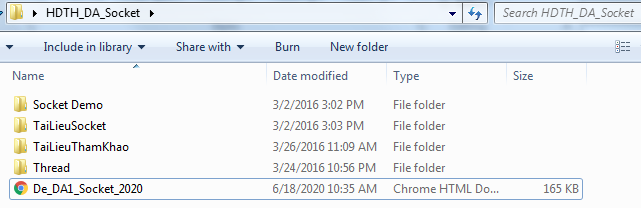
- Dữ liệu trong *“Data\_Client1.txt”* trùng với dữ liệu của Client.

Bảng phân công công việc:

|  |  |
| --- | --- |
| Thành viên | Công việc |
| Ôn Hạo Nguyên | Register + Word (Phần 1) |
| Nguyễn Trịnh An | Login + Logout + Download |
| Nguyễn Lê Quốc Tài | Upload + Word (Phần 2) |

Tài liệu tham khảo:

- Bộ tài liệu do giảng viên cung cấp:



- <https://phattrienphanmem123az.com/lap-trinh-mfc> (Giao diện MFC)

- <https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/> (Thư viện Map)