

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**

BÁO CÁO ĐỒ ÁN



LẬP TRÌNH SOCKET XÂY DỰNG ỨNG DỤNG FTP

**GIẢNG VIÊN HƯỚNG DẪN: LÊ HOÀNG CƯỜNG
THÀNH VIÊN NHÓM:**

1. **VÕ QUỐC THẮNG** - 1712162
2. **LÊ NGUYỄN NHỰT TRƯỜNG** - 1712195
3. **LÊ TUẤN ĐẠT** - 1712329

MỤC LỤC

1. GIỚI THIỆU	2
1.1. Nội dung và yêu cầu của đồ án	2
1.1.1. Nội dung	2
1.1.2. Yêu cầu	2
1.2. Phân tích đồ án	2
1.2.1. Gửi và nhận dữ liệu sử dụng Socket và multi-threading	2
1.2.2. Chức năng đăng nhập	2
1.2.3. Chức năng tải file về từ server	3
1.2.4. Chức năng tải file lên server	3
1.2.5. Tạo và Xóa thư mục trên server	3
1.3. Thông tin nhóm	3
1.3.1. Thông tin thành viên	3
1.3.2. Phân công công việc	3
2. TỔNG QUAN MÔ HÌNH	4
2.1. Ý tưởng	4
2.2. Gói tin gửi và nhận	4
2.2.1. Cấu trúc gói tin	4
2.2.2. Các loại gói tin	5
2.3. Mô hình	5
3. CÀI ĐẶT ĐỒ ÁN	6
3.1. Xây dựng kết nối giữ Server-Client	6
3.1.1. Các lớp xây dựng	6
3.1.2. Socket của server và client	6
3.1.3. Xây dựng các hàm gửi, nhận và xử lý gói tin	7
3.2. Chức năng đăng nhập	7
3.3. Chức năng tải file về từ Server	8
3.4. Chức năng tải file lên Server	8
3.5. Chức năng xóa file trên Server	8
3.6. Chức năng tạo thư mục trên Server	8
3.7. Chức năng xóa thư mục trên Server	8
3.8. Server đóng / Client log out	8
4. GIAO DIỆN VÀ DEMO	9
4.1. Hướng dẫn sử dụng giao diện	9
4.2. Link Demo	13
5. TỔNG KẾT	13
6. ĐÁNH GIÁ	13
6.1. Đánh giá đồ án	13

6.2. Đánh giá cá nhân.....	14
TÀI LIỆU THAM KHẢO.....	15

1. GIỚI THIỆU

1.1. Nội dung và yêu cầu của đề án

1.1.1. 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.

1.1.2. Yêu cầu

- Server phải cho phép nhiều client đăng nhập và tải file cùng lúc.
- Khi một client đăng nhập hoặc đăng xuất (hoặc thoát chương trình), phải có dòng thông báo trên màn hình server và mọi client.
- 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.

1.2. Phân tích đề án

1.2.1. Gửi và nhận dữ liệu sử dụng Socket và multi-threading

Server tạo một socket nằm ở một tiến trình riêng biệt để lắng nghe khi có client kết nối đến. Với mỗi client kết nối đến, server sẽ tạo ra một socket mới và một thread mới nhằm luôn lắng nghe và nhận yêu cầu từ client. Khi client ngắt kết nối, server sẽ hủy tiến trình đó và loại socket đó khỏi danh sách đang quản lý kết nối. Server sẽ dựa trên cấu trúc của gói tin để thực hiện lệnh tương ứng.

1.2.2. Chức năng đăng nhập

Server sẽ quản lý danh sách người dùng thông qua một file (*ClientDatabase.txt*). Nếu tài khoản đã được lưu trữ trong file thì báo người dùng đăng nhập thành công, ngược lại thì yêu cầu người dùng đăng ký tài khoản. Mỗi tài khoản bao gồm hai thông tin UserName và Password.

1.2.3. Chức năng tải file về từ server

Xây dựng chức năng cho phép người dùng có thể thấy danh sách file trong thư mục của server. Client sẽ chọn tên file hiển thị trong danh sách và tiến hành tải file. Lưu ý người dùng có thể bấm chọn nhiều file để tải nhiều file từ server.

1.2.4. Chức năng tải file lên server

- Xây dựng chức năng cho phép người dùng có thể thấy danh sách file trong thư mục của client (không cần yêu cầu đăng nhập). Client sẽ chọn tên file hiển thị trong danh sách và tiến hành đưa file lên server. Lưu ý người dùng có thể bấm chọn nhiều file để đưa nhiều file lên server.
- Chức năng xóa file trên server.
- Trên danh sách file của server, người dùng có thể bấm chọn để xóa file đó trên server folder.

1.2.5. Tạo và Xóa thư mục trên server

- Ngoài ra người dùng còn có thể tạo và xóa thư mục trên server.

1.3. Thông tin nhóm

1.3.1. Thông tin thành viên

MSSV	Họ và tên	Email	Vai trò
1712162	Võ Quốc Thắng	voquochang1999@gmail.com	Developer
1712195	Lê Nguyễn Nhựt Trường	truongthk62014@gmail.com	Developer
1712329	Lê Tuấn Đạt	letuandat2110@gmail.com	Tester

Bảng 1.1 Danh sách thành viên

1.3.2. Phân công công việc

Thành viên	Công việc	Mô tả
Võ Quốc Thắng	Phân công công việc cho các thành viên	Phân tích đề bài, tham khảo nhiều nguồn để đưa ra hướng đi và phân công hợp lý

	Xây dựng cấu trúc của chương trình	Cấu trúc gói tin, Xây dựng kết nối cho server và Client.
	Xây dựng giao diện của chương trình	Sử dụng thư viện MFC để thể hiện giao diện của chương trình
Lê Tuấn Đạt	Xây dựng chức năng tải file từ server	Xây dựng hàm get để người dùng tải file trên 200MB từ server
	Xây dựng chức năng tải file lên server	Xây dựng hàm put để người dùng tải file trên 200MB lên server
	Xây dựng các chức năng thêm	Bao gồm xóa một file, tạo thư mục, xóa thư mục
Lê Nguyễn Nhật Trường	Xử lý đăng nhập	Tính năng đăng nhập của hệ thống
	Xử lý đa luồng	Nghiên cứu và áp dụng đa luồng vào server-client
	Test chương trình	Thử các chức năng trong quá trình làm

Bảng 1.2 Bảng phân chia công việc

2. TỔNG QUAN MÔ HÌNH

2.1. Ý tưởng

- Người dùng tương tác trực tiếp với giao diện của chương trình, với mỗi chức năng nhận được, chương trình sẽ đóng gói (Encapsulation) thành dạng gói tin (Package) theo một chuẩn quy ước trước và tiến hành chuyển lên server. Server sau khi nhận được gói tin sẽ tiến hành giải mã (DeEncapsulation) và thực thi lệnh tương ứng, trả về kết quả cho người dùng (nếu có).
- Mỗi lần client giao tiếp với server, đều được thông báo trên màn hình của server. Server sẽ quản lý các client dựa trên kỹ thuật đa tiểu trình (Multi-Threading).

2.2. Gói tin gửi và nhận

2.2.1. Cấu trúc gói tin

- Mỗi gói tin là một chuỗi ký tự (char *) có cấu trúc như sau:



Hình 2.1 Cấu trúc gói tin

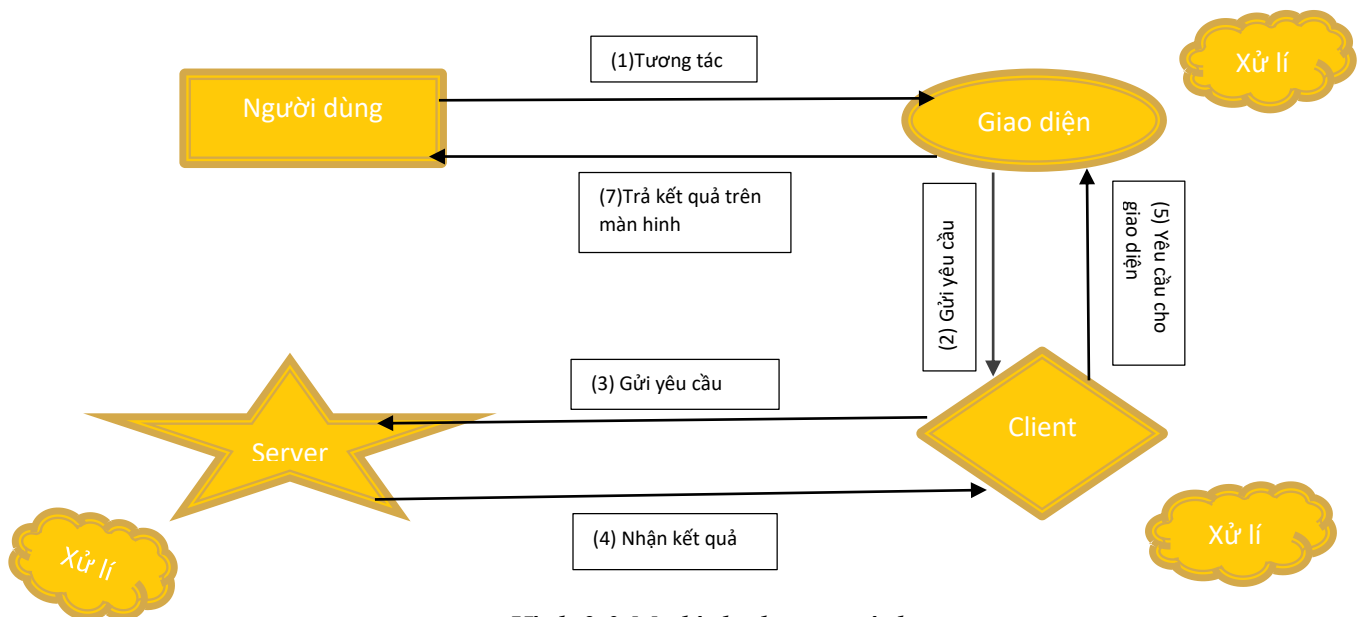
- Mỗi thông tin sẽ được ngăn cách bởi ký tự (\$) và trường FLAG sẽ chứa mã chức năng tương ứng. Message có thể không có hoặc có nhiều Message.
- Vì vậy trên tài khoản của người dùng ,chương trình không cho phép client nhập vào có chứa ký tự '\$'.

2.2.2. Các loại gói tin

FLAG	Nội dung theo sau	Ý nghĩa
LOG_IN	\$_[username]\$_[Password]	Người dùng muốn đăng nhập vào hệ thống
SIGN_UP	\$_[username]\$_[Password]	Người dùng muốn đăng kí tài khoản
LS	\$_[NameOfServerFolder]	Hiển thị thư mục chứa file của folder
PUT_FILE	\$_[NameOfFile]	Nhận file từ người dùng
GET_FILE	\$_[NameOfFile]	Truyền file cho người dùng
DELETE_FILE	\$_[NameOfFile]	Xóa file trên server
MKDIR	\$_[NameOfFolder]	Tạo thư mục trên server
RMDIR	\$_[NameOfFolder]	Xóa thư mục trên server
101	NULL	Thông báo đăng nhập thất bại
102	NULL	Thông báo đăng nhập thành công
103	NULL	Đăng ký thành công
104	NULL	Đăng ký thất bại
105	\$_[list of file names]	Danh sách tên các file có trong thư mục của server
106	NULL	Server is working on it

Hình 2.2 Bảng các loại gói tin

2.3. Mô hình



Hình 2.3 Mô hình chương trình

3. CÀI ĐẶT ĐỒ ÁN

3.1. Xây dựng kết nối giữa Server-Client

3.1.1. Các lớp xây dựng

Tên lớp	Chức năng
CServerDlg	Giao diện chính của Server
CClientDlg	Giao diện chính của Client
Tokenzier	Tách chuỗi. Sử dụng để lấy được thông tin trên gói tin nhận được
Converter	Thực hiện chuyển đổi qua lại các kiểu dữ liệu

3.1.2. Socket của server và client

- Ở Server, **CSocket** *listen* sẽ chịu trách nhiệm lắng nghe kết nối từ các client (đặt trong một tiểu trình để luôn nhận tín hiệu từ client). Khi các kết nối được Accept, Server sẽ tạo một CSocket mới để duy trì kết nối này và sẽ được lưu trữ trong một **vector** `<CSocket*> clients`
- Ở Client, **CSocket** *client* sẽ chịu trách nhiệm kết nối với server nếu có tín hiệu

3.1.3. Xây dựng các hàm gửi, nhận và xử lý gói tin

- Hàm send được xây dựng để gửi một gói tin (cấu trúc char *) dựng trên hàm Send có sẵn trong thư viện `<afxsock.h>`

```
bool send(CSocket & socket, char * ServerMsg) {
    int MsgSize;

    MsgSize = strlen(ServerMsg);

    if( socket.Send(&MsgSize, sizeof(MsgSize), 0) == SOCKET_ERROR) return 0;

    socket.Send(ServerMsg, MsgSize, 0);

    return 1;
}
```

- Hàm receive được xây dựng để gửi một gói tin (cấu trúc char *) dựng trên hàm Receive có sẵn trong thư viện `<afxsock.h>`

```
char * receive(CSocket & client) {
    int MsgSize;

    char *temp;

    if (client.Receive((char*)&MsgSize, sizeof(int), 0) == 0) {
        return "";
    } // Nếu nhận lỗi thì trả về là SOCKET_ERROR.

    temp = new char[MsgSize + 1];

    client.Receive((char*)temp, MsgSize, 0);

    temp[MsgSize] = '\0';

    return temp;
}
```

- Hàm handleMessage được xây dựng để xử lý gói tin nhận được thành các lệnh thực thi qua lại giữa server và client.

3.2. Chức năng đăng nhập

- Người dùng điền các thông tin tài khoản theo yêu cầu (Lưu ý: Tên không được trùng với dữ liệu đã có và không bao gồm ký tự '\$') Chương trình nhận được thông tin khi người dùng bấm Login / Signup. Với mỗi lệnh Server nhận được mã lệnh cùng thông tin tài khoản từ Client, Server tiến hành kiểm tra trên dữ liệu khách hàng và trả về trạng thái thành công hay không thành công cho người dùng.

- Các hàm xử lý liên quan trong chương trình: *CServerDlg::logIn(), CServerDlg::signUp*

3.3. Chức năng tải file về từ Server

- Thực hiện việc tải file từ client lên server, client sẽ mở file và đọc một lần lớn nhất là 2048 bytes và truyền đi 2048 bytes.
- Bên server sẽ nhận dữ liệu truyền từ client một lần nhận lớn nhất là 2048 bytes từ client.
- Các hàm xử lý liên quan trong chương trình: *CServerDlg::get(), CClientDlg::get()*

3.4. Chức năng tải file lên Server

- Tương tự như việc tải file từ client lên server, ở đây server sẽ đọc file nếu không có sẽ báo cho client, mỗi lần đọc lớn nhất 2048 bytes và truyền xuống cho client.
- ở client sẽ tạo một file có tên tương tự và ghi dữ liệu nhận được vào, mỗi lần nhận ghi lớn nhất là 2048 bytes.
- Lưu ý: trong quá trình tải file lên, hay tải file xuống thì client sẽ không thực hiện được các công việc khác.
- Các hàm xử lý liên quan trong chương trình: *CServerDlg::put(), CClientDlg::put()*

3.5. Chức năng xóa file trên Server

- Client sẽ gửi tên file qua cho server thực hiện lệnh xóa thông qua hàm `remove(char* filename)` của thư viện `stdio.h`, nếu như file không tồn tại server sẽ báo lại cho client biết.
- Các hàm xử lý liên quan trong chương trình:
CServerDlg::deleteFile(), CClientDlg::deleteFile().

3.6. Chức năng tạo thư mục trên Server

- Hàm tạo thư mục trên server.
- Client sẽ gửi tên thư mục qua cho server và server sẽ thực hiện lệnh tạo thư mục từ thư viện `direct.h` là `mkdir(char *name)`.
- Các hàm xử lý liên quan trong chương trình: *CServerDlg::mkdir(), CClientDlg::mkdir()*

3.7. Chức năng xóa thư mục trên Server

- Hàm xóa thư mục trên server
- Client sẽ gửi tên thư mục cần xóa cho server, nếu thư mục rỗng server sẽ thực hiện lệnh xóa thông qua hàm `rmdir(char* name)` của thư viện `direct.h`, nếu thư mục không rỗng thì server sẽ báo thư mục không xóa được.

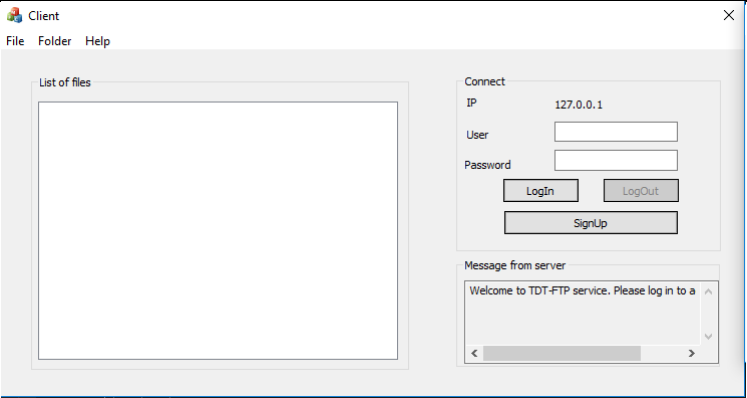
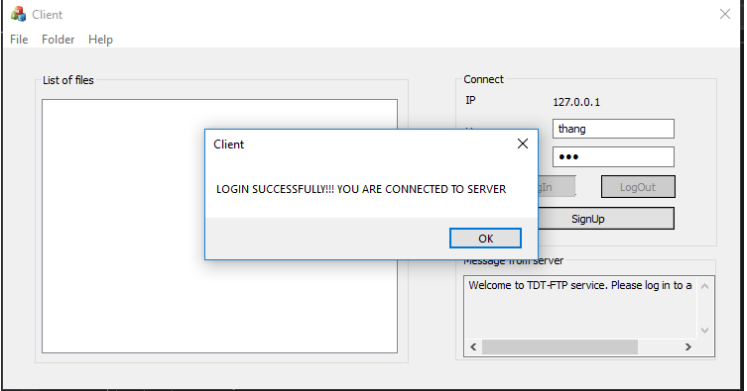
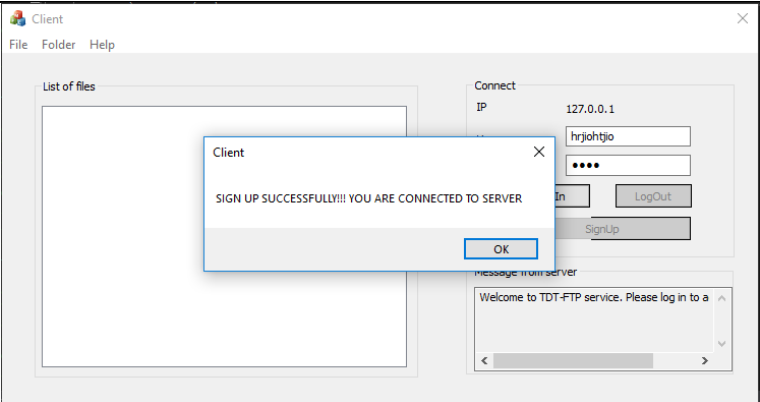
3.8. Server đóng / Client log out

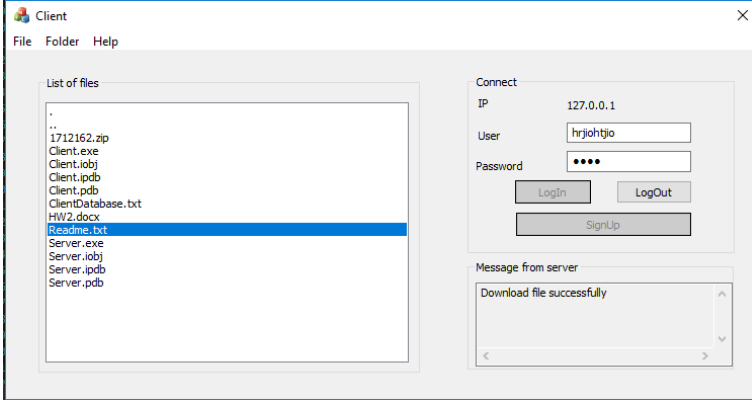
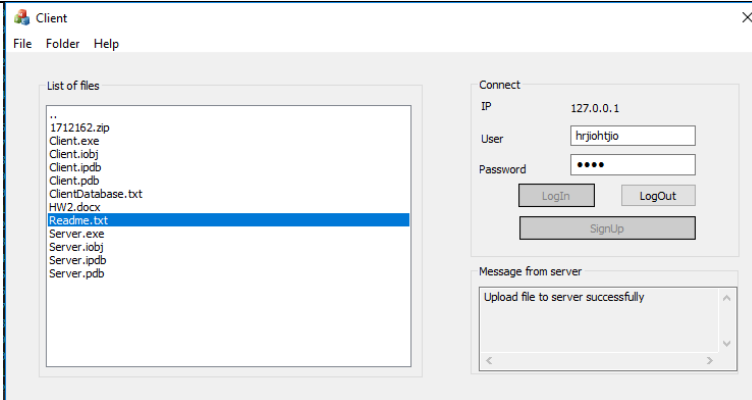
- Khi một trong hai bên đóng kết nối thì chương trình sẽ báo lại cho bên còn lại.

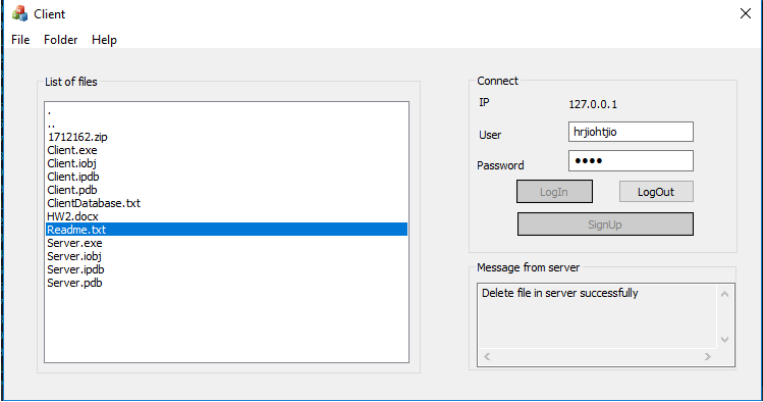
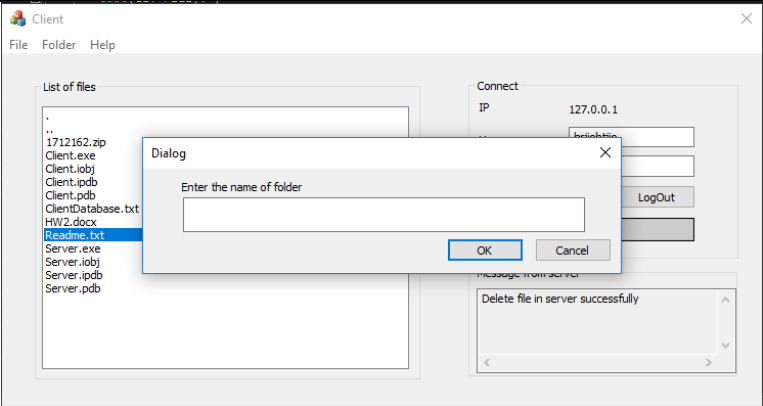
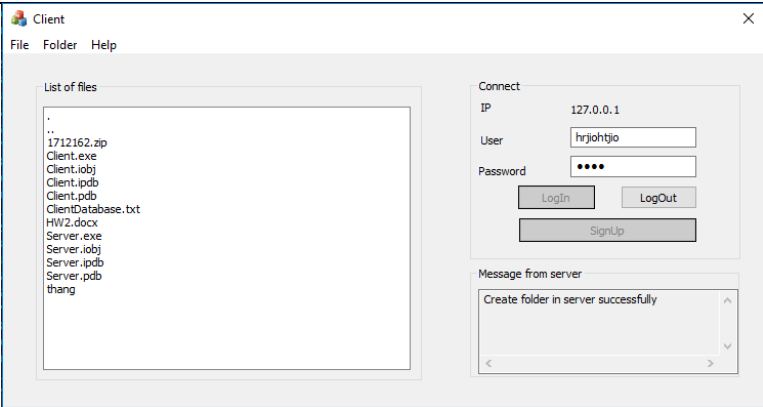
- Cụ thể là: Khi Client đóng, Server sẽ hiện thông báo lên màn hình “777 is out of connection”. Khi Server đóng, Chương trình sẽ yêu cầu người dùng chờ đến khi Server mở lại.
- Các hàm liên quan trong chương trình: *CServerDlg::OnCancel, CClientDlg::logOut*

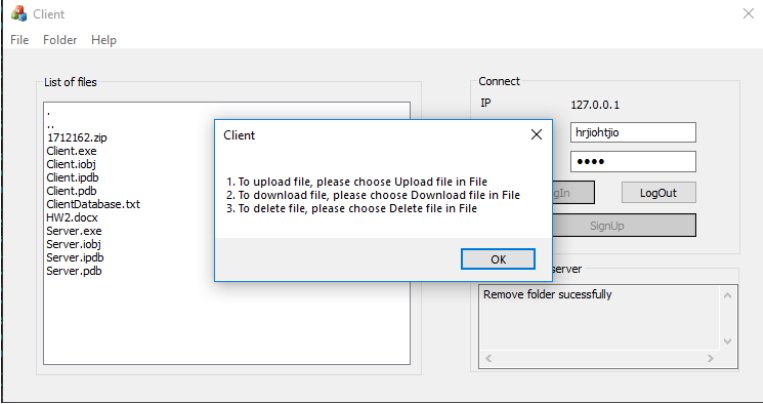
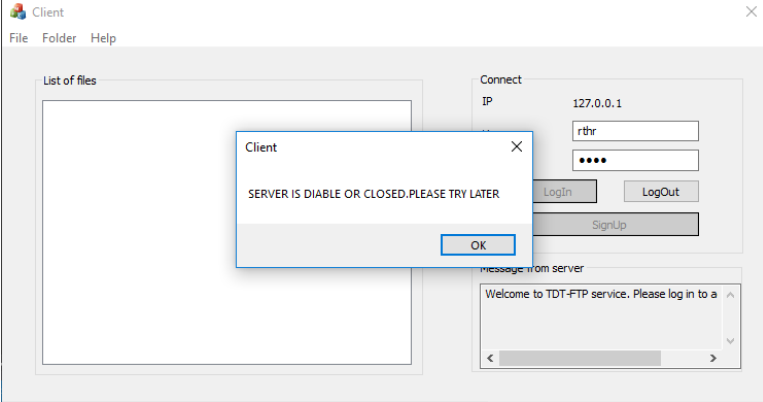
4. GIAO DIỆN VÀ DEMO

4.1. Hướng dẫn sử dụng giao diện

STT	Chức năng	Màn hình	Mô tả
1	Khởi tạo		Chương trình Client hiển thị với dòng chữ chào mừng đến với hệ thống
2	LogIn		Người dùng điền thông tin trên khung và bấm nút LogIn để truy cập vào hệ thống. Nếu thành công sẽ trả kết quả như hình
3	SignUp		Người dùng điền thông tin trên khung và bấm nút SignUp để đăng ký tài khoản. Nếu thành công sẽ

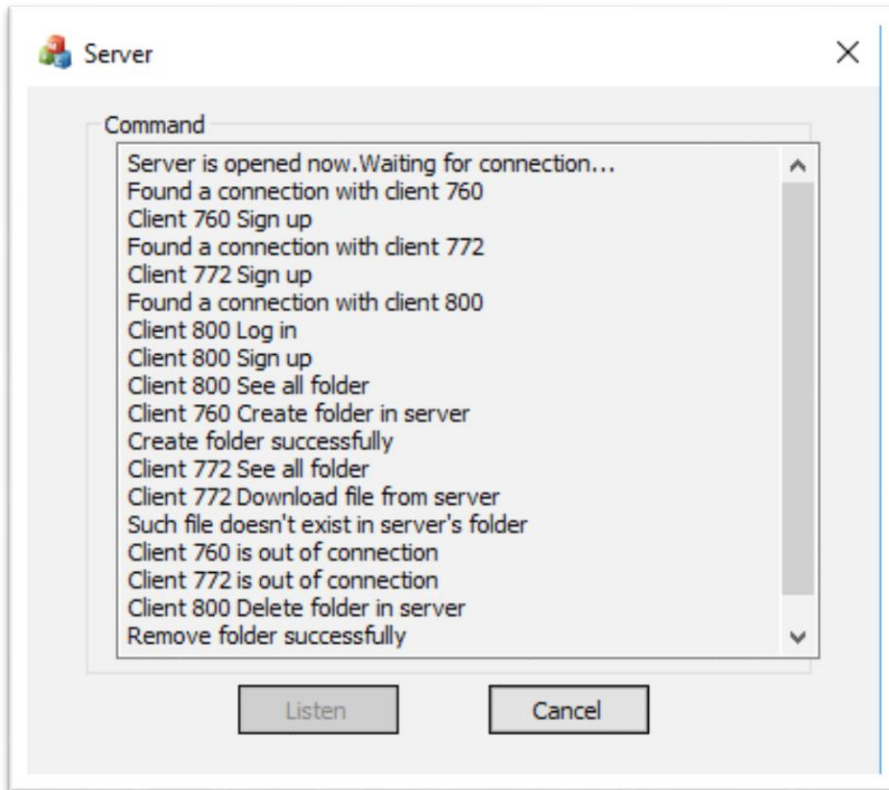
			<p>trả kết quả như hình. Lưu ý: Sau khi đăng ký thành công, bạn đã đăng nhập vào hệ thống</p>
4	Download File		<p>Bấm vào File trên thanh công cụ và bấm chọn Download File. Màn hình hiển thị danh sách file trong server. Người dùng nhấp chuột vào tên file muốn tải</p>
5	Upload File		<p>Bấm vào File trên thanh công cụ và bấm chọn Upload File. Màn hình hiển thị danh sách file trong client folder. Người dùng nhấp chuột vào tên file muốn tải file lên</p>

6	Delete file		Bấm vào File trên thanh công cụ và bấm chọn Delete File. Màn hình hiển thị danh sách file trong server. Người dùng nhấp chuột vào tên file để xóa file
7	Create / Delete file		Bấm Folder trên thanh công cụ ,chọn Create Folder để tạo thư mục trên server. Một yêu cầu xuất hiện yêu cầu người dùng nhập tên folder.
8	Create / Delete file		Màn hình trả về kết quả tạo file sau khi bấm OK.

9	Help	 <p>The screenshot shows the Client application window. It has a menu bar with 'File', 'Folder', and 'Help'. Below the menu is a 'List of files' area containing a list of files: 1712162.zip, Client.exe, Client.job, Client.pdb, ClientDatabase.txt, HW2.docx, Server.exe, Server.job, Server.pdb, and Server.pdb. To the right of the file list is a 'Connect' section with fields for IP (127.0.0.1), Username (hrjohtjo), and Password (masked with dots). There are buttons for 'Login', 'Logout', and 'SignUp'. A 'Remove folder successfully' message is visible at the bottom right. A 'Client' dialog box is open in the center, displaying instructions: '1. To upload file, please choose Upload file in File', '2. To download file, please choose Download file in File', and '3. To delete file, please choose Delete file in File'. The dialog has an 'OK' button.</p>	Bấm Help trên thanh công cụ để xem hướng dẫn sử dụng
10	Khi Server tắt hoặc chưa mở kết nối	 <p>The screenshot shows the Client application window. It has a menu bar with 'File', 'Folder', and 'Help'. Below the menu is a 'List of files' area which is currently empty. To the right of the file list is a 'Connect' section with fields for IP (127.0.0.1), Username (rthr), and Password (masked with dots). There are buttons for 'Login', 'Logout', and 'SignUp'. A 'Message from server' area at the bottom right displays the text: 'Welcome to TDT-FTP service. Please log in to a'. A 'Client' dialog box is open in the center, displaying the message: 'SERVER IS DIALBE OR CLOSED.PLEASE TRY LATER'. The dialog has an 'OK' button.</p>	Người dùng sẽ nhận biết khi sử dụng các chức năng khi gặp thông báo như hình

Bảng 4.1 Hướng dẫn sử dụng cho Client

Đối với giao diện Server. Bấm Listen để bắt đầu mở kết nối với Client. Bấm Cancel để tắt server.



Hình 4.2 Màn hình Server

4.2. Link Demo

<https://youtu.be/aFBNDXgQ4Rg>

5. TỔNG KẾT

Qua đồ án nhóm đã biết cách hai máy có thể giao tiếp với nhau thông qua **lập trình socket**, tìm hiểu về mô hình **FTP(File transfer protocol)**, có thể xây dựng một chương trình FTP đơn giản.Nắm được bản chất khi truyền dữ liệu qua Internet và cách đọc ghi dữ liệu.Ngoài ra còn trang bị thêm một số kiến thức liên quan đến lập trình giao diện **MFC** và **xử lý đa luồng**.

6. ĐÁNH GIÁ

6.1. Đánh giá đồ án

STT	Yêu cầu	Kết quả đạt được	Mức độ hoàn thành
1	Xây dựng hệ thống giao tiếp giữa server - client	Hoàn thành kết nối giữa 1 server và nhiều client.Xử lý song song	100%

2	Hệ thống đăng nhập	Cài đặt đăng ký và đăng nhập dựa trên các ràng buộc cho trước. Ngoài ra dữ liệu được lưu trong file để quản lý	100%
3	Truyền và tải file trên 200MB	Chương trình hoàn toàn có thể download hay upload file trên 200MB trong thời gian vài giây. Ngoài ra, chương trình có thể tải file lên đến 4GB trong thời gian 5 phút (Lần thử gần nhất) Lưu ý: Người dùng có thể chọn nhiều file cùng một lúc để tải/up	100%
4	Server hiển thị trạng thái hoạt động với mỗi người dùng	Màn hình server trình chiếu tất cả các hoạt động của tất cả người dùng trên màn hình, ngay cả khi người dùng rời khỏi hệ thống	100%
5	Xử lý với nhiều người dùng	Chương trình có thể xử lý với nhiều client (theo lý thuyết), Các client đều có thể tải/up file cùng một lúc	100%
6	Xóa file, Tạo / Xóa thư mục(thêm)	Cho phép người dùng xóa file trên server, tạo và xóa thư mục trên server	100%

Bảng 6.1 Đánh giá đồ án

6.2. Đánh giá cá nhân

Thành viên	Khối lượng công việc	Mức độ hoàn thành
Võ Quốc Thắng	35%	100%
Lê Tuấn Đạt	35%	100%
Lê Nguyễn Nhật Trường	30%	100%

Bảng 6.2. Đánh giá cá nhân

TÀI LIỆU THAM KHẢO

- [1] <http://www.hoangvancong.com/2017/03/03/mfc-socket-in-mfc/>
- [2] <https://docs.microsoft.com/en-us/windows/desktop/api/winsock/>
- [3] <https://docs.microsoft.com/en-us/cpp/mfc/reference/csocket-class?view=vs-2019>
- [4] <https://www.geeksforgeeks.org/socket-programming-cc/>
- [5] <https://stackoverflow.com/questions/3509011/socket-programming-in-c>
- [6] <https://www.geeksforgeeks.org/multithreading-in-cpp/>
- [7] https://www.tutorialspoint.com/cplusplus/cpp_multithreading.htm
- [8] [http://forums.codeguru.com/showthread.php?317429-Why-do-I-get-WSAEWOULDBLOCK-when-calling-recv\(\)](http://forums.codeguru.com/showthread.php?317429-Why-do-I-get-WSAEWOULDBLOCK-when-calling-recv())
- [9] <https://social.msdn.microsoft.com/Forums/en-US/home>