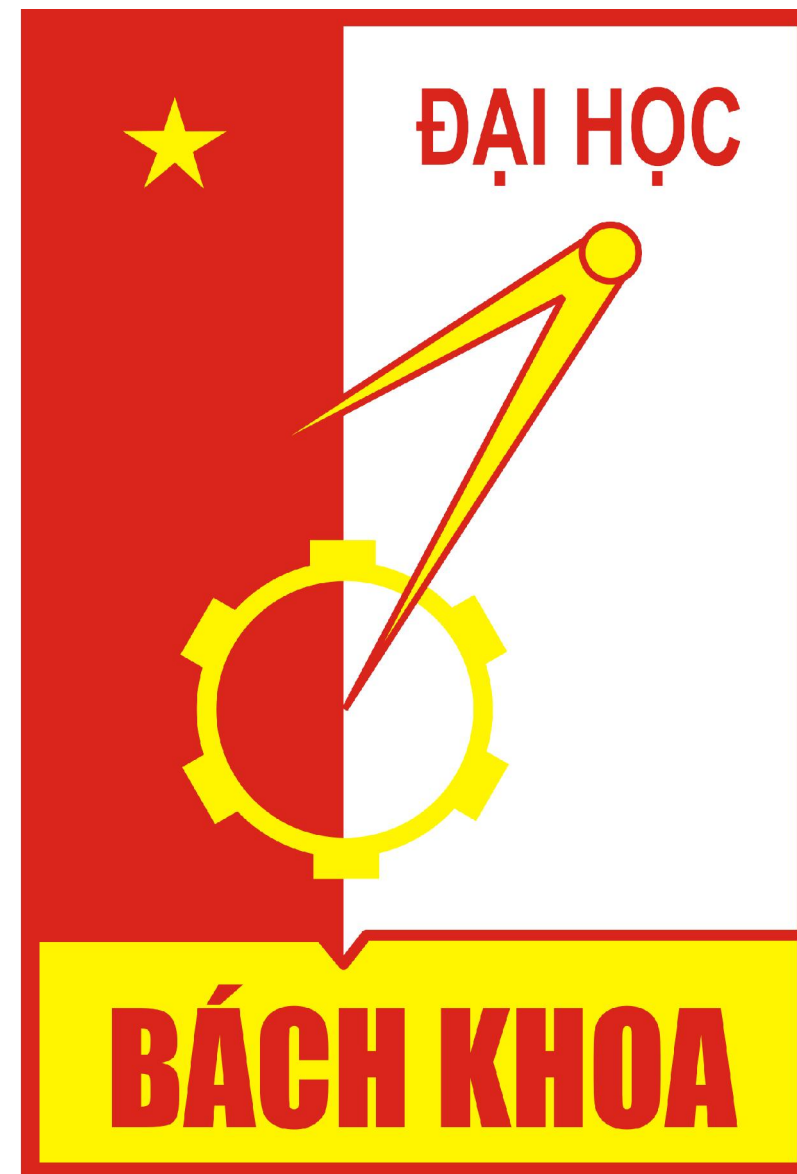


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO  
BÀI TẬP LỚN  
MÔN LẬP TRÌNH MẠNG

Đàm Minh Tiến 20156599

Vương Ngọc Quỳnh Hương



A decorative graphic consisting of several concentric circles in orange, purple, green, blue, and pink. Four large colored dots (pink, blue, green, purple) are positioned along the top arc of the circles, and three small black dots are positioned along the bottom arc.

# Đề tài: FTP Server

Mã đề tài: 2

# Mục lục

- 1 Giới thiệu về giao thức FTP
- 2 Phân tích đề tài
- 3 Thiết kế cấu trúc và các lớp đối tượng
- 4 Thực thi và demo kết quả
- 5 Kết luận

# I. Giới thiệu về giao thức FTP

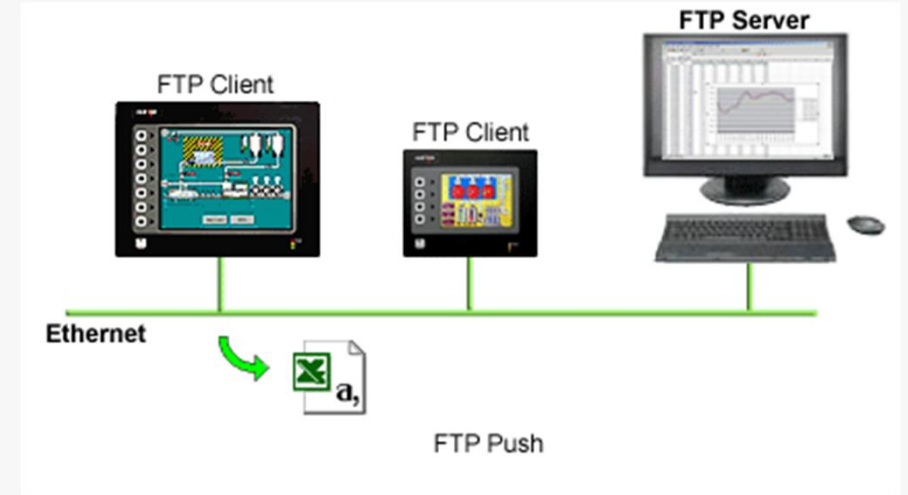
---

Chương này trình bày giới thiệu chung về khái niệm và mô hình hoạt động của giao thức FTP

FTP Server



# File Transfer Protocol ?

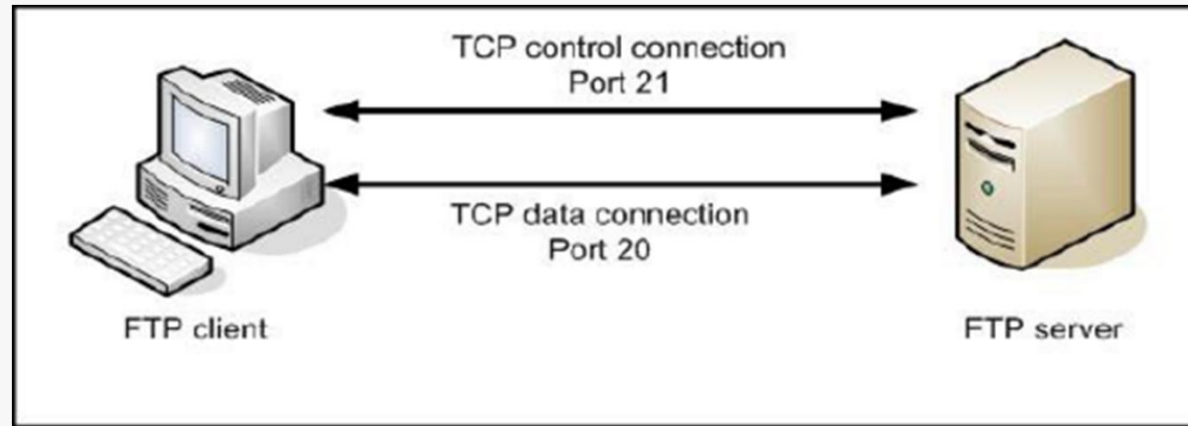


Giao thức FTP được đề xuất lần đầu tiên trong bản thảo RFC 114 (1971), và chính thức ra đời năm 1973.

FTP được tạo ra với mục tiêu chính là cho phép những máy tính trên 1 mạng gián tiếp có thể dễ dàng chuyển file từ máy tính này sang máy tính khác.

Để đảm bảo rằng các tập tin được gửi và nhận được mà không làm mất dữ liệu có thể hỏng chúng, FTP sử dụng giao thức điều khiển truyền dẫn tin cậy tại tầng giao vận.

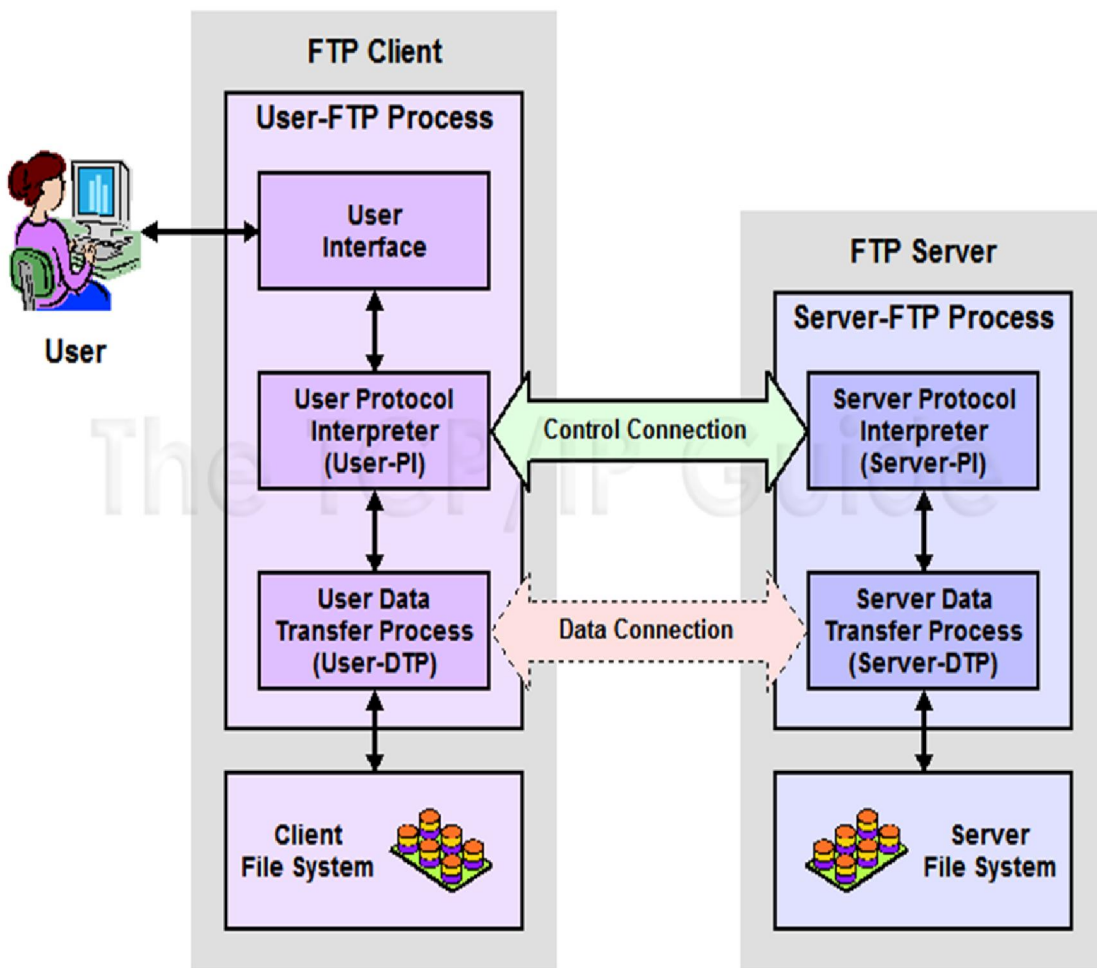
# Mô hình hoạt động



*FTP gồm hai đường: kiểm soát và dữ liệu*

- **Control connection:** Đây là kết nối TCP logic chính được tạo ra khi phiên làm việc được thiết lập. Nó được duy trì trong suốt phiên làm việc và chỉ cho các thông tin điều khiển đi qua ví dụ như lệnh và trả lời. Nó không được sử dụng để gửi dữ liệu.
- **Data connection:** Mỗi khi dữ liệu được gửi từ sever tới client hoặc ngược lại, một kết nối dữ liệu TCP riêng biệt được thiết lập giữa chúng. Dữ liệu được truyền qua kết nối này. Khi hoàn tất việc truyền dữ liệu, kết nối được hủy bỏ.





**Figure 288: File Transfer Protocol (FTP) Operational Model**

FTP is a client/server protocol, with communication taking place between the *User-FTP Process* on the client and the *Server-FTP Process* on the server. Commands, replies and status information are passed between the *User-PI* and *Server-PI* over the *control connection*, which is established once and maintained for the session. Data is moved between devices over *data connections* that are set up for each transfer.

## Tiến trình bên phía server:

**Server Protocol Interpreter (Server-PI):** chịu trách nhiệm quản lý điều khiển kết nối trên server. Nó lắng nghe yêu cầu kết nối hướng từ users trên cổng dành riêng. Khi kết nối đã được thiết lập, nó nhận lệnh từ User-PI, gửi trả lời lại và quản lý tiến trình truyền dữ liệu trên server.

**Server Data Transfer Process (Server-DTP):** Làm nhiệm vụ gửi hoặc nhận file từ bộ phận User-DTP. Server DTP vừa làm nhiệm vụ thiết lập kết nối kênh dữ liệu và lắng nghe một kết nối kênh dữ liệu từ user. Nó tương tác với server file trên hệ thống cục bộ để đọc và chép file.

# Trình tự truy cập và chứng thực FTP:

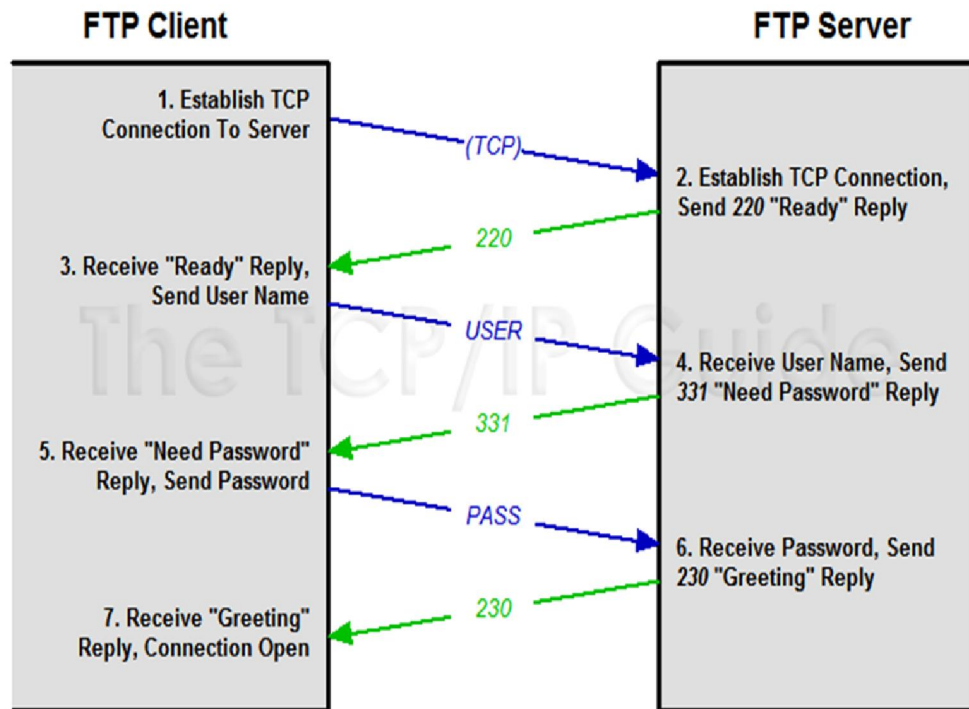


Figure 289: FTP Connection Establishment and User Authentication

An FTP session begins with the establishment of a TCP connection between the client and server. The client then sends the user name and password to authenticate with the server. Assuming that the information is accepted by the server, it sends a greeting reply to the client and the session is open.

# Quản lí kênh dữ liệu FTP

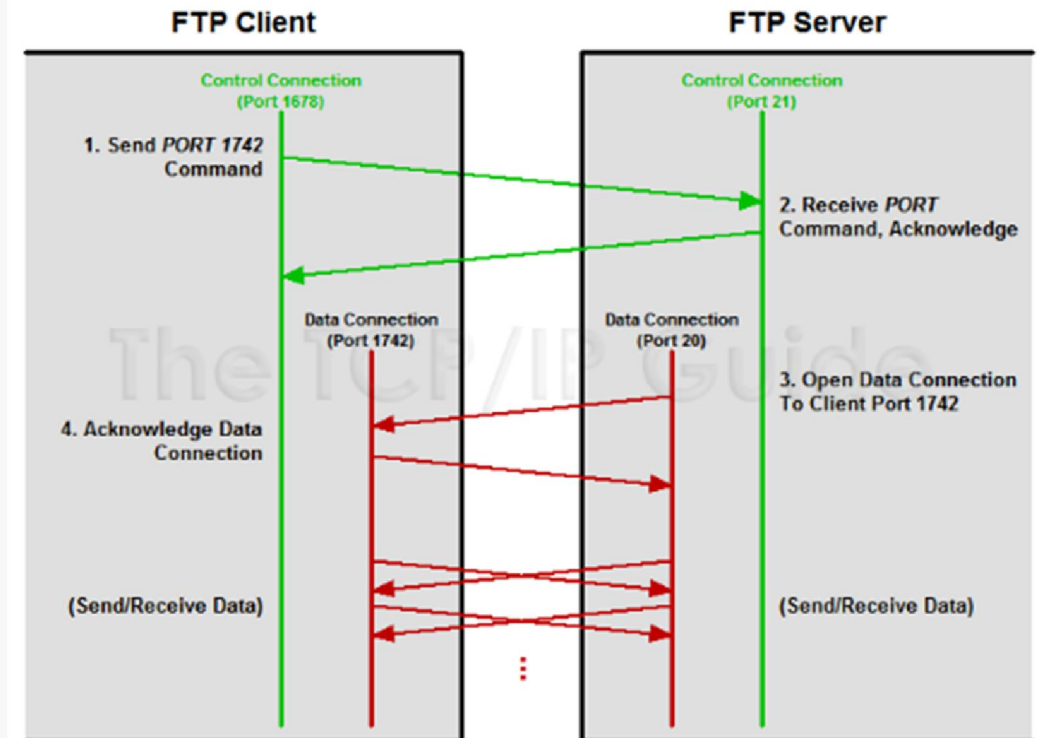


Figure 290: FTP Active Data Connection

In a conventional, or active, FTP data connection, the server initiates the transfer of data by opening the data connection to the client. In this case, the client first sends a **PORT** command to tell the server to use port 1742. The server then opens the data connection from its default port number of 20 to client port 1742. Data is then exchanged between the devices using these ports. Contrast to [Figure 291](#).



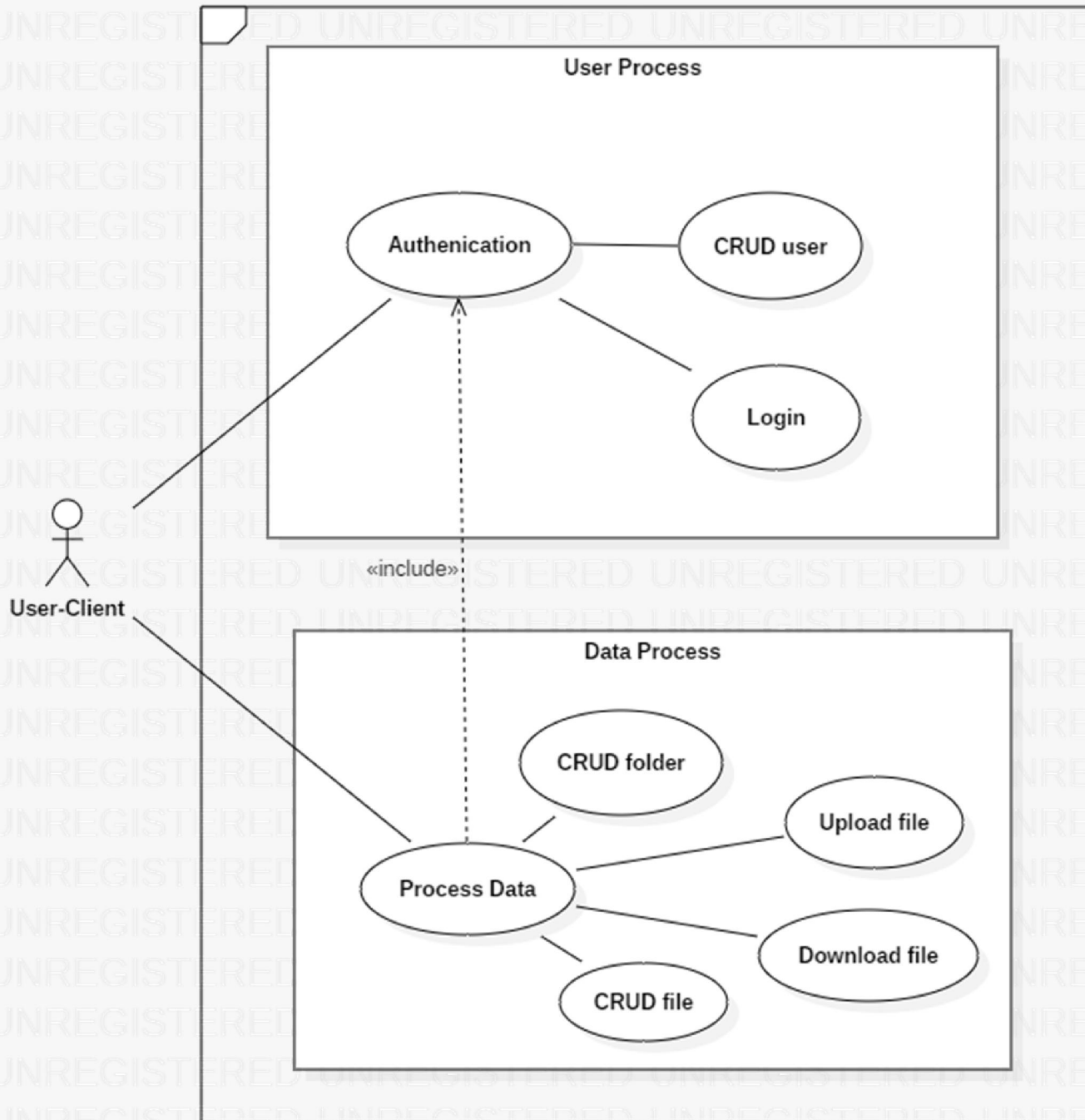
## II. Phân tích đề tài

---

Chương này trình bày phân tích đề tài theo hướng đối tượng, bắt đầu từ biểu đồ đặc tả chức năng

FTP Server





## Usecase diagram cho FTP Server

**CRUD** = **C**reate **R**ead **U**ppdate **D**elele

# Các lệnh hỗ trợ :

- ✓ **!** : xem thông tin hệ điều hành đang sử dụng.
- ✓ **?** : xem các lệnh được hỗ trợ.
- ✓ **mkdir** : tạo một thư mục
- ✓ **rmdir** : xóa 1 thư mục.
- ✓ **get** : lấy 1 file từ ftp server.
- ✓ **put** : gửi 1 file lên ftp server.
- ✓ **delete** : xóa 1 file trên ftp server.
- ✓ **rename** : đổi tên 1 file
- ✓ **bye** : thoát khỏi phiên làm việc.
- ✓ **adduser**: thêm mới một user
- ✓ **deluser**: xóa một user
- ✓ **updateuser**: thay đổi thông tin user
- ✓ **mk**: tạo một file với nội dung
- ✓ **renamedir**: đổi tên một thư mục



# III. Thiết kế cấu trúc và các lớp

---

Chương này trình bày việc thiết kế các lớp đối tượng cho giao thức FTP và cấu trúc liên quan

FTP Server



# Các lớp chính:

```
class Ftp {
private:
    int sock; /* định danh của socket */
    struct sockaddr_in ServAddr; /* lưu địa chỉ ip */
    unsigned short ServPort; /* lưu địa chỉ cổng */
    WSADATA wsaData;
    char myHostname[HOSTNAME_LENGTH];

public:
    void Init();
    int createSocket();
    unsigned long ResolveName(char name[]);
    void err_sys(char * fmt,...);

    bool file_exists(char * filename);
    bool readFileAndSend(char* name, SOCKET sock);
    bool recvFileAndWrite(char * filename, SOCKET sock);
};
```

```
class FtpServer : public Ftp {
private:
    struct sockaddr_in ServerAddr; /* Server address */
    char servername[HOSTNAME_LENGTH];
    int sock;

public:
    FtpServer();
    ~FtpServer();
    void accept_new_clients();
    bool accept_client(Client * cli); //accept client connections

    static bool check_server_status();
    static void updateStatus(int status);
    static bool disconnectClient(Client *cli);
    static void notiCmd(bool flag, char* cmd);

    static DWORD WINAPI ReceiveCmds(LPVOID lpParam);
    static bool handleFrame(Client* cli);

    static void checkLogin(Client* cli);
    static bool addUser(Client* cli, char* arg);
    static bool delUser(Client* cli, char* arg);
    static bool updateUser(Client* cli, char* arg);
    static bool doUpload(Client* cli, char* filename);
    static bool doDownload(Client* cli, char* filename);
    static bool checkOS(Client *cli);
    static bool listCmd(Client *cli);
    static bool renamefile(Client *cli, char* arg);
    static bool deletefile(Client *cli, char* filename);
    static bool makefile(Client* cli, char* arg);
    static string showCurDir(Client *cli);
    static bool makefolder(Client* cli, char* foldername);
    static bool renamefolder(Client *cli, char* arg);
```

```
};
```



### III. Thực thi và demo kết quả

---

Chương này trình bày luồng hoạt động của chương trình, demo, và phân công công việc của các thành viên trong nhóm

FTP Server



# III. Kết luận

---

Chương này tóm tắt lại các công việc mà nhóm đã làm được, kết quả đạt được, tự đánh giá và rút kinh nghiệm

FTP Server





# Chi tiết công việc:

Môi trường lập trình: Visual Studio 2017

Ngôn ngữ: C++ (OOP)

Lập trình mức SOCKET

Mô hình Overlapped (Multi Thread)

# Phân công công việc:

|                        |   |
|------------------------|---|
| Đàm Minh Tiến          | <ul style="list-style-type: none"><li>• Xử lý các lệnh cơ bản: upload, download tập tin, tạo mới , đổi tên, xóa tập tin, thư mục ...</li><li>• Viết báo cáo và thuyết trình</li></ul> |
| Vương Ngọc Quỳnh Hương | <ul style="list-style-type: none"><li>• Các thao tác quản lý người dùng: thêm, xóa, thay đổi thông tin (username, password)</li></ul>   |

Thank you!!!

