TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN MÔN: CHUYÊN ĐỀ .NET**

**Tên đề tài:**

**XÂY DỰNG CHƯƠNG TRÌNH FTP**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2016**

*Người hướng dẫn:***ThS VŨ ĐÌNH HỒNG**

**Thầy MAI VĂN MẠNH**

*Người thực hiện:* **Nguyễn Quang Huy - 51303302**

**Nguyễn Công Minh - 51303338**

Lớp: 13050303

Khóa: 17

LỜI CẢM ƠN

Trong suốt thời gian qua, nhờ sự giảng dạy tận tâm của quý Thầy Cô Khoa Công Nghệ Thông Tin, trường Đại học Tôn Đức Thắng, chúng em đã học hỏi được rất nhiều điều bổ ích và tích lũy cho mình một số kiến thức để hoàn thành bài báo cáo này. Chúng em xin chân thành cảm ơn.

Chúng em xin cảm ơn Thầy Vũ Đình Hồng và thầy Mai Văn Mạnh đã tận tình chỉ bảo chúng em qua những buổi báo cáo đồ án, Thầy đã chỉ chúng em cách thức làm bài, chỉ điểm những chỗ còn sai sót chưa phù hợp cũng như phải làm sao để trình bày bố cục đẹp. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì bài thu hoạch của chúng em cũng rất khó để hoàn thiện. Một lần nữa chúng em xin chân thành cảm ơn Thầy.

Bước đầu đi vào thực tế với nền kiến thức mở rộng, kiến thức chúng em còn hạn chế và nhiều bỡ ngỡ. Vì thế, trong quá trình biên soạn khó tránh khỏi những sai sót, chúng em rất mong nhận được những ý kiến đóng góp quý báu của quý Thầy/Cô và Các bạn để bài báo cáo hoàn thiện hơn.

CAM KẾT

Tôi xin cam đoan đây là sản phẩm bài tập lớn của riêng tôi / chúng tôi và được sự hướng dẫn của thầy Vũ Đình Hồng và thầy Mai Văn Mạnh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung bài tập lớn của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Nguyễn Quang Huy*

*Nguyễn Công Minh*

**MỤC LỤC**

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI 5](#_Toc468653216)

[I. BỐI CẢNH 5](#_Toc468653217)

[II. MỤC TIÊU 5](#_Toc468653218)

[CHƯƠNG 2. GIAO THỨC TCP/IP VÀ FTP 7](#_Toc468653219)

[I. GIAO THỨC TCP/IP 7](#_Toc468653220)

[1. Tầng liên kết 7](#_Toc468653221)

[2. Tầng mạng 7](#_Toc468653222)

[3. Tầng giao vận 7](#_Toc468653223)

[4. Tầng ứng dụng 7](#_Toc468653224)

[II. GIAO THỨC FTP 8](#_Toc468653225)

[1. Khái niệm. 8](#_Toc468653226)

[2. So sánh FTP va FTPS: 8](#_Toc468653227)

[3. Cơ chế hoạt động của FTP. 8](#_Toc468653228)

[*Hình 2.2.3.1 – Cơ chế hoạt động FTP.* 9](#_Toc468653229)

[4. Mô hình hoạt động của FTP. 9](#_Toc468653230)

[a. Tiến trình Server-FTP và User-FTP 9](#_Toc468653231)

[*Hình 2.2.4.1 – Mô hình hoạt động FTP.* 10](#_Toc468653232)

[b. Các tiến trình phía server 10](#_Toc468653233)

[c. Các tiến trình phía Client 10](#_Toc468653234)

[d. Trình tự truy cập chứng thực FTP 11](#_Toc468653235)

[e. Mở rộng bảo mật FTP 11](#_Toc468653236)

[5. Quản lý kênh dữ liệu FTP. 12](#_Toc468653237)

[a. Kết nối dữ liệu dạng chủ động 12](#_Toc468653238)

[b. Kết nối dữ liệu dạng bị động 12](#_Toc468653239)

[c. Vấn đề về phương thức kết nối 13](#_Toc468653240)

[CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG 15](#_Toc468653241)

[I. Form Main (giao diện chính) 15](#_Toc468653242)

[*Hình 3.1.1 – Giao diện chính.* 15](#_Toc468653243)

[II. Form Login (giao diện đăng nhập) 16](#_Toc468653244)

[*Hình 3.2.1 – Giao diện đăng nhập.* 16](#_Toc468653245)

[III. Form Upload (giao diện upload) 17](#_Toc468653246)

[*Hình 3.3.1 – giao diện Upload.* 17](#_Toc468653247)

[IV. Form Download (giao diên download) 17](#_Toc468653248)

[*Hình 3.4.1 – Giao diện download.* 18](#_Toc468653249)

[V. Form Rename (giao diện đổi tên) 18](#_Toc468653250)

[*Hình 3.5.1 – Giao diện đổi tên* 18](#_Toc468653251)

[VI. Form New Folder (giao diện tạo folder mới) 19](#_Toc468653252)

[*Hình 3.6.1 – Giao diện tạo folder mới.* 19](#_Toc468653253)

[VII. Form Information (giao diện thông tin) 19](#_Toc468653254)

[*Hình 3.7.1 – Giao diện thông tin phần mềm.* 19](#_Toc468653255)

[CHƯƠNG 4. KẾT LUẬN 20](#_Toc468653256)

[I. Kết quả đạt được 20](#_Toc468653257)

[1. Ưu điểm 20](#_Toc468653258)

[2. Nhược điểm 20](#_Toc468653259)

[II. Hướng phát triển 20](#_Toc468653260)

[TÀI LIỆU THAM KHẢO 21](#_Toc468653261)

# TỔNG QUAN ĐỀ TÀI

# BỐI CẢNH

Trước sự phát triển nhanh chóng của khoa học công nghệ, nhu cầu của con người ngày càng lớn về nhiều lĩnh vực. Để tạo nên sự tiện nghi đó, ngành công nghệ thông tin phải không ngừng phát triển để hỗ trợ một cách tối đa nói chung và lĩnh vực mạng nói riêng. Mô hình Clients-Server đã được phát triển từ lâu và được ứng dụng rộng rãi nhưng không phải chương trình nào là hoàn thiện, với giao thức FTP (File Transfer Protocol) trong môi trường Internet, khi cần truyền nhận file có kích thước lớn sẽ gặp trường hợp bị gián đoạn trong khi truyền nhận qua một số trình duyệt. Chính vì thế người sử dụng đã chuyển sang một số chương trình chuyên dụng như FileZilla.

Người sử dụng có xu hướng lấy đồng thời nhiều file tại một server nào đó trên Internet. Như vậy chương trình phải hỗ trợ lấy nhiều file đồng thời. Tuy nhiên không phải lúc nào chương trình cũng có thể lấy đồng thời nhiều file tại một site nếu server hạn chế số kết nối đồng thời tới một địa chỉ IP. Người sử dụng cũng có nhu cầu lấy file bằng nhiều giao thức khác nhau như FTP (File Transfer Protocol), HTTP (HyperText Transfer Protocol),…

# MỤC TIÊU

- Mục tiêu của chương trình:

Chương trình phải tạo sự tiện lợi cho người sử dụng. Chương trình phải có giao diện đồ họa thân thiện, giúp cho người sử dụng xem được nội dung các thư mục từ xa và cục bộ. Từ đó giúp cho việc upload và download dễ dàng hơn. Chương trình cũng phải hiện thực một số chức năng cơ bản của một FTP client như tạo thư mục, xóa thư mục, xóa file, thay đổi thư mục,…

- Mục tiêu của giao thức:

* Giúp đỡ việc sử dụng gián tiếp(thông qua các chương trình) các máy tính từ xa.
* Che thông tin người sử dụng khỏi những thay đổi trong các hệ thống lưu trữ file giữa các máy chủ(host).
* Truyền dữ liệu một cách tin cậy và hiệu quả.

- Mục tiêu của đề tài:

* Tạo một ứng dụng FTP chạy trên bất kỳ máy nào.
* Nghiên cứu tìm hiểu về họ giao thức TCP/IP, cách thức hoạt động, ưu nhược điểm của giao thức này.
* Nghiên cứu về ngôn ngữ lập trình C#.Net.

# GIAO THỨC TCP/IP VÀ FTP

# GIAO THỨC TCP/IP

TCP/IP là bộ giao thức cho phép kết nối các hệ thống mạng không đồng nhất với nhau. Ngày nay TCP/IP được sử dụng rộng rãi trong mạng cục bộ cũng như mạng toàn cầu.

# Tầng liên kết

Tầng liên kết (còn được gọi là tầng liên kết dữ liệu hay là tầng giao tiếp mạng) là tầng thấp nhất trong mô hình TCP/IP, bao gồm các thiết bị giao tiếp mạng và chương trình cung cấp các thông tin cần thiết để có thể hoạt động, truy nhập đường truyền vật lý qua thiết bị giao tiếp mạng đó.

# Tầng mạng

Tầng mạng (còn gọi là tầng Internet) xử lý quá trình truyền gói tin trên mạng. Các giao thức của tầng này bao gồm: IP (Internet Protocol), ICMP (Internet Control Message Protocol), IGMP (Internet Group Messages Protocol).

# Tầng giao vận

Tầng giao vận phụ trách luồng dữ liệu giữa hai trạm thực hiện các ứng dụng của tầng trên. Tầng này có hai giao thức chính: TCP (Transmission Control Protocol) và UDP (User Datagram Protocol) TCP cung cấp một luồng dữ liệu tin cậy giữa hai trạm, nó sử dụng các cơ chế như chia nhỏ các gói tin của tầng trên thành các gói tin có kích thước thích hợp cho tầng mạng bên dưới, báo nhận gói tin, đặt hạn chế thời gian time-out để đảm bảo bên nhận biết được các gói tin đã gửi đi. Do tầng này đảm bảo tính tin cậy, tầng trên sẽ không cần quan tâm đến nữa.

UDP cung cấp một dịch vụ đơn giản hơn cho tầng ứng dụng. Nó chỉ gửi các gói dữ liệu từ trạm này tới trạm kia mà không đảm bảo các gói tin đến được tới đích. Các cơ chế đảm bảo độ tin cậy cần được thực hiện bởi tầng trên.

# Tầng ứng dụng

Tầng ứng dụng là tầng trên cùng của mô hình TCP/IP bao gồm các tiến trình và các ứng dụng cung cấp cho người sử dụng để truy cập mạng. Có rất nhiều ứng dụng được cung cấp trong tầng này, mà phổ biến là: Telnet: sử dụng trong việc truy cập mạng từ xa, FTP (File Transfer Protocol).

Cũng tương tự như trong mô hình OSI, khi truyền dữ liệu, quá trình tiến hành từ tầng trên xuống tầng dưới, qua mỗi tầng dữ liệu được thêm vào một thông tin điều khiển được gọi là phần header. Khi nhận dữ liệu thì quá trình xảy ra ngược lại, dữ liệu được truyền từ tầng dưới lên và qua mỗi tầng thì phần header tương ứng được lấy đi và khi đến tầng trên cùng thì dữ liệu không còn phần header nữa. Trong hình vẽ này ta thấy tại các tầng khác nhau dữ liệu được mang những thuật ngữ khác nhau:

* Trong tầng ứng dụng dữ liệu là các luồng được gọi là stream.
* Trong tầng giao vận, đơn vị dữ liệu mà TCP gửi xuống tầng dưới gọi là TCP segment.
* Trong tầng mạng, dữ liệu mà IP gửi tới tầng dưới được gọi là IP datagram.
* Trong tầng liên kết, dữ liệu được truyền đi gọi là frame.

# GIAO THỨC FTP

# Khái niệm.

FTP là chữ viết tắc của File Transfer Protocol - Giao thức truyền file. FTP là một giao thức truyền file trên mạng dựa trên chuẩn TCP nên rất đáng tin cậy.

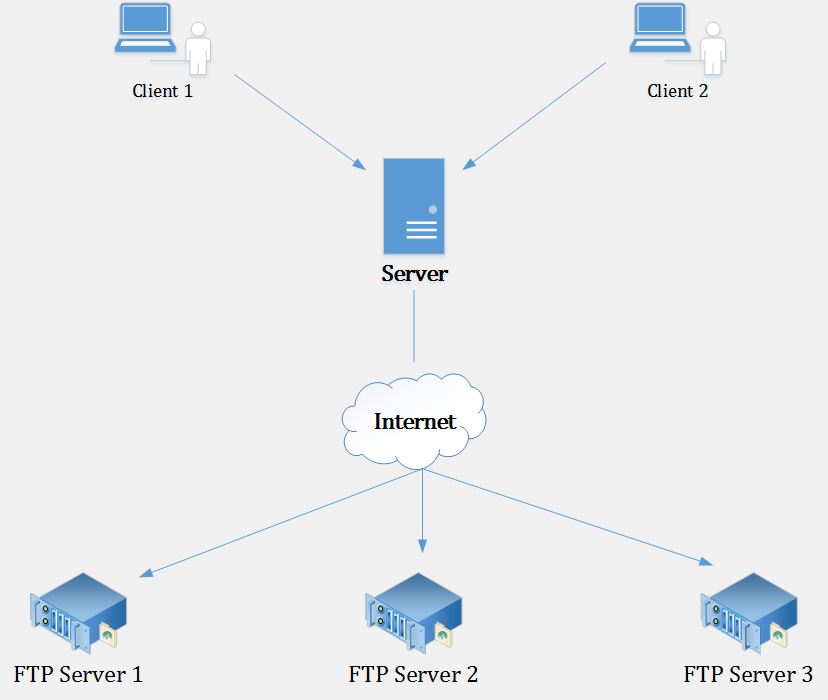
# So sánh FTP va FTPS:

FTP là một giao thức không được bảo mật, nó truyền dữ liệu mà không thực hiện mã hóa vì vậy mà người dùng có thể gặp phải những rủi ro khi sử dụng giao thức này.

Một phương pháp hữu hiệu hơn đó là sử dụng giao thức FTPS (FTP trên SSL) cung cấp khả năng mã hóa giao thức cho các dữ liệu được truyền.

# Cơ chế hoạt động của FTP.

Server xác nhận yêu cầu từ phía client => Kiểm tra yêu cầu => Sau đó là xác thực. Cuối cùng, server tạo nen kết nối FTP thực thi yêu cầu của Client.



*Hình 2.2.3.1 – Cơ chế hoạt động FTP.*

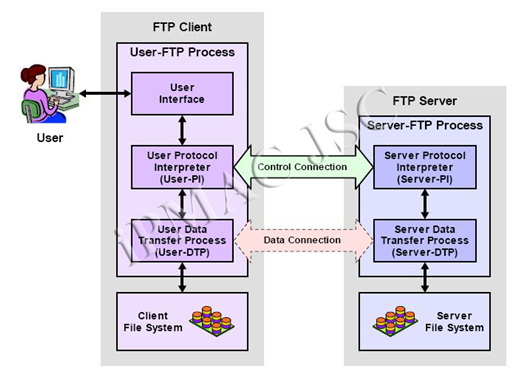
# Mô hình hoạt động của FTP.

# Tiến trình Server-FTP và User-FTP

FTP là một giao thức dạng client/server truyền thống, tuy nhiên thuật ngữ client thông thường được thay thế bằng thuật ngữ user – người dùng – do thực tế là người sử dụng mới là đối tượng trực tiếp thao tác các lệnh FTP trên các máy client. Bộ phần mềm FTP được cài đặt trên một thiết bị được gọi là một tiến trình. Phần mềm FTP được cài đặt trên máy Server được gọi là tiến trình Server-FTP, và phần trên máy client được gọi là tiến trình User-FTP.

Do các chức năng điều khiển và dữ liệu sử dụng các kênh khác nhau, nên mô hình hoạt động của FTP cũng chia phần mềm trên mỗi thiết bị ra làm hai thành phần logic tương ứng với mỗi kênh. Thành phần Protocol Interpreter (PI) là thành phần quản lý kênh điều khiển, với chức năng phát và nhận lệnh. Thành phần Data Transfer Process (DTP) có chức năng gửi và nhận dữ liệu giữa phía client với server. Ngoài ra, cung cấp cho tiến trình bên phía người dùng còn có thêm thành phần thứ ba là giao diện người dùng FTP - thành phần này không có ở phía server.

Do đó, có hai tiến trình xảy ra ở phía server, và ba tiến trình ở phía client. Các tiến trình này được gắn với mô hình FTP để mô tả chi tiết hoạt động của giao thức FTP.



*Hình 2.2.4.1 – Mô hình hoạt động FTP.*

# Các tiến trình phía server

Các tiến trình phía server bao gồm hai giao thức:

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

- Server DataTransfer Process (Server-DTP): làm nhiệm vụ gửi hoặc nhận file từ bộ phận User-DTP. Server-DTP vừa làm nhiệm 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.

# Các tiến trình phía Client

- User Protocol Interpreter (User-PI): chịu trách nhiệm quản lý kênh điều khiển phía client. Nó khởi tạo phiên kết nối FTP bằng việc phát ra yêu cầu tới phía Server-PI. Khi kết nối đã được thiết lập, nó xử lý các lệnh nhận được trên giao diện người dùng, gửi chúng tới Server-PI, và nhận phản hồi trở lại. Nó cũng quản lý tiến trình User-DTP.

- User Data Transfer Process (User-DTP): là bộ phận DTP nằm ở phía người dùng, làm nhiệm vụ gửi hoặc nhận dữ liệu từ Server-DTP. User-DTP có thể thiết lập hoặc lắng nghe yêu cầu kết nối kênh dữ liệu trên server. Nó tương tác với thiết bị lưu trữ file phía client.

- User Interface: cung cấp giao diện xử lý cho người dùng. Nó cho phép sử dụng các lệnh đơn giản hướng người dùng, và cho phép người điều khiển phiên FTP theo dõi được các thông tin và kết quả xảy ra trong tiến trình.

# Trình tự truy cập chứng thực FTP

Quy luật chứng thực trong FTP khá đơn giản, chỉ là cung cấp username/password. Trình tự của việc chứng thực như sau:

- Người dùng gửi một username từ User-PI tới Server-PI bằng lệnh USER. Sau đó password của người dùng được gửi đi bằng lệnh PASS.

- Server kiểm tra tên người dùng và password trong database người dùng của nó. Nếu người dùng hợp lệ, server sẽ gửi trả một thông báo tới người dùng rằng phiên kết nối đã được mở. Nếu người dùng không hợp lệ, server yêu cầu người dùng thực hiện lại việc chứng thực. Sau một số lần chứng thực sai nhất định, server sẽ ngắt kết nối. Giả sử quá trình chứng thực đã thành công, server sau đó sẽ thiết lập kết nối để cho phép từng loại truy cập đối với người dùng được cấp quyền. Một số người dùng chỉ có thể truy cập vào một số file nhất định, hoặc vào một số loại file nhất định. Một số server có thể cấp quyền cho một số người dùng đọc và viết lên server, trong khi chỉ cho phép đọc đối với những người dùng khác. Người quản trị mạng có thể nhờ đó mà đáp ứng đúng các nhu cầu truy cập FTP. Một khi kết nối đã được thiết lập, server có thể thực hiện các lựa chọn tài nguyên dựa vào nhận diện người dùng. Ví dụ: trên một hệ thống nhiều người dùng, người quản trị có thể thiết lập FTP để khi có bất cứ người dùng nào kết nối tới, người dùng sẽ tự động được đưa tới “home directory” của chính người dùng. Lệnh tùy chọn ACCT (account) cũng cho phép người dùng chọn một tài khoản cá nhân nào đó nếu như người dùng có nhiều hơn một tài khoản.

# Mở rộng bảo mật FTP

Giống như phần lớn các giao thức cũ, phương pháp đăng nhập đơn giản của FTP là một sự kế thừa từ những giao thức ở thời kỳ đầu của Internet. Ngày nay, nó không còn bảo đảm tính an toàn cần thiết trên môi trường Internet toàn cầu vì username và password được gửi qua kênh kết nối điều khiển dưới dạng clear text. Điều này làm cho các thông tin đăng nhập có thể bị nghe lén. Chuẩn RFC 2228 về các phần mở rộng cho bảo mật FTP đã định ra thêm nhiều tùy chọn chứng thực và mã hóa phức tạp cho những ai muốn tăng thêm mức độ an toàn vào trong phần mềm FTP của họ.

# Quản lý kênh dữ liệu FTP.

# Kết nối dữ liệu dạng chủ động

Phương thức đầu tiên đôi khi còn được gọi là kết nối kênh dữ liệu dạng thông thường (vì nó là phương pháp mặc định) và đôi khi được gọi là kết nối dạng chủ động (để đối chiếu với dạng kết nối bị động mà ta sẽ xét ở phần sau). Trong dạng kết nối này, phía Server-DTP khởi tạo kênh dữ liệu bằng việc mở một cổng TCP cho phía User-DTP. Phía server sử dụng cổng được dành riêng, là cổng 20 cho kênh dữ liệu. Trên máy client, một giá trị cổng được chọn theo mặc định chính là cổng được sử dụng đối với kênh điều khiển, tuy nhiên phía client sẽ luôn chọn hai cổng riêng biệt cho hai kênh này.

Giả sử phía User-PI thiết lập một kết nối điều khiển từ cổng bất kỳ của nó là 1678 tới cổng điều khiển trên server là cổng 21. Khi đó, để tạo một kênh dữ liệu cho việc truyền dữ liệu, phía Server-PI sẽ báo cho phía Server-DTP khởi tạo một kênh kết nối TCP từ cổng 20 tới cổng 1678 của phía client. Sau khi phía client chấp nhận kênh được khởi tạo, dữ liệu sẽ được truyền đi.

Thực tế, việc sử dụng cùng một cổng cho cả kênh dữ liệu và kênh điều khiển không phải là một ý hay, nó làm cho hoạt động của FTP trở nên phức tạp. Do đó, phía client nên chỉ định sử dụng một cổng khác bằng việc sử dụng lệnh PORT trước khi truyền dữ liệu.

# Kết nối dữ liệu dạng bị động

Phương pháp kế tiếp được gọi là kết nối dữ liệu dạng bị động. Phía client sẽ nhận server là phía bị động, làm nhiệm vụ chấp nhận một yêu cầu kết nối kênh dữ liệu được khởi tạo từ phía client. Server trả lời lại phía client với địa chỉ IP cũng như địa chỉ cổng mà nó sẽ sử dụng. Phía Server-DTP sau đó sẽ lắng nghe một kết nối TCP từ phía User-DTP trên cổng này.

Mặc định, phía client sử dụng cùng một cổng đối với cả hai kênh điều khiển và dữ liệu như trong trường hợp kết nối chủ động ở trên. Tuy nhiên, ở đây, một lần nữa phía client có thể chọn sử dụng một giá trị cổng khác cho kênh dữ liệu.

# Vấn đề về phương thức kết nối

Vấn đề phía nào là phía khởi tạo kết nối kênh dữ liệu đưa ra một câu hỏi: sự khác nhau giữa hai phương thức là gì? Điều này cũng giống như việc hỏi ai đã thực hiện một cuộc điện thoại nội bộ. Câu trả lời là sự bảo mật. Việc FTP sử dụng nhiều hơn một kết nối TCP có thể giải quyết các vấn đề về phần mềm cũng như về phần cứng mà người dùng cần phải có để đảm bảo sự an toàn cho hệ thống của họ.

Khi xem xét việc gì sẽ xảy ra trong trường hợp kênh dữ liệu chủ động như trong ví dụ phía trên:

- Đối với phía client, có một kênh kết nối điều khiển được thiết lập từ cổng 1678 client tới cổng 21 server. Nhưng kênh dữ liệu lại được khởi tạo từ phía server. Do đó, client sẽ nhận được một yêu cầu kết nối tới cổng 1678 (hoặc cổng nào khác). Một số client sẽ nghi ngờ về việc nhận được những kết nối tới như vậy, vì trong tình huống thông thường, client mới là phía khởi tạo kết nối chứ không phải đáp trả kết nối. Do các kênh kết nối TCP hướng tới có thể mang theo những mối đe dọa nhất định, một số client có thể sẽ ngăn chặn các luồng kết nối hướng tới bằng việc sử dụng tường lửa.

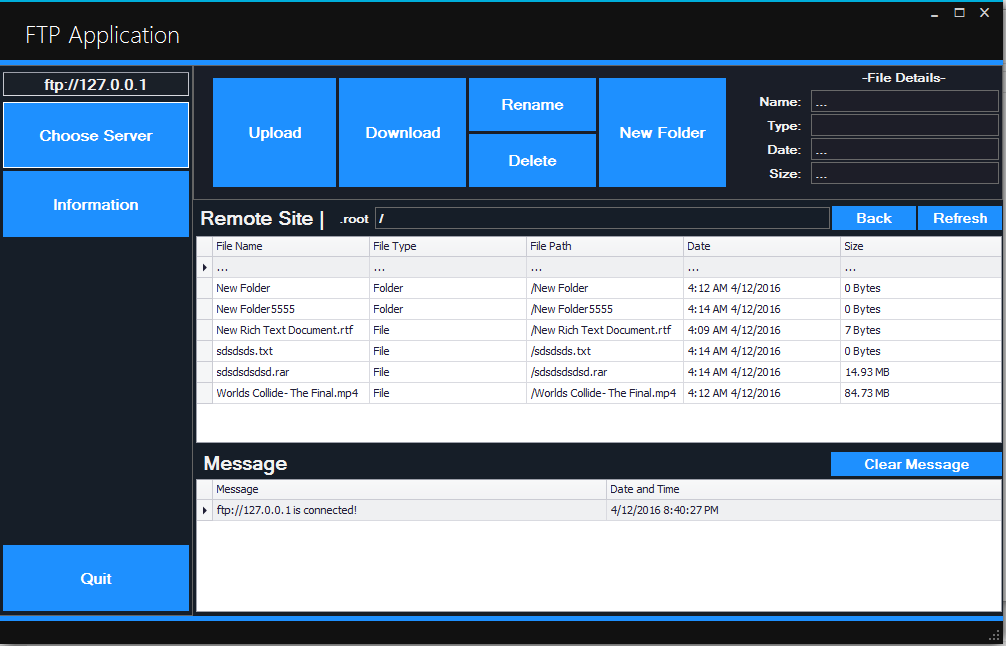
- Tại sao người ta lại không làm cho phía client luôn chấp nhận kết nối từ một chỉ số port được dùng trong kênh điều khiển? Vấn đề ở đây là vì client thường dùng các cổng khác nhau cho mỗi phiên kết nối bằng việc sử dụng câu lệnh PORT. Và tại sao điều này lại được thực hiện? Vì theo luật TCP: sau khi một kết nối được đóng lại , có một khoảng thời gian trống trước khi cổng đó có thể được sử dụng lại – điều này để ngăn ngừa tình trạng các phiên kết nối liên tiếp bị lẫn với nhau. Điều này sẽ tạo ra độ trễ khi gửi nhiều file – do đó phía client thường dùng các giá trị cổng khác nhau cho mỗi kết nối. Điều này rất hiệu quả nhưng cũng dẫn tới việc firewall của client sẽ hỏi có chấp nhận phiên kết nối tới với nhiều giá trị cổng không ổn định hay không.

- Việc dùng kết nối kiểu kênh gián tiếp sẽ giảm thiểu vấn đề này một cách hiệu quả. Phần lớn các tường lửa có nhiều vấn đề liên quan tới kết nối hướng về với các giá trị cổng bất kỳ, hơn là gặp vấn đề với các kết nối hướng đi. Ta có thể xem chi tiết hơn về vấn đề này trong chuẩn RFC 1579. Chuẩn này khuyến nghị rằng phía client nên sử dụng kết nối kiểu bị động làm dạng mặc định thay vì sử dụng kiểu kết nối dạng chủ động cùng với lệnh PORT, để ngăn chặn tình trạng block theo cổng.

- Tất nhiên, phương thức kết nối kiểu bị động không hoàn toàn giải quyết được vấn đề, chúng chỉ đẩy vấn đề về phía server mà thôi. Phía server, giờ đây phải đối mặt với việc có nhiều kênh kết nối hướng về trên hàng loạt các cổng khác nhau. Tuy nhiên việc xử lý các vấn đề bảo mật trên một nhóm nhỏ server vẫn dễ hơn nhiều so với việc phải đối mặt với một lượng lớn các vấn đề từ nhiều client. FTP server phải được cấu hình chấp nhận phương thức truyền bị động từ client, do đó cách thông thường để thiết lập trên server là thiết lập chấp nhận một số cổng kết nối hướng về trên server trong khi vẫn khóa các yêu cầu kết nối hướng về trên các cổng khác.

# XÂY DỰNG ỨNG DỤNG

# Form Main (giao diện chính)



*Hình 3.1.1 – Giao diện chính.*

- Form này chịu trách nhiệm quản lý tát cả các chức năng.

- Button “Choose Server” – Gọi giao diện đăng nhập.

- Button “Information” – Gọi giao diện thông tin.

- Button “Upload” – Gọi một Open Dialog chọn đường dẫn file, rồi gọi giao diện upload thực thi.

- Button “Download” – Chọn một File trong server gọi một Save Dialog chọn dường dẫn lưu, rồi gọi giao diện down load thực thi.

- Button “Rename”- Gọi giao diện đổi tên.

- Button “Delete” – Chọn một file hoặc một folder rồi chọn button “Delete” file hoặc folder đó sẽ bị xóa khói server.

- Button “New Folder” – Gọi giao diện tạo folder mới.

- Vùng “File Detail” – Hiển thị thông tin file hoặc folder được chọn dưới grid view quản lý Server.

- Grid view quản lý server (Grid view trên) – Hiển thị toàn bộ các file hoặc folder trên server theo địa chỉ thư mục.

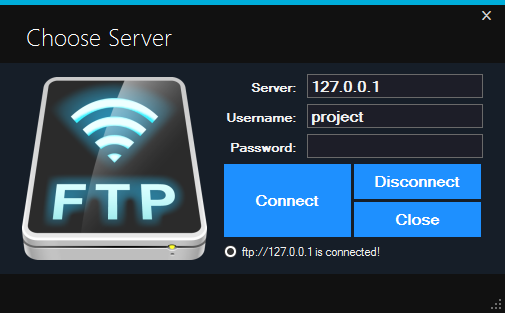
- Button “Back” – Quay về thưc mục trước thư mục hiện tại, nếu thư mục hiện tại là thư mục root thì sẽ không thực thi.

- Button “Refresh” – Load lại hiển thị các file hoặc folder trên grid view quản lý server.

- Button “Clear Message” – Xóa các message hiển thị dưới grid view quản lý message.

- Grid View quản lý message (Grid view dưới) – Hiển thị các message sau mỗi lần thực hiện chức năng.

# Form Login (giao diện đăng nhập)



*Hình 3.2.1 – Giao diện đăng nhập.*

- Kết nối hoặc ngắt kết nối đến server.

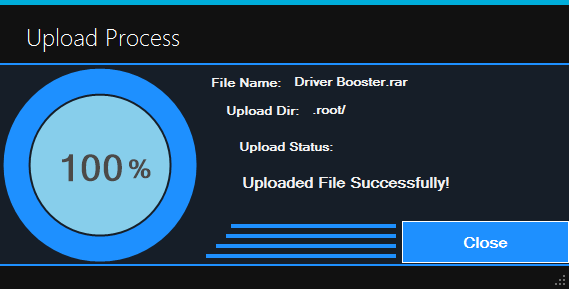
- Button “Connect” – Gọi kết nối giao thức FTP với server khi nhập thông tin đầy đủ. Nếu sai, báo lỗi.

- Button “Disconnect” – Ngắt kết nối với server.

- Button “Close” – Đóng giao diện đăng nhập.

- Radiobutton “Status” – Hiển thị trạng thái có kết nối hay không.

# Form Upload (giao diện upload)

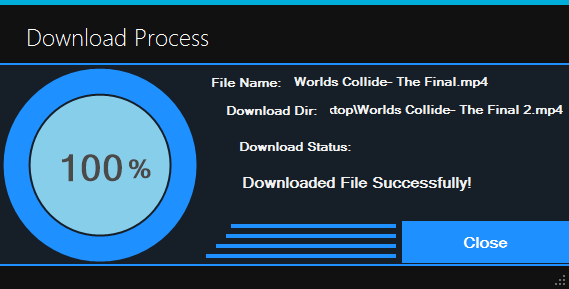


*Hình 3.3.1 – giao diện Upload.*

- Hiển thị thông tin tiến trình upload đến khi kết thúc.

- Button “Controll” – Trong tiến trình upload sẽ đóng vai trò là cancel upload, tiến trình hoàn thành sẽ đóng vai trò là đóng giao diện upload.

# Form Download (giao diên download)

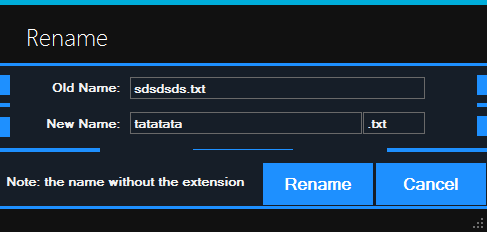


*Hình 3.4.1 – Giao diện download.*

- Hiển thị thông tin trong tiến trình download đến khi kết thúc.

- Button “Controll” – Trong tiến trình upload sẽ đóng vai trò là cancel upload, tiến trình hoàn thành sẽ đóng vai trò là đóng giao diện download.

# Form Rename (giao diện đổi tên)



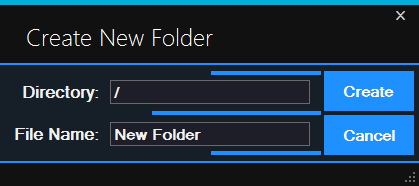
*Hình 3.5.1 – Giao diện đổi tên*

- Đặt tên mới cho file hoặc folder.

- Button “Rename” – Thực thi đổi tên file hoặc folder.

- Button “Cancel” – Đóng giao diện đổi tên.

# Form New Folder (giao diện tạo folder mới)



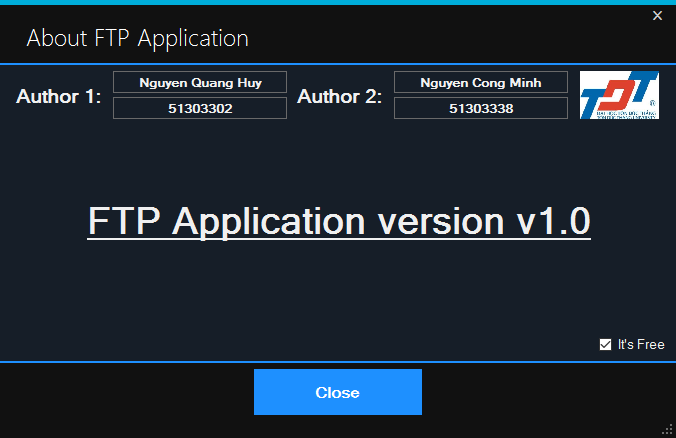
*Hình 3.6.1 – Giao diện tạo folder mới.*

- Tạo một folder mới theo tên người dùng đặt tại directory đã chọn.

- Button “Create” – Thực thi tạo folder mới.

- Button “Cancel” – Dóng giao diện tạo folder mới.

# Form Information (giao diện thông tin)



*Hình 3.7.1 – Giao diện thông tin phần mềm.*

- Hiển thị thông tin người thực hiện và phần mềm.

- Button “Close” – Đóng giao diện thông tin.

# KẾT LUẬN

# Kết quả đạt được

# Ưu điểm

- Ứng dựng với giao diện hiện đại (Metro Form), dễ sử dụng.

- Đáp ứng được nhu cầu quản lý file cơ bản với giao thức FTP.

- Sử dụng ngôn ngử C# phổ biến dễ dàng cho việc bảo trì và phát triển, với bộ công cụ DevExpress thuận lợi cho tùy biến giao diện.

- Người sử dụng không cần phải gõ phím nhiều mà đơn giản chỉ bằng những lần nháy chuột có thể sử dụng được phần mềm.

# Nhược điểm

- Vẫn con nhiều chức năng chưa đáp ứng dược nhu cầu của người dùng, chưa tối ưu tốt ở nhiều chỗ.

- Bảo mật còn kém.

# Hướng phát triển

- Upload và download nhiều file một lúc.

- Xử lý tốt hơn với các tập tin nén.

- Có thể mở được nhiều loại file trong phần mềm.

# TÀI LIỆU THAM KHẢO

1. Tìm hiểu về giao thức FTP – Vũ Thanh Lai (<http://sinhvienit.net/forum/tim-hieu-ve-giao-thuc-ftp.28754.html>).
2. Kris Jamsa, Ph.D. & Ken Cope - Internet Programming - Jamsa Press – 1995.
3. J. Postel & J.K. Reynolds - Request For Comments 959: File Transfer Protocol – 1985.
4. MSDN Library Visual Studio.