

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



BÁO CÁO BÀI TẬP LỚN MÔN CÁC HỆ PHÂN TÁN

ĐỀ TÀI: Xây dựng chương trình lưu trữ và gửi nhận file FTP

Giáo viên hướng dẫn: TS. Nguyễn Bình Minh

Sinh viên thực hiện:

- 1. Nguyễn Tiến Thiện - 20153583**
- 2. Nguyễn Hoàng Giang - 20151094**
- 3. Mai Trung Kiên - 20152055**

Năm học 2017-2018

Mục lục

1.	Mô tả bài toán.....	3
2.	Phân công công việc nhóm	3
3.	Thiết kế chương trình	3
3.1.	Phía Server.....	3
3.1.1.	Kiến trúc server trong việc quản lý các client kết nối.....	4
3.1.2.	Kiến trúc trong quản lý truyền tải dữ liệu giữa server và client	7
3.1.3.	Các phân mảnh của file	8
3.2.	Phía Client	9
3.3.	Mô hình truyền tải	9
3.4.	Xử lý dừng và tiếp tục truyền tải.....	10
3.5.	Xử lý tương tranh khi client upload file.....	10
4.	Hướng dẫn chạy chương trình.....	11
4.1.	Chạy chương trình trên netbeans	11
4.2.	Chạy chương trình bằng file .jar đã biên dịch sẵn.....	11
5.	Tài liệu tham khảo	11
5.1.	Mã nguồn tham khảo	11
5.2.	Tài liệu tham khảo	11

1. Mô tả bài toán

- Xây dựng ứng dụng cho phép lưu trữ và gửi nhận dữ liệu. Ứng dụng gồm có hai thành phần là Client và Server. Server có chức năng lưu trữ dữ liệu, truyền tải dữ liệu theo yêu cầu của client. Client có các chức năng gửi nhận dữ liệu từ Server.
- Dữ liệu được phía Client gửi lên (upload) máy Server, và ngược lại Client có thể tải (download) dữ liệu từ máy Server về.
- Cơ chế upload và download dữ liệu được thực hiện thông qua cơ chế phân mảnh (chia dữ liệu thành các phần nhỏ và tải lên hoặc tải xuống về các mảnh này, sau quá trình upload và download các mảnh hoàn tất, các mảnh được ghép lại với nhau thành file hoàn chỉnh).
- Ứng dụng các chức năng download, upload, dừng download, dừng upload, tiếp tục download, tiếp tục upload (khi chưa hoàn thành).
- Ứng dụng xử lý được tương tranh dữ liệu từ các lần upload lên Server.

2. Phân công công việc nhóm

Thành viên	Công việc
Nguyễn Tiến Thiện	-Thiết kế kiến trúc ứng dụng.
	-Xây dựng phần lắng nghe và quản lý các client phía server.
	-Xây dựng giao thức giao tiếp giữa client và server dựa trên giao thức FTP.
	-Xây dựng chức năng upload.
	-Xử lý tương tranh khi upload.
	-Xử lý tạm ngừng và tiếp tục truyền tải.
Nguyễn Hoàng Giang	-Xây dựng chức download.
	-Xây dựng giao diện client.
Mai Trung Kiên	-Xây dựng giao diện server.

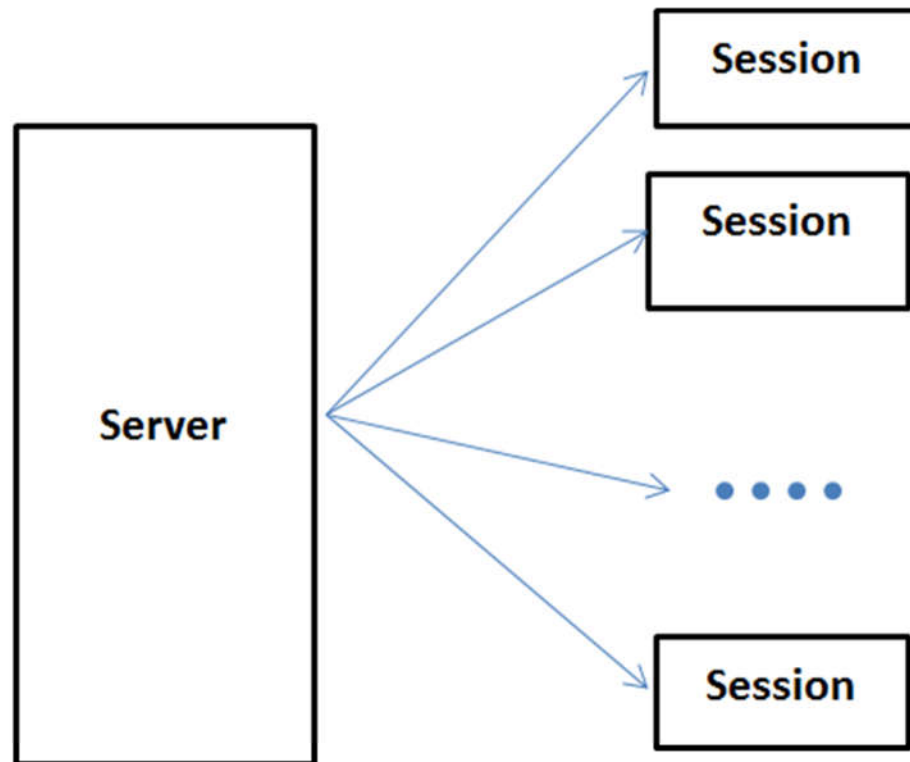
3. Thiết kế chương trình

Chương trình gồm hai phần: client và server.

3.1. Phía Server

Server lắng nghe các kết nối từ client và quản lý các kết nối.

3.1.1. Kiến trúc server trong việc quản lý các client kết nối



Hình ảnh 1. Kiến trúc tổng quan Server

Class	Giải thích
Server	-Server quản lý các phiên kết nối của client.
	-Mở socket lắng nghe kết nối từ client.
	-Khi client kết nối đối tượng Session sẽ được tạo mới tương ứng với client. Sesion mới được tạo này được Server thêm vào danh sách quản lý.
	-Số lượng client kết nối đến server được giới hạn theo thiết lập trên giao diện.
	-Khi số lượng client kết nối đã đến giới hạn mà có client mới kết nối thì server sẽ không chấp nhận xử lý yêu cầu của client. Client mới đó sẽ phải chờ cho đến khi có client ngắt kết nối với server.
Session	-Tương ứng với một phiên làm việc của client.
	-Session lưu các thông tin liên quan đến client: socket, username, thư mục làm việc hiện tại, thư mục làm việc gốc...
	-Session sẽ giao tiếp với client thông qua socket đã thiết lập từ khi client

	kết nối với server.
	-Session chờ nhận các lệnh từ client, xử lý theo các lệnh và gửi lại phản hồi.
	-Khi client đăng nhập thành công vào hệ thống, Session tạo một kết nối dữ liệu với client để trao đổi dữ liệu.

Lệnh	Giải thích
USER <param>	-Nhập tài khoản lên hệ thống. -Tham số là tên tài khoản của người dùng.
PASS <param>	-Nhập mật khẩu tương ứng với tài khoản đã nhập của người dùng. -Tham số là mật khẩu.
CWD <param>	-Chuyển thư mục làm việc hiện tại sang thư mục khác. -Tham số là thư mục đích muốn chuyển đến.
LIST	-Hiển thị danh sách chi tiết các file và thư mục có trong thư mục hiện tại.
NLIST	-Hiển thị danh sách tên các file và thư mục có trong thư mục hiện tại.
ALLO <param>	-Cấp phát bộ nhớ, chuẩn bị cho quá trình upload. -Tham số là dung lượng muốn cấp phát tính bằng byte. -Được thực hiện trước câu lệnh STOR.
SIZE <param>	-Lấy thông tin kích thước của một file. -Tham số là tên (hoặc tên đầy đủ) của file muốn lấy kích thước.
RETR <param>	-Download file về máy client. -Tham số gồm 2 phần được phân cách bởi dấu cách: + Tên (hoặc tên đầy đủ) của file. + Danh sách các offset của các mảnh của file muốn download. Các offset được phân cách bởi dấu cách. Số lượng mảnh của file được xác định dựa trên số lượng offset.
STOR <param>	-Upload file từ máy client lên server. -Tham số gồm 3 phần: + Tên (hoặc tên đầy đủ) của file. + Bit thể hiện quá trình chưa hoàn thành từ trước (1-tiếp tục quá trình chưa hoàn thành, 0-quá trình mới). + Thiết lập số mảnh muốn truyền tải.
APPE <param>	-Ghi thêm vào file đã có trên server. -Tham số là tên (hoặc tên đầy đủ) của file.
RNFR <param>	-Đổi tên file. -Tham số là tên (hoặc tên đầy đủ) của file muốn đổi tên. -Câu lệnh này được thực hiện trước RNTD.

RNTO <param>	-Đổi tên file. -Tham số là tên mới của file đã chọn từ câu lệnh RNFR.
DELE <param>	-Xóa file. -Tham số là tên (hoặc tên đầy đủ) của file.
MKD <param>	-Tạo một thư mục mới. -Tham số là tên thư mục muốn tạo.
RMD <param>	-Xóa thư mục. -Tham số là thư mục muốn xóa.
PWD	-Hiện thị đường dẫn tới thư mục hiện tại.
CDUP	-Chuyển đến thư mục cha của thư mục hiện tại.
QUIT	-Thoát phiên làm việc hiện tại.
RESUME <param>	-Tiếp tục truyền tải file chưa hoàn thành. -Tham số là mã định danh file muốn tiếp tục.
PAUSE <param>	-Dừng truyền tải file. -Tham số là mã định danh file muốn dừng.

Bảng 1. Các lệnh trong giao thức

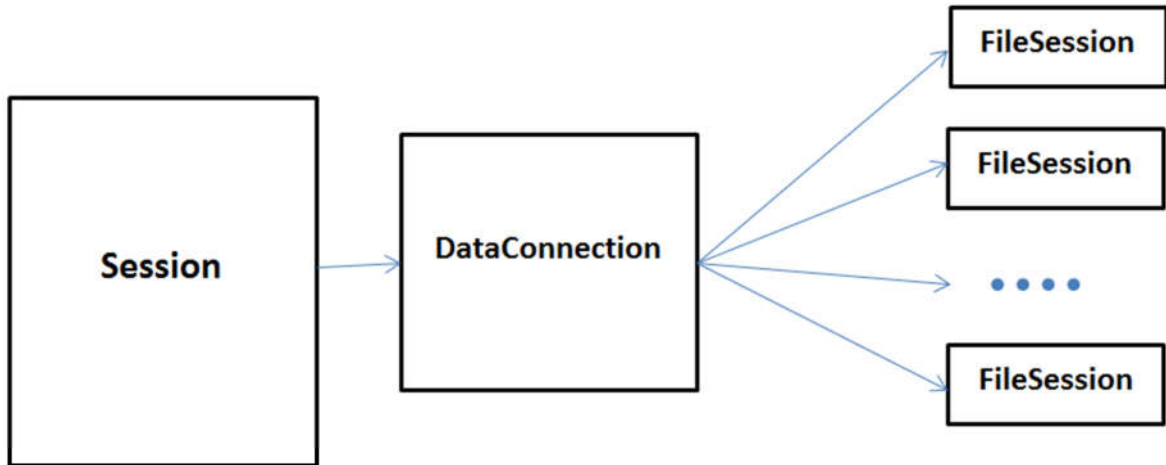
Mã phản hồi	Giải thích
150	File sẵn sàng cho truyền tải. Mở kết nối dữ liệu.
200	Lệnh được chấp nhận.
213	Trạng thái file (lấy thông tin kích thước file).
220	Chấp nhận phục vụ tài khoản mới.
221	Đóng kết nối phục vụ client.
230	Đăng nhập thành công vào hệ thống.
250	Yêu cầu thực hiện lệnh với file đã hoàn thành.
257	Yêu cầu thực hiện với thư mục thành công.
331	Tài khoản được chấp nhận. Yêu cầu nhập mật khẩu cho tài khoản.
450	Yêu cầu với file bị hủy bỏ (file không sẵn sàng phục vụ).
500	Lỗi cú pháp, hệ thống không nhận diện được câu lệnh.
501	Lỗi trong tham số của lệnh.
503	Thư tự thực hiện câu lệnh không hợp lệ.
521	Yêu cầu thực hiện với thư mục thất bại.
530	Chưa đăng nhập vào hệ thống.
550	Không tìm thấy file.
552	Yêu cầu thực hiện lệnh với file bị hủy bỏ.
553	Yêu cầu thực hiện với file không được phép.

Bảng 2. Mã phản hồi từ server

Phản hồi đặc biệt:

- Khi client gửi yêu cầu upload một file lên server, server sẽ kiểm tra có tồn tại file chưa hoàn thành tương ứng. Và server sẽ phản hồi với định dạng: 150 <offset mảnh thứ 1> <offset mảnh thứ 2> ... <offset mảnh cuối>. Với các offset là vị trí offset của các mảnh chưa hoàn thành. Nếu không có file chưa hoàn thành thì offset các mảnh bằng 0.

3.1.2. Kiến trúc trong quản lý truyền tải dữ liệu giữa server và client

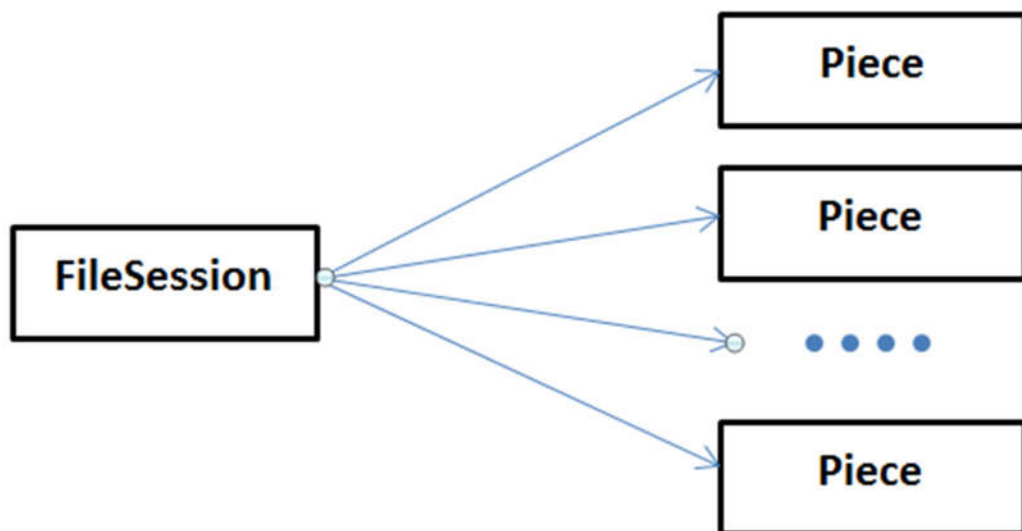


Hình ảnh 2. Kiến trúc phần truyền tải dữ liệu với client

Class	Giải thích
DataConnnection	-Quản lý trao đổi dữ liệu với client.
	-Được tạo sau khi client đăng nhập thành công vào hệ thống.
	-Luồng này chạy độc lập với Session, chờ xử lý yêu cầu từ Session
	-DataConnnection quản lý các FileSession.
FileSession	-DataConnnection gửi các dữ liệu dạng chuỗi ký tự, phục vụ cho phương thức LIST của Session.
	-Điều khiển các FileSession: tạo 1 FileSession mới, dừng truyền tải của FileSession, tiếp tục truyền tải FileSession.
	-Quản lý một phiên trao đổi trên một file.
	-Được tạo ra từ DataConnnection, khi client muốn truyền tải file.
	-Là một luồng riêng độc lập với các luồng khác.
	-Chứa các thông tin liên quan đến file truyền tải: tên, đường dẫn, số phân mảnh, độ dài file, loại truyền tải (download hay upload), các mảnh.
	-Sau khi khởi tạo, FileSession tạo các phân mảnh, thiết lập các phân mảnh và khởi chạy các phân mảnh.

	-Sau khi khởi chạy các phân mảnh, luồng FileSession chờ cho đến khi các mảnh tải xong.
	-Khi các mảnh tải xong, luồng FileSession tiếp tục hoạt động và ghép các mảnh nếu người dùng upload file.
	-FileSession có các phương thức có chức năng lưu trạng thái hiện tại của luồng để phục vụ cho việc phục hồi lại cấu hình để tiếp tục truyền tải file chưa hoàn thành.

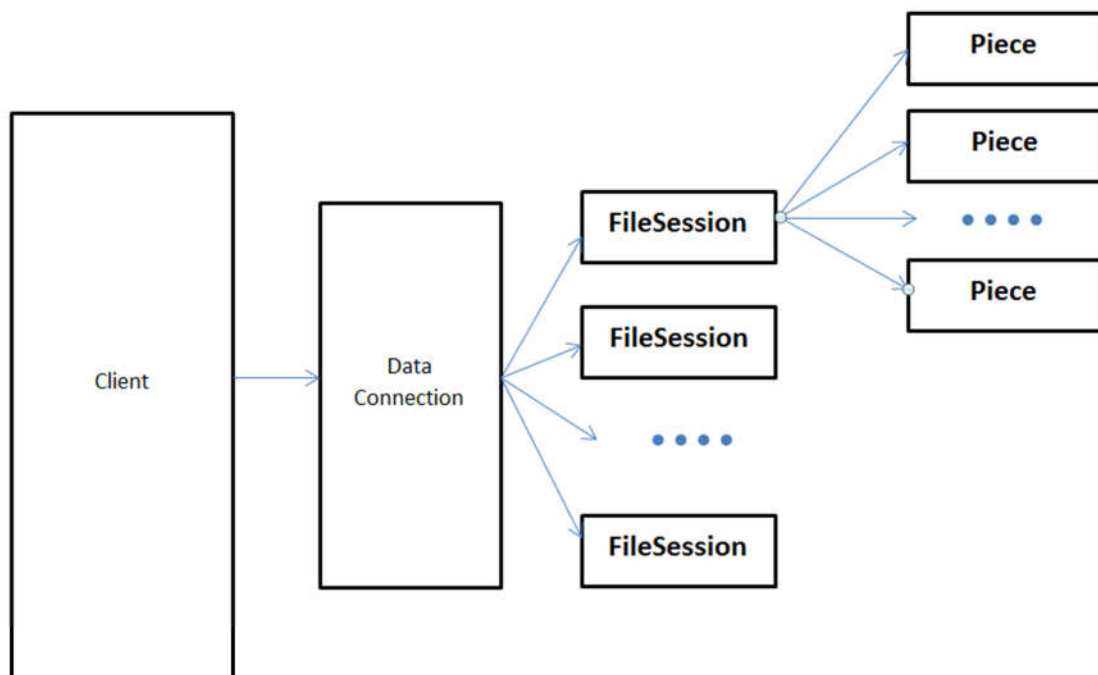
3.1.3. Các phân mảnh của file



Hình ảnh 3. Kiến trúc một phiên truyền tải file

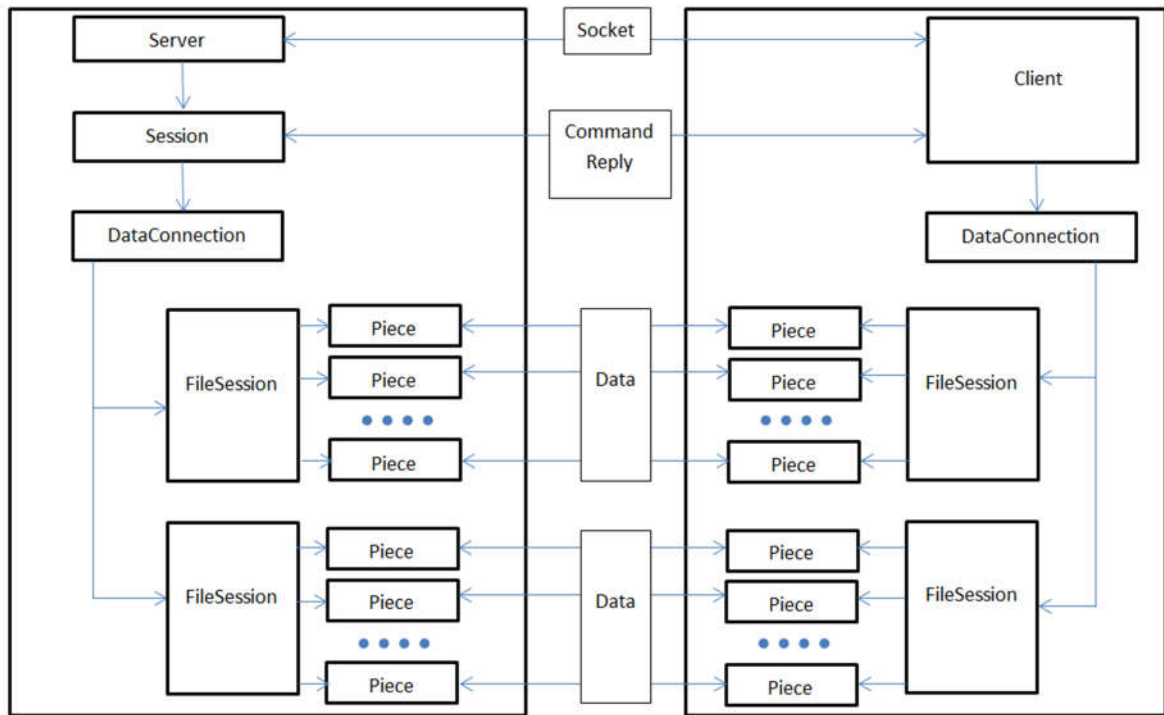
Class	Giải thích
Piece	-Quản lý trao đổi dữ liệu trên một mảnh.
	-Là một luồng chạy độc lập với FileSession và các mảnh khác.
	-Có socket trao đổi dữ liệu riêng với các mảnh khác.
	-Được tạo ra khi FileSession thiết lập các thông số.
	-Lưu các thông tin của mảnh: tên file, chỉ số mảnh, vị trí offset bắt đầu của mảnh trên file, vị trí offset đọc/ghi dữ liệu của mảnh, socket trao đổi dữ liệu.
	-Có các phương thức để truyền tải dữ liệu: receive() và send().
	-Dùng biến đối tượng lock và cờ notified để kiểm soát dừng truyền khi chương trình đang chạy.

3.2. Phía Client



- Kiến trúc của client tương tự với server nhưng ngược lại. Khi client upload thì server lấy dữ liệu, khi client download thì server truyền dữ liệu đi.
- Client gửi các lệnh lên server yêu cầu xử lý. Đồng thời client cũng thiết lập các thông số để kết nối với server.

3.3. Mô hình truyền tải



Hình ảnh 4. Mô hình truyền tải

3.4. Xử lý dừng và tiếp tục truyền tải

Có 2 trạng thái dừng / tiếp tục truyền tải file:

- Trạng thái 1: khi client chưa tắt chương trình. Client mới chỉ gửi yêu cầu dừng/tiếp tục đến server. Lúc này cả client và server sẽ chuyển các luồng của các phân mảnh vào trạng thái wait với lệnh dừng hoặc kích hoạt lại các luồng với lệnh tiếp tục. Như vậy dữ liệu cấu hình hiện tại của các mảnh không mất đi, từ đó chúng ta có thể dừng/tiếp tục dễ dàng.
- Trạng thái 2: khi client tắt chương trình, dữ liệu cấu hình các mảnh sẽ bị mất. Vì thế cần lưu lại các thông tin cấu hình của FileSession và các Piece. Khi client đăng nhập trở lại, dữ liệu lưu này sẽ được đọc lại và thiết lập các FileSession và Piece theo dữ liệu đã lưu. Từ đó, chúng ta lại chuyển sang trạng thái 1.

3.5. Xử lý tương tranh khi client upload file

Khi có 2 client cùng gửi dữ liệu lên cùng một file trong cùng thư mục, lúc này tương tranh xảy ra.

Giải pháp:

- Khi có yêu cầu upload, Server sẽ kiểm tra file có tồn tại không.

- Nếu không tồn tại, quá trình thực hiện bình thường và file được đưa vào danh sách chờ.
- Nếu file đã tồn tại, Server sẽ phản hồi lại file đã tồn tại và chờ Client quyết định ghi đè hay tạo bản copy.
- Nếu chọn tạo bản copy, Server sẽ tạo bản copy với tên có thêm “copy_” ở đầu => quá trình thực hiện tiếp bình thường.
- Nếu chọn ghi đè, Server sẽ kiểm tra trong danh sách chờ:
 - Nếu file không có trong danh sách chờ, quá trình thực hiện bình thường.
 - Nếu file có trong danh sách chờ, yêu cầu của client sẽ được đưa vào hàng đợi chờ xử lý.
 - Khi file bị xóa khỏi danh sách chờ, Server sẽ lấy yêu cầu trong hàng đợi và xử lý tiếp.
- Xác định file bị xóa khỏi danh sách chờ:
 - Khi phiên đang thao tác trên file hoàn thành, phiên sẽ thông báo lại cho Server.
 - Server sẽ xóa file khỏi danh sách chờ và lấy yêu cầu trong hàng đợi ra để xử lý.

4. Hướng dẫn chạy chương trình

4.1. Chạy chương trình trên netbeans

- Mở mã nguồn của 2 project (Client và Server) bằng trình IDE Netbeans.
- Để chạy Server: chọn class MainFrame.java trong project FTPStorageServer -> run file
- Để chạy Client: chọn class MainFrameClient.java trong project FTPStorageClient -> run file

4.2. Chạy chương trình bằng file .jar đã biên dịch sẵn

- Nếu dịch chuyển file .jar trong folder project cần copy kèm theo các file cần thiết. Đối với Server cần copy kèm theo file [ftp.properties](#)
- Nháy đúp vào file .jar để chạy chương trình.

5. Tài liệu tham khảo

5.1. Mã nguồn tham khảo

<https://github.com/rath/simple-ftp>

<http://www.java2s.com/Code/Java/Network-Protocol/>

5.2. Tài liệu tham khảo

1. https://www.w3.org/Protocols/rfc959/A3_FTP_RFCs.html

2. <http://cs.lmu.edu/~ray/notes/javanetexamples/>
3. Adrew S.Tanenbaum, Maarten Van Steen Distributed Systems principles and paradigms