

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



MẠNG MÁY TÍNH (CO3093)

BÁO CÁO BÀI TẬP LỚN SỐ 1

HK231 - Lớp: L07 - Nhóm: 3

ĐỀ TÀI:

SIMPLE FILE-SHARING APPLICATION

GV hướng dẫn: Lê Bảo Khánh

Danh sách thành viên:

| STT | Họ và tên | MSSV | Tỷ lệ đóng góp |
|-----|-----------------|---------|----------------|
| 1 | Liu Ngọc Yến | 2112899 | 25% |
| 2 | Trịnh Khải Toàn | 2113269 | 25% |
| 3 | Từ Mai Thế Nhân | 2114277 | 25% |
| 5 | Lê Hoàng Anh Vũ | 2115319 | 25% |

Mục lục

| | | |
|----------|---|-----------|
| 1 | Mô tả yêu cầu dữ liệu | 2 |
| 1.1 | Yêu cầu chức năng | 2 |
| 1.1.1 | Sever | 2 |
| 1.1.2 | Client | 2 |
| 1.1.3 | File transfer | 2 |
| 1.1.4 | Thiết kế UI/UX | 3 |
| 1.2 | Yêu cầu phi chức năng | 3 |
| 1.3 | Đặc tả các hàm | 3 |
| 1.3.1 | Phía Server | 3 |
| 1.3.2 | Phía Client | 4 |
| 1.3.3 | File Transfer | 5 |
| 1.4 | Giao thức sử dụng | 5 |
| 1.4.1 | Giao thức giao tiếp giữa client và server | 5 |
| 1.4.2 | Giao thức giao tiếp giữa 2 server | 6 |
| 1.4.3 | Giao thức chia sẻ file | 7 |
| 2 | Mô hình hóa hệ thống | 9 |
| 2.1 | Use Case Diagram | 9 |
| 2.2 | Activity Diagram | 9 |
| 2.3 | Sequence Diagram | 9 |
| 2.4 | Class Diagram | 9 |
| 3 | Hiện thực | 10 |
| 3.1 | Các công nghệ sử dụng | 10 |
| 3.1.1 | MERN Stack | 10 |

1 Mô tả yêu cầu dữ liệu

1.1 Yêu cầu chức năng

1.1.1 Sever

Quản trị viên: Quản lý server và tài nguyên của nó, chẳng hạn như thêm và xóa client, giám sát hiệu suất server và khắc phục sự cố.

Giữ lại danh sách các client kết nối và tệp tin trong repository của chúng: Server duy trì một danh sách của tất cả các client đang kết nối và các tệp tin mà mỗi client đã lưu trữ trong repository. Thông tin này được sử dụng để phản hồi các yêu cầu của client để lấy tệp tin. Máy chủ định kỳ cập nhật danh sách này dựa trên thông tin nhận được từ client.

Phản hồi phù hợp cho các yêu cầu của client: Khi client yêu cầu một tệp tin, server xác định danh sách các client khác có file đó trong repository và gửi danh sách các client đó cho client yêu cầu. Client yêu cầu sau đó có thể tải tệp tin trực tiếp từ một trong các client đó mà không cần can thiệp thêm từ server.

Command-shell interpreter: Server có command prompt đơn giản cho phép server thực hiện các nhiệm vụ khác nhau, chẳng hạn như:

- *discover hostname:* Truy xuất danh sách các tệp tin cục bộ trên client có tên là "hostname".
- *ping hostname:* Thực hiện kiểm tra trực tiếp trên server có client tên là "hostname" hoạt động hay không.

1.1.2 Client

Đăng nhập / Đăng ký: Client phải đăng nhập hoặc đăng ký với server trước khi sử dụng dịch vụ truyền tệp tin. Điều này cho phép server theo dõi client nào đang kết nối và những file mà client đó đang lưu trữ trong repository.

Thông báo cho server về các file của client: Client định kỳ thông báo cho server về các file mà client đang lưu trữ trong repository.

Gửi yêu cầu đến client để lấy file: Khi một client cần một file mà nó không có, nó gửi một yêu cầu đến server. Server xác định các client khác có file đó trong repository và gửi thông tin của chúng cho client yêu cầu. Client yêu cầu sau đó có thể tải file trực tiếp từ một trong những client đó.

Command-shell interpreter: Client có command prompt đơn giản cho phép client thực hiện các nhiệm vụ khác nhau, chẳng hạn như:

- *publish lname fname:* Tải một file cục bộ (được lưu trữ trong hệ thống tệp tin của client với tên lname) lên repository của client với tên fname và truyền thông tin này cho server.
- *fetch fname:* Tải một bản sao của file mục tiêu và thêm repository.

1.1.3 File transfer

Chuyển tệp tin giữa các client mà không cần can thiệp từ server: Sau khi server đã xác định danh sách các client khác có file được yêu cầu, client yêu cầu có thể tải tệp tin trực tiếp từ một trong những client đó mà không cần can thiệp thêm từ server.

Xử lý việc tải xuống đồng thời bằng cách sử dụng đa luồng: Code client cần được thiết kế đa luồng để cho phép tải xuống đồng thời nhiều file từ một client.

1.1.4 Thiết kế UI/UX

1.2 Yêu cầu phi chức năng

- Xử lý lỗi và độ tin cậy:
 - Xử lý hiệu quả các trường hợp lỗi như file không tìm thấy, khách hàng mất kết nối, và lỗi mạng.
 - Tính ổn định: Hệ thống phải hoạt động một cách ổn định trong các tình huống khác nhau, bao gồm đồng thời có nhiều khách hàng kết nối và tải xuống, và khách hàng tải xuống các tệp rất lớn.
- Hiệu suất và khả năng mở rộng:
 - Thiết kế hệ thống xử lý nhanh chóng các tác vụ, có khả năng mở rộng để hỗ trợ nhiều khách hàng, bao gồm đồng thời tải và tải lên tệp tin.
- Thống kê:
 - Hệ thống theo dõi và ghi nhận số liệu thống kê về tải xuống và tải lên tệp tin.
 - Ghi nhận số lượng tệp tin đã tải xuống và tải lên, tổng kích thước của tệp tin và tốc độ trung bình của quá trình tải xuống và tải lên.
- Triển khai trên trình duyệt và ứng dụng desktop
 - Hệ thống triển khai trên trình duyệt web và ứng dụng desktop.
 - Người dùng có thể truy cập và sử dụng hệ thống từ cả trình duyệt và ứng dụng desktop.
- Bảo mật
 - Hệ thống triển khai biện pháp bảo mật để bảo vệ tệp tin khỏi truy cập và thay đổi trái phép.
 - Sử dụng mã hóa để bảo vệ tệp tin trong quá trình chuyển đổi giữa các khách hàng.
- Cơ sở dữ liệu
 - Hệ thống sử dụng cơ sở dữ liệu để lưu trữ thông tin về khách hàng, tệp tin và quá trình chuyển đổi tệp tin.
 - Cơ sở dữ liệu giúp quản lý hệ thống và theo dõi hoạt động chia sẻ tệp tin.
- Thiết kế giao diện người dùng (UI/UX)
 - Hệ thống có giao diện người dùng thân thiện, dễ sử dụng và dễ điều hướng.
 - Người dùng có thể dễ dàng tìm kiếm và tải xuống tệp tin, cũng như đăng tải tệp tin lên kho lưu trữ cá nhân.

1.3 Đặc tả các hàm

1.3.1 Phía Server

| Tên hàm | Chức năng |
|---------|-------------------|
| start() | Khởi động server. |

| | |
|--|---|
| stop() | Dừng hoạt động của server, đóng tất cả các kết nối và giải phóng tài nguyên. |
| add_client(hostname, client_files) | Thêm client có tên hostname vào danh sách các client đã kết nối và lưu tên của các file trong repository. |
| remove_client(hostname) | Xóa client có tên hostname khỏi danh sách các client đã kết nối. |
| handle_discover_command(hostname) | Truy xuất danh sách các file trong repository của client có tên là hostname. |
| handle_ping_command(hostname) | Kiểm tra xem client mang tên hostname có đang kết nối đến server hay không. |
| handle_publish_command(hostname, lname, fname) | Cập nhật thông tin về tập tin trong thư mục lưu trữ của server và ghi nhận thông tin của file từ client. |
| handle_fetch_command(hostname, fname) | Hàm này xử lý yêu cầu fetch từ một client cụ thể. |
| update_client_status() | Cập nhật trạng thái của các client trong danh sách. |

1.3.2 Phía Client

| Tên hàm | Chức năng |
|---|--|
| sign_in(hostname, password) | Xác thực thông tin đăng nhập của client (hostname, password) và thiết lập kết nối. |
| sign_up(hostname, password) | Kiểm tra tồn tại duy nhất của hostname trên server ,tạo và lưu thông tin tài khoản (hostname, password) vào server. |
| connect(server_address) | Kết nối với server sử dụng địa chỉ server được cung cấp. |
| disconnect(server_address) | Hủy kết nối với server sử dụng địa chỉ server được cung cấp. |
| request_file_server(fname) | Client gửi yêu cầu lên server tìm file fname. |
| get_file_list() | Trả về danh sách các tập tin mà client đang có trong repository của client đó. |
| connect_peer(peer_address) | Kết nối 2 client với nhau thông qua địa chỉ của client. |
| disconnect_peer(peer_address) | Hủy kết nối với client có địa chỉ peer_address. |
| request_file_from_peer(fname, peer_address) | Yêu cầu file fname từ 1 client có địa chỉ peer_address. |
| send_file_with_peer(fname, peer_address) | Gửi file fname với 1 client có địa chỉ peer_address. |
| receive_file_from_peer(fname, peer_address) | Nhận file từ 1 client có địa chỉ peer_address. |
| delete_file(fname) | Xóa file fname khỏi repository của client. Nó xóa tập tin có tên tương ứng khỏi danh sách và cập nhật thông tin tập tin trên server. |

| | |
|-------------------------------------|--|
| hanle_publish_command(lname, fname) | Thêm file lname trong danh sách local file vào repository với tên fname và gửi yêu cầu đến server, thông báo về việc thêm. |
| handle_fetch_command(fname) | Tạo bản sao của file fname trong repository của client sở hữu và thêm vào repository của client yêu cầu. |

1.3.3 File Transfer

| Tên hàm | Chức năng |
|--|---|
| start_transfer(file_name, source_client, destination_client) | Bắt đầu quá trình truyền tải tập tin từ client nguồn đến client đích. |
| stop_transfer(source_client, destination_client) | Hàm này kết thúc và hủy quá trình truyền tải tập tin từ client nguồn đến client đích. |
| handle_interrupted_transfer(file_name) | Hàm này xử lý trường hợp quá trình truyền tải bị gián đoạn. Nếu quá trình truyền bị mất kết nối hoặc gặp sự cố, hàm này thực hiện các bước xử lý để khôi phục hoặc xử lý tình huống không mong muốn. |
| transfer_file(source_client, destination_client, file_name) | Hàm này được gọi bởi một client để truyền tập tin trực tiếp đến một client khác mà không thông qua server. Nó khởi tạo quá trình truyền tập tin bằng cách thiết lập kết nối trực tiếp giữa client nguồn và client đích. |
| handle_multiple_downloads(file_list) | Hàm này xử lý việc tải xuống đồng thời sử dụng đa luồng. Nó nhận danh sách các tập tin cần tải xuống và tạo các luồng riêng biệt cho từng quá trình truyền tải, cho phép nhiều quá trình truyền tải diễn ra song song. |

1.4 Giao thức sử dụng

1.4.1 Giao thức giao tiếp giữa client và server

- Các thông điệp:
 - CONNECT: yêu cầu kết nối với server
 - ASK: hỏi server về file cần truy xuất, check hostname còn sống không,....
 - RESPONSE: phản hồi yêu cầu kết nối
 - ANSWER: trả về các peer sở hữu file cần truy xuất
 - DISCONNECT: Yêu cầu hủy kết nối với server

- Cú pháp thông điệp:
 - CONNECT <Server_address>
 - DISCONNECT
 - RESPONSE <status_code>
 - ASK <do_what> <> <>Ex:
 - * ASK -help -ping: hỏi cú pháp lệnh ping.
 - * ASK -ping -YanPeer: hỏi xem hostname YanPeer còn sống không - ANSWER <for_what> <> <>
- Ex:
- * ANSWER -ping -300_alive: thông báo hostname còn sống
 - * ANSWER -file -302_ip - “NhPeer, DPeer, NPeer..”
- Quy tắc chung:
 - Về mã trạng thái:
 - * 200: Kết nối thành công
 - * 201: Kết nối thất bại
 - * 300_alive: Thông báo hostname sống
 - * 301_dead: Thông báo hostname chết
 - * 302_ip: Tìm được peer có file cần truy xuất
 - * 303_ipNF: Không tìm được peer có file cần truy xuất
 - Nội dung header:
 - * Method: <Tên thông điệp>
 - * From: Địa chỉ đã gửi.
 - * To: Địa chỉ nhận.
 - * Time: Thời gian.

1.4.2 Giao thức giao tiếp giữa 2 server

- Các thông điệp:
 - CONNECT: Khởi tạo kết nối
 - RESPONSE: Phản hồi yêu cầu kết nối
 - ASK: Yêu cầu peer gửi file
 - ANSWER: Phản hồi yêu cầu
 - NOTI:: Thông báo
 - QUIT: Thoát kết nối
- Cú pháp thông điệp:
 - CONNECT <hostname>
 - QUIT

- RESPONSE <status_code>
- ASK <File_name>
- ANSWER <ACCEPT/DECLINE>
- NOTI <Sender_address> <File_name> <File_size>

- Quy tắc chung:

- 200: Kết nối thành công
- 201: Kết nối thất bại

1.4.3 Giao thức chia sẻ file

- Các thông điệp:

- REQUESTF: yêu cầu share file
- SENDF: Gửi lại file.
- WARN: Gửi cảnh báo

- Cú pháp thông điệp:

- REQUESTF <hostname> <tên file>
- SENDF <tên file> <hostname>
- WARN <status code>

- Quy tắc chung:

- Về mã trạng thái:
 - * 500: Không thể kết nối với peer.
 - * 404: Cần nhắc nhận file.
 - * 401: Không tìm thấy file
- Nội dung:
 - * Method: <Tên thông điệp>
 - * From: Địa chỉ đã gửi.
 - * To: Địa chỉ nhận.
 - * Time: Thời gian.
- Nội dung thông điệp SENDF:
 - * Method: <Tên thông điệp>
 - * From: Địa chỉ đã gửi.
 - * To: Địa chỉ nhận.
 - * Time: Thời gian.
 - * Length: kích thước
 - * Content: 16 kí tự đầu của file
- Nội dung WARN:
 - * Method: <Tên thông điệp>



- * From: Địa chỉ đã gửi.
- * To: Địa chỉ nhận.
- * Time: Thời gian.
- * Length: kích thước
- * Warning: nội dung cảnh báo



2 Mô hình hóa hệ thống

2.1 Use Case Diagram

2.2 Activity Diagram

2.3 Sequence Diagram

2.4 Class Diagram



3 Hiện thực

3.1 Các công nghệ sử dụng

3.1.1 MERN Stack