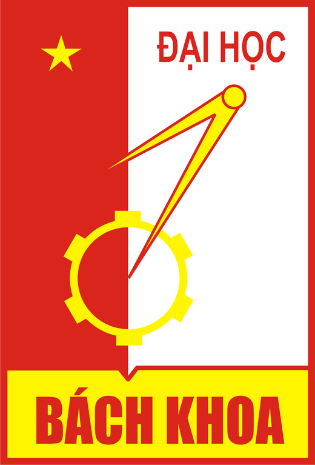
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  
VIỆN ĐIỆN TỬ - VIỄN THÔNG  
\*\*\*\*\* □&□ \*\*\*\*\***



**BÁO CÁO MÔN HỌC**

**HỆ ĐIỀU HÀNH**

**Nhóm 9**

**ĐỀ TÀI: Xây dựng ứng dụng cho phép nhiều máy đồng bộ thư mục qua một server trung gian (tương tự Dropbox)**

**Trịnh Anh Tú – 20200551**

**Nguyễn Văn Trường – 20200654**

***Hà Nội 07 - 2023***

1. **Giới thiệu đề tài** 
   1. Bối cảnh và lý do thực hiện đề tài

Đề tài “ Xây dụng ứng dụng cho phép nhiều máy đồng bộ thư mục qua một server trung gian” nhằm tạo ra một giải pháp hiệu quả để đồng bộ hóa thư mực giữa các máy tính thông qua việc sử dụng một server trung gian. Trong thế thời kỹ thuật hiện đại, việc chia sẻ và truy cập dữ liệu giữa nhiều máy tính trở nên phổ biến và cần thiết, đặc biệt là trong môi trường làm việc nhóm hoặc doanh nghiệp.

* 1. Mục tiêu của đề tài

Mục tiêu chính của đề tài là xây dựng một ứng dụng đáng tin cậy, an toàn và hiệu quả, cho phép đồng bộ hóa thư mục giữa các máy tính thông qua việc sử dụng một server trung gian. Ứng dụng này sẽ cung cấp khả năng đồng bộ hóa tự động và liên tục, giúp đảm bảo rằng dữ liệu trên các máy tính sẽ luôn được cập nhật và đồng bộ.

1. **Giới thiệu về rsync và inotify**

Rsync là một công cụ dòng lệnh được sử dụng để đồng bộ dữ liệu và sao lưu trên các máy tính khác nhau. Nó có khả năng sao chép và cập nhật chỉ những phần của tập tin đã thay đổi, giúp giảm băng thông và thời gian truyền tải. Rsync sử dụng giao thức SSH để bảo mật dữ liệu trong quá trình truyền và hỗ trợ nén dữ liệu để tối ưu hóa hiệu suất. Với bài toán như yêu cầu trên, rsync được sử dụng trong ứng dụng đồng bộ thư mục giữa các tài khoản khách (client1 và client2) với máy chủ (server).

Inotify là một công cụ được sử dụng trong Linux để theo dõi sự kiện trên hệ thống tệp và thư mục. Nó cho phép chương trình tuần tự theo dõi các thay đổi như tạo, sửa đổi hoặc xóa file và thư mục . Với đề tài này, “inotifywait” là một câu lệnh được sử dụng để theo dõi các thay đổi trên thư mục “client1\_path” và “client2\_path”. Khi có sự kiến tạo, sửa đổi hoặc xóa file trong các thư mục này, lệnh “inotifywait” sẽ bắt đầu kích hoạt và thông báo lại cho chương trình chính. Sau đó,vòng lặp “while” sẽ đọc thông báo từ “inotifywait” và xử lý các sự kiện tương ứng. Điều này cho phép các chương trình tự động đồng bộ hóa các thay đổi từ máy khách lên máy chủ và từ máy chủ xuống máy khách.

1. **Giải thích đoạn code**



A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Đoạn code trên được ứng dụng như sau:

Ban đầu, thiết lập đường dẫn thư mục nguồn và đích cho mỗi máy client và máy chủ server. Tiếp theo các hàm “sync\_clients\_to\_server” và “sync\_server\_to\_clients” trong đoạn mã để đồng bộ hóa thay đổi từ client lên server và từ server xuống clients. Với việc sử dụng vòng lặp vô hạn, ta có thể theo dõi các thay đổi trên các máy clients. Khi có thay đổi xảy ra trên một client, các thay đổi sẽ được gửi đến máy chủ server thông qua hàm “sync\_clients\_to\_server”. Sau đó, máy chủ server sẽ gửi các thay đổi tới các client khác thông qua hàm “sync\_server\_to\_clients”. Các thay đổi này sẽ được thực hiện qua giao thức rsync, giúp đảm bảo đồng bộ hóa hiệu quả và chỉ truyền các phần đã thay đổi của file thay vì toàn bộ file. Đồng thời việc sao chép và xóa file tạm trong thư mục “tmp” giúp quản lý và đảm bảo sự an toàn của các file tạm thời.

**Giải thích các phần code :**

Cấu trúc đoạn code:

#!/bin/bash //Chỉ định rằng đoạn mã được viết bằng ngôn ngữ shell bash.

# Set the source and destination paths for client1 and client2

client1\_path="/home/tuitachi/client-server-dropbox/client1"

client2\_path="/home/tuitachi/client-server-dropbox/client2"

server\_path="/home/tuitachi/client-server-dropbox/server"

tmp\_path="/home/tuitachi/client-server-dropbox/tmp"  
 //Khai báo các biến để định nghĩa đường dẫn thư mục nguồn và đích cho client1, client2, server và thư mục tạm thời (tmp)

# Function to sync changes from clients to the server

sync\_clients\_to\_server() {

local client\_path="$1" // gán giá trị của client\_path cho biến cục bộ “client\_path”

local file\_path="$2" // gán giá trị của file\_path cho biến cục bộ “file\_path”

local file\_name="$(basename "$file\_path")" // sử dụng lệnh base name để lấy tên file cuối cùng từ “file\_path” và gán giá trị cho biến cục bộ “file\_name”

local tmp\_file="$tmp\_path/$file\_name" // tạo đường dẫn của file tạm trong thư mục “tmp” bằng cách kết hợp đường dẫn của “tmp\_path” với “file\_name” và gán giá trị cho biến cục bộ “tmp\_file”

# Copy the changed file to a temporary directory

cp "$file\_path" "$tmp\_file" // sử dụng lệnh cp để sao chép file từ “file\_path” đến “tmp\_file”.

# Send the file to the server

rsync -av "$tmp\_file" "$server\_path" // sử dụng lệnh “rsync” để gửi file từ “tmp\_file” đến “server\_path”.

# Clean up the temporary file

rm "$tmp\_file" // sử dụng lệnh “rm” để xóa file tạm “tmp\_file”

}  
Định nghĩa của hàm “sync\_clients\_to\_server”. Hàm này nhận hai tham chiếu đaau vào: “client\_path” đại diện cho đường dẫn của client và “file\_path” đại diện cho đường dẫn của file bị thay đổi

sync\_server\_to\_clients() {

local file\_path="$1" //Gán giá trị của “file\_path” cho biến cục bộ “file\_path”

local file\_name="$(basename "$file\_path")" // sử dụng lênh basename lấy tên file cuối cùng từ “file\_path” và gán cho giá trị biến cục bộ “file\_name”

local tmp\_file2="$tmp\_path/$file\_name.client2" // tạo đường dẫn file tạm cho”client1” và “client2” bằng cách kết hợp “tmp\_path”, “file\_name” và client1 hoặc “”client2” và gán giá trị cho biến cục bộ “tmp\_file1” và “tmp\_file2”

if [[ "$client\_path" == "$client2\_path" ]]; then

# Send the file from client2 to the server

rsync -av "$file\_path" "$server\_path"

rsync -av "$file\_path" "$tmp\_file1"

fi // kiểm tra xem "$client\_path" == "$client2\_path" hay không. Nếu hai giá trị bằng nhau thì điều kiện trong câu lệnh “if” được thực hiện.

# Monitor changes on client1 and client2

while true; do

inotifywait -r -e create,modify,delete "$client1\_path" "$client2\_path" |

while read -r directory event file; do

file\_path="$directory$file"

# Sync changes from clients to the server

sync\_clients\_to\_server "$directory" "$file\_path"

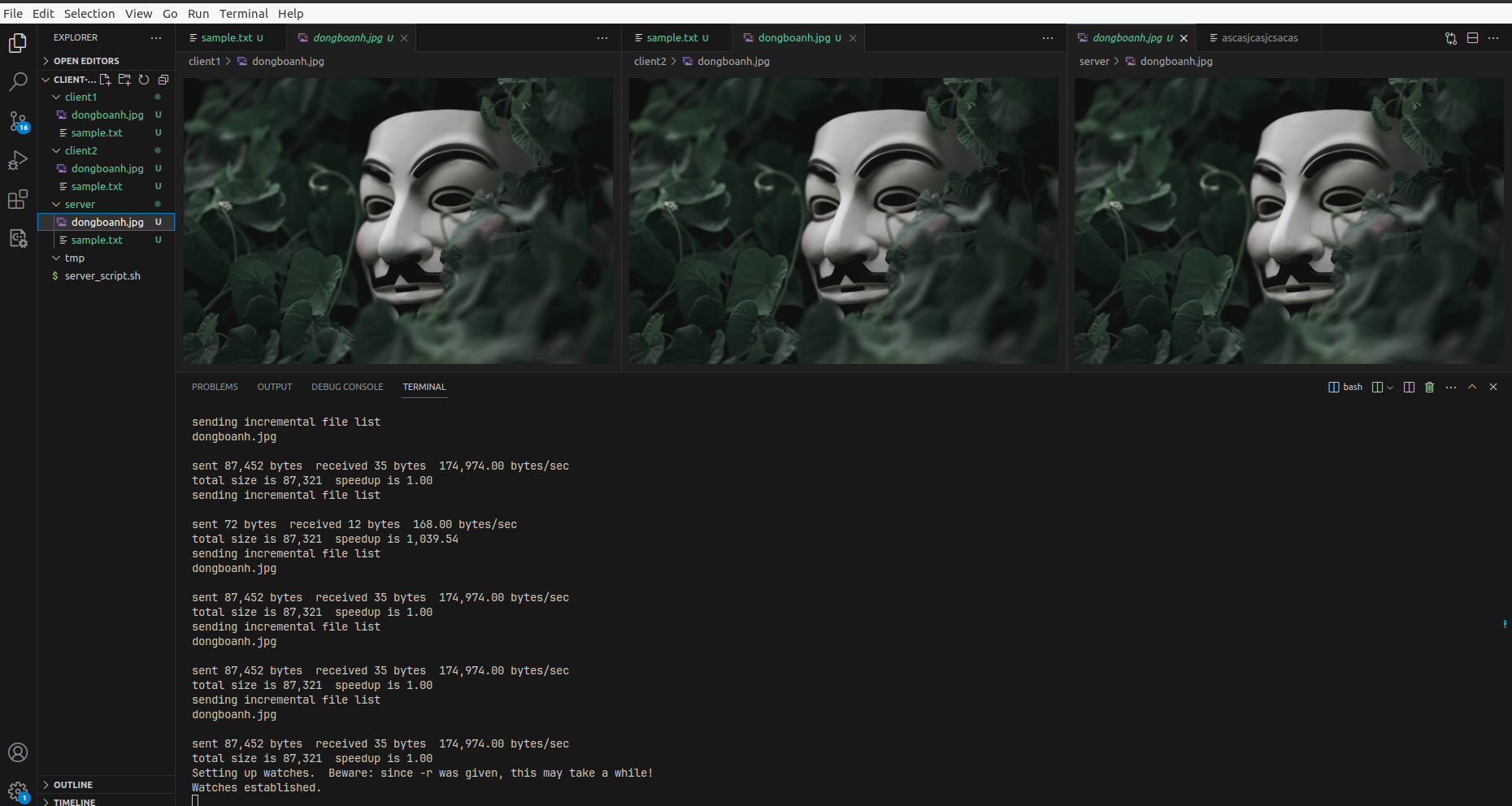
# Sync changes from the server to clients

sync\_server\_to\_clients "$file\_path"

done

done //Đoạn mã trên tạo một vòng lặp vô hạn (whiletrue) để theo dõi các thay đổi trên client1 và client2 bằng cách sử dụng lệnh inotifywait, Với cách sử dụng và inotifywait, các hàm đồng bộ hóa, đoạn mã trên đảm bảo rằng mọi thay đổi trên client1 và client2 sẽ được theo dõi và đồng bộ hóa lên máy chủ và từ máy chủ xuống các máy khách. Điều này đảm bảo rằng dữ liệu trên tất cả các máy tính luôn được cập nhật và đồng bộ nhau. Vòng lặp vô hạn trong đoạn mã cho phép tiếp tục theo dõi thay đổi mà không bị dừng lại.

1. **Kết quả chạy code**

****

**A screenshot of a computer

Description automatically generated**

Nhận xét :Dữ liệu đã được đồng bộ trên cả client1 và client2 và server

1. **Phân công công việc**

|  |  |
| --- | --- |
| Trịnh Anh Tú | Code và chạy kết quả |
| Nguyễn Văn Trường | Survey tài liệu, làm báo cáo |