

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



BÀI TẬP LỚN 1
MẠNG MÁY TÍNH (CO3093)

Đề tài
**Develop A Network Application - P2P
File Sharing System**

Giảng viên hướng dẫn: Hoàng Lê Hải Thanh

Nhóm sinh viên thực hiện:

Phạm Lê Thế Anh	- 1652026
Mai Phương Nhã	- 2111892
Phan Phạm Thi	- 2114857
Trần Hồ Thanh Vũ	- 2220031

TP. HỒ CHÍ MINH, 2023



Mục lục

1	Đề bài	2
1.1	Mục tiêu	2
1.2	Mô tả	2
1.3	Phase 1	3
2	Background	3
2.1	Ngôn ngữ lập trình	3
2.2	Các định nghĩa	3
2.2.1	Socket	3
2.2.2	Socket TCP	3
2.2.3	Thư viện	4
3	Định nghĩa hàm và giao thức	4
3.1	Server	4
3.2	Client	7



1 Đề bài

1.1 Mục tiêu

Xây dựng một ứng dụng chia sẻ tập tin đơn giản với các giao thức ứng dụng định nghĩa bởi mỗi group và dùng giao thức stack TCP/IP

1.2 Mô tả

- Một Server tập trung theo dõi những Clients nào được kết nối và lưu trữ tập tin nào (lưu trữ tên tập tin).
- Client thông báo cho Server biết tệp nào được chứa trong kho lưu trữ cục bộ (local repository) của nó nhưng không thực sự truyền dữ liệu tệp đến Server.
- Khi Client yêu cầu một tệp không thuộc kho lưu trữ của nó, một yêu cầu sẽ được gửi đến Server. Server xác định một số Client khác lưu trữ tệp được yêu cầu và gửi thông tin (tên) của chúng đến Client yêu cầu. Client sẽ chọn một source node (một client/peer có file yêu cầu) thích hợp và sau đó tệp sẽ được Client yêu cầu tìm nạp trực tiếp từ node có bản sao của tệp mà không yêu cầu bất kỳ sự can thiệp nào của Server.
- Nhiều Client có thể tải xuống các tệp khác nhau từ một Client mục tiêu tại một thời điểm nhất định. Điều này yêu cầu mã Client phải đa luồng (multithreaded)
- Client có một trình thông dịch command-shell đơn giản được sử dụng để chấp nhận hai loại lệnh:
 - **publish lname fname**: một tệp cục bộ (được lưu trữ trong hệ thống tệp của Client tại lname) được thêm vào kho lưu trữ của Client dưới dạng tệp có tên fname và thông tin này được chuyển đến Server.
 - **fetch fname**: tìm nạp một số bản sao của target file và thêm nó vào kho lưu trữ cục bộ
- Server có trình thông dịch command-shell đơn giản như sau:
 - **discover hostname**: xem danh sách các local files của các host có tên là hostname
 - **ping hostname**: kiểm tra trạng thái online của host có tên là



- Điều quan trọng cần lưu ý là cơ sở hạ tầng kết nối không được thể hiện trong Hình 1. Tất cả các thiết bị đều được kết nối với nhau thông qua Internet. Việc tách chúng ra là quan điểm hợp lý về các hoạt động của giao thức.

1.3 Phase 1

- Xác định các chức năng cụ thể của ứng dụng chia sẻ tập tin.
- Xác định các giao thức giao tiếp được sử dụng cho từng chức năng.

2 Background

2.1 Ngôn ngữ lập trình

Để hoàn thành bài tập lớn 1 môn Mạng Máy tính với chủ đề ứng dụng chia sẻ tập tin, nhóm đã quyết định dùng ngôn ngữ lập trình Python để hiện thực.

2.2 Các định nghĩa

2.2.1 Socket

Socket cung cấp phương tiện liên lạc FIFO hai chiều qua mạng. Một socket kết nối với mạng được tạo ở mỗi đầu của giao tiếp. Mỗi socket có một địa chỉ cụ thể. Địa chỉ này bao gồm một địa chỉ IP và số port. Ban đầu, Server tạo một socket, gán nó vào một địa chỉ IP - port sau đó đợi Client liên hệ với nó. Client tạo một socket và sau đó cố gắng kết nối với socket bên Server. Khi kết nối được thiết lập, quá trình truyền dữ liệu diễn ra.

2.2.2 Socket TCP

Ban đầu, ta tạo một đối tượng socket bằng cách sử dụng `socket.socket()`, chỉ định loại socket là `socket.SOCK_STREAM` - Mặc định được sử dụng là Transmission Control Protocol (TCP) . Tại sao sử dụng TCP?

- Reliable: Các packets bị loss trong network được sender phát hiện và truyền lại.
- In-order data delivery: Ứng dụng đọc dữ liệu theo thứ tự được sender gửi.



Trong khi đó, các socket User Datagram Protocol (UDP) được tạo bằng `socket.SOCK_DGRAM` không đáng tin cậy và dữ liệu mà người nhận đọc có thể không đúng thứ tự so với thứ tự của người gửi. TCP giúp ta không lo lắng về việc mất packet, dữ liệu đến không đúng thứ tự và các cạm bẫy khác luôn xảy ra khi giao tiếp qua network.

2.2.3 Thư viện

Python hỗ trợ nhiều thư viện hỗ trợ cho việc hiện thực giao thức giao tiếp giữa client và server cũng như giữa các client/peer như `socket` (standard library), `thread` (đa luồng), ... Vì các lý do trên, nhóm đã quyết định chọn ngôn ngữ Python cho bài tập lớn

3 Định nghĩa hàm và giao thức

3.1 Server

Main: `Handle_request()`

Mô tả: Duyệt các request từ client như `publish`, `unpublish`, `fetch`, ..., xử lý các request này dựa trên `Request_type` của gói tin gửi từ client bằng các hàm liên quan

1. Function "`rely_register()`"

- Chức năng chính: Server nhận gói tin đăng ký chứa thông tin về client.
- Câu lệnh thực thi: `rely_register()`
- Mô tả cụ thể: Sau khi Server nhận được gói tin đăng ký được gửi từ client đến server qua kết nối TCP/IP đã thiết lập, server sẽ thực thi câu lệnh **`rely_register()`** xác nhận thông tin từ gói tin và lưu trữ thông tin gồm `username` và `password` vào cơ sở dữ liệu để xác minh cho lần đăng nhập sau. Sau đó server sẽ phản hồi về cho client rằng thông tin đăng nhập đã được ghi vào cơ sở dữ liệu thành công và client có thể đăng nhập với thông tin đã đăng ký.

2. Function "`removeClientInfo`"

- Chức năng chính: Xóa thông tin đăng nhập cũng như các thông tin về `hostname`, `file name` của Client ra khỏi cơ sở dữ liệu.
- Câu lệnh thực thi: `removeClientInfo`

- Mô tả cụ thể: Sau khi client đăng nhập thành công và thực thi lệnh disconnect với command "disconnect" thì server sẽ thực thi lệnh *remove-ClientInfo* và xóa tất cả thông tin đăng nhập và dữ liệu về hostname và file name mà client đã ghi vào cơ sở dữ liệu. Sau đó Server sẽ thông báo về cho client xóa thành công hoặc thất bại. Nếu thành công màn hình client sẽ trở về trang login và không thể đăng nhập với username và password trước đó nữa.

3. Function "Ping hostname"

- Chức năng chính: Server check trạng thái hoạt động của host có tên là hostname. Trạng thái hoạt động là online khi host đang mở máy và đang trong application. Trạng thái hoạt động là offline khi host đang không mở máy hoặc đang không trong application.
- Câu lệnh thực thi: *ping hostname*
- Server thực hiện truy vấn đến hàm Ping để phục vụ cho việc tương tác với client trong các hàm khác. Ví dụ: hàm ***Reply_request_file*** gọi hàm ***Ping hostname*** để kiểm tra xem các peers có chứa file được yêu cầu của requesting client có đang online hay không. Nếu có, server sẽ thêm vào danh sách được trả về cho client yêu cầu, Nếu không, server sẽ loại file của peer đó ra khỏi danh sách trả về cho client yêu cầu.
- Mô tả cụ thể: Sau khi server và client tạo socket để kết nối với nhau, server thực thi câu lệnh "ping" thông qua hệ điều hành (VD: ping -n 1 hostname) để thực hiện lệnh check và gửi một tín hiệu cho client, nếu kết quả trả về là dạng "Received: 0" tức trạng thái hoạt động của client hiện tại đang không online, hàm Ping(hostname) trả về kết quả "False" và server sẽ xóa các thông tin như hostname, IP address của client này ra khỏi danh sách client đang online (available). Ngược lại, nếu kết quả là dạng "Received: 1", có nghĩa sau khi gửi 1 yêu cầu thì cũng nhận được 1 trả lời của client, hàm Ping(hostname) trả về "True", Client đang online.
- Ghi chú: Có thể gọi hàm Ping với tham số hostname truyền bằng command line (VD: ping 127.0.0.1) hoặc truyền qua hàm để phục vụ cho các tác vụ liên quan.

4. Function "ReplyPublishedView(hostname)"

- Chức năng chính: Trả về danh sách các file mà client có tên là hostname đã publish khi client đó gửi yêu cầu ViewPublishedFile.

- Hàm thực thi: function *replyPublishedView(hostname)* với tham số là hostname - tên của client yêu cầu xem danh sách file đã publish.
- Mô tả cụ thể: Sau khi nhận được request từ client, Server truy cập vào cơ sở dữ liệu

5. Function "Reply_publish()"

- Mô tả: Server phản hồi lại lệnh publish của client, lưu thông tin của file vào database, gửi về client thông báo thành công/thất bại
- Gửi về lại client Message gồm:
 - (a) Respond_code: gồm các giá trị {200, 201, 202}
 - (b) Respond_type: gồm các giá trị {"Thành công", "Đã chia sẻ trước đó", "Trùng tên file"} lần lượt tương ứng với các giá trị Respond_code

6. Function "Reply_unpublish()"

- Mô tả: Server phản hồi lại lệnh unpublish của client, xóa thông tin của file khỏi database, gửi về client thông báo thành công/thất bại
- Gửi về lại client Message gồm:
 - (a) Respond_code: gồm các giá trị {200, 203}
 - (b) Respond_type: gồm các giá trị {"Thành công", "Không tồn tại file"} lần lượt tương ứng với các giá trị Respond_code

7. Function Reply_request_file(filename, client_id)

- Chức năng: Server trả về danh sách các peer chứa file lời yêu cầu của client.
- Thuộc tính:
 - client_id: id của client yêu cầu file.
 - peer_list: mảng chứa danh sách id các peer có file đó và đang online.
- Mô tả:
 - Server nhận yêu cầu từ client có tên filename
 - Server tìm trong database những peer có file tương ứng
 - Server lần lượt **ping** đến từng peer để check trạng thái hoạt động
 - Server trả về danh sách những peer có file đó và đang online



3.2 Client

Main: `Handle_sendRequest()`

Mô tả: Gửi các request cho server như publish, unpublish, fetch, ..., dựa trên loại request mà client command.

1. Function "Login"

- Chức năng chính: Client đăng nhập vào server khi có username và password được lưu trong cơ sở dữ liệu.
- Câu lệnh thực thi: *login*
- Mô tả cụ thể: Nếu client đã có username và password thì đăng nhập vào server thành công. Còn nếu không thì sẽ hiển thị username và password không tồn tại và yêu cầu client thực hiện lệnh *registry* để đăng ký username và password. Sau khi đăng ký và server phản hồi về thành công thì màn hình client sẽ chuyển về màn hình login và client có thể đăng nhập thành công vào server với username và password mình đã đăng ký trước đó.

2. Function "Register"

- Chức năng chính: Đăng ký username và password vào Database khi username và password cảnh báo không tồn tại khi login.
- Câu lệnh thực thi: *register*
- Mô tả cụ thể: Sau khi Server kiểm tra username và password khi client nhập vào và nhận thấy username và password chưa tồn tại thì sẽ thực thi câu lệnh ***register*** để tạo một gói tin chứa thông tin đăng ký bao gồm username của client và password của nó. Gói tin đăng ký sẽ được gửi từ client đến server qua kết nối TCP/IP.

3. Function "disconnect"

- Chức năng chính: Ngắt kết nối giữa client và server.
- Câu lệnh thực thi: *disconnect*
- Mô tả cụ thể: Sau khi đăng nhập thành công, client dùng command "disconnect" để gửi request ngắt kết nối cho Server và chờ Server phản hồi câu lệnh ***removeClientInfo*** để xóa thông tin của client ra khỏi database.

4. Function "ViewPublishedFile"



- Chức năng chính: Client có nhu cầu xem danh sách các file mà mình đã publish, gửi request cho Server và nhận phản hồi là danh sách các file.
- Câu lệnh thực thi: *view*
- Mô tả cụ thể: Sau khi đăng nhập/đăng ký thành công, client dùng command "view" để gửi request xem những file mà client đã publish cho Server và chờ Server phản hồi danh sách file được hiện trên giao diện cmd.

5. Function "Publish (lname, fname)"

- Mô tả: Client công bố chia sẻ file ở địa chỉ local là lname, lưu trên hệ thống với tên fname
- Tương ứng với command line **publish lname fname**
- Gửi tới server trung tâm Message gồm:
 - (a) Request_type: Publish
 - (b) Payload: fname, lname, file_size

6. Function "Unpublish (fname)"

- Mô tả: Client ngừng chia sẻ file lưu trên hệ thống với tên fname
- Tương ứng với command line **unpublish lname fname**
- Gửi tới server trung tâm Message gồm:
 - (a) Request_type: Unpublish
 - (b) Payload: fname

7. Function Request_file(client_id, file_name)

- Chức năng: Client gửi yêu cầu lấy file **file_name** đến server.
- Thuộc tính: **client_id** là id của client yêu cầu, **file_name** là tên file cần tải.

8. Function Fetch_file(file_name)

- Chức năng: tìm nạp một số bản sao của target file và thêm nó vào kho lưu trữ cục bộ
- Tương ứng với command line **fetch fname**
- Thuộc tính:
 - **file_name**: tên file cần tải.

- Quá trình gửi yêu cầu và tải file:
 - Client "C1" gửi yêu cầu lấy file filename bằng lệnh `request_file("file1.jpg")`
 - Server trả về danh sách peer bằng hàm `reply_request_file("file1.jpg", "C1")`
 - Client "C1" nhận được danh sách peer có file là ["P1"]
 - Client chọn một peer từ danh sách ["P1"] và kết nối đến peer đó Peer "P1" sử dụng hàm `send_file("C1", "file1.jpg")` để gửi file cho Client "C1"
 - Client "C1" nhận được file và lưu trữ

9. Function `Send_file(client_id, file_name)` - *Peer*

- Chức năng: Gửi nội dung file từ peer sang cho client.
- Thuộc tính:
 - `client_id`: ID của client yêu cầu file.
 - `file_name`: Tên file cần gửi.
- Thực hiện:
 - Nhận `peer_id` và `file_name` từ tham số hàm.
 - Tìm kiếm và mở file có tên `file_name` trên hệ thống tập tin của peer.
 - Lấy địa chỉ IP của client thông qua `peer_id`.
 - Mở socket kết nối với client theo địa chỉ IP.
 - Đọc nội dung file theo đơn vị bloc và gửi qua socket.
 - Luân phiên giữa đọc file và gửi dữ liệu để đảm bảo tốc độ truyền.
 - Đóng file và socket sau khi gửi xong toàn bộ nội dung file.
 - Trả về thông báo hoàn tất gửi file.



Tài liệu tham khảo

- [1] harleenk_99, *Socket in Computer Network*. <https://www.geeksforgeeks.org/socket-in-computer-network/>
- [2] Nathan Jennings, *Socket Programming in Python*. <https://realpython.com/python-sockets/#background>