ĐỒ ÁN THỰC HÀNH 1

Ứng dụng Client-Server truyền và nhận file qua giao thức TCP

Tổng quan

Nhóm sinh viên sẽ xây dựng một ứng dụng client-server sử dụng giao thức TCP, cho phép nhiều clients cùng kết nối đến một server để thực hiện các thao tác truyền và nhận file. Ứng dụng sẽ sử dụng socket để quản lý kết nối, và server phải có khả năng xử lý đồng thời nhiều yêu cầu từ các clients.

Mục tiêu

Xây dựng ứng dụng giúp sinh viên hiểu rõ về:

- Cơ chế giao tiếp client-server sử dụng socket và giao thức TCP.
- Xử lý nhiều kết nối đồng thời bằng đa luồng (multithreading) hoặc bất đồng bộ (asynchronous programming).
- Truyền file qua mạng một cách hiệu quả và xử lý lỗi cơ bản trong quá trình truyền.

Yêu cầu

1. Ứng dụng phía Server:

- **Kết nối (0.5 điểm)**: Sử dụng socket TCP để lắng nghe và chấp nhận các kết nối từ nhiều clients.
- **Xử lý đa luồng hoặc bất đồng bộ (0.5 điểm)**: Mỗi khi có một client kết nối, server phải tạo một luồng mới hoặc xử lý bất đồng bộ để đáp ứng yêu cầu mà không làm gián đoạn các client khác.
- **Chức năng upload (1 điểm)**:
    - Server nhận file từ client và lưu trữ vào một thư mục chỉ định trên server (configurable).

    - Tạo tên file được thay đổi để đảm bảo tính duy nhất nếu file đã tồn tại (ví dụ: thêm số thứ tự hoặc timestamp).
- **Chức năng download (1 điểm)**:
    - Khi client yêu cầu tải xuống một file cụ thể, server kiểm tra sự tồn tại của file.
    - Nếu file tồn tại, server truyền file đó cho client; nếu không tồn tại, server gửi thông báo lỗi.
- **Xử lý lỗi (0.5 điểm)**: Server phải xử lý được các tình huống lỗi như file không tồn tại, mất kết nối đột ngột, yêu cầu không hợp lệ, v.v.

2. Ứng dụng phía Client:

- **Kết nối đến server (0.5 điểm)**: Client kết nối đến server qua socket TCP.
- **Giao diện dòng lệnh (CLI)**:
    - Người dùng nhập lệnh qua CLI để thực hiện các thao tác.
    - Các thao tác bao gồm:
        - `upload <tên_file_hoặc_đường_dẫn>`: Gửi file lên server. (1 điểm)
        - `download <tên_file_hoặc_đường_dẫn>`: Yêu cầu server gửi file về. (1 điểm)
    - Giao diện CLI phải thông báo kết quả cho người dùng (thành công hay lỗi) và các lỗi khác có thể xảy ra. (0.25 điểm)
- **Đóng kết nối (0.5 điểm)**: Đảm bảo kết nối được đóng sau khi truyền file hoặc khi không còn sử dụng để tránh rò rỉ tài nguyên.
- Client cần cập nhật theo thời gian thực cho người dùng biết về tiến độ tải file theo tỷ lệ phần trăm (progress update). (0.25 điểm)

3. Các yêu cầu kỹ thuật:

- Sử dụng thư viện `socket` của chính ngôn ngữ lập trình để lập trình kết nối TCP. Không sử dụng thư viện bên thứ 3 thay thế cho thư viện socket.
- Server phải có khả năng phục vụ nhiều clients đồng thời bằng cách sử dụng đa luồng (mỗi client là một luồng riêng) **hoặc** xử lý bất đồng bộ (asynchronous I/O) tuỳ chọn. Nếu ứng dụng chỉ cho phép 1 client - 1 server thì nhóm sinh viên sẽ không được điểm tối đa.
- **Xử lý và thông báo lỗi**: Cả client và server đều phải thông báo chi tiết khi có lỗi (ví dụ: lỗi kết nối, lỗi file không tồn tại…).

- **Ngôn ngữ lập trình**: Python, Java, C# hoặc C++... tuỳ chọn môi trường Windows, macos, Linux.
- **Lưu ý: Không** lập trình ứng dụng Web (HTTP, WebSocket).

4. Tiêu chí đánh giá:

- **Đúng chức năng**: Đảm bảo các chức năng như tải lên, tải xuống hoạt động chính xác.
- **Xử lý đồng thời**: Server có khả năng phục vụ nhiều client cùng lúc mà không bị gián đoạn.
- **Clean code** (1 điểm): Code cần có chú thích rõ ràng, tuân theo chuẩn code và dễ bảo trì.
- **Xử lý lỗi tốt**: Ứng dụng phải xử lý được các lỗi có thể xảy ra trong quá trình truyền file và kết nối.

Yêu cầu mở rộng (không bắt buộc, cộng thêm điểm):

1. **Giao diện đồ họa người dùng (GUI)**: (0.5 điểm)
    - Cả client và server có thể có giao diện GUI thân thiện với người dùng, giúp dễ dàng thao tác các chức năng tải lên và tải xuống file. GUI cho phép người dùng chọn file từ hệ thống thay vì nhập tên file thủ công.
    - Có thể sử dụng các thư viện GUI bên thứ 3 (3rd party library) để xây dựng giao diện.
2. **Xác thực người dùng**: (0.5 điểm)
    - Thêm chức năng xác thực đơn giản như yêu cầu người dùng nhập mã PIN hoặc mật khẩu trước khi kết nối.
    - Đảm bảo mã PIN/mật khẩu được kiểm tra và thông báo lỗi khi xác thực thất bại.
3. **Ghi nhật ký (log) tại phía server:** (0.5 điểm)
    - Ghi nhật ký các hoạt động tại phía server ra file, gồm các hoạt động như: thời điểm server khởi động, thời điểm restart, thời điểm dừng hoạt động, các kết nối từ phía client, các yêu cầu download/upload tập tin…
4. **Cho phép upload một thư mục (folder):**
    - Cho phép upload một thư mục gồm nhiều tập tin trong thư mục đó.
    - Hỗ trợ upload tuần tự các files trong thư mục (0.25 điểm).
    - Hỗ trợ upload song song cùng lúc nhiều files trong thư mục (0.5 điểm).

Yêu cầu báo cáo

Sinh viên phải nộp một báo cáo mô tả (1 điểm):

- Cấu trúc chương trình.
- Cách thức hoạt động của mỗi phần chính (client, server).
- Quy trình xử lý các yêu cầu tải lên, tải xuống và lỗi.
- Các thư viện và công nghệ đã sử dụng.
- Các vấn đề gặp phải trong quá trình thực hiện và cách khắc phục.

Yêu cầu nhóm sinh viên

- Nhóm sinh viên gồm 3 sinh viên (tối thiểu 2).
- Phân chia công việc đều giữa các thành viên.

Hình thức đánh giá

- Vấn đáp nhóm sinh viên và từng cá nhân. (1 điểm)
- Demo trực tiếp chương trình.